MEASUREMENT AND MANAGEMENT OF THE IMPACT OF MOBILITY ON LOW-LATENCY ANONYMITY NETWORKS

S.DOSWELL

Ph.D

2016

Measurement and management of the impact of mobility on low-latency anonymity networks

Stephen Doswell

A thesis submitted in partial fulfilment of the requirements of the University of Northumbria at Newcastle for the degree of Doctor of Philosophy

Research undertaken in the Department of Computer Science and Digital Technologies, Faculty of Engineering and Environment

October 2016

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others. Any ethical clearance for the research presented in this thesis has been approved. Approval has been sought and granted by the University Ethics Committee on 23rd March 2012.

I declare that the word count of this thesis is no more than 40,718 words.

Stephen Doswell
October 2016

Acknowledgements

I would like to express my sincere appreciation and thanks to my supervision team:
Dr. Nauman Aslam, Dr. David Kendall, and Dr. Graham Sexton for their guidance, knowledge, and not least patience. A special thanks to my friends and family and finally the medical professionals who helped me recover from serious illness to be able to return to my studies and submit this thesis.

Anyone who sacrifices their privacy for security will end up with neither.

Benjamin Franklin (1706 - 1790)

Abstract

Privacy, including the right to privacy of correspondence, is a human right. Privacy-enhancing technologies, such as the Tor anonymity network, help maintain this right. The increasing use of Tor from mobile devices raises new challenges for the continued effectiveness of this low-latency anonymity network. Mobile Tor users may access the Internet from a range of wireless networks and service providers. Whenever a wireless network hands-off a mobile device's connection from one access point to another, its external Internet Protocol (IP) address changes, and the connection to the Tor network is dropped. Every dropped connection requires the Tor circuit to be rebuilt. The time required to rebuild the circuit negatively impacts client performance. This research is the first to highlight this negative impact and to investigate the likely extent of the impact for typical usage scenarios and mobility models. The increased network churn caused by circuit rebuilding also negatively impacts anonymity. A novel metric (q-factor) is proposed here to measure the trade-off between anonymity and performance over the duration of a communication session. Two new solutions to the problems of managing mobility in a low-latency anonymity network are proposed in this thesis. The first solution relies on adaptive client throttling, based on a Kaplan-Meier estimator of the likelihood of a mobile network hand-off. The second solution relies on the use of a static bridge relay (mBridge) that acts as a persistent 'home' for a mobile Tor connection, so avoiding the need to recreate the Tor circuit whenever the mobile device is handed-off. The effectiveness of these solutions has been measured using the new q-factor metric. Both solutions provide better performance for mobile Tor clients than the standard Tor client implementation, although some performance reduction by comparison with static Tor clients remains. The bridge relay solution (mBridge) has been shown to offer better performance than client throttling, but is more vulnerable to certain types of attack. A strength of both solutions is that changes are restricted to client devices, the existing algorithms and protocols of the interior Tor network are unaffected.

List of publications

Published:

Stephen Doswell, Nauman Aslam, David Kendall and Graham Sexton (2015) A longitudinal approach to measuring the impact of mobility on low-latency anonymity networks. In: Proceedings of the 2015 International Wireless Communications and Mobile Computing Conference (IWCMC 2015 Security Symposium). IEEE, 24th-28th August 2015, Dubrovnik, Croatia.

Stephen Doswell, Nauman Aslam, David Kendall and Graham Sexton (2013) Please Slow Down! The Impact on Tor Performance from Mobility. In: 3rd Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM), 8th November 2013, Berlin, Germany.

Stephen Doswell, Nauman Aslam, David Kendall and Graham Sexton (2013) The novel use of bridge relays to provide persistent Tor connections for mobile devices. In: 2013 IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications: Mobile and Wireless Networks (PIMRC'13 - Mobile and Wireless Networks), 8th-11th September, 2013, London.

Stephen Doswell (2013) Internet anonymity with mobility - key challenges for the future. In: Northumbria Research Conference (NRC'13), 15th-16th May 2013, Newcastle.

Glossary

This list contains domain-specific terms that are not ambiguous but, rather, may be unknown to the reader. It provides a general meaning of these terms.

Additive-increase / multiplicative-decrease (AIMD): refers to the algorithm best known for its use in Transmission Control Protocol (TCP) congestion management.

Anderson's rule: refers to a principle formulated by Ross J. Anderson that if a system designed for ease of access it becomes insecure; if made too secure it becomes impossible to use.

Average bitrate (ABR): is the measurement of the average amount of data transferred per unit of time, usually per second.

Bonini's paradox: explains the difficulty in constructing models or simulations that fully capture the workings of complex systems.

Botnet: a number of Internet-connected computers communicating often used to send spam email or participate in distributed denial-of-service attacks.

Braess's paradox: is a proposed explanation for why improvements to a network can sometimes impede traffic through it, generating worse overall performance.

Economy: the ratio between good and bad data transferred.

Dissent: is a research project to create a practical anonymous group communication system offering strong, provable security guarantees with reasonable efficiency.

Garlic routing: a variant of onion routing that encrypts multiple messages together and uses separate outbound and inbound paths, to make it more difficult for attackers to perform traffic analysis.

Global (adversary): both a theoretical and non-theoretical adversary who has infinite, or finite but large amount of capability, to perform an attack.

Hand-off: refers to the process of transferring communications from one network to another.

Indinymity: a 'distinguishability' based metric, based on distinguishing features for adversaries to make probabilistic 'guesses'.

Invisible Internet Project (I2P): an overlay network and 'darknet' that allows applications to send messages to each other pseudonymously and securely.

Kaplan-Meier estimator: also known as the product limit estimator, is used to estimate the survival function from lifetime data. In medical research, it is often used to measure the fraction of patients living for a certain amount of time after treatment.

Man-in-the-Middle (MitM): in cryptography and computer security, a man-in-the-middle attack is where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other.

Mix network: first described by David Chaum, mix networks are routing protocols that create hard-to-trace communications by using a chain of proxy servers known as mixes.

Onion routing: is a technique for anonymous communications over a computer network, where messages are encapsulated within layers of encryption, analogous to layers of an onion.

Onion, the: is a conceptual representation of the multilayered encapsulation of the onion routing datagram.

Orbot: is software that provides anonymity on the Internet from a Google Android smartphone. It acts as an instance of the Tor network on such mobile devices and allows traffic routing from a device's web browser, e-mail client, etc., through the Tor network, providing anonymity for the user.

Privacy-enhancing technologies (PET): a technology that enhances the privacy of an individual!

Possinymity: a ‘possibilistic’ based measurement of an anonymity set size motivated by plausible deniability arguments.

Quality of experience (QoE): a qualitative measurement of a system’s effectiveness, usually from the user’s perspective, such as ease of use, functionality.

Quality of service (QoS): a quantitative measurement of a system’s effectiveness, such as for performance, its reliability, speed.

Roaming: the term here refers to handing-off between networks that incurs a change to the external Internet Protocol (IP) address of the user.

Sybil: is an attack where the adversary subverts reputation-based security system by forging identities, within a network.

Tails: the Amnesic Incognito Live System (Tails) is a security-focused Debian-based Linux distribution aimed at preserving privacy and anonymity, with all its outgoing connections forced to go through Tor by default.

Tor: derived from the original software project name The Onion Router, Tor is a free software and a low-latency anonymity network for enabling anonymous communication, based on onion routing.

Nomenclatures

This list defines notations, terms and symbols that are ambiguous:

HOT: an acronym of 'Hand-Off Time', the interval between physical network hand-offs.

h: is the shortened notation for a HOT.

θ: the threshold, for example the threshold for anonymity (a) is notated as θ_a.

Θ: the upper threshold if two layers of thresholds are used. For example, the lower threshold for performance (b) remains notated as θ_b, and the higher performance threshold as Θ_b.

q**-factor**: is the value of the conjunction of v_a and v_b, that is $q = v_a \cdot v_b$, where the boolean v is based on whether anonymity (a) and performance (b) measurements meet their respective thresholds (θ).

g: is goodput, that is, the total amount of application level data received by the client between hand-offs.

$\lfloor g \rfloor$: is the total number of successful file transfers, derived from goodput, between hand-offs.

b: is the total value of successful file transfers, achieved during the scenario time, presented as an average bitrate (ABR.

o: is the overhead, the value of remaining application level data that is does not form part of a successful file transfer, such as partial downloads, data remaining in-flight at hand-off.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Privacy, including the right to privacy of correspondence, is a human right. In 1948, the Universal Declaration of Human Rights (UDHR) was ratified by the United Nations [1]. Among the articles is Article 12, “No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence” [1]. Although not a legally binding document, the declaration forms part of customary international law. A couple of years later, in 1950, all members of the Council of Europe signed the European Convention on Human Rights (ECHR) [2]. The ECHR is a legal instrument. Members of the European Union (EU) are legally bound to protect the human rights, including privacy, of their citizens. This was later supplemented with a specific reference to the communications via the Internet [3]. As with the UDHR, the ECHR includes an article on protecting privacy. Article 8 provides an individual the right to a “private and family life, his home and his correspondence”. In contrast to the UDHR, the ECHR adds “in accordance with law…necessary in a democratic country” [2]. In the United Kingdom (UK), the ECHR is legislated within the Human Rights Act 1998 [4]. The UDHR, however, also affords an individual “the right to life, liberty and security of person” (Article 3) [1]. This is again supported by the ECHR (Article 5) as the right of “liberty and security”, and consequently within UK legislation [2] [4]. The balance between supporting the right to privacy while protecting society is an ongoing debate. This is further highlighted by recent disclosures (by Edward Snowden) of large-scale blanket surveillance of personal communications including Internet usage [5]. The social context continues to be debated but is out of scope for this work [6]. How this relates to technology, however, in particular Internet communications, forms the basis of this research.

Privacy enhancing technologies (PET) encompass a wide range of tools and techniques to help maintain an individual’s right to privacy, and therefore security [7]. PETs includes cryptography, location privacy, obfuscation-based privacy, privacy in mobile devices, reliability of privacy systems, anonymous communications and censorship resistance [8].

Anonymous communications' systems, such as the Tor anonymity network, help maintain an individual's right to privacy (of correspondence) by providing anonymity for their Internet traffic [9]. During the 10 years since its public release, the Tor anonymity network has gained a degree of infamy. The Silk Road online black market, selling illegal drugs around the world, is one example of the sometimes nefarious use of Tor [10] [11] [12] [13] [14]. Consequently, governments are known to have an uneasy relationship with Tor. For example, there are a number of countries that block Tor [15]. China has outlawed the use of Tor altogether while Saudi Arabia and United Arab Emirates previously blocked Tor's website and consequently the downloading of the Tor software [16] [17]. Russia, rather than blocking Tor, has offered a bounty to anyone able to break the anonymity of Tor [18]. Other countries, however, are less transparent. The intelligence and security services of the United Kingdom (Government Communications Headquarters (GCHQ)) and the United States (National Security Agency (NSA)) are reported to have made repeated attempts to develop attacks against Tor [19]. Leaked documents confirm that the NSA operates and collects traffic from some routers on the Tor network [19]. NSA also tracks individuals who only search for and download Tor [20]. The Federal Bureau of Investigation (FBI) has also admitted having taken control of the largest hosting provider of Tor services to identify criminals distributing child pornography [21].

Maintaining the optimal balance between anonymity and performance is critical for low-latency anonymity networks [22]. Design considerations such as number of 'hops' within a circuit and the selection of paths with the highest available bandwidth are just two examples of how the need to maintain this balance has influenced Tor's design [23] [24]. Research has also shown that poor performance, by deterring usage, can negatively impact anonymity by reducing the number of concurrent users where eventually the number of concurrent users offers little or no anonymity [22].

The development of smartphone technology is increasing Internet usage from wireless-enabled devices. Mobile users may access the Internet from a range of networks and service providers (cellular, Wi-Fi). Even for the same Internet service such as BT Wi-Fi, which, provides more than five million hotspots within the United Kingdom, the service allocates a different external Internet Protocol (IP) address after each hand-off [25] [26]. The challenges of both cross-domain and inter-domain network handovers can cause problems in maintaining a connection for Internet services that use Session Initiation Protocol (SIP) such as Voice over Internet Protocol (VoIP). Due to the current design of Tor, the connection to the Tor network also breaks whenever a user's external IP address changes, requiring an extended time to recovery while Tor builds new circuits [9]. Although there are several mobility management protocols at different layers such as Mobile IP (RFC5944) binding and Wi-Fi pre-authentication, these cannot provide seamless handover if used in their current form. An additional optimization mechanism is needed to prevent the loss of in-flight packets transmitted during the mobile node's binding update procedure and to achieve seamless handovers. RFC6252 is a proposed framework for media-independent, pre-authenticated, secure handover optimization scheme that works over any link layer and with any mobility management protocol, including Mobile IP and SIP [27]. However, although interesting, the request for comments is at a status of 'information only' and not currently implemented. The effect on the Tor's secure session management is unknown and at this juncture, the support of mobility on Tor requires to address the current technology.

A limited amount of research has so far examined the use of anonymity networks, such as Tor, from a mobile device. Two studies have explored the impact of mobility on low-latency anonymity networks in both cases using Tor as a case study [28] [29]. The first study assessed the performance of running a Tor client on a cellular network connected device, a 'standard' mobile phone [28]. An experiment compared the performance, download times of files via the Tor network, from both a wired and wireless Internet connections. As expected, performance from the faster wired connection was significantly better but generally more consistent, where the results from the wireless connection was found to vary greatly between different test cases. It is important to note that the experiment was based on the device remaining stationary. Although the study provided an early indication of performance connecting to Tor from a mobile device, testing while roaming would have been more beneficial due to factors such as the impact of hand-offs. The second study examined the impact of 'mobility' on Tor but this time in the context of providing 'location privacy' [29]. Initially the case study proposed a "travelling businessman" with a requirement for "downloading streaming content on a train". The focus, however, changed to maintaining anonymity ("checking the status of the stock market") at different locations while away from home. Therefore, the research moved from understanding the impact of a more 'dynamic' mobile Internet connection, such as 'roaming' Wi-Fi hotspots, to maintaining location privacy, while staying relatively 'static' at a hotel. The authors conclude that Tor "does not support mobility", adding, implementing a simple mobility solution, such as Mobile IP, to an anonymity network such as Tor, "location privacy is lost" [29]. Although an important starting point, these previous studies do not address the impact of mobility on both anonymity and performance, over time, from mobile users recycling their connections.

In 2010, the Guardian Project released the Orbot Android application [30] [31]. Orbot enables onion routing (via the Tor network) on the Android operating system, and supports web browsing via Orfox, a new bespoke application replacing the original vulnerable Orweb browser [32]. By 2016, Orbot has so far recorded over five million installations [30].

Alongside the reported two million daily users of Tor, this are not merely 'underground' technology [33]. Mainstream support from Internet services such as Facebook could potentially open up the Tor network to a billion users in the future [34] [35] [36]. The predicted parity of the number of mobile and desktop Internet connections and the potential impact of mobility on current anonymity network design suggests new research is long overdue.

During the investigation, the current approaches used for measuring anonymity networks predominately only measure either anonymity or performance. This reflects that solutions often primarily focus on either securing Tor (that is enhancing anonymity) or usability (enhancing performance). However, these are not mutually exclusive and each can have an effect on the other. To maintain this critical balance and building upon the work of Panchenko, Lanze, and Engel [37], a more appropriate approach to measuring both anonymity and performance is also investigated as part of research. A new metric, named *q*-factor, is evaluated for both its effectiveness for measuring anonymity and performance over time but also the proposed solutions to mitigate the impact of mobility.

1.1 Research methodology

To explore the effect of mobility on low-latency anonymity networks, the research follows three distinct phases: 'Exploratory', 'Constructive', and 'Empirical'. The *exploratory* phase of the research further defines the research problem and questions from the original research proposal. Referring back to the title of the research: *The Measurement and Management of the Impact of Mobility on Low-Latency Anonymity Networks*, the key aim of the early work is to confirm the impact of mobility on low-latency anonymity networks such as Tor. However, this research is not aiming to improve mobility modelling nor through changing mobility patterns, in this case human behaviour, to 'fit' a mobility solution. In fact, the complete opposite is the case. The aim is not to shape user behaviour, which, itself could be counterproductive in terms of maintaining anonymity of the mobile user as discussed later, but react to the increasing mobility of users. The next phase is the *constructive* research, that is, based on the literature and findings of the impact analysis, one or more solutions are proposed. Finally, the *empirical* phase will first test the theory underpinning the new metric, and if suitable, assess the feasibility of each of the solutions using the new metric. The metric uses quantitative measures and described in detail during Chapter 4. Each of the phases are supported by distinct research processes based on the common 'DNA' approach of: question, hypothesis / prediction, experimentation, and analysis. Table 1.1 shows the alignment of both the research phases and processes for the core chapters.

	Chapter	**Phase**	**Process**
2	Privacy Enhancing Technologies and Anonymity Networks	Exploratory	Generate research questions
3	The Impact of Mobility on Performance	Constructive	Test the ‘impact’ hypothesis / propose solutions
4	Measuring Anonymity and Performance in Dynamic Network Topologies	Empirical	Experimentation and analysis
5	Throttling the Impact of Mobility	Empirical	Experimentation and analysis
6	mBridge: Persistent Connections for Mobility and Anonymity	Empirical	Experimentation and analysis

Table 1.1 The research methodology and phases for each of the core chapters.

1.2 Research questions

Mobility, and its potential impact on anonymity and performance, for low-latency anonymity networks such as Tor has not been re-examined since 2007. The development of smartphone technology and subsequent increase in mobile Internet usage raises the following questions:

1. What is the impact of mobility on performance using Tor while roaming?

2. What is the effect of mobility, and increased network churn, on anonymity?

3. What is the best approach to mitigate the impact of mobility while maintaining acceptable anonymity and performance?

4. How should anonymity and performance be best measured in dynamic environments?

These questions will be answered during the course of the research and reviewed together in the Conclusions (Section 7.1) in the final chapter.

1.3 Contributions

In this research, the effect of mobility on low-latency anonymity networks, such as Tor, is explored. Through experimentation and modelling, the measurement and management of mobility are investigated to help maintain an effective anonymity service. The key contributions, supported by a number of publications (please refer to the *List of Publications* on page xi) are a follows:

1. The first study to highlight and quantify the negative impact of mobility on low-latency anonymity networks, such as Tor.

By applying a range of approaches: field experimentation, network simulation, and mathematical modelling; while a mobile Tor user is roaming, the time required to rebuild the Tor circuits after each hand-off negatively impacts client performance. This research is the first to highlight, and quantify the extent of this weakness within the design of Tor (and its underpinning onion routing) in supporting mobility efficiently.

2. A new and more effective, metric (q-factor) to measure the trade-off between anonymity and performance in dynamic network topologies.

The increased network churn generated by mobility not only impacts negatively performance but can also affect anonymity. By applying the new q-factor metric, not only gains in performance are measured, but its effect on anonymity is now also considered when evaluating mobility solutions. During the experiments, the longitudinal nature of q-factor is shown to help the operator manage the network more efficiently, while also providing the ability to make better strategic design decisions.

3. An original application of a Kaplan-Meier estimator used in adaptive client throttling for supporting mobility.

Adaptive client throttling was previously proposed to mitigate the impact on performance of bulk downloads on the Tor network. If parity is reached between the number of mobile and desktop Tor connections, the increased network churn and additional congestion, could potentially generate similar performance issues. Existing algorithms used in medicine to estimate the survival rates of patients, such as the Kaplan-Meier estimator, have previously been successfully applied to network connections. However, this is the first known application of the Kaplan-Meier estimator to predict the likelihood of a network hand-off, based on a user's recent mobility history, and adapt the level of client throttling accordingly. This novel use of the Kaplan-Meier estimator is found to the most effective adaptive throttling scheme for supporting mobility by reducing congestion, while also maintaining reasonable client performance.

4. A novel application (mBridge) of an existing solution, bridge relays, used for circumventing the blocking of Tor connections, to also support mobility.

The previously identified lack of persistence of connection to the Tor network, when breaking connections to the Internet, and consequently the time required to rebuild the circuits, suggests the existing design of Tor (and its underpinning onion routing) cannot support mobility efficiently. The use of an existing feature of Tor (bridge relays), in maintaining a persistent connection while mobile, is found to provide the best overall performance for supporting mobility.

1.4 Outline

A summary of the remaining chapters is as follows:

Privacy Enhancing technologies and anonymity networks (Chapter 2)

This chapter explores the existing literature on privacy enhancing technologies and anonymity networks, such as Tor, and the known attacks against them. The literature review also discusses the measurement of anonymity and performance and the relationship between them. The initial aim of the research is to identify, as a gap in knowledge, whether there is potentially a new and emerging issue. In this particular case, whether there is a weakness within the design of Tor (and the underpinning onion routing) in supporting mobility efficiently. For example, a mobile Tor user may access the Internet from a range of wireless networks and service providers and, if a hand-off occurs, whether this requires the Tor circuit to be rebuilt. If this is the case, it is predicted mobility will negatively impact client performance for mobile Tor users while roaming. To test this hypothesis in Chapter 3, the Tor anonymity network is selected as the initial focus of this research for the following reasons:

1. Tor is a popular and (most) widely used low-latency anonymity network.
2. The code is open source and therefore available for review, if required.
3. An Android application, Orbot, supports Internet connections through Tor.
4. Tor (and privacy) is a ‘hot’ topic and actively researched within academia.

The impact of mobility on performance (Chapter 3)

Based on the review of onion routing and Tor, it is predicted, but not yet proven, that there will be a negative impact on client performance from mobility. In Chapter 3, by applying a range of approaches: field experimentation, network simulation, and mathematical modelling while a mobile Tor user is roaming, the time required to rebuild the Tor circuits after each hand-off, negatively impacts client performance. This research is the first to highlight empirically the negative impact of mobility. Also, as onion routing and consequently Tor were originally designed for static wired Internet connections, the anonymity metrics have evolved accordingly. On reviewing the existing approaches to the measurement of anonymity and performance, weaknesses in the metrics used for low-latency anonymity networks are also identified. A new metric (q-factor) is proposed to measure the trade-off between anonymity and performance over the duration of a communication session. A range of solutions, including the new metric, are constructed based on this analysis and evaluated in the subsequent chapters.

Measuring anonymity and performance in dynamic network topologies (Chapter 4)

In Chapter 4, through network simulation and mathematical modelling, the q-factor metric is presented and proven to be more effective at monitoring low-latency anonymity networks by identifying critical events, in particular for anonymity, from the increased network churn. At the mid-point of this research, to avoid excluding interest from development other than Tor, a generic low-latency anonymity network is used as the case study. Two technical solutions to the problems of managing mobility in a low-latency anonymity network are proposed. The first is the use of adaptive client throttling to mitigate the impact of mobility and the other is providing a persistent connection to the anonymity network for a mobile user. The effectiveness of the proposed solutions are measured in chapters 5 and 6 using the q-factor metric.

Throttling the impact of mobility (Chapter 5)

Adaptive client throttling has been previously proposed to mitigate the impact on performance of bulk downloads on the Tor network. If parity is reached between the number of mobile and desktop Tor connections, the increased network churn and additional congestion, could potentially generate similar performance issues. A range of throttling schemes are evaluated to mitigate the impact of mobility, that is, to reduce congestion, while also maintaining reasonable client performance. The previously proposed schemes are found unsuitable for mobility. On further investigation, algorithms used in medicine to estimate the survival rates of patients, such as the Kaplan-Meier estimator, can also be successfully applied to network connections. By adapting the level of client throttling to the likelihood of hand-off, based on a user's recent mobility history, the novel Kaplan-Meier approach is identified as empirically the most effective adaptive throttling scheme for supporting mobility.

mBridge: persistent connections for mobility and anonymity (Chapter 6)

The use of adaptive client throttling offers one possible solution to mitigating the impact of mobility. However, the key issue remains the extended time to recovery after each network hand-off. Chapter 6 explores the novel application of an existing solution (bridge relays), used for circumventing the blocking of Tor connections, empirically provides the best overall performance in supporting mobility. By using mBridge, the mobility of users can be supported more efficiently by providing a persistence of connection to the Tor network while roaming. However, the delicate balance between anonymity and performance is illustrated with the mBridge solution. Although the mBridge solution offers better performance than the adaptive client throttling it is more vulnerable to certain types of attack.

Finally, in **Chapter 7 (conclusions and further work)** it is concluded low-latency anonymity networks, such as Tor, need to review their current design to support the increasing mobility of Internet users. A strength of both of the solutions proposed in this research, the Kaplan-Meier estimator based adaptive client throttling and Mobile IP / bridge relays (mBridge), is that changes are client-side and therefore the existing algorithms and protocols of the interior Tor network are unaffected. However, a bespoke anonymity and mobility protocol, as speculated with mDissent, may be required as part of a longer-term strategy. The new q-factor metric, also presented for the first time in this work, will help developers and operators of low-latency anonymity networks make more informed design and real-time network decisions in the future, and achieve the goal of supporting anonymity and mobility.

Chapter 2

Privacy Enhancing Technologies and Anonymity Networks

2.1 Privacy and technology

A privacy enhancing technology (PET), as the name suggests, helps maintain an individual's right to privacy [7]. PETs include techniques such as cryptography, or tools to circumvent censorship such as proxies.

A widely used implementation of cryptography is Hypertext Transfer Protocol Secure (HTTPS) that provides secure communications over the World Wide Web (WWW or the Web) [38]. HTTPS applies transport layer security (TLS) to encrypt the data transmitted between a user (Alice) and the destination, as an example, Bob's website.

A proxy, to act on behalf of another as the general term implies, is an intermediary service that receives requests from a client and forwards them on to the destination. On receiving a reply, the proxy sends the response back to the client. Proxies, such as reverse proxies, are used for load balancing web servers to maintain performance [39]. Proxies are also commonly used for monitoring and filtering Internet communications but, coincidentally, can also act as a circumvention tool [40].

Web proxies can help provide Alice with anonymity on the Web. For a website based proxy service, often only the website address (Unified Resource Locator (URL)) is required [41]. A non-website based Hyper Text Transfer Protocol (HTTP) proxies require Alice to configure her connection within the web browser network settings [42]. Socket Secure (SOCKS) proxies are similar to HTTP proxies, however, they also allow other Internet applications such as email through a secure proxy interface for example, localhost [43].

A virtual private network (VPN) is another way for Alice to circumvent Internet censorship. The VPN also acts like a proxy by establishing a virtual point-to-point connection on the Internet through network tunnelling [44].

A proxy or VPN can offer a degree of anonymity by masking Alice's 'real' external IP address. These proxy services have a number of weaknesses. If the IP address of the proxy can be detected, simply blocking the address is enough to prevent the circumvention. Also, a proxy usually only adds a single step ('hop') for masking Alice's connection details. This information may be readily available to an adversary (Charlie) on the proxy's server logs. Even if the operator of the proxy claims that no logs are kept, Charlie could instead retrieve the information from the proxy's own Internet service provider (ISP). By correlating the timings between Alice's connection (and specific traffic), the proxy server, and on Bob's web server logs, Charlie may gather enough evidence to implicate Alice. A final category, 'onion routing', most notably implemented by the Onion Router (Tor), will be explored in more detail during the remainder of this chapter.

2.2 Anonymous communications

To support privacy, in particular the anonymity component, many anonymous communications' systems have been developed [45]. Anonymous communications' systems, like any other system, have design trade-offs [46]. Chaum's mix network design helps prevent traffic analysis [47]. The mix network is suitable for high-latency anonymous communications, such as email, achieving near perfect anonymity. A number of mix network based anonymous communications' systems have been implemented, such as Mixminion, Babel, and Mixmaster [48] [49] [50]. Although, considered highly reliable and secure, the original mix network design is considered unsuitable for low-latency communications such as browsing the World Wide Web (WWW or the Web). Early development on low-latency anonymous communications, such as MorphMix, had mixed success [51] [52] [53].

Researchers at the Naval Research Laboratory (NRL) first presented 'onion routing' in 1996 [54]. In contrast to high-latency anonymous communications' networks, such as Mixminion, the prototype provided support for Hypertext Transfer Protocol (HTTP) proxies, potentially offering anonymous browsing on the Web. Due to a lack of funding, and the departure of key researchers, research on onion routing was suspended and the prototype network shutdown in 2000. Independent research on onion routing continued, first without financial support, and then in 2001, receiving funding from the Defense Advanced Research Projects Agency (DARPA). The aim of the project was to develop a workable low-latency anonymity network based on onion routing. In 2002, work on the first attempt was abandoned, however, the second attempt culminated in the launch in 2004 of Tor, officially presented as “Tor: The Second Generation Onion Router” [9]. The development of Tor continued with funding primarily from DARPA until the end of 2004. It was temporarily supported by the Electronic Frontier Foundation (EFF) [55], and finally becoming a 501(c)(3) non-profit organization (“The Tor Project”) at the end of 2006 [56]. Over 10 years since its launch Tor remains in active development. Tor has received financial and technical support from a wide range of sources including: DARPA, EFF, Google, Microsoft, National Science Foundation, Privacy International, and academic institutions such as the University of Cambridge and Massachusetts Institute of Technology (MIT) [57].

2.3 Tor anonymity network

The Tor anonymity network is one of the most popular and widely used anonymous communications’ systems [58]. Tor is free to use, open source, and has over two million daily active users [33]. The network has over 6000 volunteer-run routers relaying anonymous Internet traffic. Tor is a “distributed overlay network designed to anonymize TCP-based applications like web browsing, secure shell and instant messaging”, or more simply, a low-latency anonymity network [9]. As previously mentioned, Tor uses a technique known as onion routing, which, itself is based on Chaum’s original mix network design [47] [54]. The Tor anonymity network is available through a number of methods but the Tor Browser is a popular and user-friendly installation for desktop with multi-platform support [59].

There are a number of key differences between Tor and other proxy solutions. Firstly, the connection from the Tor client (used by Alice), to a destination such as Bob’s blog (website), in addition to the routing on the Internet, travels along multiple hops across the Tor network, as shown in Figure 2.1. A web proxy is usually single-hop, whereas, Tor is a multi-hop proxy service. Secondly, rather than a single layer of encryption, such as Hypertext Transfer Protocol Secure (HTTPS) implemented with a secure web proxy, Tor applies multilayered transport layer security (TLS) encryption. Consequently, the resulting Tor messages resemble an ‘onion’, as illustrated in Figure 2.2, hence the original name “The Onion Router”. The relationship of the onion alongside both the Open Systems Interconnection (OSI) model and TCP-IP stack can be found in Figure 2.3.

To browse the World Wide Web (WWW or the Web) anonymously, Alice first requests a list of available routers from one of the Tor directory servers. A list of the current Tor routers is also publicly available [60]. Once the file containing details, such as Internet Protocol address and public key information, of all the Tor relays is received, Alice attempts to build a circuit path. To help maintain security, *only* the Tor client can actually choose the routers to use for a circuit. The algorithm used for path selection is not, however, entirely random.

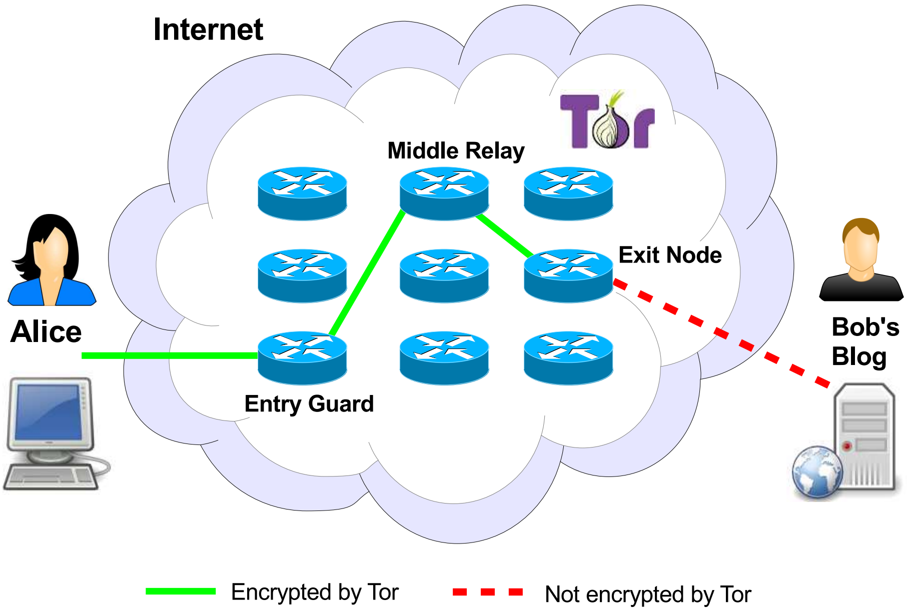


Fig. 2.1 The basic architecture of the Tor network showing both the encrypted and unencrypted links

A number of factors, for example, the reliability and available bandwidth of the Tor relay are taken into account [61]. The path for each new circuit is selected before being built, with the exit node first, followed by the other routers in the circuit. All paths generated must adhere to all the following rules, including [62]:

1. No router in the same ‘family’ as another in the same path can be used. Two routers are in the same family if each one lists the other in the family entries of its descriptor.
2. Not to choose relays that *may* belong to the same operator (i.e., not declared as ‘family’), such as a relay is in the same /16 subnet.
3. The first node must be a designated ‘trusted’ node, that is, an entry guard or bridge relay.
4. Honour the torrc client configuration that may have some user-specific settings about which exit relays not to choose.

Alice initiates a path by sending a message (*control cell*, *create circuit*) to the first router [9]. By applying Diffie-Hellman key exchanges, a session key is generated between the Tor client and the entry guard. The circuit is then extended, one router at a time (‘hop’), incrementally

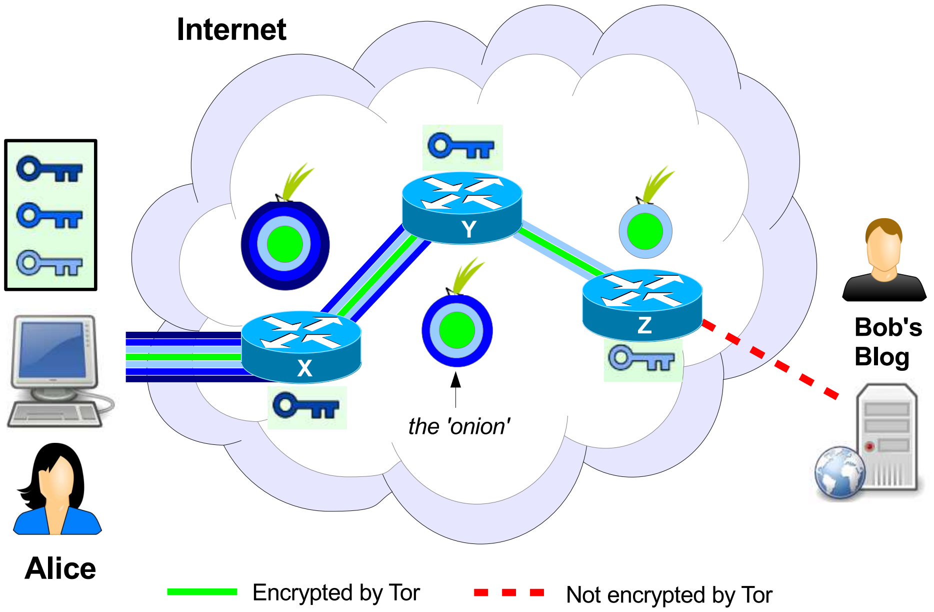


Fig. 2.2 A schematic of a circuit within the Tor network alongside the underpinning ‘onion routing’

extending (*control cell*, *extend circuit*) twice further (3 routers in total†), with the established session keys for the previous hop, as shown in Figure 2.4. This technique is also known as ‘telescoping’ as illustrated conceptually in Figure 2.5 [63].

The structure of the Tor onion can be notated as follows: for the core message M sent from Alice (A) to Bob (B) via routers X, Y, and Z, it leaves the client (A) entering the Tor network at the entry guard (X) as $EX(Y\|EY\ (Z\|EZ(B\|M)))$. The entry guard (X) peels off the outermost layer of encryption (E), determines that router Y is the next hop, and forwards the remaining message ($EY\ (Z\|EZ(B\|M))$) to Y. This process is repeated until the core message (M) arrives at its final destination (B), as shown in Figure 2.6. If Bob’s website also uses a secure connection (e) for example, using Hypertext Transfer Protocol Secure (HTTPS), the resulting traffic leaving the exit node (Z) would be $B\|eM$.

Once the circuit build process is complete Tor is available for browsing the Web ‘anonymously’. The time to build a circuit, in theory, is approximately 3 seconds, but due to network conditions, often longer in reality. If Alice wants to retrieve a page from Bob’s website via Tor, the Hypertext Transfer Protocol (HTTP) GET request is encapsulated within the encrypted layers containing the relevant session keys for each router along the circuit, creating the onion. The onion is forwarded through the Socket Secure (SOCKS) proxy

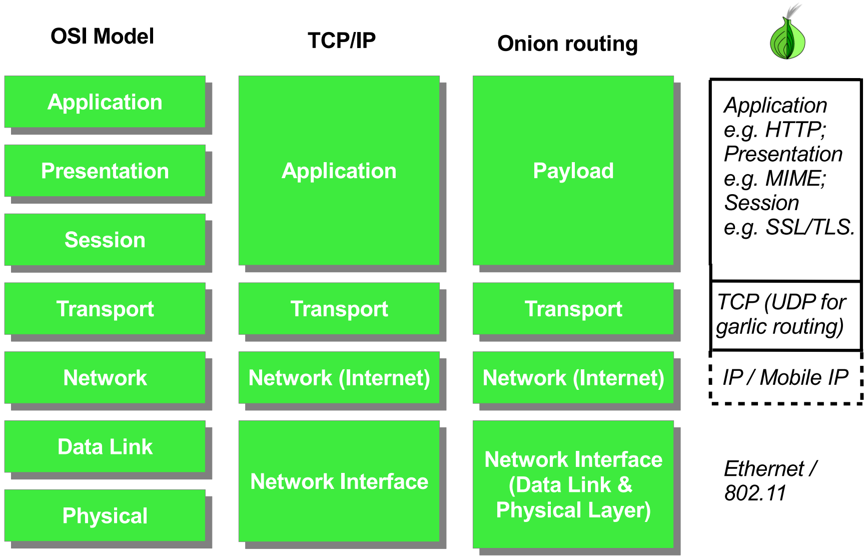


Fig. 2.3 The relationship of the Tor 'onion' layers with both the Open Systems Interconnection (OSI) model and Transmission Control Protocol (TCP) and Internet Protocol (IP), TCP-IP stack

interface (localhost), and then relayed by one of the available circuits, as data streams, via multiplexed Transmission Control Protocol (TCP) connections between paired Tor routers. It is then 'unpeeled' incrementally by the Tor router at each hop, revealing the next layer until Alice's HTTP GET request is finally exposed at the exit router and sent to, and processed by Bob's web server, as shown in Figure 2.7. As each Tor router within the path only knows its predecessor or successor and no other routers within the circuit, and the session keys also being ephemeral, this contributes to both the underpinning onion routing and consequently Tor's strength.

†A different process is used to build circuits, using rendezvous points (RVP), to access Tor 'hidden services' (.onion) [64]. Due to potential ethical issues, that is, the reported content of many of the hidden services, an early decision was made for hidden services to remain out of scope for this research [65]. It is worth noting that not all uses of hidden services are nefarious. The online social networking service Facebook has been available, to Facebook's daily active users (DAU) at an average 936 million (as at March 2015 [35]), as a .onion site since 2014 [34]. The year-over-year increase of 31% of the mobile DAU, at an average 798 million (as at March 2015 [35]), has prompted Facebook to also allow their Android application to connect through Orbot, therefore the Tor network [36].

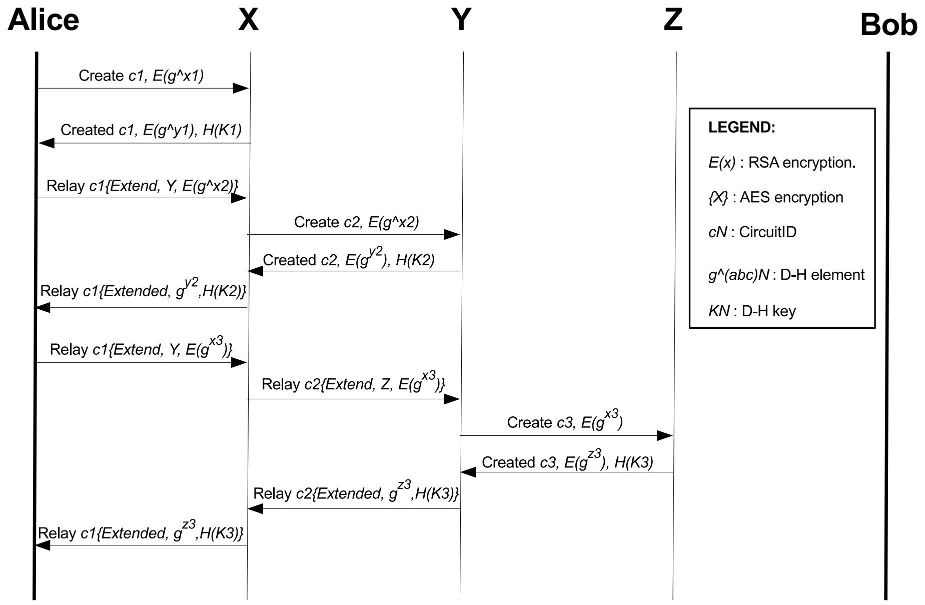

Alice
X
Y
Z
Bob
Create c1, E(g^x1)
Created c1, E(g^y1), H(K1)
Relay c1{Extend, Y, E(g^x2)}
Create c2, E(g^x2)
Created c2, E(g^y2), H(K2)
Relay c1{Extended, g^y2,H(K2)}
Relay c1{Extend, Y, E(g^x3)}
Relay c2{Extend, Z, E(g^x3)}
Create c3, E(g^x3)
Created c3, E(g^z3), H(K3)
Relay c2{Extended, g^z3,H(K3)}
Relay c1{Extended, g^z3,H(K3)}
LEGEND:
E(x) : RSA encryption.
{X} : AES encryption
cN : CircuitID
g^(abc)N : D-H element
KN : D-H key

Fig. 2.4 The Tor circuit build process where the path is extended, one router at a time (‘hop’), incrementally extending with the established session keys for the previous hop

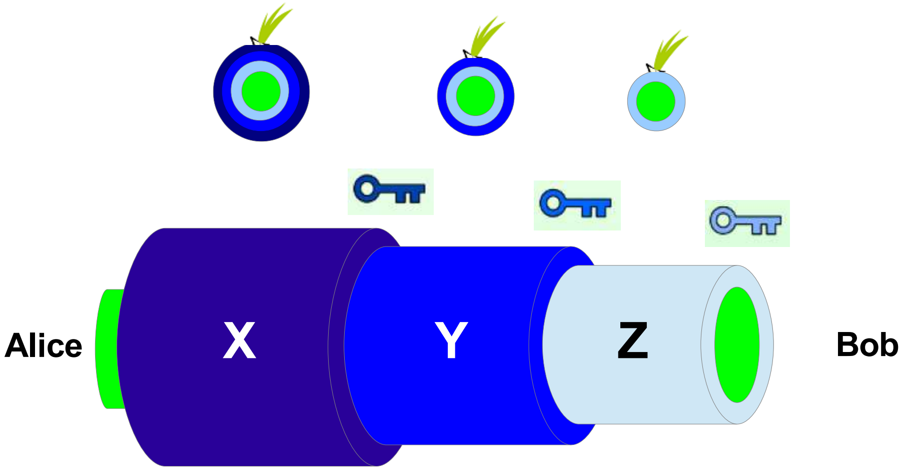
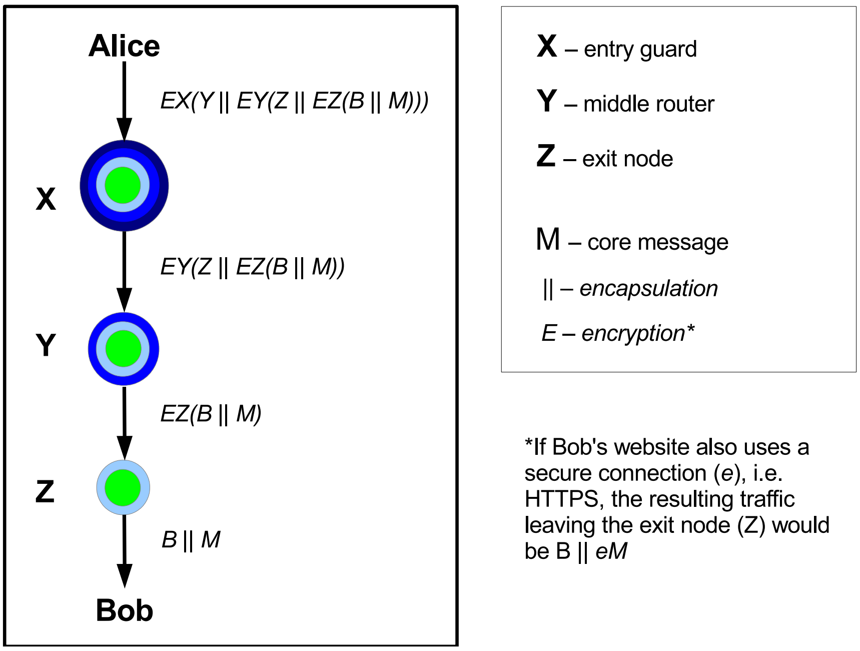


Fig. 2.5 The Tor circuit build process conceptualize from the term ‘telescoping’



*If Bob's website also uses a secure connection (*e*), i.e. HTTPS, the resulting traffic leaving the exit node (Z) would be B || *eM*

Fig. 2.6 The onion being 'unpeeled' through the Tor network

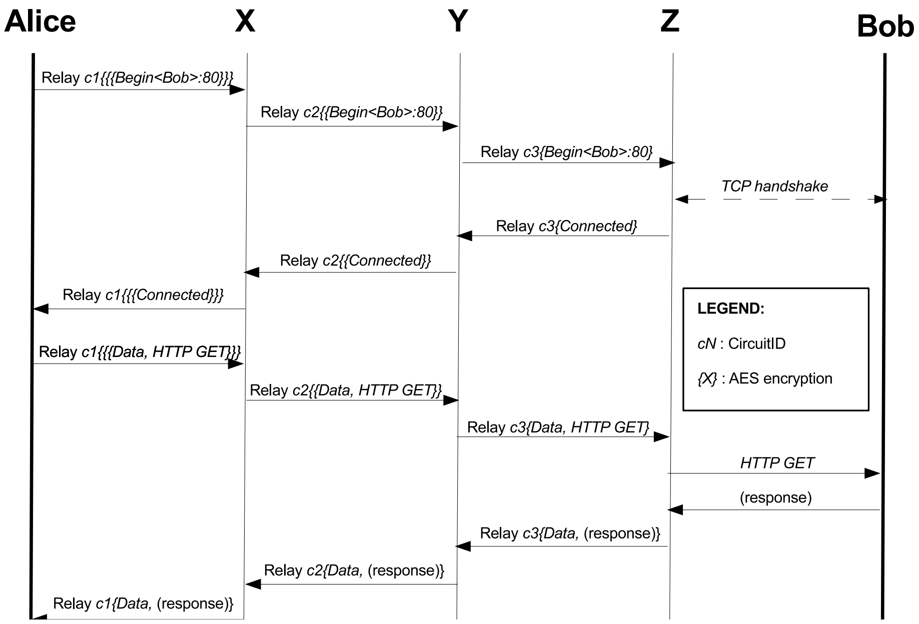


Fig. 2.7 A Transmission Control Protocol (TCP) handshake enables the transfer of Internet traffic once the Tor circuit is built

2.4 Attacks on anonymity and Tor

An initial security evaluation of both onion routing and the prototype Onion Router had already been published before Tor was released to the public [66]. The known (and predicted) limitations of Tor are outlined within the original design document [9]. These limitations, as well as any new threats, are listed on the Tor website [67]. Attacks on anonymity is a thesis in its own right but even the term 'attack' is not entirely correct. Attacks may not necessarily be a direct 'attack' by an adversary (Charlie). Issues such as poor reliability, although not malicious, can potentially be just as devastating for anonymity [22] [68]. Other external issues, such as the Heartbleed bug, can also add to the mix [69] [70] [71].

After 10 years of operation, Tor, and its use, sometimes nefarious, generates diverse opinion. In particular governments are known to have an uncomfortable relationship with Tor, and PETs in general. A prime example is the Prime Minister of the United Kingdom recently wanting to ban encryption to maintain security [72]. The irony is that encryption provides security for online shopping, banking and messaging, and without it there is no feasible digital economy. Leaked National Security Agency (NSA) documents argue that it may also be counterproductive to deter targets from using Tor [73]. Ironically, law enforcement agencies also use Tor to protect their own anonymity [74]. Tor allows them to access websites and services without leaving recognizable tracks, for example, by using a known range of government Internet Protocol (IP) addresses. It can also useful for infiltration, as in the recent takedown of the Silk Road online marketplace, where law enforcement continued to operate the hidden service on the Tor network [11] [75].

Privacy creates additional challenges in measuring the effectiveness, from attack, of anonymity networks. Accurately measuring anonymity is also a long-standing research problem. There are a plethora of anonymity metrics that can be used with differing levels of generality and complexity [76]. The well known metrics of anonymity set size, κ-anonymity, and entropy have been extensively used in wired Internet anonymity networks [77] [78] [79]. As Internet access has been extended from wired to wireless connections, the metrics have formed the basic building blocks for measuring anonymity in wireless networks [76]. However, evolving specialized metrics, such as effective anonymity set size, individual anonymity degree, and real-time anonymity, are predominately applied in wired environments [76]. Kelly et al. state these "wired-based anonymity metrics have limited applicability to mobile, wireless environments" [76]. This highlights the need for more suitable real-time anonymity metrics for the increasing mobility of users. This will be explored further, later in Chapter 4, prior to presenting a new metric, q-factor. The remainder of this chapter will outline a range of attacks from an adversary and technical perspective only.

Diaz et al. developed a simple adversary profile [77]. The model includes key characteristics of an adversary (Charlie), and the extent Charlie can control and observe the network, as part of his attack:

1. local *vs.* global
2. internal *vs.* external
3. active *vs.* passive

Firstly, whether Charlie is a *local* or *global* adversary is a key element. A local adversary, such as an individual or group may attempt to compromise Tor, for example, to receive a bounty. The resources of a local adversary may be limited both financially and technically. If global, an adversary may have the resources to take a full or 'significant' control of the system. What is defined as significant is still open to debate. Some research suggests that it is possible to compromise Tor with a relatively small number of malicious routers [80]. Other research states that if Charlie is able to observe both the entry and exit points, then low-latency anonymity networks, such as Tor, cannot defend against this [81].

Secondly, where the adversary is positioned either *external* or *internal* can provide differing options for attack. An internal attack, as previously discussed, may include controlling a number of routers within the anonymity network [80]. External attacks may include traffic analysis that only can observe Tor communications at points on the Internet for example, Alice's connection to her Internet service provider (ISP), Bob's web server connection [81].

Finally, *active* and *passive* depends on the adversary's level of interaction. Passive attacks, such as traffic analysis, over time can be highly effective [81]. Active attacks, such as being able to manipulate router selection can be even more effective [80]. The characteristics of the adversary can also be multiple for example local, *and* internal, *and* active. This is reflected by the attacks, which can be, and many are, carried out combinatorially for greater effect. Salo categorizes known attacks against Tor into five key types [82]:

1. probabilistic attacks
2. routing selection attacks
3. traffic-based attacks
4. protocol vulnerabilities
5. global-level attacks

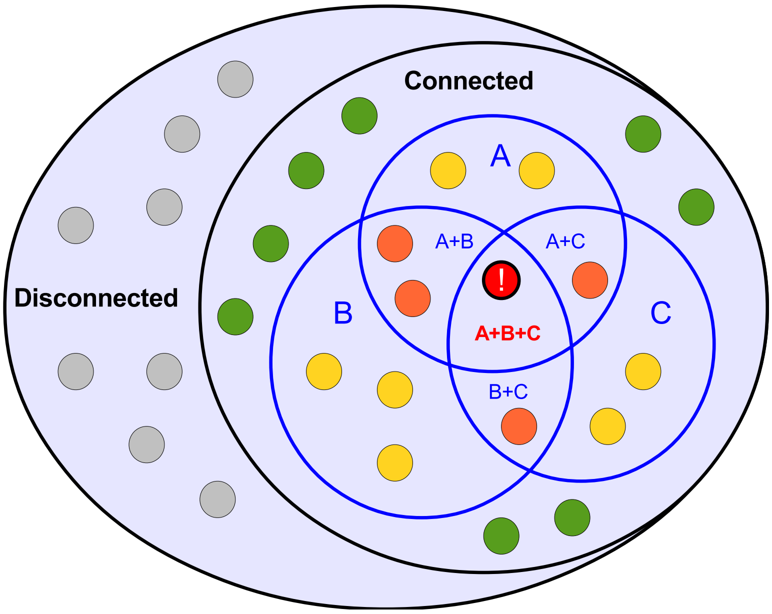


Fig. 2.8 A Venn diagram conceptualizing an attacker's approach to partitioning users into smaller subsets (A, B, C), through a series of attacks, with the aim to eventually expose a user (A + B + C)

Probabilistic attacks are based on mathematical models with the aim is to measure and provide certain information about the network [83]. This can be used for 'partitioning' type attacks where Charlie probabilistically reduces the anonymity set over time into progressively smaller and smaller subsets, until Alice is finally identified, as shown conceptually in Figure 2.8.

Probabilistic attacks can be supported by other attacks, for example, *routing selection attacks*. A denial of service on 'honest' nodes increases the probability of selection by the Tor client, Alice, of a malicious node controlled by Charlie, as shown in Figure 2.9 [22]. This in turn can support another active attack, the 'Sybil' attack, where colluding malicious nodes can control and observe Alice's circuit end-to-end, as shown in Figure 2.10 [84].

A more subtle version is to introduce traffic onto the Tor network and observe whether Alice, from a negative impact on her performance, is sharing one or more of the same relays within the circuit. This is commonly referred to as a 'congestion' attack, as illustrated in Figure 2.11 [85].

Passive attacks are generally *traffic-based* observation known as traffic analysis or 'sniffing' [86] [87]. The effectiveness of traffic analysis depends greatly on Charlie's capabilities. For example, the previous active attacks, where Charlie can influence Alice's traffic, can

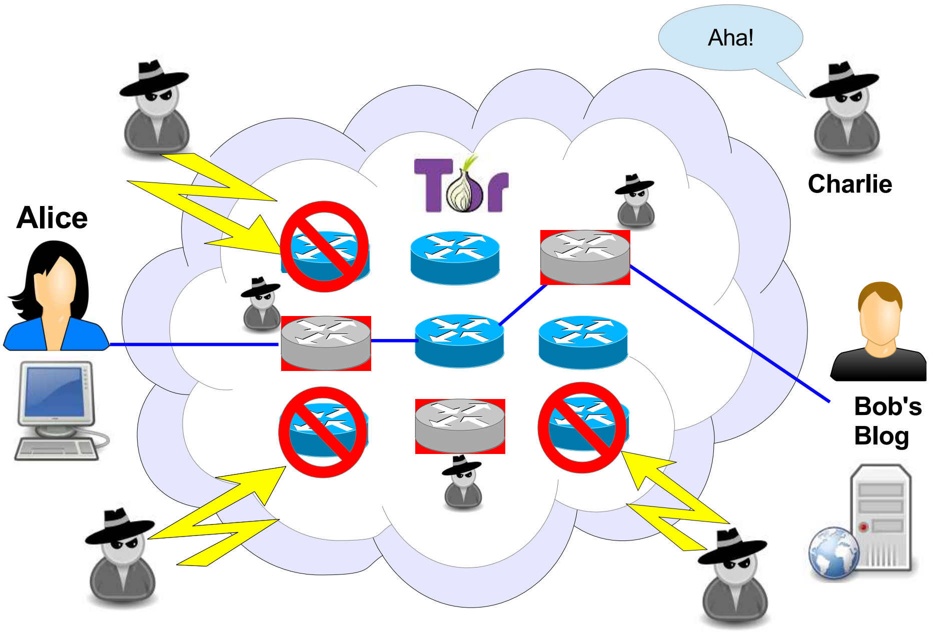


Fig. 2.9 A denial of service attack. The reduction in available honest routers for path selection, increases the likelihood Alice uses one of Charlie's malicious routers

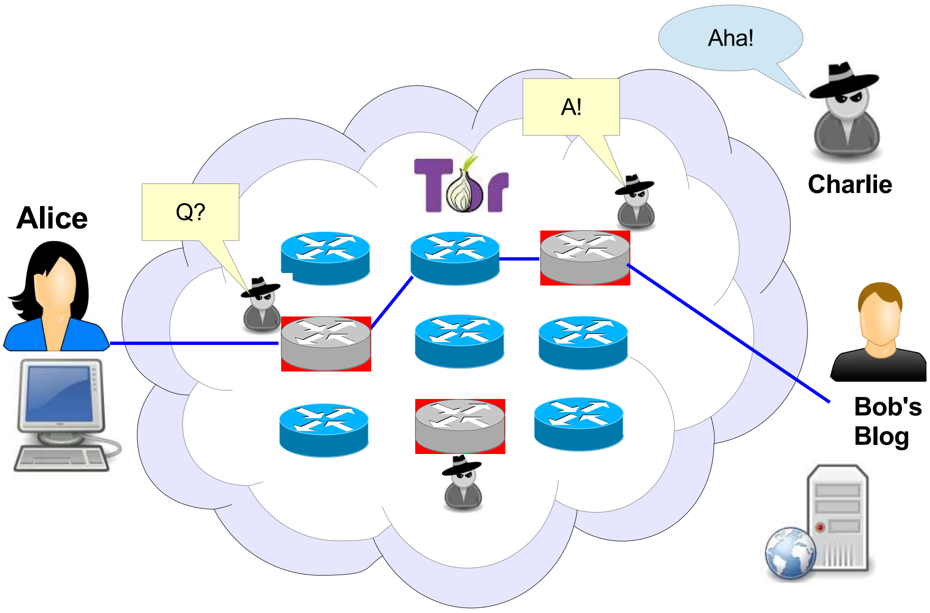


Fig. 2.10 A Sybil attack where colluding malicious nodes can control and observe Alice's circuit end-to-end

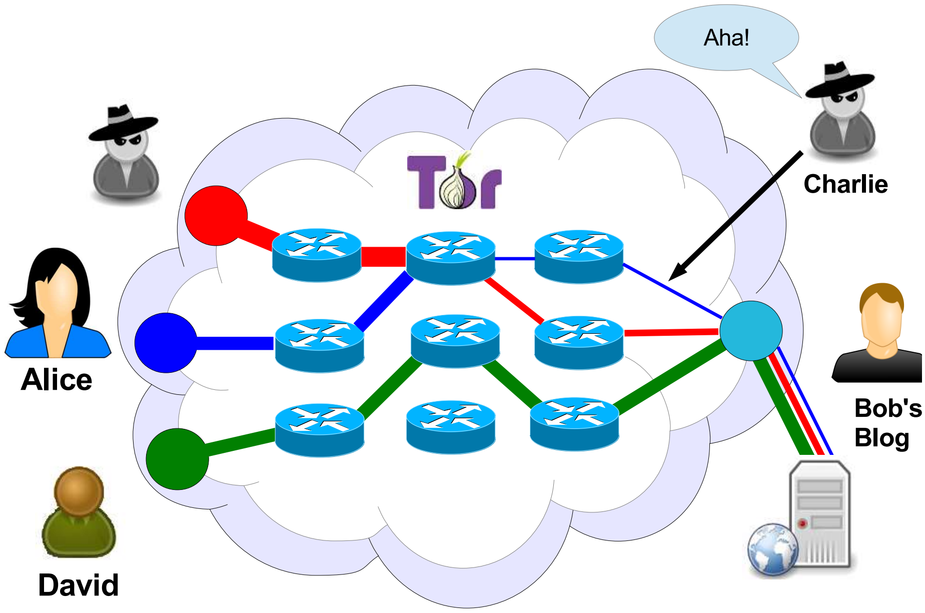


Fig. 2.11 A ‘congestion’ based attack where Charlie floods the Tor network to observe whether Alice, from any negative impact on her performance, is sharing one or more of the same relays within the circuit as Charlie

greatly reduce the amount of traffic analysis required. Timing analysis can also be used to correlate Alice and Bob’s traffic, therefore identifying a link between the sender and receiver [88]. Over time the opportunity to find a link increases, commonly referred to as an ‘intersection’ attack, as illustrated in Figure 2.12 [89].

If Charlie can observe both the entry and exit point to the Tor network he may be able to perform an end-to-end positive correlation, also known as a ‘confirmation’ attack, as shown in Figure 2.13 [90]. This attack has been observed in the wild on the Tor network [91]. If Charlie can also control Alice’s data stream, he may be able to tamper with, by ‘marking’ or ‘staining’, Alice’s traffic, akin to a man-in-the-middle (MitM) attack. By actively manipulating Alice’s traffic, this further increases the probability that Alice is the sender, linking her to Bob’s website, as illustrated in Figure 2.14. As examples, Chakravarty (2014) presented two novel approaches to actively manipulating Alice’s traffic: a) injecting deliberate traffic fluctuations, b) injecting ‘dummy’ traffic such as packets with small time-to-live (TTL) values, on Alice’s Internet connection and observing the traffic leaving the Tor network [92]. For the attack to be successful, the traffic fluctuations would require to be statistically distinct enough. If too distinct, that is, an obvious slowing down, this may cause Alice to suspect that her connection is being tampered with and take defensive measures such as resetting

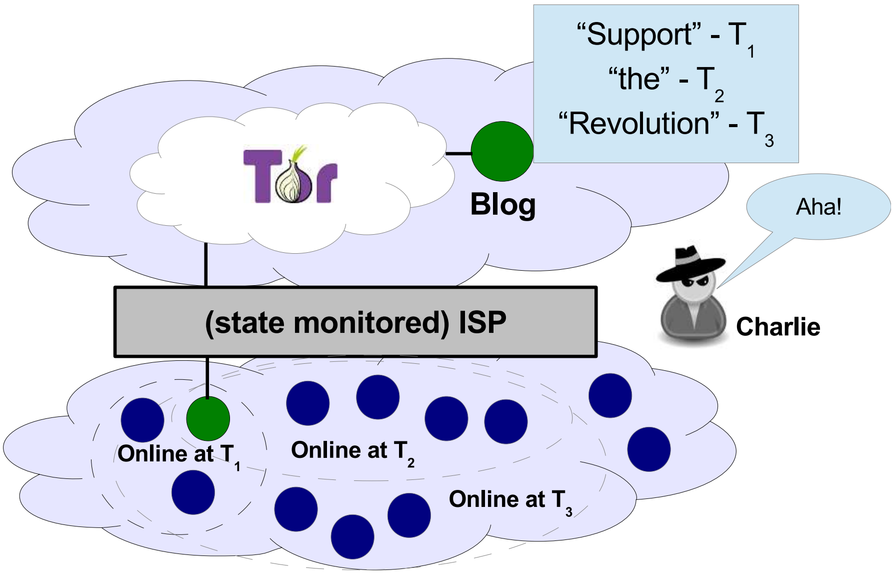


Fig. 2.12 An 'intersection' attack, facilitated by network churn, over time revealing the identity of the sender of message: "Support the Revolution"

her connection. However, with the other method of attack, the small TTL packets will be automatically discarded, therefore distort the observed network statistics, without degrading the Alice's performance.

Protocol vulnerabilities may arise from poor design decisions, coding, and testing. Existing bugs, either disclosed but not yet fixed or, worse still, a zero-day vulnerability, can significantly lower the anonymity of Alice [93]. A major bug in the software may cause Alice's Tor connection to bypass the proxy settings and reveal her real identity. The Tor software may even be fake [94] [95]. Alternatively, if Charlie, through a malicious website, is able to access Alice's Tor browser history remotely, it may reveal whether Alice has visited censored sites or her approximate location.

Finally, as previously stated, *global-level attacks* cannot be defended against [81]. The Tor project explicitly state this fact to warn its users [67]. If Charlie is a global adversary and has access to Internet Exchange Points (IEP/IXP), he can de-anonymize any given user within three months of regular Tor use, with a 50% probability, and over 80% probability within six months [96]. For some adversaries, identifying an individual as a Tor user is enough to prosecute, such as in Ethiopia, where the penalties are up to 15 years imprisonment [97].

A number of defences have been introduced to Tor to mitigate known attacks and vulner-

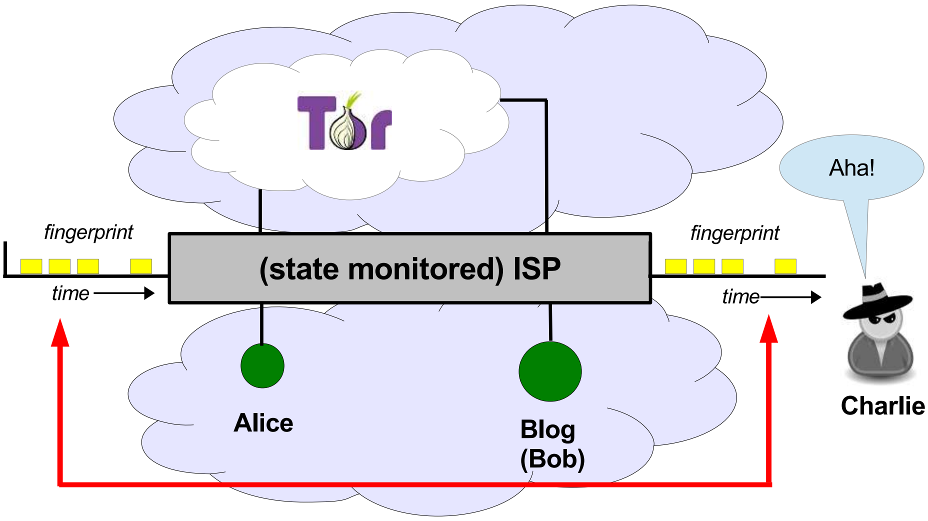


Fig. 2.13 An end-to-end correlation / ‘confirmation’ attack based on network traffic analysis

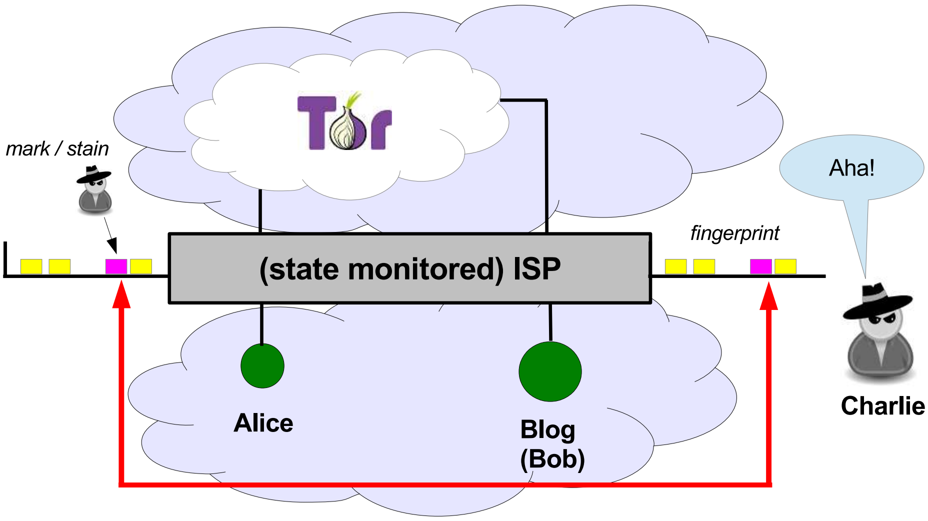


Fig. 2.14 Charlie ‘marking’ / ‘staining’ network traffic to enhance end-to-end correlation / confirmation based attacks

abilities. A few examples from the previously cited documentation, and related research, are as follows:

1. Circuits not to contain nodes from the same subnet/n range of IP addresses, unless ‘trusted’ by the Tor project(!), as a ‘honest’ family of relays. Note: This does not stop a global adversary with nodes from a larger geographical spread.

2. The introduction of ‘trustworthy’ nodes at the entry points to the Tor network. Note: Research has shown that an adversary can become a ‘trusted’ guard node in 7 days, with a stable, reasonably performing node.

3. Rebuild circuits every 10 minutes, once a circuit has been used and all data streams are closed. This aims to reduce opportunities for traffic analysis including long-term link-ability. Note: More persistent connections may be required for Internet Relay Chat (IRC), Voice over IP (VOIP), that will keep the data stream, therefore circuit, to remain open longer than 10 minutes.

4. Verifying the self-reported bandwidth of Tor relays by operators. This is to mitigate false advertisement for the purposes of attempting to control full or part circuits. Note: This could be circumvented by controlling legitimately high-performing nodes operated by a global adversary.

5. The obfuscation of Tor traffic to appear similar to Skype. Note: This may not be effective if Skype is also blocked.

6. Implementation of ‘private’ bridge-relays at the entry points of the network that are not included in the public Tor bridge directory listing, and therefore less chance of Internet Protocol (IP) address harvesting and blocked. Note: Tor’s traffic signature is detectable using Deep Packet Inspection (DPI).

7. JavaScript, originally turned off in the Tor browser, is enabled by default for better usability . . . but worse security?!

A review of the threat model is constantly required. The Tor development team maintain a working draft copy of the threat model for Tor [98]. This is important as, in a real-world implementation, the threats are continually evolving. For example, the original implementation of ‘public’ Tor bridges, aiming to circumvent censorship if Tor is blocked, has already been compromised by the Chinese government. By nefariously requesting and harvesting all public bridge IP addresses, they were able to block them at the Great Firewall of China [99]. This led to the development of ‘private’ bridges that are not available to be

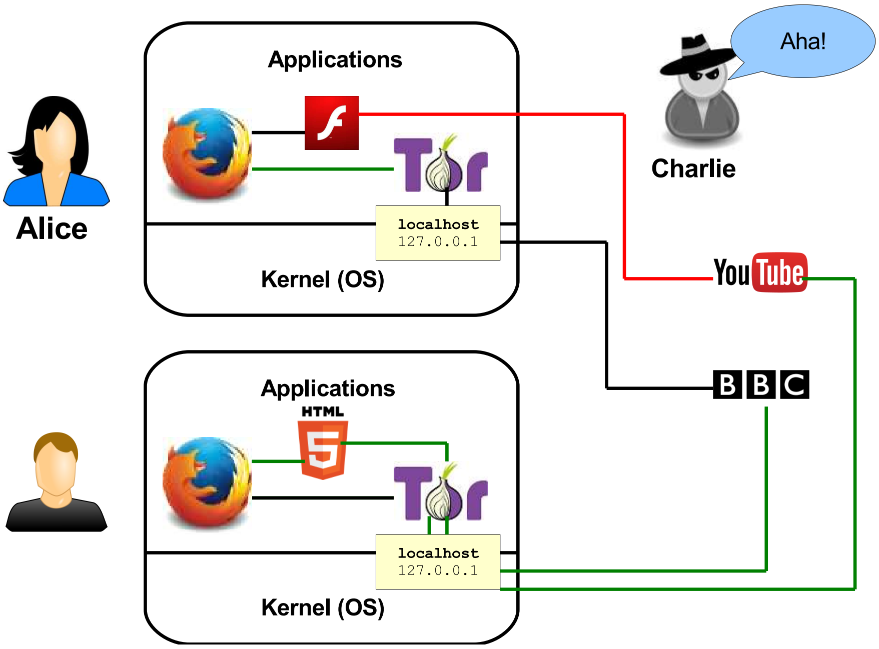


Fig. 2.15 Not all software applications honour proxy settings!

publicly requested. If the connection to the private Tor bridge cannot be detected, such as with Deep Packet Inspection (DPI), and blocked, a connection to Tor from China can be established successfully.

Although, the Tor project continually reminds its users of the limitations of using Tor, user misconceptions can lead to issues. A common misconception, although less so now, is that a Tor user's traffic is 'secure' end-to-end from the Tor browser to the website. This is an incorrect assumption. Tor traffic is unencrypted between the exit node and final destination. The unpeeling of the final layer of the onion reveals the IP header to allow routing of the core message on to Bob. However, unless Bob's web server supports Hypertext Transfer Protocol Secure (HTTPS) and Alice remembers to configure her Tor browser to enforce a HTTPS connection to Bob, the Tor exit relay will transmit as HTTP-only. A malicious exit node, operated to 'sniff' sensitive data, is able to compromise email accounts using HTTP basic authentication with a HTTP POST of username and password [100]. Other user mistakes include using Tor for peer-to-peer (P2P) file sharing applications such as BitTorrent. BitTorrent is known to ignore proxy settings of the Tor browser and create direct connections to other BitTorrent or P2P users [101].

The Tor project tries to prevent these user mistakes and misconceptions. As previously discussed, the Tor browser implements HTTPS Everywhere, developed in conjunction with the Electronic Frontier Foundation (EFF), to automatically enforce full end-to-end encryption for the user, that is, both inside and outside the Tor network, but only if supported by the destination webserver [102]. The Tor browser also blocks the use of Adobe Flash and the Java Runtime Environment (JRE), to prevent proxy settings being bypassed, but allows HTML5 for streaming media websites such as YouTube, as shown in Figure 2.15.

These features do not mean that there is no forensic evidence left behind of the Tor usage [103] [104]. An alternative, more secure in theory, implementation of Tor is through "The Amnesic Incognito Live System" (Tails) [105]. Tails is a bootable Linux distribution that runs in a computer's memory. Tails strictly enforces all traffic through the Tor network and does not write to disk by default.

Finally, Tor can even try to hide being Tor from Deep Packet Inspection (DPI). The implementation of 'pluggable transports' can offer obfuscation of Tor usage [106] [107] [108]. For example, pluggable transports can adapt the Tor traffic between the client and a private bridge to resemble a Skype conversation [109]. However, as previously discussed, if Skype is also blocked, so will be Tor. Also, these pluggable transports can still be vulnerable to traffic analysis [110].

2.5 Anonymity and performance

As outlined in the previous section, attacks can compromise the anonymity but Tor, as with any other Internet facing service, is also exposed to 'economics', that is, the supply and demand of network resources, as shown in Figure 2.16 [111]. The difficulty experienced in maintaining this balance is often referred to as Braess's paradox [112]. Braess originally applied the theory to road networks. If more capacity is added for example, a new lane or bypass, traffic can actually become worse due to opportunistic user (driver) behaviour. This principle has also been applied to anonymous network communications [113]. Therefore, not only malicious intent, such as denial of service attacks, can impact a service, but poor operator and/or user behaviour can negatively impact the degree of anonymity provided. An impact on reliability can be just as devastating as a denial of service attack [22]. For example, if a software release has a critical bug, causing 50% of the Tor relays to become inoperable, this can decrease anonymity through a general reduction of available relays for Alice to build circuits. This may also overload the remaining working relays, create bottlenecks, traffic congestion, and potentially an exponential self-generation of additional traffic through re-requests for example, 'F5, F5, F5, Tor is too slow!'. If performance degrades to a level that users are deterred from using the network, the number of concurrent users may eventually offer little or no anonymity, 'Tor sucks, I'm off elsewhere!'.

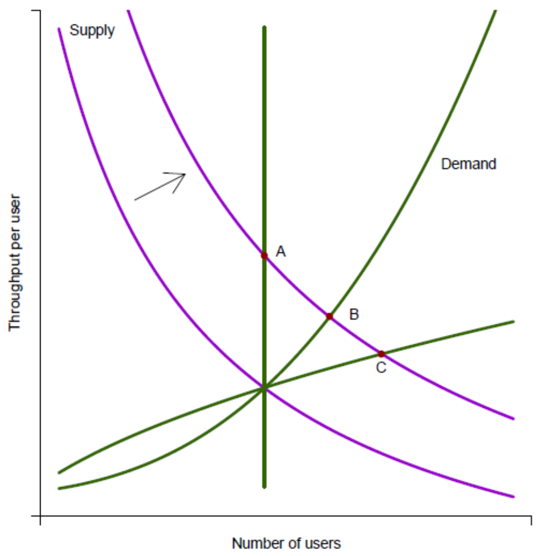


Fig. 2.16 The supply and demand of Tor network resources. *A* corresponds to no increase in users, whereas points *B* and *C* represent more users connecting to use the new capacity [111]

The relationship between security and usability, within computing, is referred to as Anderson’s rule [114] [115]. The operator of an anonymity network have to make design and operational decisions based on maintaining a degree of anonymity (security), while also providing an acceptable level of performance (usability). Therefore, for every decision, an assessment of anonymity versus performance is required.

As with existing, standard, approaches to computer network management to provide the best overall network performance, effective traffic flow control management is essential. The Tor network is an ‘open’ network operating over 6000 routers, 3000 public bridges, unknown private bridges, transporting an average of 60 gigabits per second (Gbit/s) for over two million daily users [33]. Tor uses the ‘reliability’ of the Transmission Control Protocol (TCP) as its transport layer [9]. Also, to prevent congestion, Tor relays internally operate input / output (I/O) data buffers to manage traffic, on a first in, first out (FIFO) arrangement. Data streams are selected for relaying using a standard round-robin approach across the different circuits. The design of Tor may actually degrade performance. It is found Tor does not always keep a steady traffic flow of data in-flight [116]. If a file is downloaded via a circuit consisting of a 10 Mbit/s entry and exit routers and a 128 Kibit/s middle router, the exit router can read data from the web server faster than writing to the outgoing connection with the middle router. This results in some circuits having to wait for a slower data stream to complete.

Poor traffic flow, on the Tor network, is compounded by user behaviour. An increased use of Tor for bulk downloads is negatively affecting performance [101]. A number of solutions have been proposed. These include adaptive circuit prioritization for 'burstier' circuits or the migration to a 'lighter' User Datagram Protocol based transport (μTP) [117] [118] [119] [120] [121].

The current architecture of Tor and the use of TCP as its transport layer, is arguably more suited for stable wired networks [122]. Mobile users, such as Alice, connecting to the Tor network from mobile Internet connections may potentially also generate similar congestion issues ('circuit clogging') as the P2P and bulk download traffic [123]. If Alice is roaming and hands-off between service providers, or even different networks of the same provider, any half-open TCP connections and data remaining in-flight, may not only impact Alice but also cause increased congestion on Tor network.

The potential impact of mobility on, low-latency anonymity networks, such as Tor, is not yet fully understood. Failure to identify and address these issues may pose a threat to the future of an anonymity service [124].

2.6 Measuring anonymity and performance

Privacy creates additional challenges in measuring the effectiveness of anonymity networks [125]. One reason is that user feedback and technical error reporting may be less forthcoming in contrast to 'normal' Internet services. As a result, network-based metrics are often relied on to measure overall performance rather than user data [33].

As previously outlined, one key factor in maintaining anonymity and performance is the circuit path selection. Early onion routing prototypes selected routers uniformly at random [66]. As the popularity of Tor grew, load balancing was introduced to maintain a fair distribution of bandwidth for users. Tor weights each router in proportion to its perceived bandwidth capacity. This weighting was originally self-reported but latter concerns a malicious router operator can use this to attract users to compromise circuits caused a change in approach [61] [126]. Tor now actively probes each Tor router to estimate its actual capacity [127]. Tor takes this measure to help assign each relay a weighting based on the bandwidth available. The observed available bandwidth is then used for path selection, during circuit building, to distribute load toward relays with the available network resources [61] [126]. The current load balancing algorithm provides a simple trade-off between performance and anonymity [128]. More refined techniques, such as the MATor framework, are proposed to maintain anonymity on the Tor network. For example, the MATor framework assesses how anonymity is affected on a user-level over time by Tor's path selection algorithm [129].

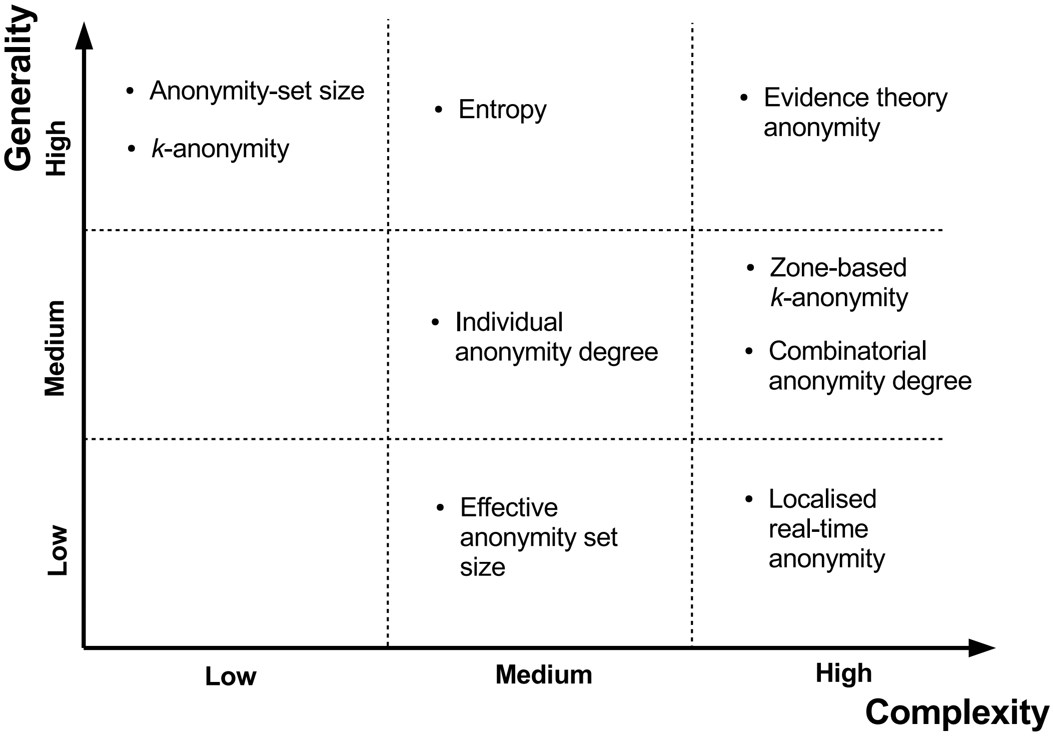

Generality
High
Medium
Low
• Anonymity-set size
• *k*-anonymity
• Entropy
• Evidence theory anonymity
• Individual anonymity degree
• Zone-based *k*-anonymity
• Combinatorial anonymity degree
• Effective anonymity set size
• Localised real-time anonymity
Low
Medium
High
Complexity

Fig. 2.17 Different types of anonymity metrics with their levels of generality and complexity, as defined by Kelly et al. [76]

However, Tor still continues to experience poor performance [111]. A number of factors that contribute to Tor's performance problems include inter-circuit interference due to TCP's congestion control, suboptimal flow control at the application layer imperfect load balancing, which causes lower bandwidth routers to handle too much traffic [116] [130]. Those users who require high anonymity would be better having the circuit paths distributed uniformly across all the router. However, those users for whom improved performance can be balanced with a lower (not low) level of anonymity, they would prefer high-bandwidth routers. However, the inflexibility of the current algorithm is "by using the same path selection algorithm for both of these, the Tor router selection algorithm sacrifices the needs of both classes" [61]. Other recent work has also suggested selection algorithms that incorporate users' trust to help maintain the anonymity and performance balance [131]. However, measuring anonymity alone provides challenges, and as shown in Figure 2.17, there are a plethora of anonymity metrics that can be used with differing levels of generality and complexity.

The well known metrics of anonymity set size, κ-anonymity, and entropy have been extensively used in wired Internet anonymity networks [77] [78] [79]. As Internet access

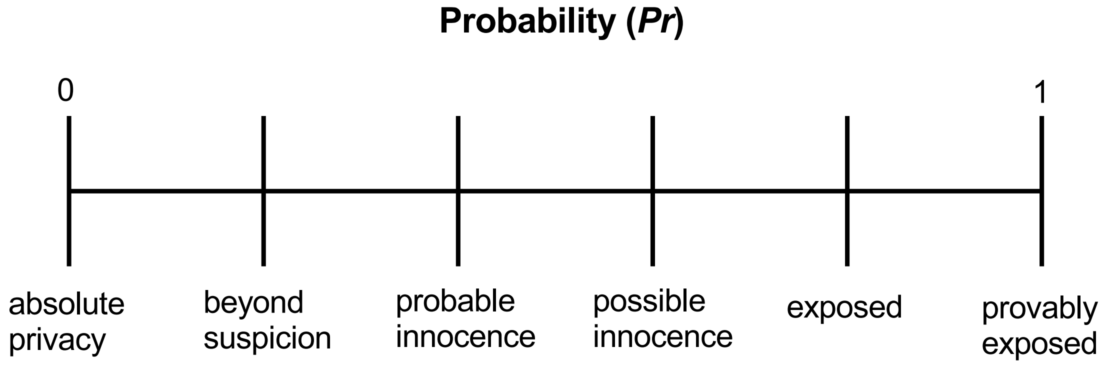


Fig. 2.18 An anonymity scale, proposed by Reiter and Rubin, in which the degree of anonymity depends on the probability that an agent is engaged in some communications' event [132]

has been extended from wired to wireless connections, the metrics have formed the basic building blocks for measuring anonymity in wireless networks [76]. However, evolving specialized metrics, such as effective anonymity set size, individual anonymity degree, and real-time anonymity, are predominately applied in wired environments [76]. Kelly et al. state these "wired-based anonymity metrics have limited applicability to mobile, wireless environments" [76]. This highlights the need for more suitable real-time anonymity metrics for the increasing mobility of users.

Accurately measuring anonymity is a long-standing research problem. As previously discussed, PETs cover a wide range of technologies including well established data security techniques, such as cryptography, but also anonymous communications. So what is meant by 'anonymity' in relation to anonymous communications? According to the Oxford English Dictionary, "anonymity' is 'the condition of being anonymous, anonymous being "not identified by name of unknown name" [133]. Communication is "imparting or exchanging of information by speaking, writing, or using some other medium" [133]. Therefore, data security, such as provided by encryption, can provide 'confidentiality' for 'what' is being communicated, whereas anonymity aims to make the sender, the 'who', unidentifiable.

Early research on anonymous communications focussed on electronic mail but now also includes wider communications on the Internet, especially, the World Wide Web (WWW or the Web). The Web has a client-server architecture, and therefore, often many individuals (clients) will access the same website (server). This one-to-many relationship forms the concept of what is known as the 'anonymity set'. Pfitzmann defines anonymity as the state of being not identifiable within a set of subjects, that is, the anonymity set [78]. According to

Diaz et al., although the anonymity set is widely adopted within PETs research, they suggest anonymity can be further broken down into two main categories [77]:

1. *"Data* anonymity: as with data security, whether the information that can be extracted from the data exchanged."

2. *"Connection* anonymity: refers to the identities of the sender and receiver during the data transfer."

Although, the term more commonly used now for 'data anonymity' is data 'confidentiality' or 'privacy'. Additionally, connection anonymity is also classified as either:

1. *"Sender* anonymity: an adversary cannot determine the sender of a given message."

2. *"Receiver* anonymity: an adversary cannot determine the receiver of a given message."

The simplest way to measure anonymity is using the anonymity set (AS). The degree of anonymity provided by AS is directly proportional to the size of its set, that is, the number of agents the group AS contains. For example, a user, Alice, is a member of a group AS with a size of N, with an eavesdropper observing a message originating from the group. In the absence of any further information, the probability (Pr) that Alice is the sender of the message is $\frac{1}{N}$, therefore $Pr = \frac{1}{N}$. The probability, alongside indicative descriptors, can be useful to provide a qualitative view of what the quantitative value means, as the example proposed by Reiter and Rubin shown Figure 2.18 [132].

Andersson and Panchenko, applying the Reiter and Rubin model, also assessed user anonymity (A_i) at different potential attack points on an anonymity network, such as Tor, from both a sender and receiver perspective [28]. The term *probable innocence* is where $A_i \leq 0.5$ based on an anonymity set with an absolute minimum of two senders. Another example is *beyond suspicion*, the user (A_i) appearing no more likely than any other user within an anonymity set of (S), that is $S = (A_i \ldots)$ being linked to the particular message $A_i = max(A_i \ldots)$. The full table is available in Andersson and Panchenko (2007) [28].

The AS model, however, has its critics. Diaz et al. state although the AS model is useful to get an idea of the anonymity provided, it does not provide information on how distinguishable any user is within the anonymity set [77]. The metric only determines the *maximum* degree of anonymity achievable if the eavesdropper believes everyone within AS is equally probable of being the sender of the message. Therefore, simply measuring the size of an anonymity set alone may not be enough if two senders do not have equal probability of having sent a particular message. Entropy, where revealing 'bits' of information are added to the calculation, can add weighting to the probability. An example is the use of entropy to determine the client software (web browser) originating a request, from information revealed by the 'User-Agent' header within the Hypertext Transfer Protocol (HTTP) [134]. If a particular message reveals information (bits) about the web browser used, and only 25% of the *AS* use this browser, then the members of the *AS* do not all have the same probability of sending the message. This can be extended to other information. For example, the Accept-Language HTTP header, to determine the correct language setting for a user accessing a web-page, reveals the language packs installed. How many users have a *EN-CY* (English-Cypriot) language pack installed when accessing the Internet from the University of Northumbria's external IP address?

As with the attacks outlined in Chapter 2, the research by Diaz et al. for measuring anonymity, and work published concurrently by Serjantov and Danezis, also adopt an attacker's perspective [77] [135]. For a probabilistic attack, the assessed level of anonymity is based on the information obtainable from different attack points and the resources available for an adversary, that is, 'global vs. local', 'external vs. internal', 'active vs. passive'. The approach used by Diaz et al. applies entropy (*H*) using Boltzmann's H-theorem. The entropy is based on the number of identifying 'bits' of information gathered by the adversary. This is used to probabilistically measure the degree (*d*) of anonymity provided, not anonymity (*A*) as before.

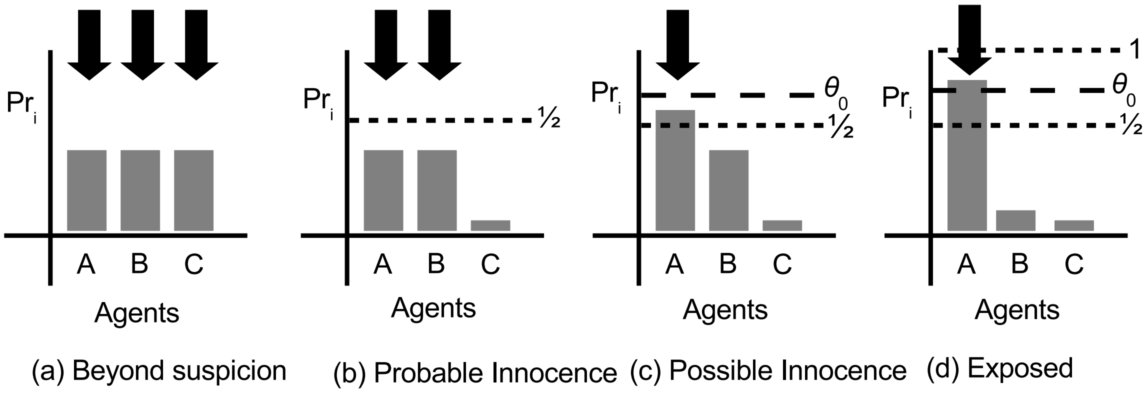
To calculate the degree (*d*) of anonymity, Diaz et al. set the pre-attack conditions. *N* is the starting anonymity set, *H* the entropy, and *max* is the maximum entropy afforded by the anonymity network prior to the attack:

$$H(max) = log_2(N)$$

A discrete random variable (X) is applied from the probability mass function $p_i = Pr(X = i)$ where *i* corresponds to an individual user, such as Alice. The variable *X* denotes the entropy gained during the attack. Therefore, $H(X)$ denotes the entropy after the attack has taken place for *N*.

$$H(X) = -\sum_{i=1}^{N} p_i log_2(p_i)$$

Once an attack has taken place, the subset of the anonymity set, which contains possible senders is $S = 1 \leq S \leq N$. Assuming the adversary cannot assign different probabilities that users belong to this subset, the probability is equally measured, therefore $p_i = 1/S$. It is also important to note that this is a 'snapshot', that is, a measure of anonymity for a particular message at a specific point in time. For each and every subsequent message, the anonymity set and subset may change, which is, highly likely in a large and dynamic anonymity network



(a) Beyond suspicion (b) Probable Innocence (c) Possible Innocence (d) Exposed

Fig. 2.19 Applying Reiter and Rubin anonymity descriptors: pre (a) and post-attack (b, c, d), to a set of users until the agent A is eventually *exposed*, adapted from Kelly et al. (2012) [76]

environment. The calculation for the individual anonymity (a) is:

$$a = 1 - \frac{H(max)-H(X)}{H(max)}$$

To maintain an anonymity of $a = 0.8$ the size of each resulting subset requires $S \geq 5$. This is often referred to as κ-anonymity [79]. κ-anonymity refers to the minimum number of agents required within the AS to maintain the required level of anonymity [136] [137].

Entropy-based metrics provide more precision in calculating anonymity compared to AS based models. This is especially true when low-level information, such as User-Agent headers, can be observed. Entropy-based metrics, however, are not always the best approach for measuring anonymity [138] [139]. Two identical d values may have a “very different qualitative anonymity” and therefore, in a real-world implementation, high entropy does not necessarily mean Alice has a higher probability to have sent the message than the rest of the anonymity set [140]. It could be that some entropy bits are more valuable than others in revealing Alice’s true identity. Some researchers believe that more combinatorial approaches for measuring anonymity should be adopted [141] [142]. Other research has also started to consider whether current approaches are still suitable and whether anonymity metrics need to measure anonymity over time [143]. Measuring anonymity in a live implementation, such as Tor, generates a number of challenges. All the models calculate the anonymity provided for a particular attack at a specific point in time. However, ‘open’ anonymity networks will have an ever-changing anonymity set. A popular anonymity network, such as Tor, may have hundreds of thousands, if not millions, of users at any time. The anonymity set may actually change while Alice's message transverses the anonymity network. The ability to calculate anonymity accurately within a highly dynamic environment is still an open issue [135] [138]. An example of anonymity dynamics is shown in Figure 2.19 and described below [76]:

1. Absolute privacy: the probability, Pr_x, that an agent $x \in$ AS sent the message is 0. This would be the case when the sending of a message was unobserved by the attacker.

2. Beyond suspicion: agent x is no more likely to have sent the message than anyone else. Also, known as total, perfect, or strongly probabilistic anonymity, as shown in 2.19(a).

3. Probable Innocence: agent x is no more likely to have sent the message than not to have sent the message. In 2.19(b), agents A and B have $Pr_A = Pr_B = 0.45$, and so are probably innocent, but C is beyond suspicion since $Pr_C = min(Pr_i) = 0.10 < Pr_A$.

4. Possible innocence: there is a non-trivial probability that an agent other than x sent the message. In 2.19(c), θ_0 is a parameter chosen to specify the threshold of non-trivial probability, and $\theta_0 > Pr_A = \max(Pr_i) > 0.5 > Pr_B > Pr_C$. As Pr_A is slightly above 0.5 but is below the threshold, θ_0, there is a non-trivial probability that some other agent sent the message, and therefore agent A is regarded as possibly innocent. Agent B is probably innocent, while the agent C is beyond suspicion.

5. Exposed: it is likely agent x is the sender of the message or $Pr_x = max(Pr_i) \geq \theta_0$. As 2.19(d) shows, agent A is exposed.

6. Provably exposed: The attacker knows agent x sent the message or $Pr_x = 1$.

The difficulty of accurately measuring anonymity, in particular within dynamic environments, has recently generated further debate. Finding the illusive 'provable' anonymity continues to drive new development within the field, such as the Dissent project [144]. Rather than passively measuring anonymity, what is particularly interesting with the Dissent approach, is mitigating actions are also introduced to help maintain provable anonymity. Gedik and Liu (2008) proposed a flexible "privacy personalization framework" to support location κ-anonymity to enable a mobile user to specify the minimum level of anonymity that they desire [145]. The Dissent model uses a more refined version of κ-anonymity. The Dissent protocol proposes two new variations of existing anonymity metrics [146]:

1. Possinymity: a probabilistic measurement of an anonymity set size motivated by plausible deniability arguments.

2. Indinymity: a 'distinguishability' metric, based on distinguishing features for adversaries to make probabilistic 'guesses'.

Possinymity is abbreviated from ‘possible anonymity’ [146]. Possinymity is the probability (Pr) an agent (x) sent the message (Pr_x). Possinymity simply captures plausible deniability based on the size of the anonymity set $Pr_x = \frac{1}{N}$. However, the passive measurement of anonymity is expanded to also maintain possinymity [146]. Maintaining possinymity aims to provide plausible deniability by ensuring an arbitrary threshold (θ_a) of anonymity is maintained throughout the communication. Due to the inherent nature of dynamic network environments, network churn may cause sudden disconnection of users and risk possinymity falling below the threshold. A simple monitoring policy could be, at the next round (interval) to compute the new possinymity that N generates. If possinymity is at, or below, the threshold, suitable remedial action is undertaken. This could vary from, in the worst case scenario, shutting down the communications’ channel completely, or some other intervention such as redirecting users to the failing channel from a high-performing channel. Applying the κ-anonymity principle, the minimum number of agents (κ) required to maintain possinymity with an anonymity threshold of $\theta_a = 0.8$ would be $\kappa = \frac{1}{1-\theta_a}$, again a minimum of 5 agents [146].

Indinymity, in contrast, is more akin to entropy. The ‘distinguishability’ aims to measure, and initiate defences against probabilistic attacks, to guarantee minimum indinymity [146]. As with possinymity, operator should ensure a minimum indinymity is maintained during the communications. If this is not the case, as previously with possinymity, then intervention is required to protect their indistinguishability. However, the overhead of intervention needs to be balance with any potential impact on performance. If the intervention is not appropriate to the risk, then the negative impact on performance and any consequent reduction in the size of the anonymity set (N), attempting to maintain anonymity may paradoxically impact anonymity. Measuring indinymity relies on how any probabilistic analysis may increase the probability that Alice sent the message. If Alice (A) is in the same set N as another user Bob (B), during an observation, from the resulting probabilistic analysis, the adversary must initially assign identical probabilities to Alice and Bob of sending the message. Therefore, every observation (i), whereby $(A = Pr_i) \equiv (B = Pr_i)$, then Alice and Fred are probabilistically indistinguishable from each other, hence equally likely to have sent the message. If a subset of N users (S) are probabilistically indistinguishable from Alice, then under the analysis each user in $S_N(A)$ has an individual probability no greater than $\frac{1}{S_N(A)}$ of being the sender of the message.

Research	Description	Gap in knowledge
Andersson and Panchenko [28]	Investigated the performance of Tor on mobile devices using different wireless network types while remaining *static*.	Question: How does movement (mobility) affect performance?
Wiangsripanawan, Susilo, and Safavi-Naini [29]	Examined Tor and mobility from a fixed *location privacy* perspective only.	Question: How does mobility (movement) and consequently roaming between networks affect performance?
Panchenko, Lanze, and Enkel [37]	Constructed a combinatorial anonymity and performance metric to evaluate routing algorithms	Question: How can this approach be enhanced to capture mobility and increased network churn over time?

Table 2.1 A summary of the key related research and the associated gaps in knowledge.

2.7 Summary

In this chapter, the background and development of privacy enhancing technologies, anonymous communications, and the Tor anonymity network is examined. Key concepts relating to anonymity networks such as the challenges of measuring anonymity and performance, as well as the critical relationship between them is also discussed. Mobility, and its potential impact on anonymity and performance, for low-latency anonymity networks such as Tor is still an open question. Previous research has attempted to address some this question and related gaps in knowledge, as summarised in Table 2.1.

Although the previous work was an important step in understanding the issues, the predicted parity of the number of mobile and desktop Internet connections and the potential impact of mobility on current anonymity network design suggests new research is long overdue.

The formal research questions outlined within the previous chapter in Section 1.2 will now be addressed in the remainder of the thesis. In Chapter 3, the concept of mobility is also introduced to address the gap in knowledge whether, and to what extent, mobility has an impact on client performance, and consequently, potentially also on anonymity. The outcome of this impact analysis should help steer the requirements for constructing a suitable solution to support mobility efficiently. However, the measurement of anonymity and performance in more dynamic environments, with higher network churn generated by mobile Tor users, also needs to be considered. Only once both these points have been addressed, the proposed solutions be evaluated.

Chapter 3

The Impact of Mobility on Performance

The Tor anonymity network is one of the most popular and widely used anonymous communications' systems. Tor's underpinning onion routing was conceived at the time of, and consequently developed for, persistent static Internet connections. Since 2010, users have been able to access the Tor network from mobile devices using the Orbot Android application. The ability to access the Tor network from a mobile device led to the following question: When accessing the Tor network while mobile, or more specifically 'roaming' between networks, does this generate a negative impact on performance? This is important to identify, not only for performance, but also its potential effect on anonymity. Poor performance may deter usage where eventually the number of concurrent users offers little or no anonymity.

Based on the previous review of the design of Tor and its underpinning onion routing, it is expected, but not yet quantified, there will be a negative impact on performance from mobility, as shown in Figure 3.1. By using a range of approaches: field experimentation, network simulation, and mathematical modelling; the time required to rebuild the Tor circuits, while a mobile Tor user is roaming, directly and negatively impacts client performance. This research is the first to highlight this negative impact and to investigate the likely extent of the impact for typical usage scenarios and mobility models.

3.1 Field study

The author's daily walk to university is a convenient, yet valid as any other, initial case study. During each walk a trace is generated. The trace is relatively linear in nature, as the example shown in Figure 3.2. The brisk 35 to 40-minute walk (3.7 km) covers five different Internet service providers (ISP) usually generating 15 hand-offs in total, as illustrated in Figure 3.3.

The initial case study highlights a number of points regarding Internet provision within the United Kingdom. One of the benefits of using Wi-Fi hotspot services, such as provided

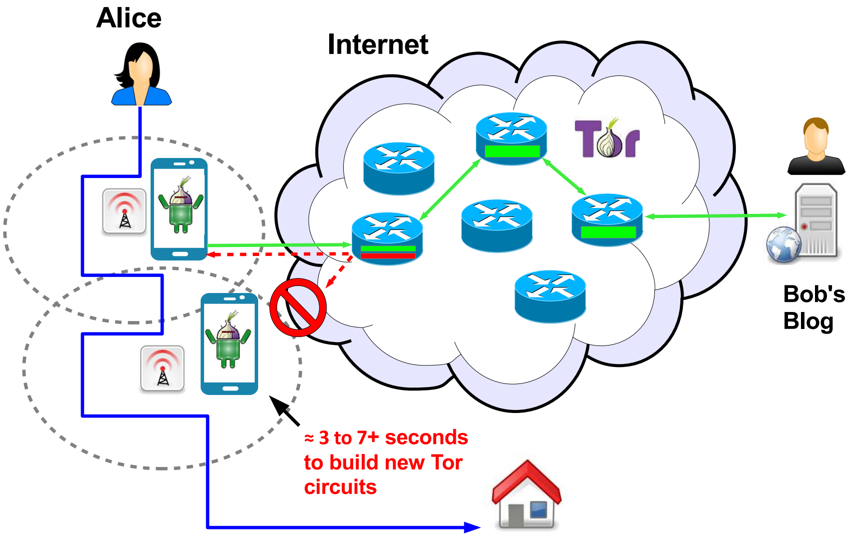


Fig. 3.1 The issue of a roaming mobile user (Alice) breaking connections to the Internet, and consequently Tor network

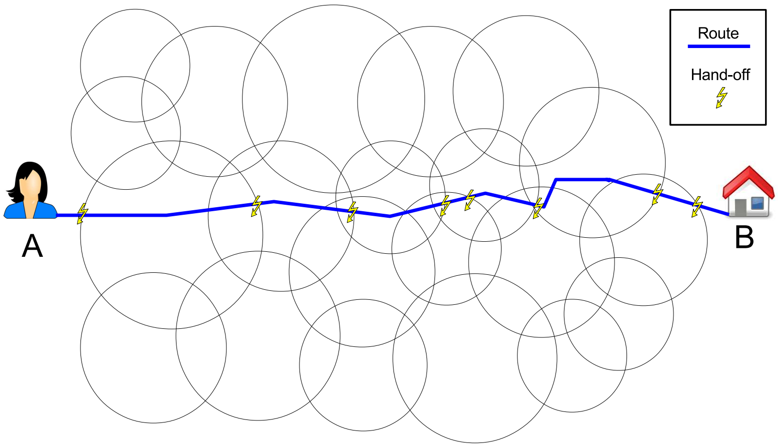


Fig. 3.2 An example trace of near-linear mobility with indicative hand-offs

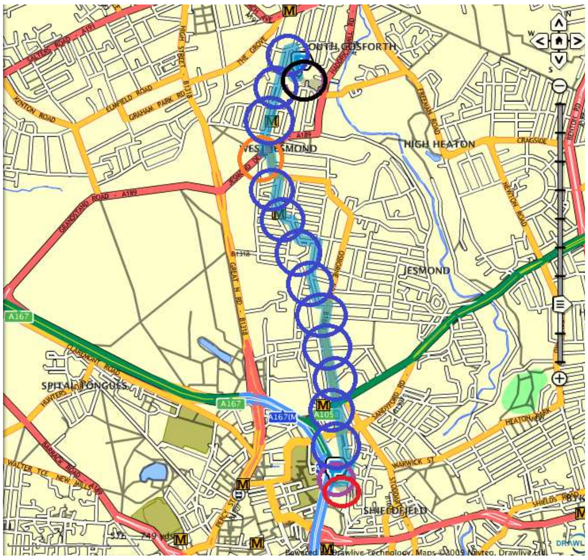
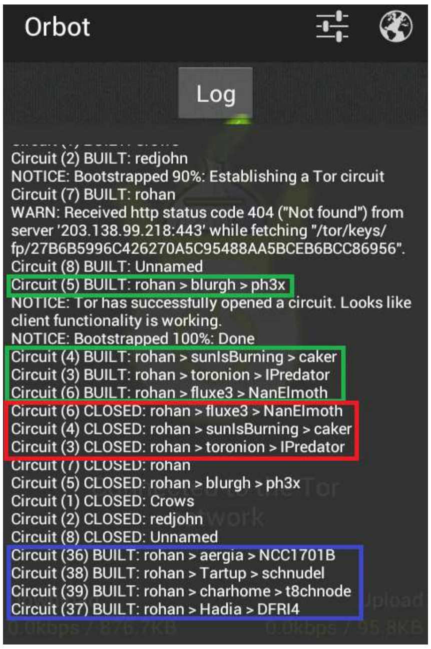


Fig. 3.3 The author’s daily walk to university with approximate location and provider of each hotspot - black: Three ‘home’ Wi-Fi, blue: BT Wi-Fi, orange: EE cellular, purple: University of Northumbria Wi-Fi, red: S3 computer laboratory Wi-Fi. Original map courtesy of walkit.com [147])

by BT, is the delivery of high performance Internet access at relatively low cost compared to cellular networks. For the over five million Wi-Fi hotspots operated by BT, each one is allocated a different external IP address [25] [26]. Hutchison 3G (trading as Three / 3), one of the largest mobile operators within the United Kingdom, tries to provide its customers with ‘seamless’ connectivity, by offering the ‘inTouch’ service [148] [149]. The inTouch service allows switching between Three’s 3G cellular network and any Wi-Fi hotspot nominated by the customer. Unfortunately, neither of these services are fully integrated. BT does not offer a cellular-based service after the sale of O2, and Three does not provide a public Wi-Fi hotspot service, only home Wi-Fi. Therefore, for both of these leading operators within the United Kingdom, neither a horizontal hand-off (between the same network) or vertical hand-off (within different networks) is achieved gracefully. Instead, the lack of integration generates a ‘hard’ hand-off leading to the unwanted dropping of voice and data calls [150]. Therefore, based on its design, it is expected the connection to the Tor network will also break causing an extended time to recovery while the Android application, Orbot, is building new Tor circuits.

Throughout this research, observations were regularly undertaken to check the operational status of the Orbot application and the indicative quality of service achievable through Tor. The first field experimentation, undertaken early in 2012, provided some surprising results.



```
Circuit (2) BUILT: redjohn
NOTICE: Bootstrapped 90%: Establishing a Tor circuit
Circuit (7) BUILT: rohan
WARN: Received http status code 404 ("Not found") from
server '203.138.99.218:443' while fetching "/tor/keys/
fp/27B6B5996C426270A5C95488AA5BCEB6BCC86956".
Circuit (8) BUILT: Unnamed
Circuit (5) BUILT: rohan > blurgh > ph3x
NOTICE: Tor has successfully opened a circuit. Looks like
client functionality is working.
NOTICE: Bootstrapped 100%: Done
Circuit (4) BUILT: rohan > sunIsBurning > caker
Circuit (3) BUILT: rohan > toronion > IPredator
Circuit (6) BUILT: rohan > fluxe3 > NanElmoth
Circuit (6) CLOSED: rohan > fluxe3 > NanElmoth
Circuit (4) CLOSED: rohan > sunIsBurning > caker
Circuit (3) CLOSED: rohan > toronion > IPredator
Circuit (7) CLOSED: rohan
Circuit (5) CLOSED: rohan > blurgh > ph3x
Circuit (1) CLOSED: Crows
Circuit (2) CLOSED: redjohn
Circuit (8) CLOSED: Unnamed
Circuit (36) BUILT: rohan > aergia > NCC1701B
Circuit (38) BUILT: rohan > Tartup > schnudel
Circuit (39) BUILT: rohan > charhome > t8chnode
Circuit (37) BUILT: rohan > Hadia > DFRI4
```

Fig. 3.4 Orbot logs showing the tear down of Tor circuits on losing wireless network connection, and the subsequent rebuilding on reconnection (v14.1.4 on 2nd June 2015)

The Orbot application was found to have a critical error. Consequently, Orbot was unable to support roaming at all, that is, the application required to be restarted after each hand-off, for the user to be able to continue to use the Tor network. This bug was later independently verified in 2013 by the Guardian Project as “Orbot seems unable to cope with roaming Wi-Fi” [151]. This issue made anonymously browsing the World Wide Web (WWW or the Web), using the Tor network, impractical for a mobile user.

The bug was fixed in 2014 and now Orbot automatically builds a new set of Tor circuits on reconnection to the Internet, as shown in Figure 3.4. Although the critical error was resolved, on breaking a connection to the Internet, the circuits on the Tor network are still torn down ungracefully. Due to an existing issue with the time it takes to build Tor circuits, this is expected to negatively impact performance [111] [127], as shown in Figure 3.5.

During each of the scheduled field experiments, the Tor circuits are consistently observed being torn down at every break in connection to the Internet and, consequently, an extended time to recovery incurred while roaming between networks. However, what is inconsistent, over the last couple of years of field experiments, is the time taken to rebuild circuits to re-establish a connection to the Tor network. In the real-world environment, earlier Tor circuit build timings were observed at 10 to 20 seconds and often timing out at 30 seconds. However, since 2013, a number of fixes have been implemented by the Tor project that have

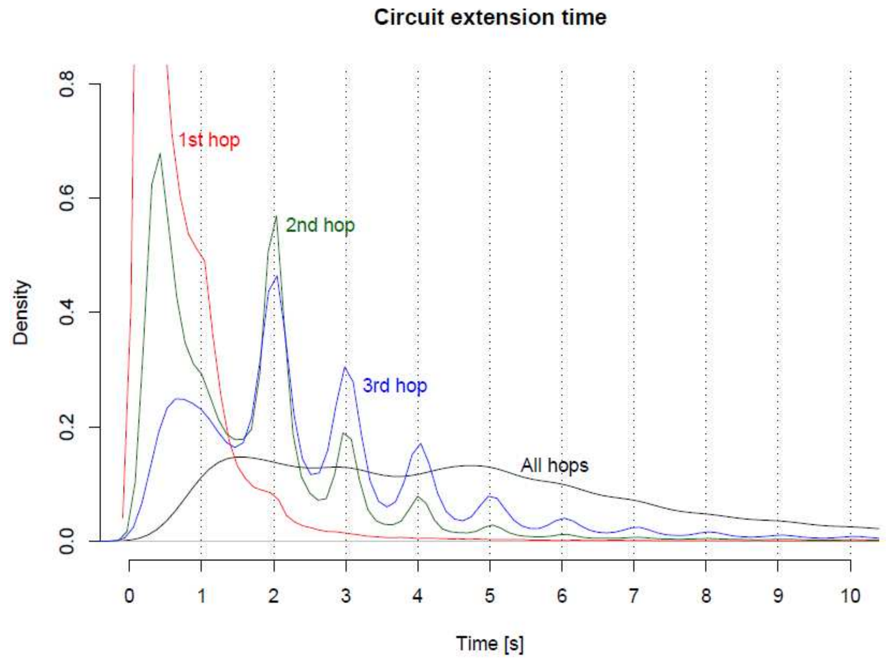


Fig. 3.5 The mean time taken to establish each hop of a 3-hop circuit [111]

improved the circuit build process to make it more reliable and quicker, more akin to the published mean circuit build timings [111].

The field study achieved its objective by initially understanding how Tor handles mobility of its users. The field experimentation regularly required to be adapted. Not only to incorporate the new Internet services being introduced such as BT Wi-Fi hotspot service (2012), Three inTouch (2015), but over the course of the observations, changes to both Orbot and Tor. As environmental and network conditions at the time of each observation, such as windy conditions, can affect performance, although, the field study can provide general information on whether the service is in-use and generally usable, it is unlikely, if not impossible, to be able to provide detailed performance data. This is often the case within a live environment as many of these external factors (or variables) cannot be easily controlled. To try to replicate even a scaled-down Wi-Fi service of five million hotspots and a popular anonymity network, such as Tor, itself with over two million users and 5,000 relays, is practically impossible. Therefore, even though regular observations of Orbot and the Tor network are useful, the use of modelling, whether mathematically or through simulation, is required to provide more scientific rigour and is used for the remainder of this research.

Simulator	A	B	C	D	E	Notes
Chutney	✓	X	X	✓	✓	Deprecated, not supported, use not advised by developer
Experimentor	✓	X	X	✓	✓	Deprecated, no longer actively developed
Matlab	†	†	✓	✓	†	†Yes - with scripting and/or Simulink libraries
NS	X	✓	✓	✓	✓	**Shortlisted** - usability issues?
OMNET++	X	✓	✓	✓	✓	**Shortlisted**
OPNET	X	✓	✓	X	X	Academic license withdrawn
Shadow	✓	X	X	✓	✓	Limited support, installation issues

Table 3.1 A summary of the evaluation of the different simulation tools for this research.

3.2 Network simulation

Although the field study was useful, it lacked control and scope for further detailed analysis. The need for a bespoke Tor simulator is an ongoing area of research and development [152]. Tor simulators including Experimentor and Shadow are a step forward in accurately simulating the Tor network with reasonable scalability [153] [154]. However, based on their current design specifications, neither Experimentor nor Shadow can support the key simulation requirement of mobility. The requirements of the simulator are coincidentally similar to Tor's design goals of "Deployability, usability, flexibility, simple design" and other features of being 'free software' and the code also open source. The key requirements are as follows:

A - does the simulator support the Tor protocol?
B - does the simulator support mobility, simulate wireless hand-offs?
C - is the simulator vendor supported, or well-supported community?
D - is the simulator free-to-use, includes academic licence?
E - is the simulator code open source?

As seen in Table 3.1 there is no clear 'winner' in terms of supporting the simulation. With the OPNET academic licence under review in 2012, the two shortlisted candidates are NS and OMNET++. NS was rejected based on some installation and usability issues an OMNET++ selected as having extensive support libraries. As will be discussed later in the next section on mathematically modelling, Matlab will be used to reflect the Tor functionality such as circuit window size. The experimental environment begins to evolve into a two-stage approach during later evaluation of the proposed solutions.

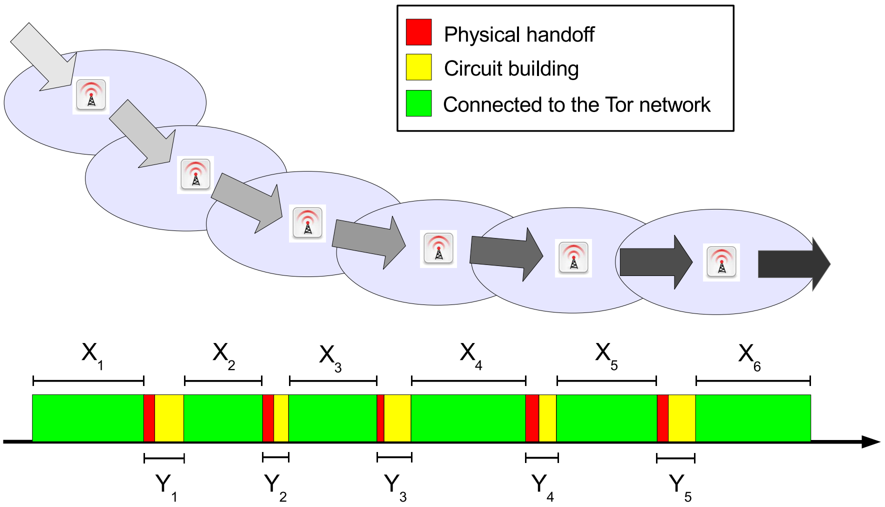


Fig. 3.6 A schematic of ‘roaming’ and the availability derived from mean time between failure (MTBF) and mean time to recovery (MTTR)

Therefore, to replicate the field study within a simulated environment, quickly at the beginning of the research, a generic network simulator is used. Although OMNeT++ is a simulation framework rather than a simulator, specific application areas, such as wireless networks, are supported by various simulation project libraries [155]. For example, the INET project library is a communications network simulation package for the OMNeT++ simulation framework supporting wireless networks and mobility modelling [156]. The OMNeT++ framework alongside the INET project library is chosen for the initial network simulation.

The aim of the network simulation is to provide an indicative assessment of the level of impact of mobility. Therefore, the key parameters for the scenario are: mobility (speed and direction), the location and range of the wireless network access points, and an important Tor-specific variable, circuit build time, as part of the overall time to recovery. To reflect the author’s walk to university again, the simulation consists of one mobile Tor user (the author), travelling at constant speed with synthetically generated linear mobility between access points at fixed ranges. As the behaviour of the mobile Tor user, and consequently the frequency of hand-offs, is uniform, this should help provide clearer results of the impact.

The mobility of a hypothetical user, while roaming, is shown conceptually in Figure 3.6. There are three distinct phases as a mobile Tor user while roaming. Firstly, when the user is handing off between physical (wireless) networks, they have no connection to the Internet, and consequently Tor. Once a connection to the Internet has been established, the service through the Tor network needs to be created by building the circuits. Only once this is completed, the mobile Tor user is able to use the Internet, such as for browsing the Web anonymously. It is possible to calculate a standard quality of service (QoS) performance metric, availability, derived from mean time between failures (MTBF), and mean time to recovery (MTTR): availability = $\frac{MTBF}{MTBF+MTTR}$. The MTBF and MTTR values are generated from $\sum \frac{X_n}{n}$ and $\sum \frac{Y_n}{n}$ respectively.

To calculate the impact, an experiment is undertaken using OMNeT++ / INET. A mobile user is simulated travelling between wireless network access points set at 75 m apart. The mobility speeds analyzed are average walking pace (1.2 m/s), ‘commute’ speed (10 m/s), ‘highway’ speed (30 m/s), and stationary (0 m/s) [157]. Each mobility speed is assessed alongside a range of Tor circuit build times. Based on the parameters used in previous studies, the user downloads a file (300 kB webpage) at intervals of 2 seconds between each download [158] [159] [160]. The performance metric used is average bitrate (ABR), in Kibit/s, achieved by the mobile user over a time frame of 600 seconds. In addition to the simulated physical hand-off, an artificial delay is also introduced, to reflect the performance overhead of using the Tor network based on previous research, including 2 seconds ‘wait’ time and 9 seconds to complete each 300 kB download [158] [159] [160]. Finally, the expectation is that higher mobility speeds and Tor circuit build times will reduce the performance achievable by the mobile Tor user.

Figure 3.7 shows performance as the ABR achieved, over the time frame of 600 seconds, by a mobile Tor user stationary (0 m/s), at average walking pace (1.2 m/s), ‘commute’ speed (10 m/s), ‘highway’ speed (30 m/s). Performance is also assessed at different circuit build times between 0 (a theoretical baseline for not using Tor) and 30 seconds as the outer range of usability, with indicative markers of 3, 7, and between 15 and 20 seconds, having been previously cited as ‘good’, ‘average’ and ‘slow’ circuit build timings [161]. For the remainder of the thesis, the key parameters are summarized using a bespoke coverage matrix, as first shown in Table 3.2. For each experiment, the relevant experimental section is highlighted. Although early the experimentation does not include attacks on anonymity, from Chapter 4 onwards selected attacks are undertaken, with a full analysis of anonymity in Chapter 6. If required, descriptions of the full range of the attacks A-1 to A-7 can be found together on pp.110-116.

As predicted, the introduction of mobility alone reduces performance. The physical network hand-off introduces an interruption in service and therefore reduces overall service availability. However, while using Tor, each break in the physical network connection additionally requires the rebuilding of the Tor circuits. The negative impact of this increased

	Experiment in §:	**3.2**	3.3	3.4	4.2	5.3	6.2
ANONYMITY:							
Possinymity:	Attack A-0	-					
	Attack A-1	-					
	Attack A-2	-					
	Attack A-3	-					
	Attack A-4	-					
	Attack A-5	-					
Indinymity:	Attack A-6	-					
	Attack A-7	-					
PERFORMANCE:							
MTTF:	Speed (mps)	0, 1.2, 10, 30					
	HOTs (s)	-					
MTTR:	Physical (s)	1.5					
	Circuit (s)	1...30					
	Redirection (s)	-					
Load:	File size (KiB)	300					
	Wait (s)	2					
	Request (s)	2					

Table 3.2 A coverage matrix of the key parameters used for the network simulation.

time to recovery is significant. At user mobility speeds above walking pace, for example commute and highway speed, the reduction in performance from the baseline is dramatic, at 85% and 97% respectively. Therefore, the use of Tor is impractical for these particular scenarios. Even at walking pace, the negative impact on performance is still considerable, with a 66% reduction in ABR for ‘good’ and a 77% reduction for ‘slow’ circuit build times. At circuit build times of 30 seconds the reduction in performance is on average 91%. This is similar to the negative impact experienced at high mobility speeds.

In summary, based on the results of the network simulation, the negative impact on client performance, that is, on a mobile Tor user, is significant while roaming. For cases with high mobility speeds and circuit build times, the impact is so significant the use of Tor is impractical. In addition to the direct impact to the mobile user, data remaining in-flight at the break in connection and any resubmitted failed requests, generates additional congestion on the Tor network, causing a negative impact on performance for all users, whether mobile or static.

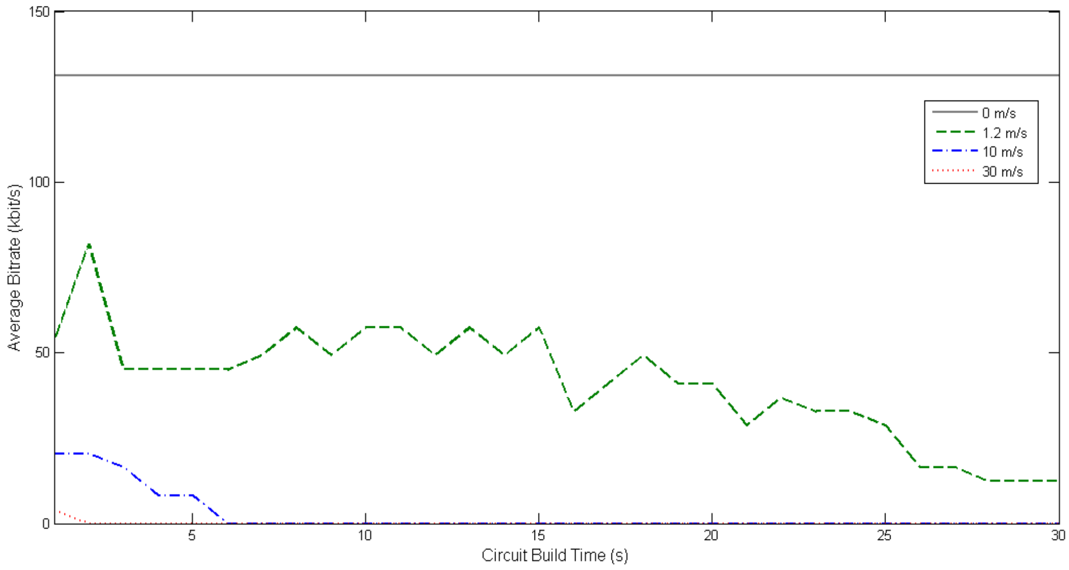


Fig. 3.7 The performance impact on a roaming mobile Tor user, at different mobility speeds and circuit build times, based on a network simulation using OMNeT++/INET

3.3 Mathematical modelling - Part 1

The network simulation is useful to begin to quantify the impact of mobility. As previously discussed, the lack of a tool that can support the simulation of both mobility and the Tor network led to the initial selection of a generic network simulator, OMNeT++. Although OMNeT++ provides realistic simulation of the network layers, it is unable to replicate the inner application workings of Tor. The next approach investigates the impact of mobility on the performance of Tor clients by modelling a variety of scenarios using MATLAB [162]. MATLAB provides the ability to model specific Tor features, such as application-level circuit windows and algorithms for circuit path selection, that Tor employs for its traffic flow and management, which has a significant role in optimizing performance [163].

As previously shown, any break of connection to the Internet, and consequently to the Tor network, has a negative impact on the performance achieved by the mobile Tor user. In contrast to the previous simulations, the hand-off timings are randomly generated (UNIFORM) based on the mobility speed and network ranges. This is to extend the modelling away from the initial more rigid mobility. The overall time to recovery variable comprises a physical network hand-off of one second and a range of Tor circuit build timings again based on previous research [111] [127]. The key parameters are outlined in Table 3.3.

	Experiment in §:	3.2	**3.3**	3.4	4.2	5.3	6.2
ANONYMITY:							
Possinymity:	Attack A-0		-				
	Attack A-1		-				
	Attack A-2		-				
	Attack A-3		-				
	Attack A-4		-				
	Attack A-5		-				
Indinymity:	Attack A-6		-				
	Attack A-7		-				
PERFORMANCE:							
MTTF:	Speed (mps)		0, 1.2, 10, 30				
	HOTs (s)		-				
MTTR:	Physical (s)		1				
	Circuit (s)		1...30				
	Redirection (s)		-				
Load:	File size (KiB)		320				
	Wait (s)		2				
	Request (s)		2				

Table 3.3 The key parameters used for part one of the mathematical modelling.

The key performance measure is average bitrate (ABR), that is, the total number of successfully completed downloads of 320 kB over 600 seconds represented as Kibit/s. To reflect the performance overhead of using the Tor network based on previous research, taking 11 seconds to complete each download, a latency of 2 seconds for the first byte received and 9 seconds for time to complete, are introduced for each request, again based on previous research [159]. The ABR achieved is classified as 'good' traffic and any remaining data, either incomplete downloads or data still in-flight at the time of hand-off, classified as 'bad' data. The expectation is that any increase in mobility speed and/or the time to recovery will negatively, and significantly, impact performance as an overall reduction of goodput in the ABR. Resubmitted requests and data lost in-flight not only directly impacts the mobile user, but could also potentially impact the Tor network, and consequently other users, through the additional network traffic and congestion.

The results of the modelling are shown in Figure 3.8. At mobility speeds above walking pace, the negative impact on performance is so significant the use of Tor is impractical. As previously discussed, the key issue is that the time to rebuild Tor circuits is many times longer than the physical network hand-off. Therefore, while commuting at a mean speed of 10 m/s alongside a circuit build time of over 4 seconds, the user fails to complete a successful

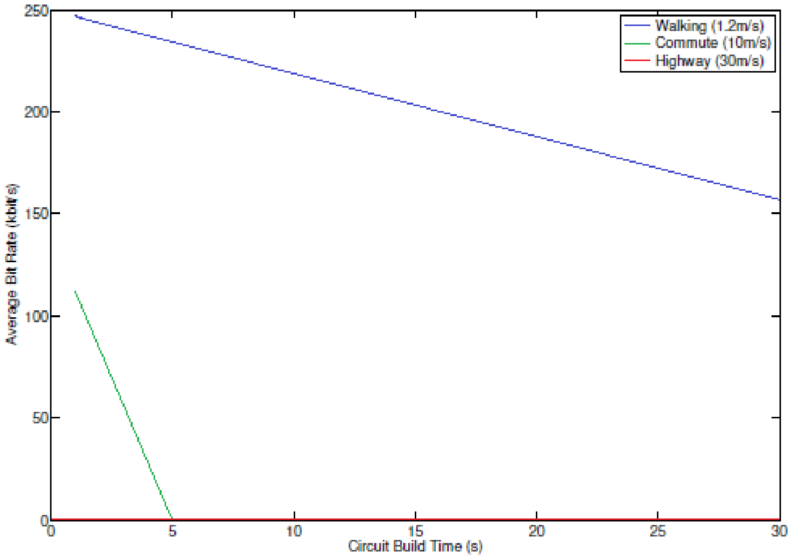


Fig. 3.8 The impact of mobility speed and circuit build times, on Tor client performance, using mathematical modelling (MATLAB)

download between hand-offs. At 30 m/s, no downloads can be completed at any circuit build time. Again, even at an average walking pace of 1.2 m/s, the negative impact on performance remains significant. For example, at an average circuit build time, of 7 seconds, there is a 14.45% reduction in the ABR than if the mobile Tor user remains stationary (0 m/s). Even at a 'good' circuit build time of 3 seconds, mobility still generates a 9.45% reduction in performance. For longer circuit build times of 15 and 20 seconds the negative impact on performance is 23.43% and 29.18% respectively. Also, a roaming mobile user, at average walking pace, generates on average nearly 7% of all traffic as bad data. Travelling at higher mobility speeds and with longer circuit build times, this can generate significant amounts of bad traffic, which, in some cases surpasses the total amount of goodput achieved.

The results show again, that even at average walking pace, there is a significant negative impact on performance achievable by the mobile user while roaming. As we expect, the negative impact on performance increases with higher mobility speeds and longer circuit build times, in some cases making the use of Tor no longer practical for a roaming mobile user. Although the results compare relatively to the previous experimentation using a generic network simulator, the mean goodput generated is significantly higher (almost double) than the previous study. The reason for this anomaly will be discussed during the following section.

3.4 Mathematical modelling - Part 2

The results from the field study, generic network simulation, and initial mathematical modelling, all suggest Tor cannot support mobility efficiently. In some circumstances, the negative impact on performance is so significant the use of Tor is impractical. The amount of 'bad' traffic generated by mobile Tor users, breaking their Internet connections while roaming, is also a concern. The concern being that this may cause further congestion and therefore have an impact on the Tor network and consequently other Tor users.

The previously mentioned anomaly, between the results from the network simulation and mathematical modelling, is due to different interpretations of the implementation of the variable: 'time between requests'. In the previous modelling, the requests were scripted at regular two-second intervals instead of once the previous download has been completed, matching the previous generic network simulator. Although, the results from the network simulation and mathematical modelling both show an impact, there is a significant difference in the performance achieved, even though the rate of degradation remains similar, as shown in Figures 3.7 and 3.8 on pages 46 and 48. This observation has been useful and therefore, to ensure some consistency for subsequent modelling, the variables that need capturing and formally notated.

The simulation cannot run indefinitely and therefore again requires to have a fixed length of time (t). During the scenario, based on the current case study (the author's walk to university), the user remains mobile for the duration of the time frame. The approach to generating the hand-offs will be dependent on the mobility modelling, bit the interval between each hand-off, is referred to hereafter as h or HOT (the latter an acronym of Hand-Off Times).

The next key component is the time to recovery, which as previously discussed, consists of the time taken for the physical network hand-off and to complete the build of circuits on the Tor network. Again, depending on mobility modelling approach, this also allows both or either of the time to recovery components to be scripted as an array for the modelling if required, so each break in connection may incur a different time to recovery. Once an array of hand-offs is generated, each requires adjusting to generate a net figure (of availability), that is, the time available for downloading files once the physical hand-off (y) and circuit build (k) times are subtracted. Next, traffic load needs to consist of the time between each *completed* request (r), and the size of the requested file. The file size, for example a webpage, is denoted as S to distinguish from seconds (s), which is already used for average bitrate (ABR). The time to first byte (f), to reflect the 'think time' of a web server, and the mean time taken to download a file (d), also need to be included. These times are to based on empirical data from the same previous study [159]. This delay is to reflect again the reported slower of performance of using Tor, rather than a 'normal' Internet configuration, for browsing the

Web. The initial calculation for goodput (g), application-level data, includes both completed and partial downloads achieved during the interval between specific hand-off (i), as described in Equation 3.1:

$$g(i) = \frac{h-(y+k)}{r+f+d} \tag{3.1}$$

The number (floor) of successfully completed downloads is calculated as $g \rightarrow \lfloor g \rfloor$. This is then applied to all $h(i,\ldots,n)$ where n is the total number of intervals recorded between hand-offs. This generates the total number of successfully completed downloads during the overall time frame. The overall performance (b) is simply derived by the number of downloads multiplied by the request size and with the factor (x), to convert to kibibit (Kibit). Finally, divided by t to provide the mean goodput as an ABR in Kibit/s. Therefore, goodput reflects the performance achieved as an ABR delivered to the application layer, for example Hypertext Transfer Protocol (HTTP) traffic, but exclusive of all protocol overhead, data packets retransmissions, as notated in Equation 3.2:

$$b = \frac{\sum_{i=n}(\lfloor g \rfloor(i)) \cdot S \cdot x}{t} \tag{3.2}$$

The calculation for total amount of 'bad' traffic as orphaned data (o) generated, that is incomplete downloads or data remaining in-flight, as described in Equation 3.3:

$$o = \sum_{i=1}^{n}(g) - \sum_{i=1}^{n}(\lfloor g \rfloor(i)) \cdot S \tag{3.3}$$

Based on the results of the network simulation, the levels of negative impact shown on client performance at higher mobility speeds, the modelling will focus on the mobile user travelling at walking pace. To complete the analysis on the impact of mobility, the existing case study of the author's walk into university is used for the final time. As the walk is at a brisk, rather than average, walking pace, the range of hand-off times generated will fall between 60 and 180 seconds alongside a Gaussian distribution of σ = 25, to focus the distribution around a mean hand-off of 120 seconds. Additionally, based on empirical data provided by Jansen, Syverson, and Hopper, more realistic timings and traffic loads are assessed during this additional modelling [159]. The different levels of indicative congestion on the Tor network are introduced, categorised as: low, medium, high, adjusting the time to first byte received as shown in Table 3.4 and consequently the mean time to download values.

The hypothesis is slightly changed from the previous modelling as only walking pace is assessed. As before, the hypothesis is that any increase in the time to recovery, or more specifically increases in circuit build times, will have a negative impact on performance. Any increase in congestion is also expected to have a negative impact on performance.

	Experiment in §:	3.2	3.3	**3.4**	4.2	5.3	6.2
ANONYMITY:							
Possinymity:	Attack A-0			-			
	Attack A-1			-			
	Attack A-2			-			
	Attack A-3			-			
	Attack A-4			-			
	Attack A-5			-			
Indinymity:	Attack A-6			-			
	Attack A-7			-			
PERFORMANCE:							
MTTF:	Speed (mps)			-			
	HOTs (s)			60…180			
MTTR:	Physical (s)			1			
	Circuit (s)			3, 7, 17.5			
	Redirection (s)			-			
Load:	File size (KiB)			320, 5000			
	Wait (s)			3, 5, 7			
	Request (s)			2			

Table 3.4 The key parameters used for part two of the mathematical modelling.

The results of the modelling provide some further, and interesting, insight into the impact of mobility and the wider performance issues on the Tor network. Congestion on the Tor network is shown to have a considerable impact on client performance supporting other research. By comparing results in Figures 3.9 to 3.11, the total amount of data successfully downloaded drops by approximately half for each level of congestion, that is, light, medium and heavy. The impact of congestion for 5 megabyte (MB) bulk downloads is even worse. As illustrated in Figure 3.12, any worse than light congestion dramatically impacts the ability of a mobile Tor user to successfully complete any downloads, and even more significant is the amount of bad traffic generated by these failed downloads. Even with light congestion, the amount of bad traffic generated, shown in Figure 3.13, exceeds goodput at mean circuit build times of 7 seconds. Slower circuit build times can generate twice as much bad traffic than goodput. This supports trying to deter bulk downloads on the Tor network due to the performance issues it causes for other Tor users trying to browse the web anonymously.

Although somewhat overshadowed by the effect of congestion on performance, the negative impact from the time taken to build the circuits at each hand-off remains critical. Another finding is the relationship between good (goodput) and bad traffic. As previously alluded to, at a network hand-off, downloads may still be in progress, and only partially

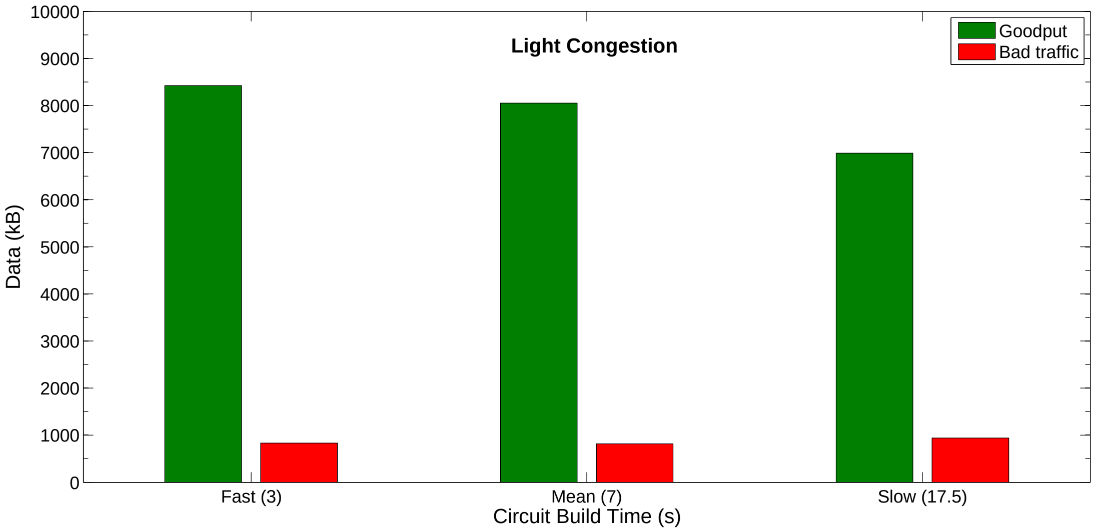


Fig. 3.9 The total data received downloading 320 kB files with light congestion at different circuit build times

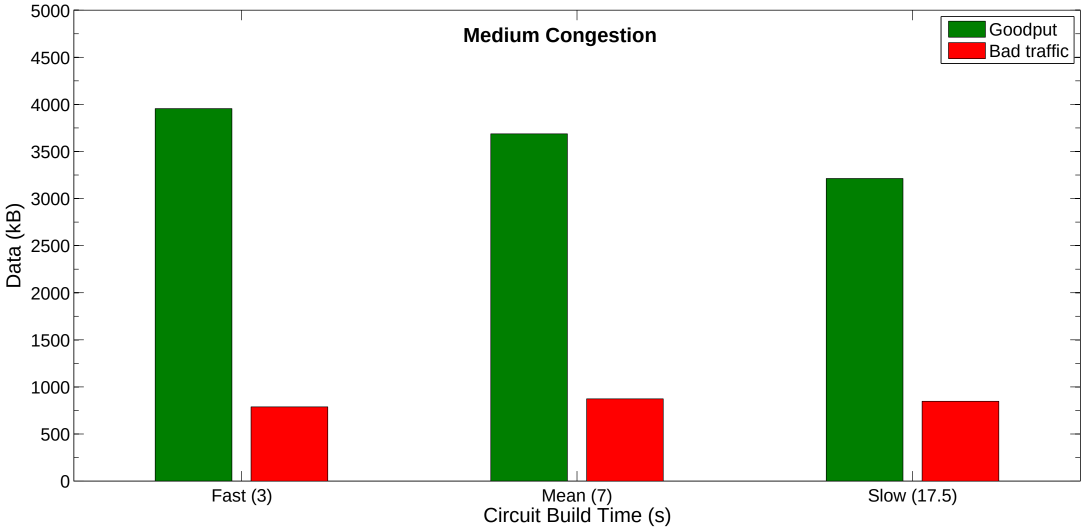


Fig. 3.10 The total data received downloading 320 kB files with medium congestion at different circuit build times

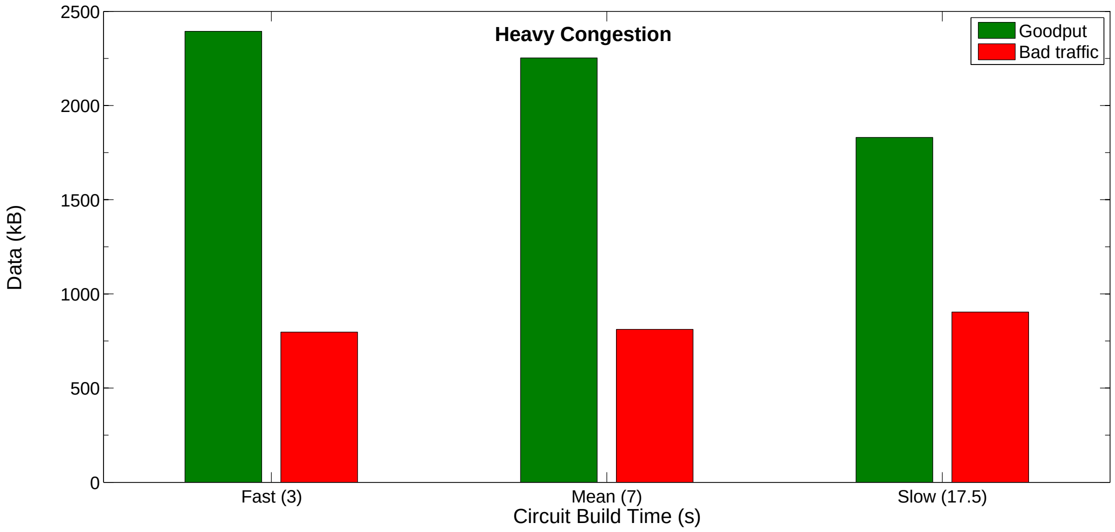


Fig. 3.11 The total data received downloading 320 kB files with heavy congestion at different circuit build times

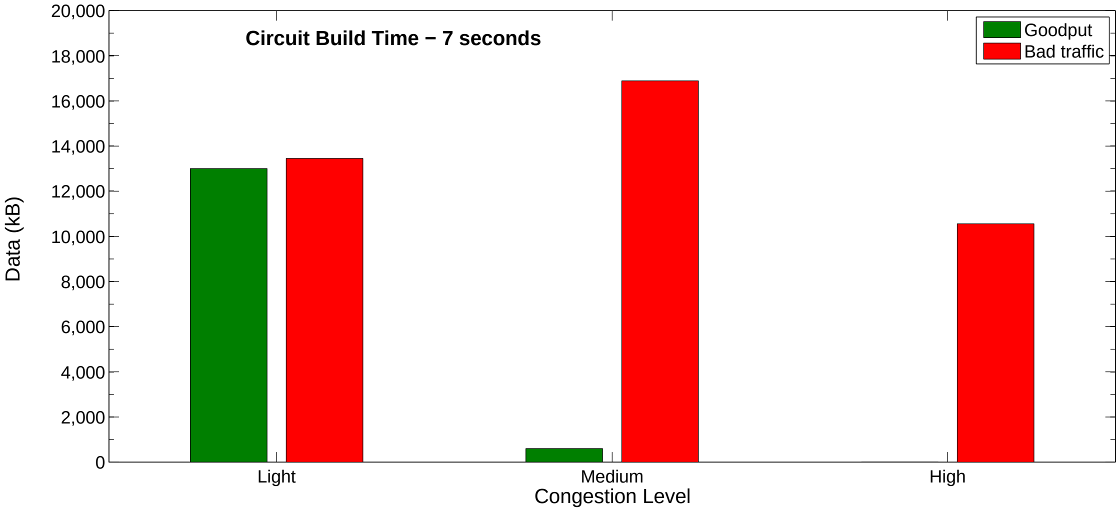


Fig. 3.12 The total data received downloading 5 MB files at a circuit build time of 7 seconds, across different levels of congestion. No files successfully downloaded with heavy congestion

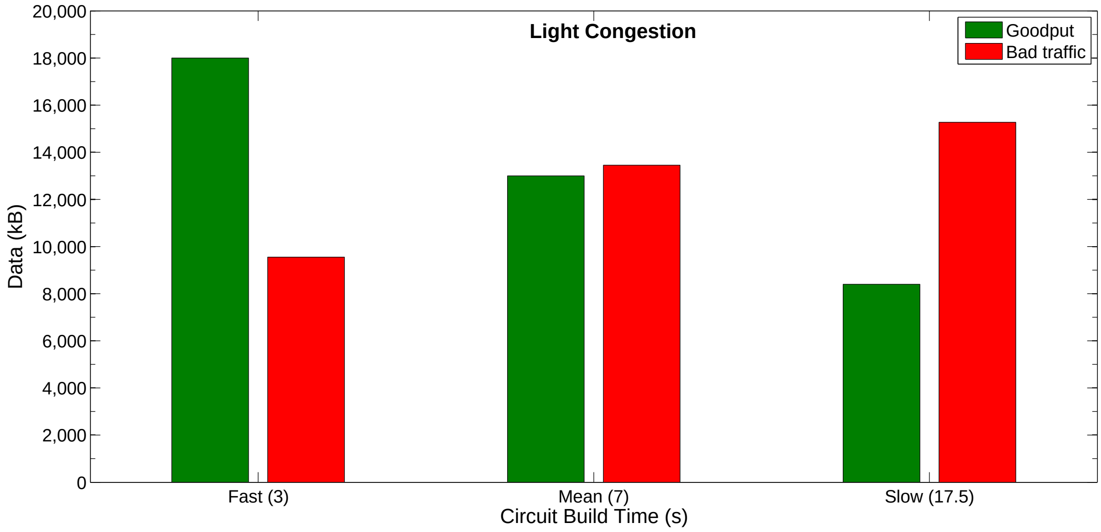


Fig. 3.13 The total data received downloading 5 MB files with light congestion, across different circuit build times

complete when the network connection is broken ungracefully. This suggests that any approach to mitigate the impact of mobility needs to reduce the amount of data left in-flight at the hand-off while maintaining a reasonable level of client performance.

3.5 Summary

The results from the field study, network simulation, and mathematical modelling, all support the hypothesis that mobility has a negative impact on performance. Increases in mobility speed and circuit build time directly influence the level of impact, in some cases, making the use of Tor no longer practical for a mobile user. The amount of bad traffic generated, especially for bulk downloads, is far greater than expected. This supports other research aiming to mitigate the impact of peer-to-peer file sharing and bulk download traffic on the Tor network. Based on the findings so far, the existing design of Tor is unsuitable for supporting the increasing mobility of its users. This suggests that some form of impact mitigation and/or a solution to provide a persistence of connection to the Tor network, for mobile Tor users, is required. The solution will require not only to maintain an acceptable level of performance for the mobile user, but also mitigate the potential impact on the Tor network from increased network churn, and while also maintaining anonymity. These requirements are not trivial and how any proposed solution is evaluated will be covered during the next chapter.

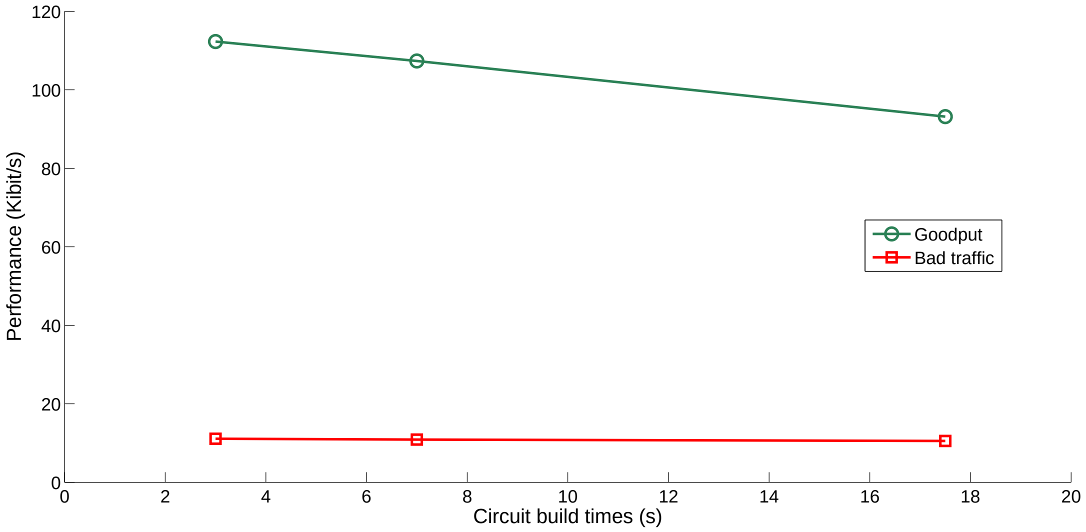


Fig. 3.14 Performance, as an average bitrate (Kibit/s), across different circuit build times with light congestion

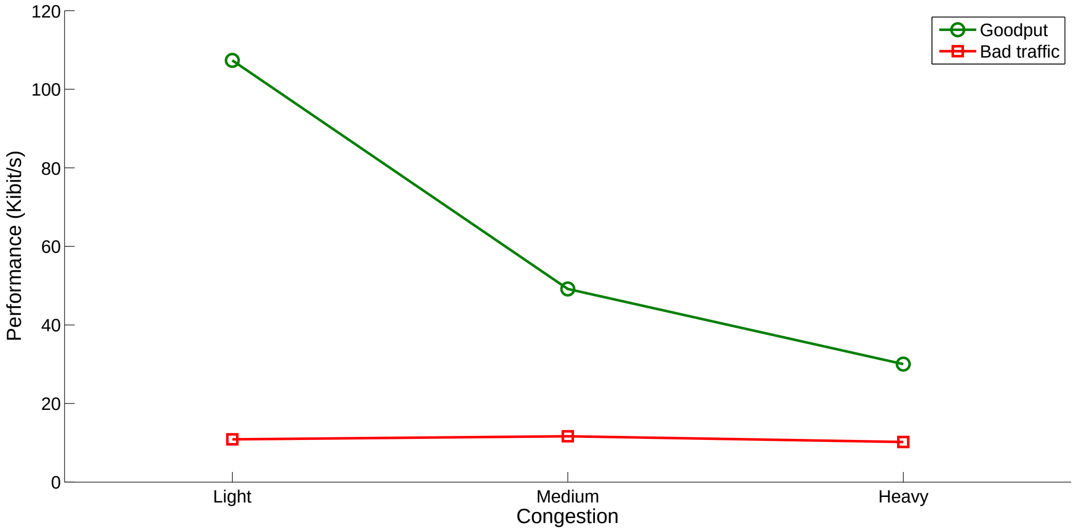


Fig. 3.15 Performance, as an average bitrate (Kibit/s), across different levels of congestion, at a circuit build time of 7 seconds

Chapter 4

Measuring Anonymity and Performance in Dynamic Network Topologies

Maintaining an appropriate balance between anonymity and performance is critical for low-latency anonymous communications' systems, such as the Tor anonymity network [9] [111]. The number of 'hops' chosen for a circuit and the selection of paths with the highest available bandwidth are two key examples of how the need to maintain this balance has influenced Tor's design [23] [61]. It is not only design choices that can affect this balance. Research has also shown that threats to the reliability of an anonymity network, such as a denial of service attack on Tor routers, can simply decrease anonymity through an overall reduction in the number of routers available for circuit-building [22]. Additionally, poor performance on the Tor network for example, caused by bulk downloads, can also have an impact, deterring usage, where eventually the number of concurrent users offers little or no anonymity [101]. The development of smartphone technology is increasing Internet usage from wireless-enabled devices. In 2010, Orbot was released enabling access to the Tor network from Android devices [30]. Anonymity networks, and the underpinning onion routing, are designed for persistent wired Internet connections. Mobile users, while roaming, may access the Internet from a range of networks and service providers (cellular, Wi-Fi) generating a different external IP address after each hand-off. Due to its design, the connection to the Tor network breaks whenever a client's external IP address changes, requiring an extended time to recovery while building new Tor circuits. In this chapter, by modelling mobility and the consequent network churn, the impact of mobile usage *over time* on both anonymity and performance is illustrated. A new metric (q-factor) is presented that supports a combined measurement of both anonymity and performance. The metric is shown to enable more effective network management to maintain the optimal balance between anonymity and performance for low-latency anonymity networks such as Tor.

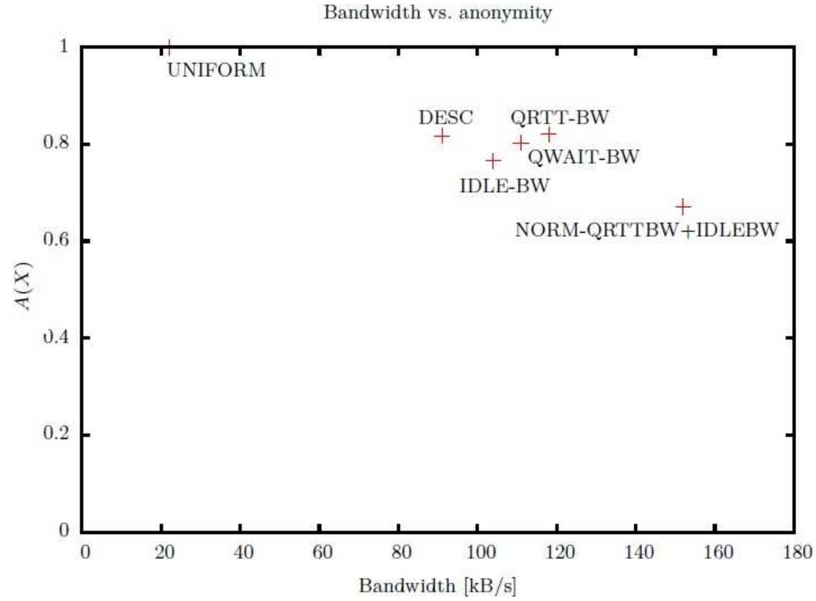


Fig. 4.1 Panchenko, Lanze, and Enkel's original mapping of both anonymity and performance for different circuit building path selection algorithms [37]

4.1 Introducing q-factor

It has been shown that the maintenance of both anonymity and performance is critical for low-latency anonymity networks. Anonymous browsing of the World Wide Web is an application that attracts millions of users to anonymity networks such as Tor [33]. In general, the higher the number of users, the larger the anonymity set (*AS*) generated, and potentially the anonymity provided. This reflects the old adage of 'safety in numbers'. If performance degrades to a level where users are deterred from using the network, then the anonymity set will reduce in size, perhaps to the point where eventually the number of concurrent users offers little or no anonymity. This suggests it is more appropriate to adopt a holistic approach, measuring anonymity and performance together, rather than separately.

Panchenko, Lanze, and Enkel examined the critical balance between anonymity and performance while assessing different circuit path selection algorithms [37]. The results are shown in Figure 4.1. The solution names are not important but illustrate that, although the mapping approach shows a comparison between different solutions, the mapping does not show whether any solution meets the minimum requirements for anonymity and performance. If thresholds are applied, in this case arbitrarily, it is easier to see whether the minimum requirements for anonymity and performance are met, as illustrated in Figure 4.2. The current design of Tor (DESC), although performing better overall than randomly allocating routers (UNIFORM), still does not meet the minimum requirements compared to the proposed solution (SOLUTION). This is also shown in Figure 4.3 with 'traffic lights' colours.

Adopting this one-off 'snapshot' approach still requires care. Any proposed solution could initially appear to provide the best overall balance between anonymity and performance, but when observed over time, may fall below one or both of the required thresholds. In addition, it should be noted that, although a mean value can often provide useful information about performance, it is likely to give a misleading view of anonymity. The issue is that any occurrences where the level of anonymity falls below the required threshold (θ_a), that is, a critical event, may not be captured. For example, a mean anonymity of $a = 0.67$, above $\theta_a = 0.50$, may at certain points in time drop below the threshold. Therefore, the anonymity provided by applying Reiter and Rubin's descriptors, rather than being *beyond suspicion*, is at best *possible innocence* or even *exposed* / *provably exposed* in the worst case scenario [132].

The risk of using the existing approaches to measuring low-latency anonymity networks is that strategic design and dynamic network management decisions are made, which at best deliver poor performance, or in the worst case scenario could compromise anonymity. Therefore, any new approach to measuring the effectiveness of low-latency anonymity networks, should consider both anonymity and performance, and the effect of network churn over time generated by mobile users recycling their connections.

Building upon the previous work of Panchenko, Lanze, and Enkel, if either anonymity or performance falls on or below the threshold at any point during the scenario, this event needs to be captured. To achieve this, a snapshot of both anonymity and performance are calculated periodically at an individual user, path, or network-wide level. This generates boolean values (0 and 1) for both anonymity (v_a) and performance (v_b), based on specified thresholds, θ_a and θ_b, for anonymity and performance, respectively, where:

$$v_a \equiv a > \theta_a \quad \text{and} \quad v_b \equiv b > \theta_b$$

The anonymity and performance is calculated as simply the conjunction of v_a and v_b, that is $q = v_a \cdot v_b$, as shown in Figure 4.4. A value of 0 indicates intervention is required, otherwise, the network can continue to operate as it is. However, in some real world environments, the network may be too volatile for a single-layered approach to remain effective. For example, if a q value of 0 is observed, it may already be too late to intervene, and the 'damage' already done especially if anonymity has already fallen below the required threshold.

An additional layer of 'states' is, therefore, proposed for highly dynamic scenarios. A 3 × 3 matrix of binary nibbles (q-nibbles) is implemented to allow the operator a longer time to intervene. The aim remains the same, that is, to maintain a q-factor value towards to

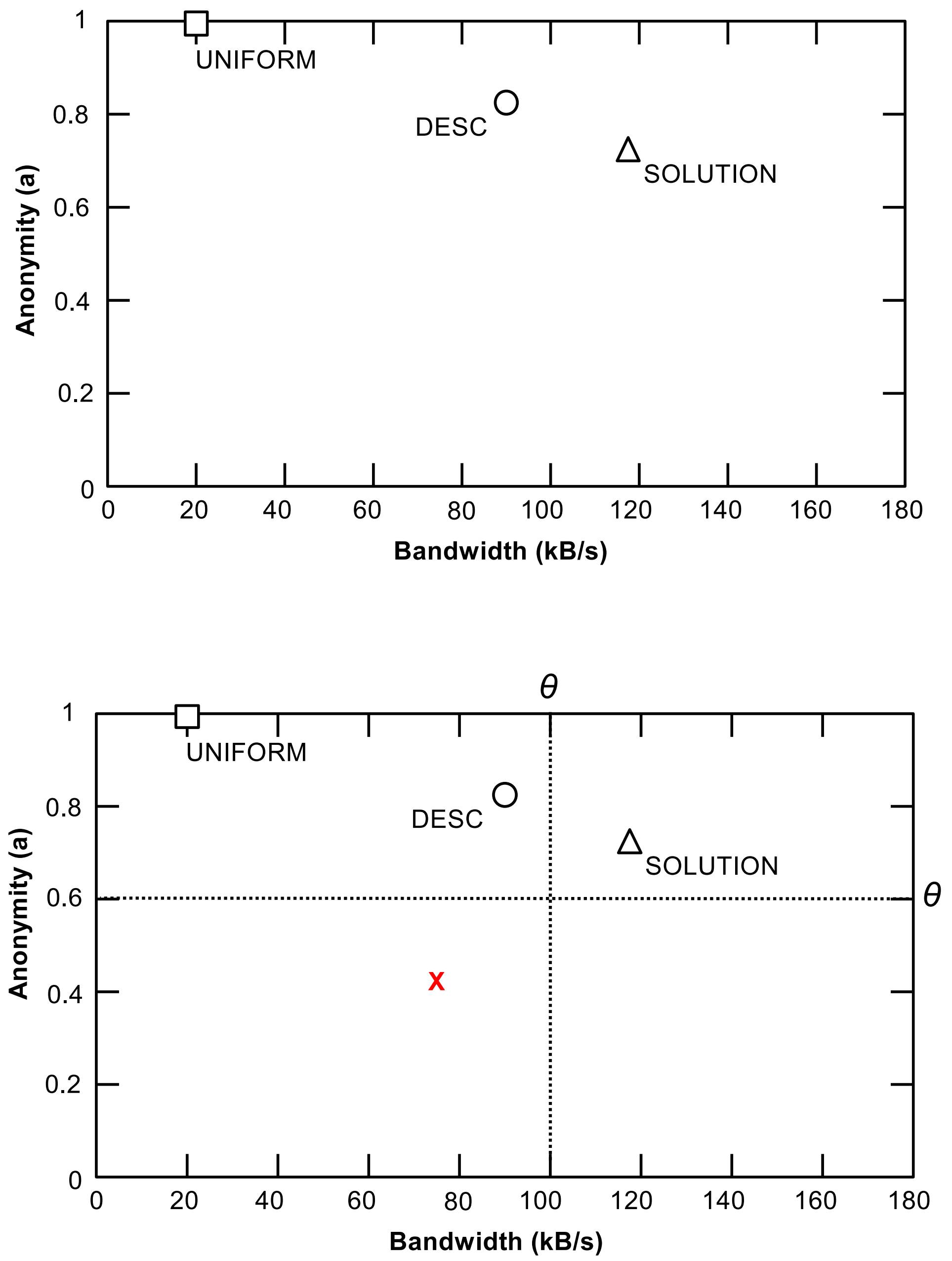


Fig. 4.2 Mapping anonymity and performance, adapted from Panchenko, Lanze, and Engel, with arbitrary thresholds (θ) applied, and an indicative marker X to illustrate a 'failed' solution

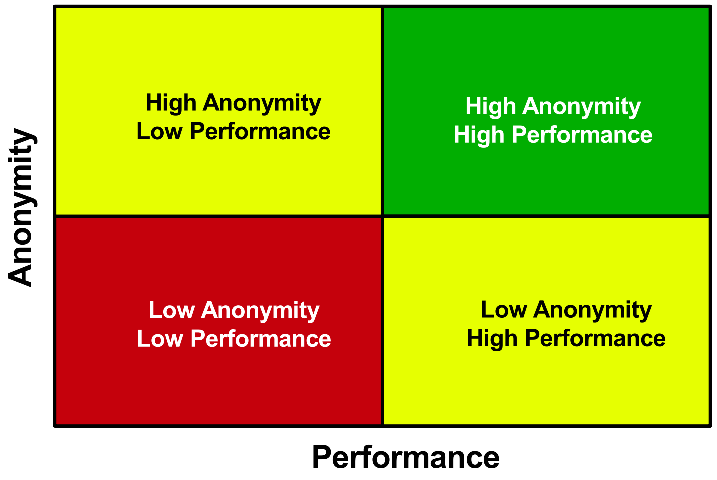


Fig. 4.3 A conceptual application of a traffic light colour system to classify each mapping of anonymity and performance

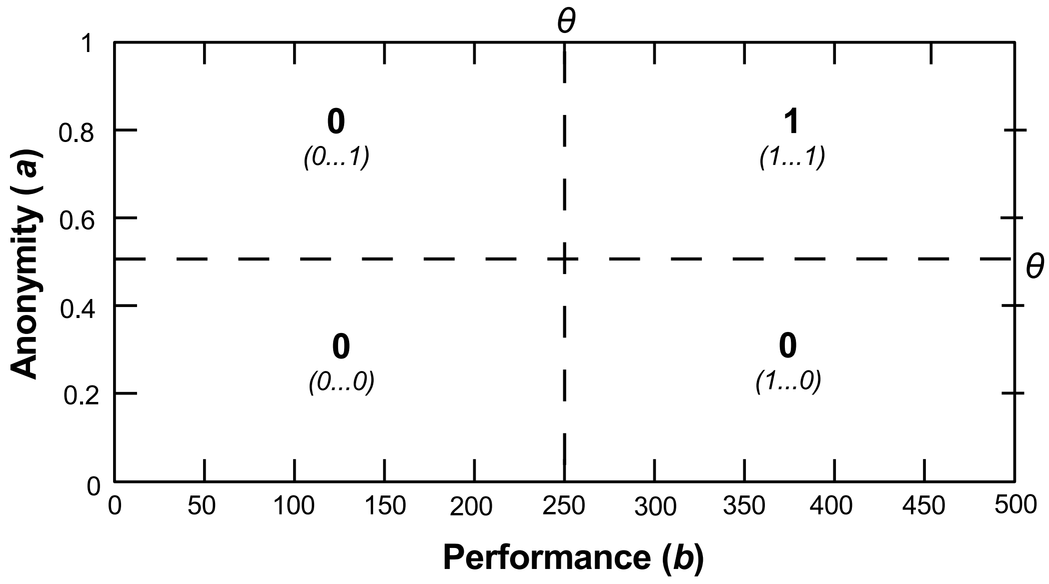


Fig. 4.4 The calculation of q-factor, as the conjunction of anonymity and performance, based on specified thresholds θ

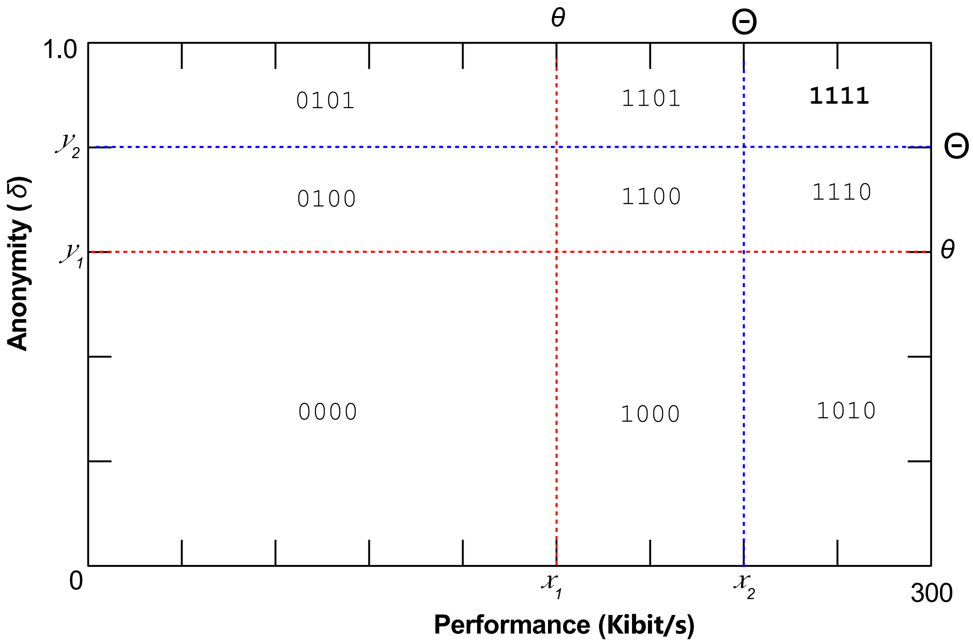


Fig. 4.5 The extended double-layered version of the q-factor metric with 9 states

the far upper-right ('goldilocks') zone. However, the q-nibbles form 9 states instead of the previous 4 with the value of '1111' being the highest, and '0000' the lowest, as illustrated in Figure 4.5. These values alone do not provide anything in addition to an observed status. Therefore, it is suggested these nine states form a set of risk levels to support the framework, are applied to the q-factor metric in Figure 4.6.

The actions undertaken by the operator depends on the specific requirements, that is, the most appropriate action for a particular service. A list of example actions that could be taken for different levels of risks are as follows:

- $\Sigma = 4$ - green: no action required, but keep monitoring.
- $\Sigma = 3$ - amber: assess possible intervention options, implement if feasible.
- $\Sigma = 2$ - red: intervention required.
- $\Sigma = 1$ - red, red: urgent intervention required.
- $\Sigma = 0$ - red, red, red: 'pull the plug'?

The proposed metric, known onwards as q-factor, single-layered (v1) and double-layered (v2), for measuring both anonymity and performance, over time, is presented. The following

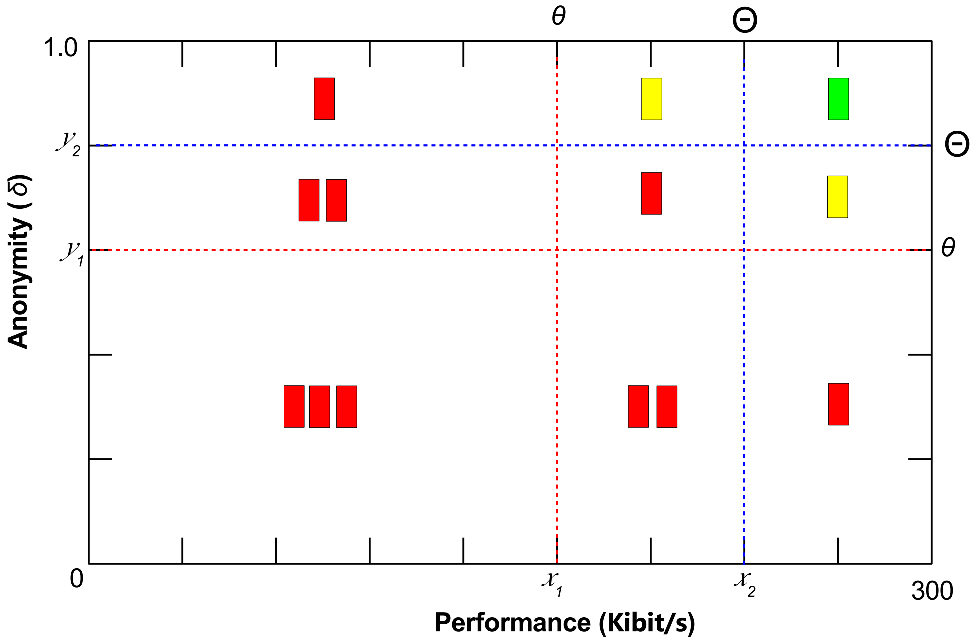


Fig. 4.6 The 9 states of q-nibbles and assigned risk levels

modelling aims to illustrate how the q-factor metric can quickly help make more effective strategic design and dynamic network management decisions.

4.2 Experimental Design

To focus on evaluating the q-factor approach and avoid excluding interest from development other than Tor, a more generic simulation is adopted, as shown in Figure 4.7. The modelling implements two alternative paths through the anonymity network to provide rudimentary load-balancing. The two channels have different fixed bandwidth capacities, of 2000 kB/s and 3000 kB/s to model varying performance across the network. A small set ($N = 20$) of users is chosen to ensure that anonymity falls below the required threshold during some runs of the model. The users are configured to randomly connect and disconnect at intervals of approximately 2 to 3 minutes during the scenario, reflecting a typical mobile user moving at walking pace. A circuit window of 500 kB/s is implemented client-side. This acts as application-level client throttling (bandwidth cap) to provide a fair distribution of resources and maintain steady traffic flow across the network.

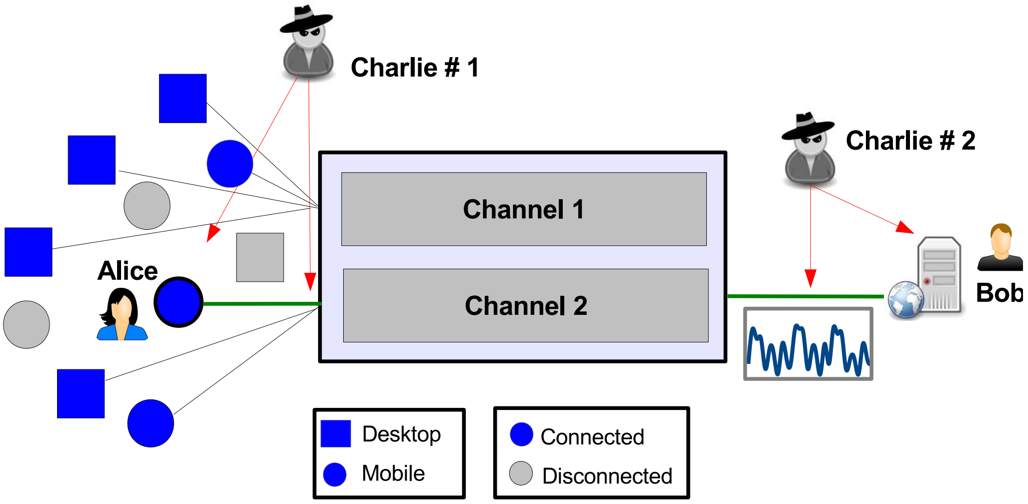


Fig. 4.7 A two-channel anonymity network used to assess the effectiveness of the q-factor metric, also illustrating the pervasive dynamics for Alice's communications with Bob, observed by an adversary, namely Charlie

The measurements ('snapshots') are taken at one-second intervals for both anonymity and performance during the scenario of 600 seconds. Based on the combined network capacity of 5000 kB/s, user set size (N), and circuit window size (w), θ_b = 250 kB/s is chosen as the lower threshold of performance. This is to allow a range of performance levels, on either side of the threshold, to be observed. An anonymity (possinymity) threshold of $\theta_a = 0.50$ is applied, as an 'acceptable' degree of anonymity at *probable innocence* or above, to be maintained during the scenario.

For the single-layered q-factor, if intervention is required, that is $q = 0$, the remedy for anonymity is simply to redirect one user to the failing channel. For poor performance, the exact opposite is required, that is, to redirect one user away to the other channel. At the next scheduled interval, the anonymity network is then re-evaluated, intervention applied (again) if required, to maintain a constant q-factor of $q = 1$, continuing each interval thereafter until the end of the scenario. To assess the effectiveness of this first approach, the following four schemes are compared:

- scheme 1: no intervention ('stock')
- scheme 2: intervention (anonymity $\underline{\text{only}}$)

- scheme 3: intervention (performance only)
- scheme 4: intervention (anonymity and performance)

In essence, scheme 1, providing no intervention ('stock'), acts as a baseline reflecting the approach of Tor [9]. Schemes 2 and 3, assess how applying only one intervention, either anonymity or performance, affects q-factor. Finally, scheme 4 assesses how intervention for both anonymity and performance affects q-factor.

The modelling is extended to evaluate how the double-layered version of q-factor (v2) performs compared to the single-layered version (v1). The same parameters are used with the key difference is how the status is interpreted and the most appropriate intervention taken. Where before only a single threshold for anonymity and performance were required (θ_a, θ_b), there now needs to be an upper threshold, Θ_a and Θ_b, to now determine the state. For anonymity, the thresholds are $\theta_a = y_1$ and $\Theta_a = y_2$, and for performance $\theta_b = x_1$ and $\Theta_b = x_2$. For anonymity $y_1 \geqslant 0.5$ and $y_2 \geqslant 0.75$, and for performance $x_1 \geqslant 250$ and $x_2 \geqslant 375$.

The logic in q-factor (v2) first checks the upper thresholds then lower. If both the upper thresholds are met, $\Sigma(q_n) = 4$, then no further action is required so do nothing (as with the value the of $q = 1$ with the previous version). For the value $\Sigma(q_n) = 3$ (previously $q = 0$) the available options require checking. If the intervention, on the current number connections, cause the other channel to fail, then the intervention is cancelled. Finally, for $\Sigma(q_n) = 2$, 1, or 0 (also previously $q = 0$), intervention will be automatically undertaken. A table containing the key parameters can be found in Table 4.1.

4.3 Results

The impact of mobility and increased network churn is shown as X in Figure 4.8. At a high level, using the default stock configuration without any form of intervention (scheme 1), the overall percentage of q-factor being maintained ($q = 1$) is lowest at 85.70%, as shown in Figure 4.9. Applying schemes 2 and 3 of the single layered version (v1) of q-factor, this figure increases to 93.86% and 92.31%, an improvement of 9.52% and 7.71% respectively against the baseline figure from scheme 1. Scheme 4, with full intervention for both anonymity and performance, provides the best overall performance at 98.32%. This is a 14.73% increase against the baseline and 4.75% better than its nearest competitor. The network-wide capacity utilization is shown in Figure 4.9. The results are less clear-cut for capacity utilization with scheme 1 again providing the lowest performance and each of the other schemes providing at around a 5% improvement over the baseline.

	Experiment in §:	3.2	3.3	3.4	**4.2**	5.3	6.2
ANONYMITY:							
Possinymity:	Attack A-0				✓		
	Attack A-1				✓		
	Attack A-2				-		
	Attack A-3				-		
	Attack A-4				-		
	Attack A-5				-		
Indinymity:	Attack A-6				-		
	Attack A-7				-		
PERFORMANCE:							
MTTF:	Speed (mps)				-		
	HOTs (s)				0...600		
MTTR:	Physical (s)				-		
	Circuit (s)				-		
	Redirection (s)				-		
Load:	File size (KiB)				300		
	Wait (s)				-		
	Request (s)				-		

Table 4.1 The key parameters used for the evaluation of the q-factor metric.

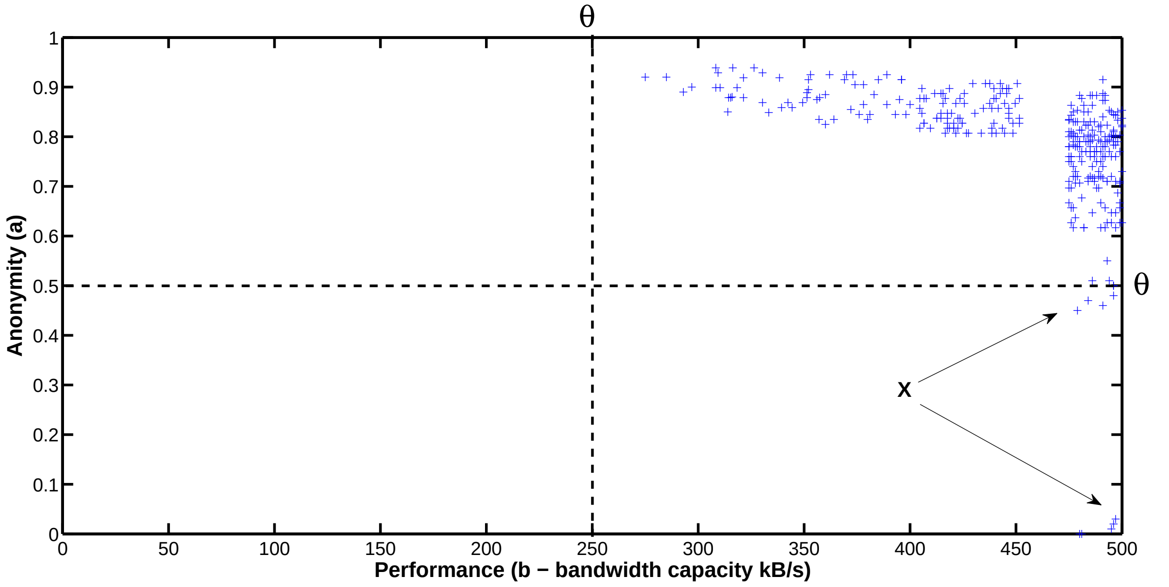


Fig. 4.8 The effect on anonymity and performance, from the network churn of mobile users recycling connections, showing the distribution of q with critical events ($q = 0$) in bottom-right quartile (highlighted by X)

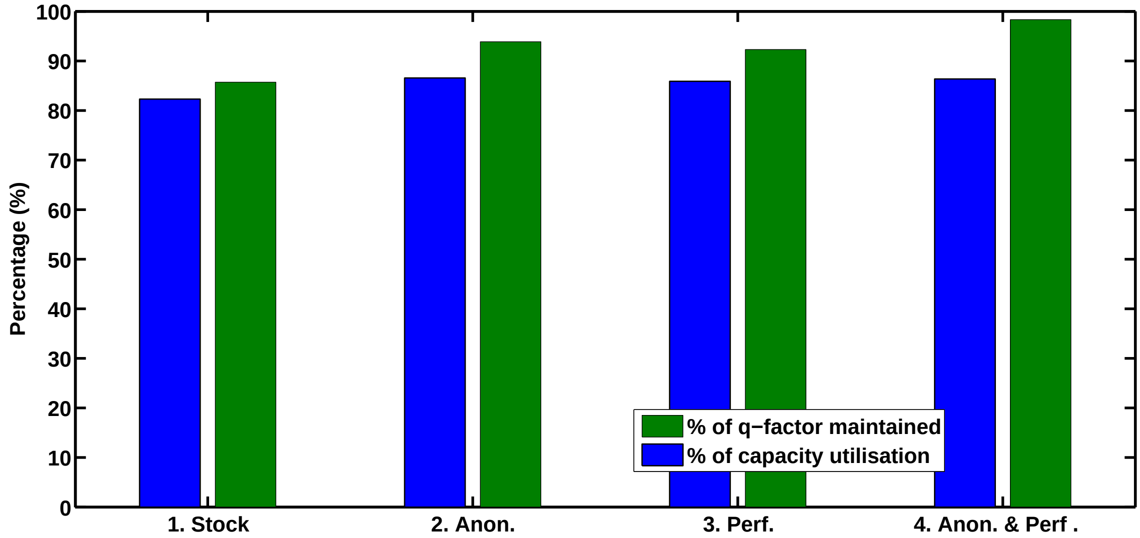


Fig. 4.9 A summary of the overall performance, for each of the single-layer (v1) schemes, as the percentage of positive q-factor achieved and network capacity utilization

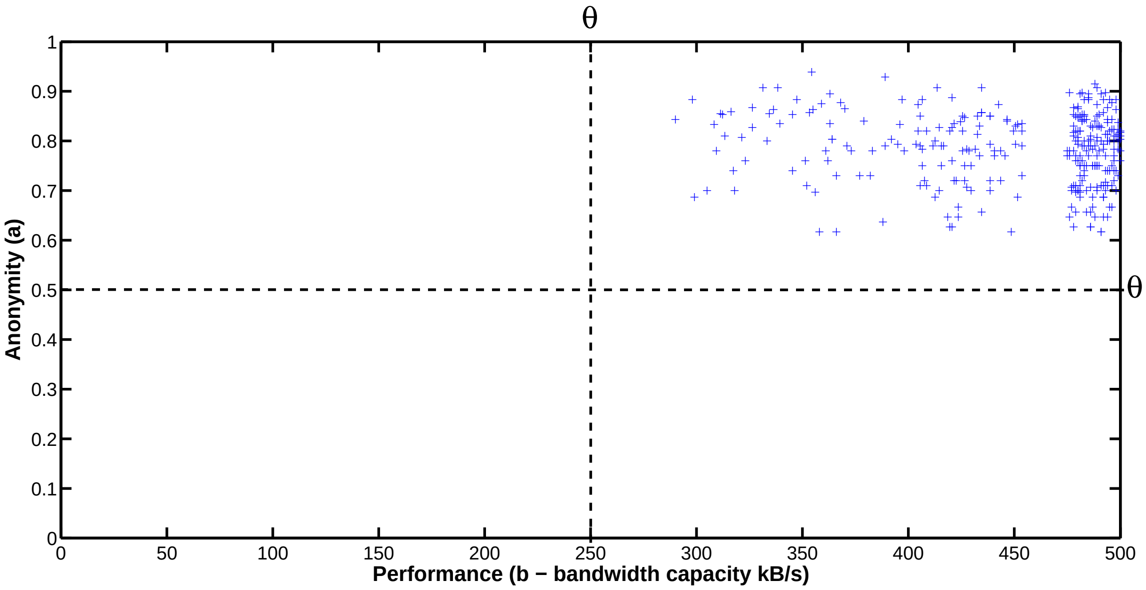


Fig. 4.10 The results from scheme 4 (v1), with full intervention for both anonymity and performance, showing the elimination of the previous critical events in Figure 4.11

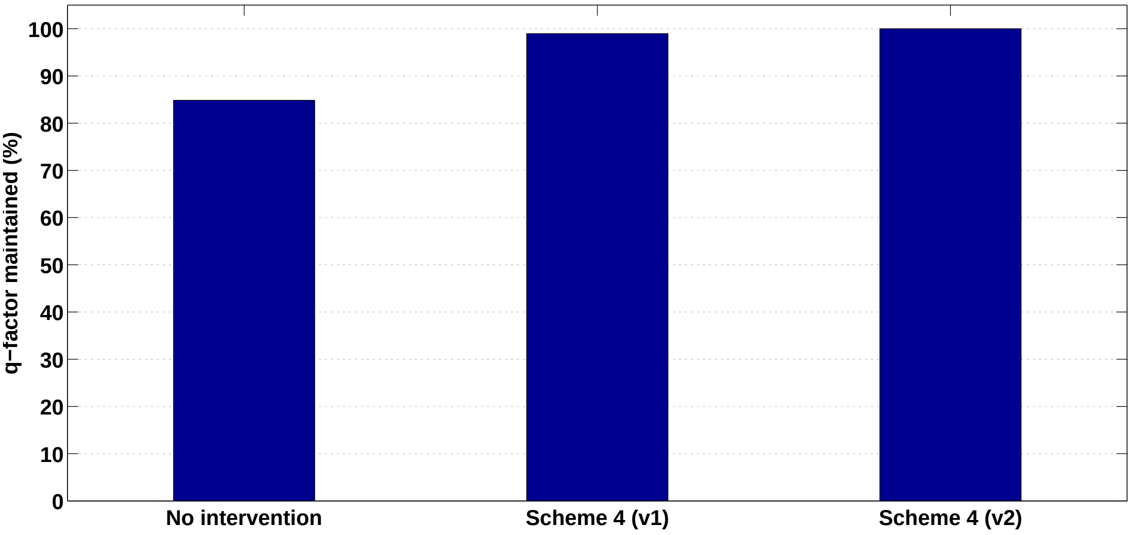


Fig. 4.11 The percentage level of positive q values maintained by scheme 4 of the double-layered version (v2) of q-factor compared to the single-layered version (v1) and 'no intervention' (scheme 1)

In Figure 4.10, the effectiveness of full intervention is shown. The cases previously highlighted (X) in Figure 4.8 have been eliminated with the distribution now completely secured within the top-right quartile (q = 1). The overall percentage of borderline cases where the level of anonymity is either stabilized at the threshold (a = 1) or fully recovered (a > 1) is 83.92%. This compares to no intervention where only 1.25% of cases are resolved 'organically' through natural network churn. In summary, scheme 4, with intervention for both anonymity and performance, provides the best overall performance, and stock (scheme 1), with no intervention, performing worst.

The results of the double-layered version (v2) of q-factor provides a final step up in effectiveness compared to the single-layered version. The 'buffer zone' delivers more timely intervention as shown in the results. The enhanced q-factor resolves the previous cases where q-factor was not maintained, as shown in Figure 4.11. The percentage of observations where q-factor is maintained increases from 98.32% to 100.00%. A breakdown of the status distribution of observations is shown in Figure 4.12. The 'goldilocks' zone (green) is maintained in 83.70% of the observations. In the remaining observations, 13.03% fall within amber and the remaining 3.27% in single red. It is important to note that a single red flag maintains a positive q-factor even though deemed at risk. However, this is just an

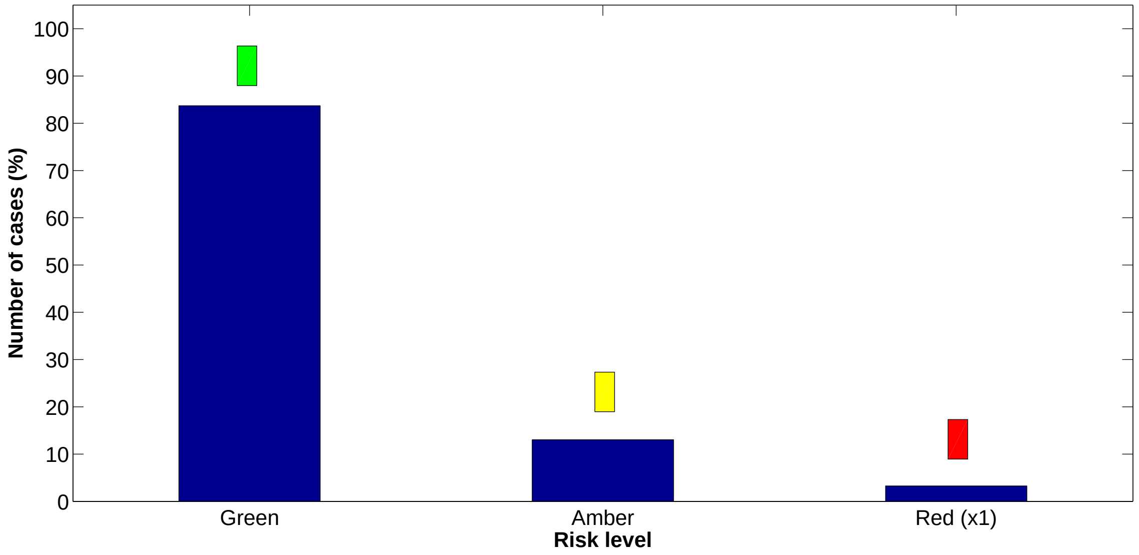


Fig. 4.12 The risk levels of cases where positive q-factor is maintained, at the lower thresholds $v_a \equiv a > \theta_a$ and $v_b \equiv b > \theta_b$, that includes a single red status as previously shown in Figure 4.9

example and may not be beneficial depending on operator's requirements if the overhead of the intervention is greater, say than, the predicted reduction in performance.

4.4 Discussion

The impact of mobility, while also trying to maintain the critical balance between anonymity and performance, is a clear issue for low-latency anonymity networks such as Tor. Increasing network churn, generated by a growing mobile client base, negatively impacts performance, and affects anonymity. The current intervention used by Tor, only applied to maintain performance through path selection algorithms for circuit building, and only undertaken reactively, is still a valid approach, due to the relationship between performance and anonymity, and based on the findings of the evaluation of q-factor, better than no intervention at all. The q-factor metric, presented for the first time within this research, contributes to a solution to these problems. The q-factor metric allows the operator to combine anonymity and performance into a single measurement that weighs the trade-off between the two.

A final consideration is the potential cost of intervention. In version 1 of the q-factor metric, during the scenario, the mean number of interventions generated was 28.24 per user. In version 2, this figure is 26.06% lower, at 20.88 interventions per user. This reduction is likely due to the fewer observations at the lower thresholds initiating mandatory intervention. However, if the redirection overhead is one second, based on the previous number of interventions, this may reduce the system availability around 4%. The decision of the operator is whether the benefit of any intervention exceeds the cost. Therefore, for the subsequent evaluations in chapters 5 and 6, the more basic version (v1) of the q-factor metric will be applied to illustrate its effectiveness as a metric at its most pessimistic. Additionally, q-factor will be used just as a metric, that is, **no** intervention is undertaken to provide a clean assessment of the effectiveness of each of the solutions.

4.5 Summary

The increasing mobility of Internet users is becoming an emerging issue for low-latency anonymity networks. The network churn, arising from an increasing mobile client base, can generate a negative impact on performance, and affect anonymity, when observed over time. Metrics used for low-latency anonymity networks, such as Tor, are limited and do not reflect an increasing mobile client base. A new metric, q-factor, is presented that combines measurements of anonymity and performance into a single measurement that is suitable for evaluating the quality of highly dynamic networks. By applying q-factor, it is possible to anticipate and significantly reduce the number of these critical events in which either anonymity or performance falls below an acceptable level. The wider impact of any proposed model, whether q-factor or a similar approach, however, needs careful consideration. For example, the overhead of redirecting users, as in Tor, may be a few seconds for the client to build a new circuit on a different path. Intervention may also generate 'odd' behaviour on the network. During the evaluation, the oscillation of users between channels, at each interval, was observed. This may significantly impact performance if the redirection overhead is too high, and therefore, the configuration of q-factor needs to be appropriate for the situation. However, adopting a more proactive approach to counteracting the impact of mobile usage, using real-time intervention, low-latency anonymity networks such as Tor will be able to continue to provide an essential privacy enhancing service in the foreseeable future.

Chapter 5

Throttling the Impact of Mobility

Performance issues, such as those arising from bulk downloads, continue to be an active area of investigation for anonymity network operators. For example, only 3% of connections generate over 40% of Tor network traffic [164]. Also, the use of the Tor network by 'botnets', to undertake distributed denial of service (DDoS) attacks, temporarily increased the number of users to nearly six million within three weeks [165]. This significantly impacted network performance by doubling download times for a 50 kB file from 1.5 seconds to 3.0 seconds [165]. Blocking 'misbehaving' users can be more difficult while trying to maintain anonymity [166] [167]. Adaptive throttling, to prevent some clients from using too much of the Tor's network resources, has already been proposed [159] [168]. If parity is reached between mobile and desktop connections, this will increase network churn on the Tor network and generate similar performance issues. This leads to whether the use of client throttling can also help mitigate the impact of mobility. In this chapter, a Kaplan-Meier estimator is evaluated for its suitability in mitigating the impact of mobility [169]. The Kaplan-Meier estimator, originally applied in medicine for predicting patient survival rates, provides a more refined adaptive throttling scheme, by aligning the probability of hand-off with the throttling.

5.1 Adaptive client throttling

Adaptive client throttling has been examined by AlSabah et al. to improve the overall traffic flow on the Tor network [116]. A proposal was to replace Tor's existing fixed circuit window with a more dynamic approach, that is, using an additive increase multiplicative decrease (AIMD) feedback control algorithm. The solution utilizes the existing *sendme* message Tor 'control' cell that acts like a heartbeat for each circuit to keep it alive (connected) if no data is transmitting. By measuring the round trip time (RTT) of the *sendme* message, the level of congestion is estimated. The algorithm initially sets a small, fixed circuit window

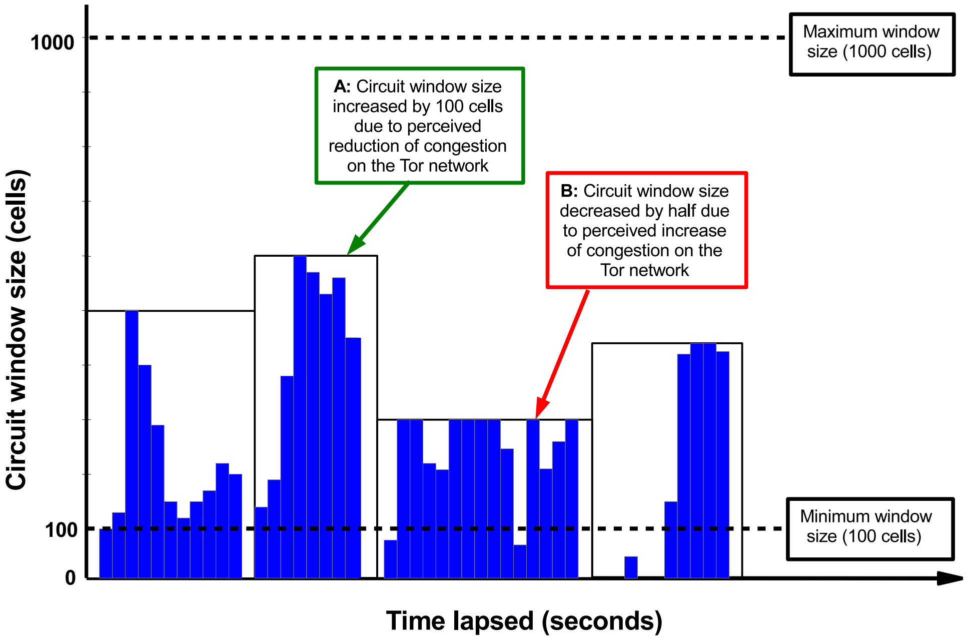
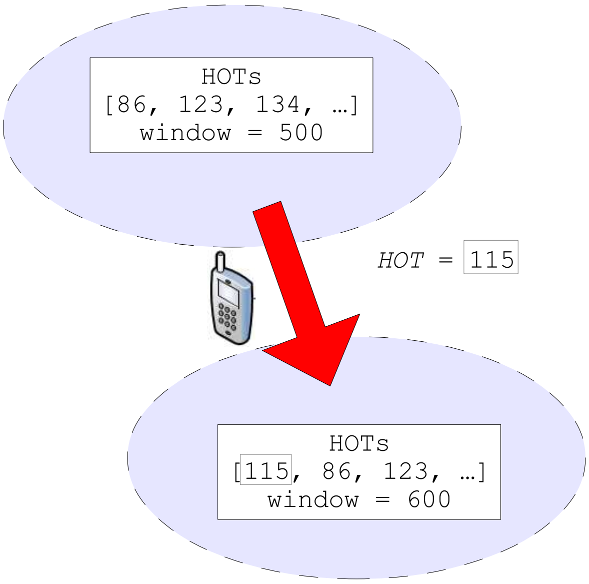


Fig. 5.1 Tor circuit window size adjusted by applying an AIMD-based throttling approach, as originally proposed by AlSabah et al. [116]

size of 100 cells (approximately 50 kB). Based on the perceived level of congestion, the circuit window size is adjusted accordingly between 100 cells and 1000 cells, as shown in Figure 5.1. AlSabah et al. evaluated the algorithm's effectiveness by measuring the time taken to download a 300 kB file. Two metrics used to measure performance were: 'time-to-first-byte' and 'time-to-complete' downloads. The performance from the adaptive throttling algorithm was compared against no throttling, and 'heavy' throttling using a fixed 100-cell circuit window. The results show that the AIMD throttling scheme provides a 40% improvement in overall performance when compared with the default stock Tor configuration with no throttling applied [116].

This prompted an investigation of the application of client throttling to the problem of the increasing mobility of Tor users. A simple throttling approach is to limit all Tor connections from mobile devices to a small, fixed-size circuit window of 100 cells, namely 'heavy' throttling. The heavy throttling approach should significantly reduce the mean total amount of data left in-flight across the Tor network when mobile users break their physical network connection.

A more dynamic approach is to adaptively throttle the circuit window using the AIMD algorithm. However, rather than based on congestion, a user's connection history including the number of, and time between hand-offs, could be a used a predictor of future reliability,



```
$\alpha = 0.25.$
$T = (1 - \alpha) \times HOT_{min} + \alpha \times HOT_{max}$
if $HOT \geq T$ then
    $window = window_{old} + 100$
else
    $window = window_{old} \times 0.5$
end if
```

Fig. 5.2 The adaptive throttling of Tor circuit window size, using an AIMD-based approach, based on hand-off intervals (HOT) instead of round trip times (RTT)

and therefore perceived ‘risk’ of hand-off. The idea is to assess the likelihood of a hand-off and to try to reduce the amount of data that potentially could be lost in-flight, at the next hand-off, by throttling the current data-flow proportionately to the level of risk. As shown in Figure 5.2, by simply replacing the RTT value with the time lapsed since the last hand-off (HOT), and comparing the latest HOT to values stored within the user’s history, the circuit window size can be adjusted accordingly. The hand-off history stores all HOTs, during the current Tor session, up to 600 seconds, the maximum life of a circuit, once used and all downloads complete. This is to maintain a degree of currency. The latest HOT is triggered, client-side, by the in-built Orbot logs that identify when a wireless network connection is lost and puts Tor into *sleep* mode, as shown in Chapter 2. On generating the latest HOT value, the history is reviewed, and in most cases, the oldest HOT value will be dropped from history and replaced with the latest HOT. The throttling algorithm is then invoked to calculate the new circuit window size as appropriate, to be applied once a connection to the Tor network is made again. The algorithm applies the constant (α) at $\alpha = 0.25$, as used by AlSabah et al., but could be adapted to suit any specific threshold requirements. It is important to note the inequality operator applied here ($\geq$) is a reverse of the operator ($\leq$) used in AlSabah et al. [116]. This is because, if the HOT is lower, then the perception is one of increased mobility and greater risk of hand-off and the algorithm needs to decrease the circuit window size. The adaptive throttling of circuit window sizes using an AIMD-based approach,

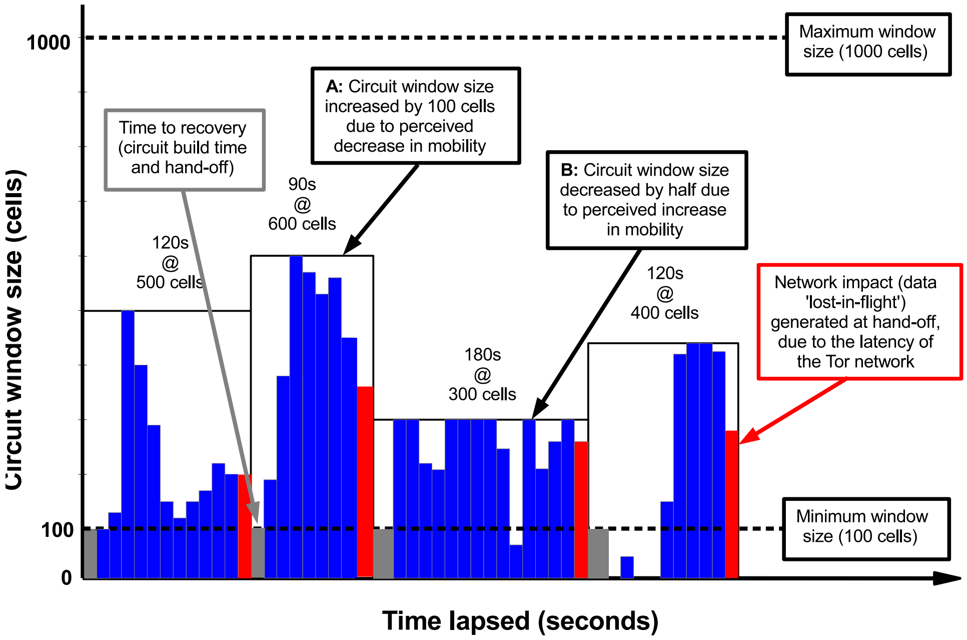


Fig. 5.3 A schematic illustrating the adaptive throttling of Tor circuit window sizes using an AIMD-based approach, based on hand-off intervals (HOT) while roaming

based on hand-off intervals (HOT) within a roaming context, is shown in Figure 5.3. The circuit window size is increased or decreased accordingly, within the range of 100 and 1000 cells, in relation to the change of mobility, and consequently risk of hand-off.

5.2 Introducing Kaplan-Meier estimator

Inspired by the challenges mentioned above, this section describes the development of a novel application of a Kaplan-Meier estimator for its suitability in mitigating the impact of mobility on low-latency anonymous communications [169]. The Kaplan-Meier estimator, originally applied in medicine for predicting patient survival rates in clinical trials, has had limited use in other domains, including computer science, in estimating the lifespan of network measurements on delay tolerant networks [170].

As previously mentioned, a simple approach to throttling is to limit all connections from mobile devices to a small, fixed-size circuit window of 100 cells, as previously proposed for bulk downloads. The current aim, however, is not to deter mobile Tor users but to mitigate the impact of the increased network churn. A dynamic approach to throttling, such as using an AIMD-based algorithm, although may be more appropriate, it could be too generalized and reactive for this particular issue. Therefore, it is decided to adopt a more sophisticated

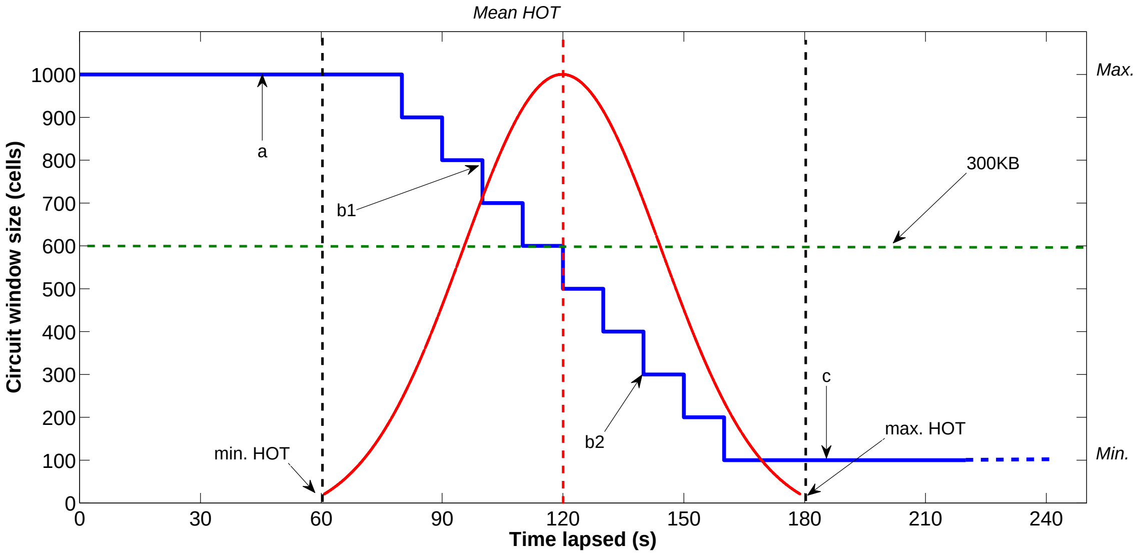


Fig. 5.4 A front-loaded Kaplan-Meier estimator based adaptive client throttling showing the three phases with indicative hand-offs as: a - an early miss, $b1$ - an early break, $b2$ - a late break, and c - a late miss

approach by predicting the probability of hand-off and throttle accordingly.

The proposed Kaplan-Meier estimator based approach has three distinct, sequential phases of 'risk' between each hand-off, as shown in Figure 5.4. The first phase is defined as the time elapsed to the lowest HOT value stored within the user's history. A mobile user is usually expected to maintain a connection for the full duration of this phase. Therefore, it can be considered relatively safe, during this phase, to afford the mobile user the maximum, and stock, circuit window size of 1000 cells. During the next phase, the 'zone of uncertainty' (incorporating the mean HOT value), the circuit window size is slowly reduced, at specified intervals for example, every second, until either a hand-off occurs or the maximum stored HOT is reached. The calculation is based on probability of a hand-off, during the next interval, occurring based on the minimum and maximum HOT values held within the history. For example, if the difference between the minimum and maximum HOT timings is 100 seconds, at the first interval after the minimum stored HOT has been reached, then the probability of hand-off is 0.01. The logic is shown in Algorithm 1. The probability continues to steadily decline at each interval for example, 0.02, 0.03, until either the connection is broken or the final phase is reached. The user is then considered at high risk of hand-off and subsequently heavily throttled at 100 cells.

The aim of the Kaplan-Meier estimator based approach is to maintain 'reasonable' client performance, while also reducing the amount of impact generated, while the user is roaming

Algorithm 1 The Kaplan-Meier estimator based approach for adjusting the circuit window size between hand-offs

```
$WINDOW_{min} = 100$
$WINDOW_{max} = 1000$
loadHOThistory()
elapsed = 0
**while** connection_alive = **true**
every INTERVAL **do**
    elapsed = elapsed + INTERVAL
    **if** elapsed $\leq HOT_{min}$ **then**
        window = $WINDOW_{max}$
    **else**
        **if** elapsed $\in [HOT_{min}, HOT_{max}]$ **then**
            $window_{old} = window$
            window = $window_{old} - (\frac{INTERVAL}{HOT_{max}-HOT_{min}} \cdot window_{old})$
        **else**
            window = $WINDOW_{min}$
        **end if**
    **end if**
**end while**
updateHOThistory(elapsed)
```

and breaking connections to the Tor network. The approach adopts a number of techniques to achieve this goal. Front-loading of network resources ensures that even if the mobile user is roaming, they still operate ‘normally’ at the maximum circuit window size for a significant period between hand-offs. The aim of this front-loading is to increase the number of successful downloads achieved during the scenario, as shown in Figure 5.5. If the next ‘riskier’ phase is reached and the throttling initiated, the slow reduction in circuit window size, ensures that it is more than often large enough ($\approx$ 600 cells) to support the light browsing of webpages of approximately 300 kB in size. These techniques also help reduce the potential impact on the Tor network. The front-loading ensures there are more successfully completed downloads between hand-offs and therefore less resubmitted requests causing unnecessary additional load on the Tor network. If a hand-off occurs and the break in connection interrupts an ongoing download, the data remaining in-flight is on average probabilistically lower, and consequently the potential impact generated. This is due to efficient alignment of the circuit window size and likelihood of hand-off, and the slow tapering of network resources. This slow tapering of network resources is in contrast to the AIMD algorithm. After a few decreases, the AIMD-based approach may provide inappropriately low resources immediately after a hand-off, even when there is little or no risk of imminent hand-off.

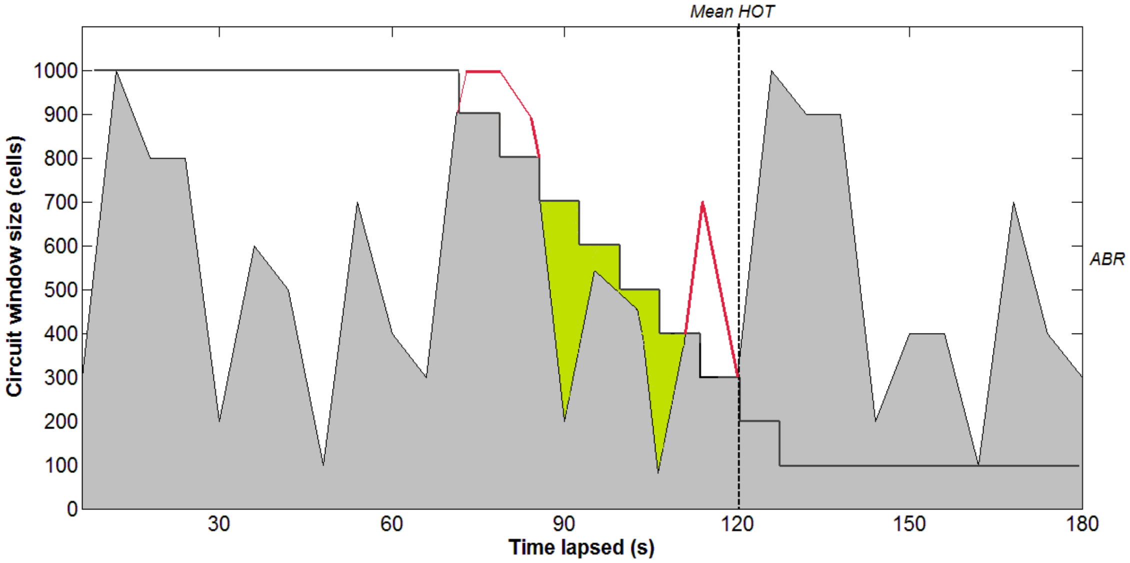


Fig. 5.5 The Kaplan-Meier estimator adaptive client throttling showing sample bursty traffic and indicative shaping

This could potentially lower client performance unnecessarily and may deter usage where eventually the number of concurrent users offers little or no anonymity. Additionally, the level of impact generated at hand-off, while using the AIMD algorithm, is more uncontrolled in contrast to the Kaplan-Meier approach.

5.3 Experimental Design

An extended version of the previous modelling, used to assess the impact of mobility, is adopted for this evaluation of client throttling. During this experiment, mobility speed is only applied at average walking pace, due to the previous findings showing, at higher mobility speeds, the use of Tor is no longer considered viable.

A range of schemes are assessed: ‘Stock’, ‘Heavy’, AIMD, and the new Kaplan-Meier estimator based approach. Stock reflects the existing default configuration of Tor with no client throttling applied using a static circuit window size of 1000 cells. The heavy throttling scheme uses a static circuit window size of 100 cells. As the aim is not to completely deter mobile Tor users, the AIMD scheme will also be applied at a more ‘optimistic’ starting window size of 1000 cells (AIMD-1000), in addition to the previously proposed 100 cells (AIMD-100). Based on the same theory, the Kaplan-Meier estimator based approach will also have a starting circuit window size of 1000 cells. Therefore, unless a break in connection occurs, the circuit window will remain at 1000 cells until the minimum hand-off time (HOT) stored is reached. After then, if still connected, reducing the circuit window size if until the maximum HOT in history, at which time the connection is then heavily throttled at 100 cells, until the next hand-off or the end of the scenario.

The scenario is again the author's daily walk to university. The brisk 35 to 40-minute walk (3.7 km) covers five different Internet service providers (ISP) and a number of wireless access points within those networks, usually generating 15 hand-offs in total, approximately every couple of minutes. Although the duration of the walk is longer, the analysis will only run for 10 minutes, due to an existing security feature of Tor, whereby all circuits that have been used and downloads are completed, are recycled. As a reminder, this feature is also incorporated within the design of the adaptive throttling algorithms (AIMD, Kaplan-Meier estimator) which store a maximum HOT history of 600 seconds to maintain currency of the user's perceived current level of mobility. During the same time frame (t) of 600 seconds, the user is mobile and travelling at a constant velocity (speed and linear direction) as broadly in-line with the real-life scenario. The intervals between hand-offs (notated as h within the formula) are randomly generated by applying a Gaussian distribution. The distribution uses a standard deviation of $\sigma = 25$, within the range of 60 and 180 seconds, again closely emulating the hand-offs during the walk. The same algorithm is used to generate an indicative hand-off history before the scenario commences.

The next key component is the time to recovery, which consists of both the physical hand-off between networks and the time to rebuild of circuits on the Tor network. Once an array of HOTs is generated, each item in the array requires adjusting to generate a net value, that is the actual time available for downloading files after each time to recovery is subtracted. For each net HOT, a number of factors need to be considered and implemented. Traffic load needs to consist of the time between, or frequency of, each request (r) and request size, the latter denoted as S, to distinguish from seconds (s) which is used as a component of the average bitrate (ABR) metric. A time-to-first-byte (f), to reflect the 'think time' of the web server, and the time-to-download (d) are also applied based on empirical data from other research on Tor performance issues [159]. The introduction of throttling also requires weighting to be additionally applied to the previous modelling. For example, if a 300 kB file is downloaded using the default stock Tor circuit window size of 1000 cells (approximately 500 kB), it is expected this will be quicker to complete than if downloaded with a 100-cell circuit window. This is because a significantly smaller amount (50 kB) of data is allowed in-flight for the circuit at any given time. Therefore, the core download time is adjusted (a range from 0.6 to 6 $\times$) based on the size of the current circuit window size (w), and to also compensate the increased delay in successfully receiving the file, the time between requests will also be adjusted accordingly. The parameters are a time-to-first-byte of f = 3 seconds and the unweighted time-to-download of d = 7.5 seconds. The modelling focuses on indicative circuit build times of 3, 7, and 10 seconds, as timings for 'good', 'average' and 'slow' respectively [127].

The calculation of client performance, measured in goodput (g), for a specific HOT (i), including any partial downloads, is described in Equation 5.1:

$$g(i) = \frac{h-(y+k)}{r+f+(d\cdot(\frac{S}{w}))} \tag{5.1}$$

The number of successful downloads is calculated, using the *floor* function, to transform the real number to the previous smallest integer, as $g \rightarrow \lfloor g \rfloor$. This rounding down is applied to all HOTs, both full and partial, to generate the total number of downloads during the time frame (t) of the scenario. Finally, the overall performance (b) is derived by the number of downloads multiplied by the request size (S), and a factor (x) and divided by t to generate the average bitrate (ABR). The calculation to determine the overall performance during the scenario is described in Equation 5.2:

$$b = \frac{\sum_{i=n}(\lfloor g \rfloor(i))\cdot S\cdot x}{t} \tag{5.2}$$

The indicative threshold for performance is an ABR of 150 Kibit/s or above, with the theoretical maximum at just under 250 Kibit/s. Although the performance threshold is arbitrary, it is a fair representation of an acceptable level of performance required to comfortably undertake normal web browsing, normal here described as downloading files of an average webpage size of approximately 300 kB.

In addition to the level of client performance achieved, the overhead or total amount of bad data (o), generated also needs to be calculated, that is, incomplete downloads and data remaining in-flight, which is calculated as described in Equation 5.3:

$$o = \sum_{i=1}^{n}(g) - \sum_{i=1}^{n}(\lfloor g \rfloor(i))\cdot S \tag{5.3}$$

The primary aim of the evaluation is to compare the overall effectiveness of different throttling approaches in mitigating the impact of mobility while also maintaining an acceptable level of performance for the mobile Tor user. In addition to the number of successfully completed downloads, represented as an ABR, this trade-off is also evaluated as the economy. Economy is basically the ratio of 'good' (g) and 'bad' (o) bits of data. The adaptive throttling approaches are expected to offer better economy than the static approaches, with the most sophisticated approach, that is, the front-loaded Kaplan-Meier estimator, providing the best overall effectiveness of all the throttling schemes. The schemes to be compared are as follows:

1. **Stock**: No throttling applied, a fixed 1000-cell circuit window size.

2. **Heavy**: Throttling applied at a fixed 100-cell circuit window size.

3. **AIMD-100**: Adaptive client throttling based on the current level of mobility and subsequent risk of hand-off. Dynamic circuit window size between 100 and 1000 cells, starting 'pessimistically' at 100 cells.

4. **AIMD-1000**: Adaptive client throttling based on the current level of mobility and subsequent risk of hand-off. Dynamic circuit window size between 100 and 1000 cells, starting 'optimistically' at 1000 cells.

5. **Kaplan-Meier**: A Kaplan-Meier estimator based approach with adaptive, and predictive, client throttling based on the probability of hand-off. Dynamic circuit window size between 100 and 1000 cells directly aligned to the probability of hand-off, starting 'optimistically' at 1000 cells.

The potential effect on anonymity also needs to be assessed as part of the overall effectiveness of each of the throttling schemes. Low-latency anonymity networks, such as Tor, can leak timing information [171]. In this example, traffic shaping through client throttling, lends itself to assessing the degree of 'indinymity' as the measure of anonymity. The distinguishability based metric, indinymity, allows the attacker to make probabilistic guesses from distinguishing feature or features between schemes [146]. The highest recorded peak of 'bursty' traffic may not be suitable. This is because both the Kaplan-Meier estimator and AIMD-1000 throttling schemes, alongside desktop users, are equally capable of achieving the maximum circuit window size of 1000 cells during the scenario. This may not help an attacker undertake the partitioning of mobile and desktop users into different subsets. Therefore, the attacker decides that the time taken to download the file, and variance between the performance achievable from a desktop connection (unthrottled) and mobile (throttled), will be used. Throttling schemes may also generate distinctive features, for example, the AIMD-based approaches halving the circuit window size. Therefore, rather than using the mean, a cumulative approach may be more appropriate. If AIMD-based throttling scheme is used, within a few re-calculations the circuit window size may recover sufficiently to appear no different from a desktop connection. The resulting variance of the time taken over the

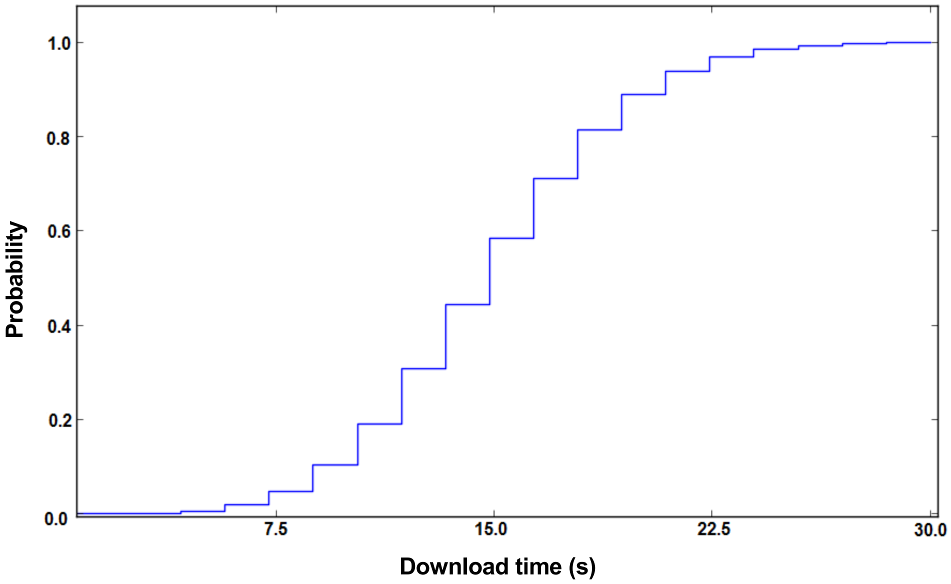


Fig. 5.6 The probability of a throttled (and mobile) user based on download times by applying a cumulative distribution function (CDF). The training data is based on the parameters previously used in Chapter 3

desktop, will be calculated by applying this cumulative based approach. Coincidentally, the Kaplan-Meier estimator adopts an inverse cumulative distribution function (CDF), that is, the survival rate diminishes over time. This feature, alongside the front-loading component, may also benefit the resistance to traffic analysis.

The attacker undertakes the preparatory analysis to generate a 'training' data set. From this data, the attacker determines a desktop connection will take on average just over Δ_x seconds and from a throttled mobile device Δ_y. The attacker begins to observe the connection and compares the actual download times and assigns a probability based on its value between Δ_x and Δ_y. In Figure 5.6, the training data distribution, with indicative markers, is shown from which the attacker applies the probability. The thresholds applied for this first stage of the evaluation are again $b = 150$ Kibit/s for client performance, and anonymity of $a = 0.5$. The threshold for anonymity infers that the connection from the mobile user appears no different from a desktop connection. A table containing the key parameters can be found in Table 5.1.

	Experiment in §:	3.2	3.3	3.4	4.2	**5.3**	6.2
ANONYMITY:					-		
Possinymity:	Attack A-0					✓	
	Attack A-1					-	
	Attack A-2					-	
	Attack A-3					-	
	Attack A-4					-	
	Attack A-5					-	
Indinymity:	Attack A-6					✓	
	Attack A-7					-	
PERFORMANCE:							
MTTF:	Speed (mps)					-	
	HOTs (s)					0...600	
MTTR:	Physical (s)					1	
	Circuit (s)					3, 7, 10	
	Redirection (s)					-	
Load:	File size (KiB)					300	
	Wait (s)					2	
	Request (s)					2	

Table 5.1 The key parameters used for the evaluation of the client throttling schemes.

5.4 Results

As expected, the existing default stock Tor configuration, with no throttling applied, achieves the highest level of client performance, as shown in Figure 5.7. The aim of this evaluation, however, is to identify a throttling approach that provides both a reasonable level of client performance while also reducing the potential for impact on the Tor network, from data left in-flight and resubmitted requests. As previously mentioned, although the performance threshold is arbitrarily set, the heavier throttling schemes suffer greatly from a performance perspective. In fact, for both the Heavy and AIMD-100 throttling schemes, none of their test cases achieved the required performance threshold, compared to 80% for AIMD-1000 and 100% achieved by the Kaplan-Meier estimator based approach.

At a circuit build time of 7 seconds, the application of heavy throttling, as previously proposed for deterring bulk downloads, generates a substantial 85.65% drop in client performance compared to the default stock configuration of Tor. Applying adaptive throttling provides better performance. However, using the AIMD-100 scheme still appears to be too aggressive for mobility with over a 71.43% drop in performance. After the AIMD algorithm is modified, to start at 1000 cells, performance improves by a factor of 2.8 compared to the AIMD-100 scheme. The best performance provided by the throttling approaches is from the

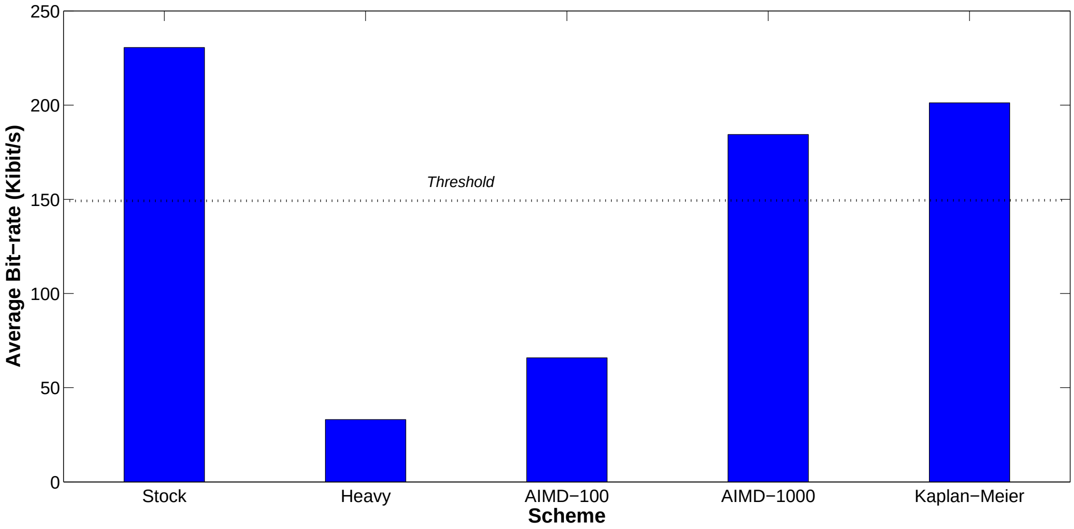


Fig. 5.7 The client performance achieved for each scheme at a circuit build time of 7 seconds against a threshold of 150 Kibit/s

Kaplan-Meier estimator, performing 10.90% better than its nearest competitor (AIMD-1000).

In Figure 5.8, the impact of different circuit build times on client performance is shown. The results are as expected, with the circuit build times indicative of 'good', 'average', and 'slow', showing the longer the circuit build time, the lower the performance achieved.

In Figure 5.9, the chart also presents the goodput and bad data generated (as total data received), side-by-side, for each of the schemes. The Kaplan-Meier estimator based approach generates 8.33% lower amounts of bad traffic than Stock.

Figure 5.10 provides an alternative view on the data for the economy each scheme provides. Economy is the ratio of good versus bad bits of traffic generated. The Kaplan-Meier estimator based approach provides the best economy at 19.06:1 nearly that of Stock (20.02:1) but with lower bad data generated, as previously highlighted in Figure 5.9. It is followed by AIMD-1000 at 16.09:1, then dropping considerably to 5.80:1 for AIMD-100 and just 2.85:1 for the heavy throttling scheme.

The results of the impact on anonymity, mapped alongside performance, from the best performing throttling schemes are shown in Figures 5.11 and 5.12 (N = 200 per scheme). The Kaplan-Meier estimator based approach offers the best anonymity protection at 100% of all cases where the anonymity threshold is achieved. This, compared to its best competitor (AIMD-1000), which for this particular attack, only manages to maintain anonymity above the threshold for 55.5% of the observations during the scenario. Another point to note is that although AIMD-1000 achieves an overall mean anonymity of a = 0.50, there are cases where

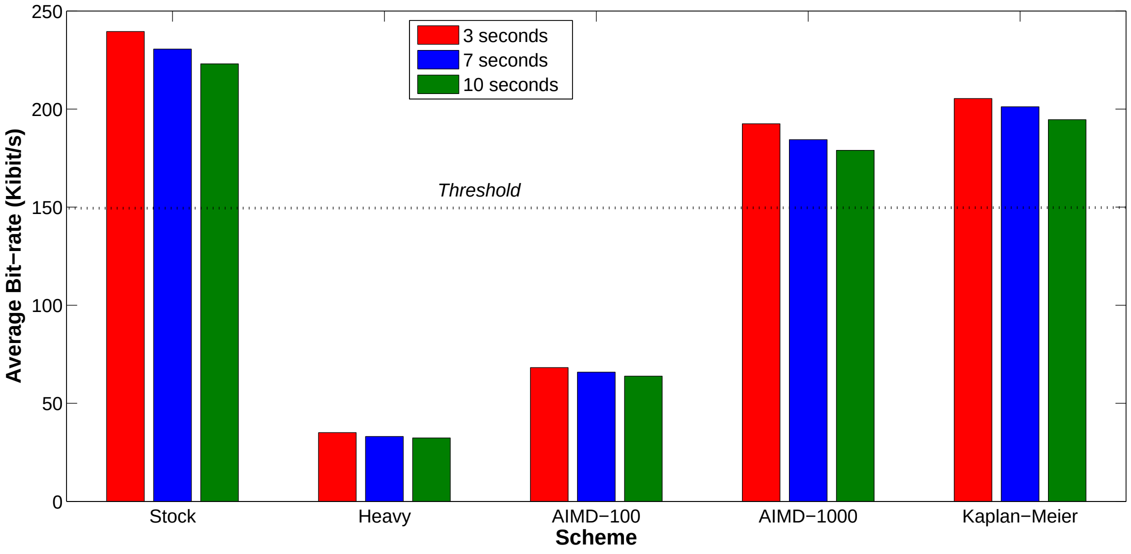


Fig. 5.8 The client performance achieved for each scheme at different circuit build times

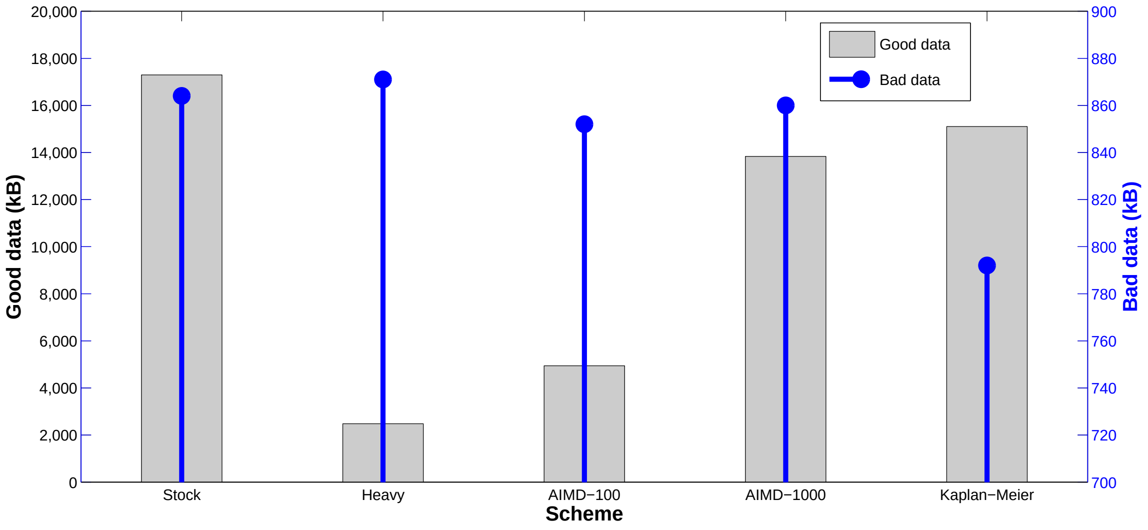


Fig. 5.9 The mean total amount of good and bad data generated for each scheme at a circuit build time of 7 seconds

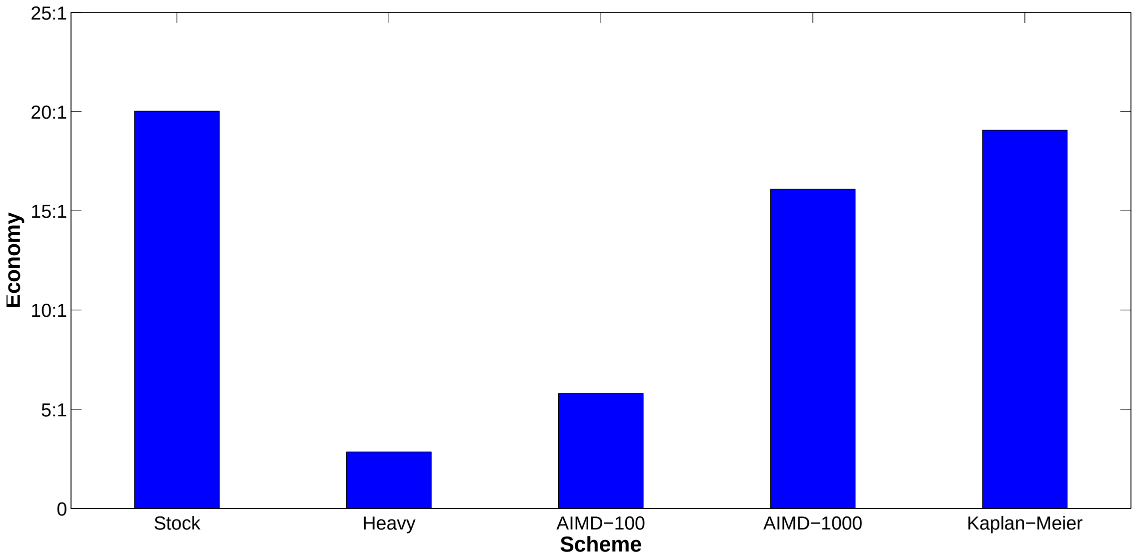


Fig. 5.10 The mean economy, the ratio of good and bad data, generated for each scheme at a circuit build of 7 seconds

anonymity falls below the required threshold, and the user is at risk of being exposed. This again highlights the key consideration, as previously discussed, when evaluating solutions relating to anonymity, and the benefit of applying more suitable metrics such as q-factor.

5.5 Discussion

In summary, the Kaplan-Meier estimator based approach provides the best overall solution to mitigate the potential impact of mobility on the Tor network, while maintaining reasonable client performance. As expected, the existing default stock Tor configuration, with no throttling applied, achieves the highest level of client performance, but also generates the highest impact than the adaptive (not heavy) throttling schemes. The heavier throttling schemes (Heavy, AIMD-100) have such a high impact on performance, it is predicted the throttling may unnecessarily deter mobile users from using the Tor network, and deemed unsuitable for supporting mobility. The likely reason behind the poor performance of the AIMD-100 scheme is illustrated in Figure 5.13. A sample of circuit window sizes during the scenario found that the circuit window size rarely exceeds halfway the maximum 1000 cells. Although the AIMD-1000 scheme fairs better, unfortunately the algorithm still generates too many occasions where the circuit window size is halved. This consequently requires a degree of chance to reach a reasonable level again by the time the scenario has completed. The

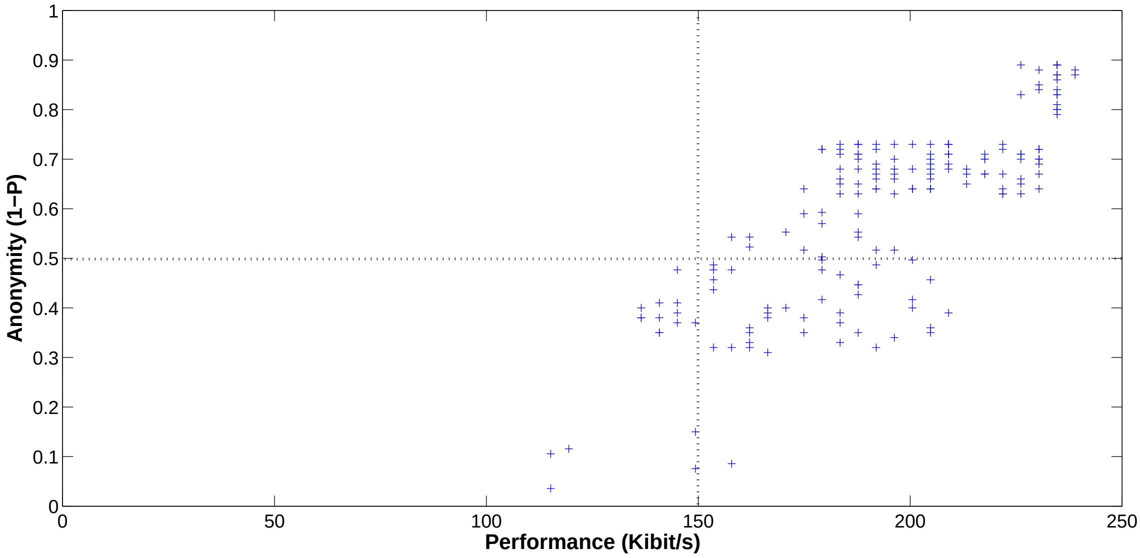


Fig. 5.11 Mapping of anonymity and performance, at a circuit build time of 7 seconds, for the AIMD-1000 scheme

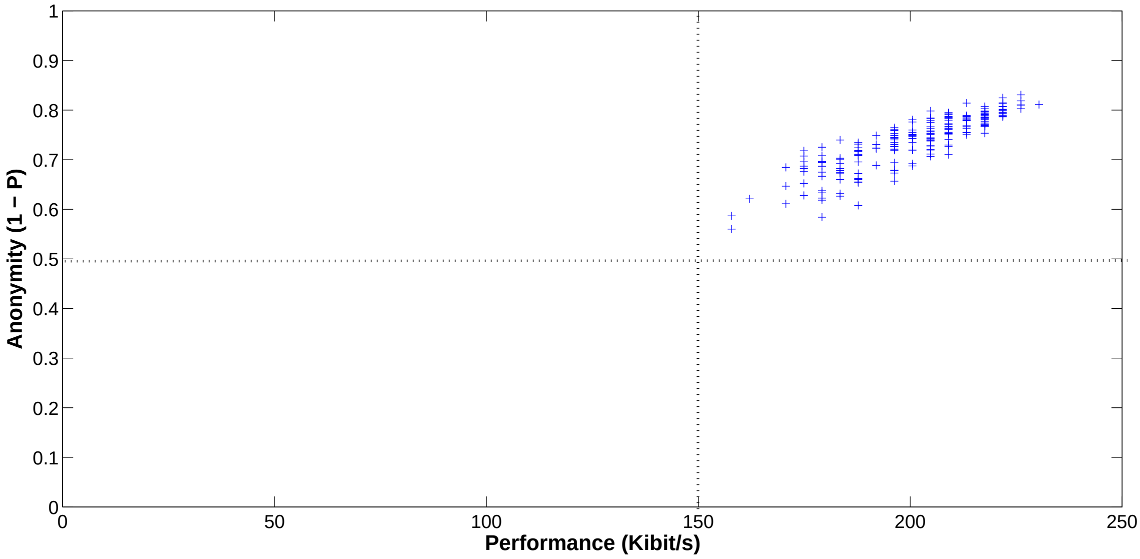


Fig. 5.12 Mapping of anonymity and performance, at a circuit build time of 7 seconds, for the Kaplan-Meier scheme

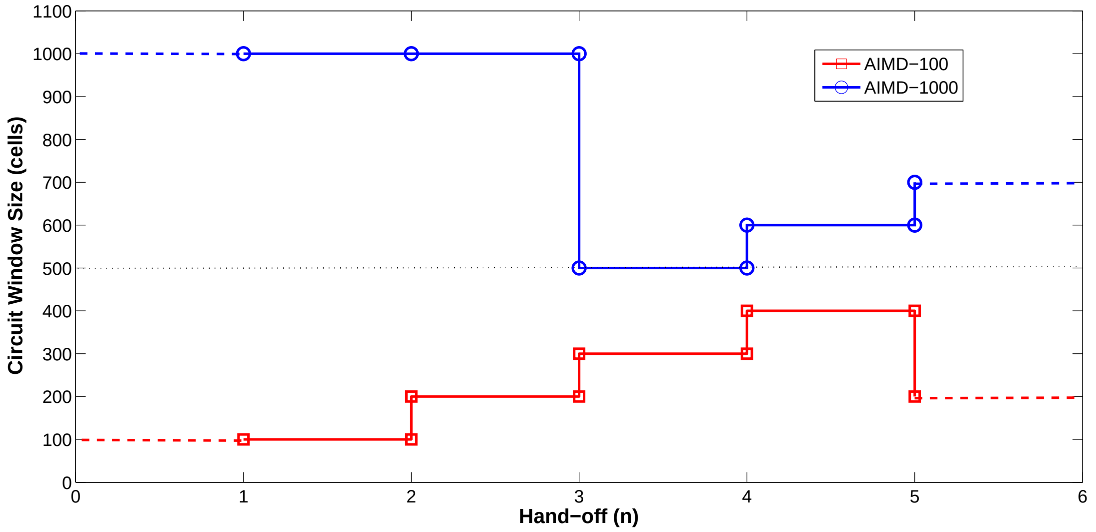


Fig. 5.13 An indicative sequence showing the AIMD-100 scheme failing to exceed halfway (500 cells) and the AIMD-1000 scheme suffering heavily from a multiplicative decrease at hand-off

Kaplan-Meier estimator based approach appears to be more stable, and appropriate, when applied to mobility.

The client performance achieved by the more optimistic adaptive throttling schemes (AIMD-1000, Kaplan-Meier estimator), although performed far better than the heavier throttling schemes, did not match performance of the default stock Tor configuration. As previously discussed, the aim of this study is not to achieve the highest level of performance. It is to identify an approach that can provide an acceptable balance, or economy, between maintaining reasonable performance and mitigating the potential impact on the Tor network from the increasing mobility of users.

The Kaplan-Meier estimator based approach, as an adaptive throttling scheme, in theory, is effective in mitigating the impact of mobility. The Kaplan-Meier solution, however, may be more suited to highly dynamic environments, that is, higher mobility than average walking pace of humans. The weakness, in certain scenarios of the current Kaplan-Meier based implementation is illustrated in Figure 5.14. The algorithm is unable to efficiently handle the lower mobility. If Alice stops handing-off during the scenario, for a significant amount of time, then her connection is heavily throttled even though she may be at no imminent risk of handing-off. Consequently, this causes unnecessary throttling and a significant reduction in client performance. It may be that this is enough to deem what is considered reasonable performance unreasonable. To mitigate this, any subsequent implementation of

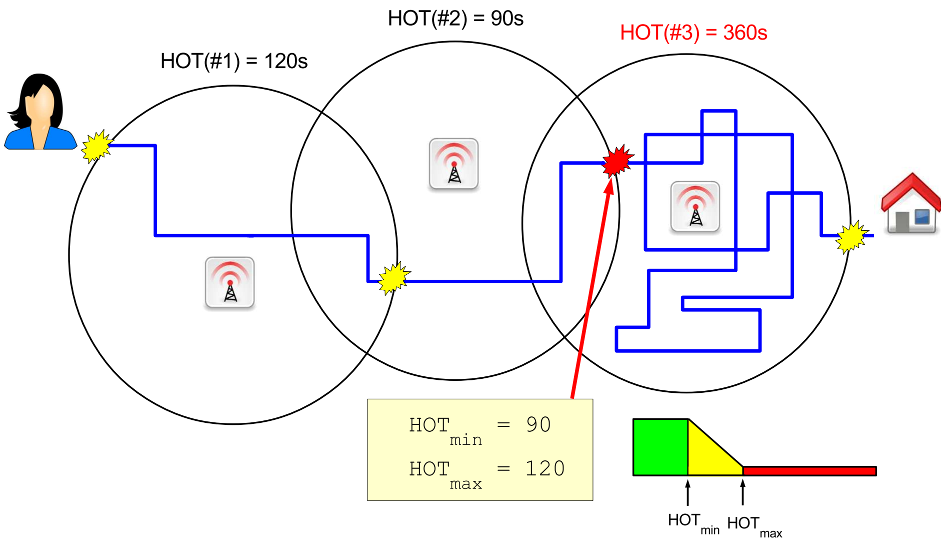


Fig. 5.14 A hypothetical scenario of Alice walking around within the same geographical area and the consequences of over-throttling by the Kaplan-Meier algorithm using the *HOT* history (min./max.) against current connection time

the Kaplan-Meier algorithm would also likely need to identify if the mobile user is static (or slowing down considerably) and adjust the circuit window size accordingly.

Based on the preliminary evaluation of the Kaplan-Meier based implementation, the following enhancements are proposed:

1. A manual on/off switch for roaming mode to be added to the Kaplan-Meier solution, and/or;

2. Interface with the smartphone's in-built geolocation/Global Positioning System (GPS) to identify when Alice has stopped or, in more complex versions, is slowing down, and/or;

3. As an additional check within the algorithm, review if the interval between hand-offs extends a certain threshold beyond the maximum *HOT* stored within the history, and adapt the throttling accordingly.

These are only proposals at this stage but if a suitable implementation can be identified, then one or more are worth considering for evaluation. For the first proposed enhancement, the requirement for any human intervention is not always an ideal solution. The old adage remains

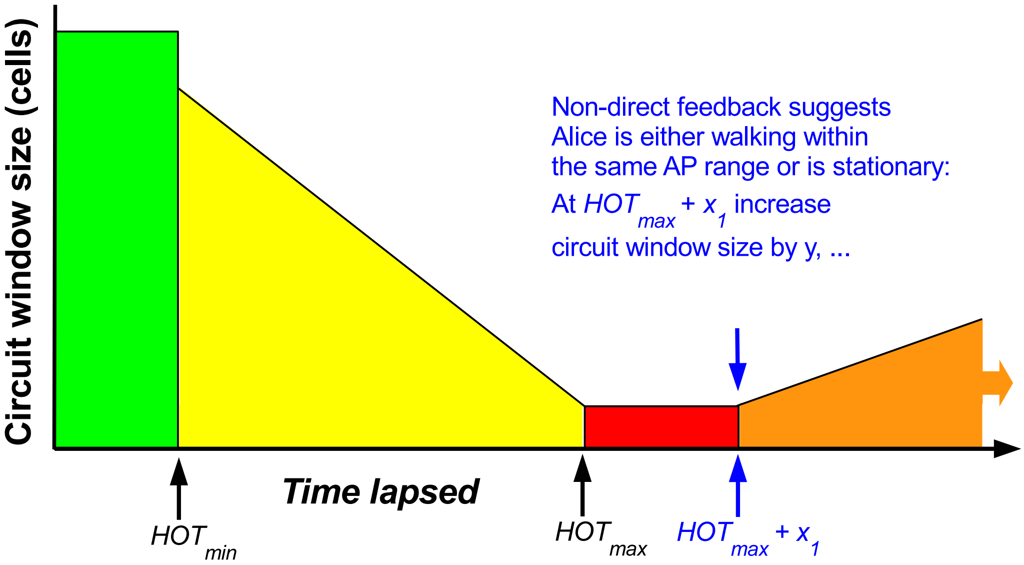

Circuit window size (cells)
Non-direct feedback suggests
Alice is either walking within
the same AP range or is stationary:
At $HOT_{max} + x_1$ increase
circuit window size by y, ...
Time lapsed
HOT_{min}
HOT_{max}
$HOT_{max} + x_1$

Fig. 5.15 A proposed enhancement to the Kaplan-Meier based scheme where *indirect* feedback suggests Alice is either walking within the same area or is stationary, allowing the circuit window (resources allocated) to be adjusted higher during the red phase

that usually the weakest link in any technological solution is the user (human). Interfacing a smartphone's in-built geolocation/Global Positioning System (GPS) may provide enough information regarding the level of mobility to make more effective decisions. This does, however, raise privacy concerns from the additional location tracking. Also, the potential larger attack surface, that is, as previously seen with recent high profile cases of Tor having been exploited through operating system (Linux) and software (web browser) vulnerabilities, add further concerns. For example, if there is a vulnerability within the GPS functionality that is exploitable by an attacker, there is little the anonymity network operator can do to protect its users, until the third-party can fix the vulnerability.

Alternatively, implementing changes to the existing algorithm could improve the effectiveness of the Kaplan-Meier adaptive throttling approach. Firstly, an additional check could be introduced to flag if the time that the connection has been heavily throttled exceeds a threshold (x_1), as shown in Figure 5.15. However, it still remains difficult to establish whether the user, Alice, is about to hand-off any time soon, if walking slowly around the same area, that is, within the same access point range, or has stopped. The key is maintaining a balance between providing Alice more resources (reasonable performance) while also being careful not to provide too much in case the user is about to hand-off.

5.6 Summary

The use of client throttling, previously proposed to deter bulk downloads and botnet abuse on the Tor network, is evaluated for mitigating the impact of mobility of users. The previously proposed algorithms are unsuitable, throttling too heavily, to the extent they could deter mobile users unnecessarily. Applying a Kaplan-Meier estimator based approach, by aligning the probability of the next network hand-off directly to the level of throttling, reasonable client performance is maintained, impact on the Tor network reduced, while also providing resistance to timing-based traffic analysis. The advantage of this approach, over other solutions, is that one of the key principles of the Tor design is maintained, that is, the user will continue to retain full control of any decisions over its traffic management and flow for example, the circuit window size. Although, more sophisticated approaches to adaptive client throttling, such as the Kaplan-Meier estimator, may offer a solution for mitigating the impact of mobility, there is some doubt whether this is the right developmental path. To maintain a reasonable level of client performance, throttling schemes need to become more and more complex. Also, any reduction in impact may be less beneficial than expected, as in reality, whether 1 kB or 100 kB is blocking the relay buffer may be fairly irrelevant. Therefore, although an interesting idea, adaptive client throttling may not offer the most effective solution in mitigating the impact of mobility. An architectural change, that offers a persistence of connection to the Tor network for roaming mobile users, may be more appropriate, and will be examined during the next chapter.

Chapter 6

mBridge: Persistent Connections for Mobility and Anonymity

The use of adaptive client throttling, previously proposed to deter bulk downloads and botnet abuse on the Tor network, can also be applied for mitigating the impact of mobility. A front-loaded Kaplan-Meier estimator based approach was identified as the most effective throttling scheme for supporting mobility. By aligning the probability of the next network hand-off directly to the level of throttling, reasonable client performance is maintained, impact on the Tor network reduced, while also providing resistance to timing-based traffic analysis. Although, more sophisticated approaches to adaptive client throttling, as seen with the Kaplan-Meier estimator based approach, may offer a solution in mitigating the impact of mobility, there remains a doubt whether this is the most appropriate solution. To maintain effectiveness the throttling schemes need to become even more complex. The reduction in impact may be less beneficial than expected, as in reality, whether 1 kB or 100 kB is blocking the relay buffer may be fairly irrelevant. Therefore, although an interesting idea, it is accepted at this juncture than adaptive client throttling may not be the most effective solution in mitigating the impact of mobility. An architectural change, that offers a persistence of connection to the Tor network for a roaming mobile user, may be more appropriate.

In this chapter, a novel application (mBridge) of the existing architecture for circumventing the blocking of access to the Tor network is presented. The aim of the mBridge solution is to provide a persistence of connection to the anonymity network for mobile users while they are roaming. The solution is evaluated for both performance and anonymity using the q-factor metric, compared to a default stock configuration and the Kaplan-Meier estimator based adaptive client throttling.

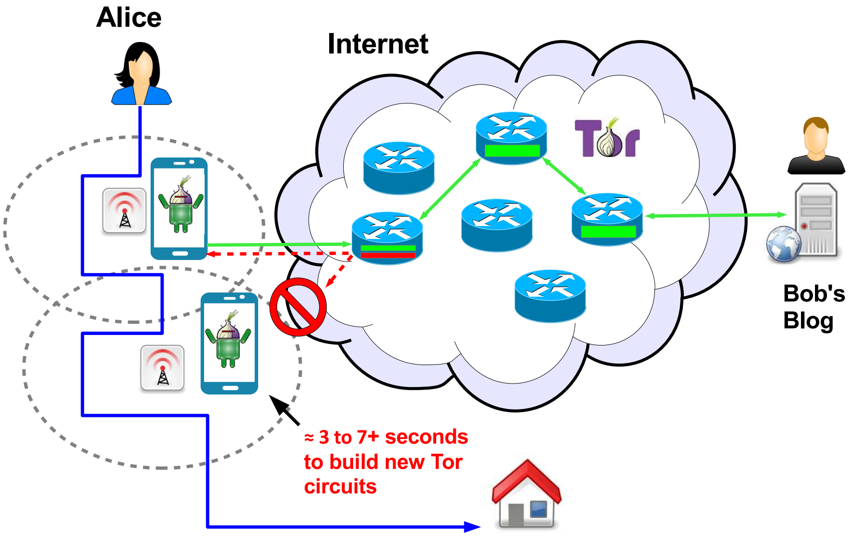


Fig. 6.1 The existing issue of a roaming mobile user breaking connection to the Tor network

6.1 Introducing mBridge

Anonymity networks, such as Tor, that use onion routing are unable to support mobility efficiently. Consequently, there is a significant negative impact on client performance for a mobile user, such as Alice, as illustrated in Figure 6.1. The negative impact not only reduces client performance for a roaming mobile Tor user. The data left in-flight and resubmitted requests after each hand-off may cause further congestion on the Tor network negatively impacting all users whether mobile or not. Wiangsripanawan, Susilo, and Safavi-Naini proposed three solutions for the supporting mobility of Tor users [29]:

1. A simple Mobile IP based approach, using ‘home’ and ‘foreign’ agents, redirecting data streams between an existing persistent Tor connection at ‘home’ and Alice’s ‘care of’ Internet Protocol (IP) address.

2. Changes to the Tor network at the entry point, to track changes in Alice’s external IP address, with supporting stop/resume control commands, to redirect the data streams as appropriate.

3. Changes at the exit point again providing the same stop/resume commands control commands based on a unique circuit identifier rather than Alice’s external IP address.

The first solution, as shown in Figure 6.2, although provides effective mobility management, does not afford location privacy, as the ‘care of’ address is revealed without appropriate

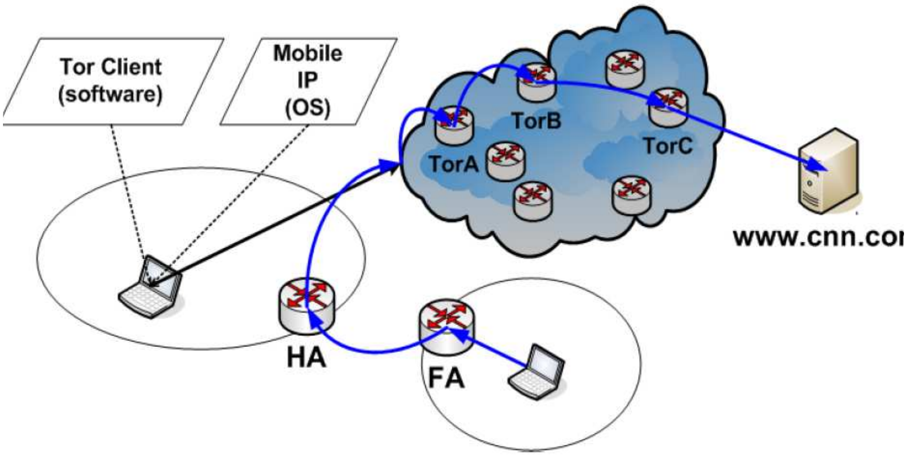


Fig. 6.2 A basic Mobile IP based architecture attached to the Tor network [29]

sanitization. Another concern is the additional ‘hops’ required for maintaining the connection end-to-end and whether this would significantly, and negatively, impact performance.

The second solution, also affords mobility management, with changes to Alice’s external IP address managed at the entry guard node, as shown in Figure 6.3. The entry guard undertakes an additional control process for roaming mobile users. A *STOP* control command is executed if Alice breaks her connection. Allowing the hand-off process to complete, on receipt of Alice’s new external Internet Protocol (IP) address, the entry guard executes a *RESUME* command, and redirects Alice’s outstanding data stream. This solution prevents half-open Transmission Control Protocol (TCP) connections, subsequent time-outs and additional congestion on the Tor network. As with the first solution, the authors raise concerns regarding privacy as the entry guard can directly ‘track’ the physical movements of Alice.

The authors state the third solution supports both anonymity and mobility, as *location privacy*, as shown in Figure 6.4. A number of significant concerns could be raised with this solution. Firstly, in contrast to the second solution, if Alice’s data is still in-flight within the Tor network at hand-off then this data will be lost. This could also add to the existing congestion issues with the current Tor architecture as previously discussed. Secondly, one of the key security features of the original Chaum’s mix network design and subsequently onion routing, and therefore consequently Tor, is each node within the network only knows the previous and next node in the connection. However, this solution requires Alice to directly

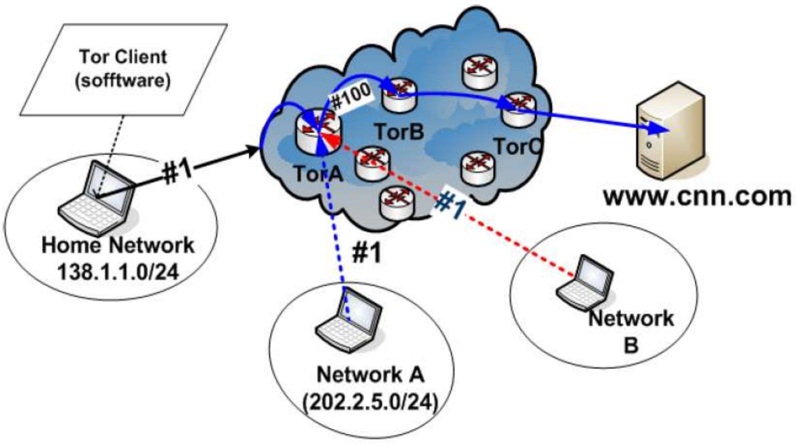


Fig. 6.3 Changes made to the entry guard of Tor network to track a mobile user's change in external IP address, with supporting stop/resume control commands, to redirect the data streams as appropriate [29]

contact, not only the entry guard, but the exit node to provide a status update for her unique circuit identifier. It could be argued that this is a potentially dangerous approach and goes against the fundamental principle of Chaum's mix network design, regardless of any security features such as encryption employed. The issue is that, as already discussed, one of the top adversarial goals is correlating ('confirming') a link between the sender and receiver. Therefore, if Alice interacts directly with both the entry and exit points to the anonymity network, this could have disastrous implications if exploitable.

The first and third solutions raise significant concerns. The first solution, implementing a basic Mobile IP solution shown in Figure 6.5, has the potential to significantly reduce performance with the additional hops and processing. The third solution appears to contravene the fundamental principle of Chaum's mix network design. The second solution provides mobility management, however, the researchers also raise concerns about maintaining privacy. This is because the entry point (entry guard) to the Tor network is able to directly track the physical movements of the mobile Tor user. For the second solution to be viable the Tor entry node must be a trusted node, however, according to Wiangsripanawan, Susilo, and Safavi-Naini "this is very unlikely to happen" [29]. But what if the entry point to the anonymity network is explicitly trusted?

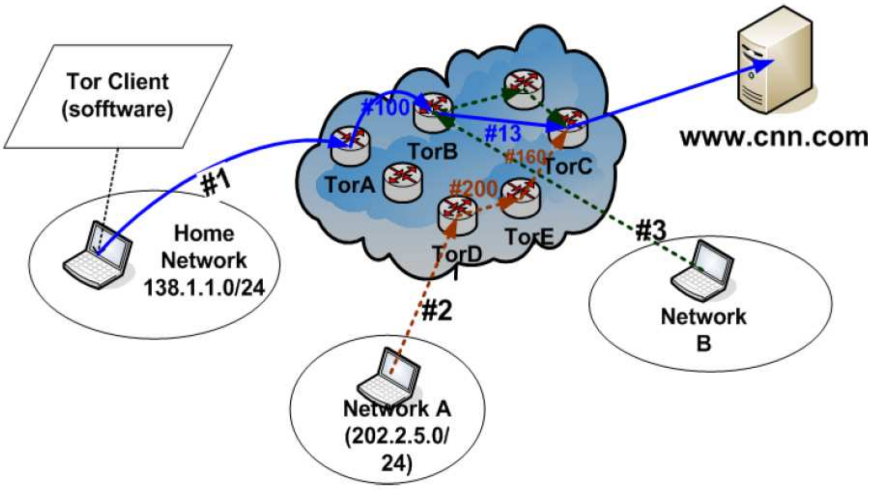


Fig. 6.4 Changes made at the exit point of the Tor network, using the same stop/resume control commands, based on the unique circuit identifier rather than Alice's external IP address [29]

The solution illustrated in Figure 6.6 builds upon the second proposal. The entry guard provides a persistence to the anonymity network for the roaming mobile Tor user, namely Alice. In this version, the entry point is a 'trusted' bridge relay, either operated by Alice herself (or an *explicitly* trusted third-party operator). After a hand-off and re-establishment of a connection to a wireless network, a command control cell (*RESUME*) is issued by Alice to her 'home' bridge relay. On receiving the *RESUME* command, the bridge relay then updates the stored Internet Protocol (IP) address for each of the existing circuit identifiers (*CIRCUITID*). Once the IP table is updated, the bridge relay will resend any outstanding data streams to Alice, alongside any new requests thereafter, to her new location. To consider this solution further, the following key points need to be assessed:

1. The performance benefit of using the mBridge solution.

2. The potential effect on anonymity for the mobile user.

3. Any wider impact on the anonymity network.

For anonymity, as a starting point, it is worth considering the different types of bridge relay operation within Tor. A bridge relay replaces the standard entry point, the guard node (entry guard), to the Tor network. A 'public' bridge is published through the bridge

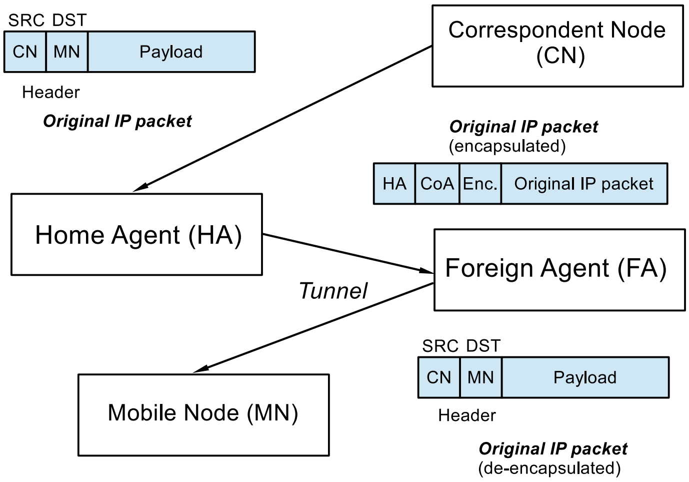


Fig. 6.5 The standard Mobile IP (v4) architecture and process

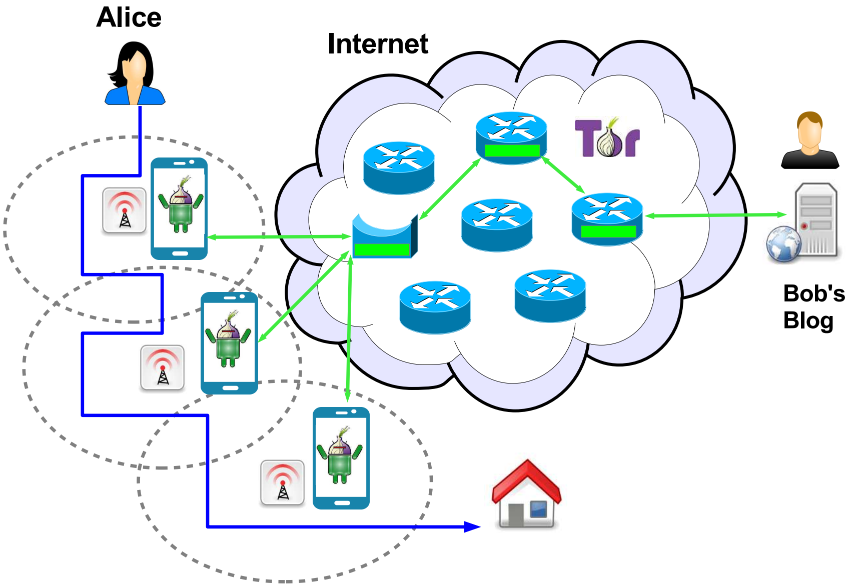


Fig. 6.6 The use of a bridge relay (mBridge), adopting the Mobile IP architecture, to provide a roaming mobile user, Alice, with a persistent connection to the anonymity network

authority (BridgeDB) and is shared by default. If, a roaming mobile Tor user is also the operator of a public bridge, their traffic should, at a high level, appear no different to the rest of the Tor network. Based on the anonymity descriptors previously proposed by Reiter and Rubin, the anonymity achievable would be classified at some point between *absolute privacy* and *beyond suspicion* [28] [132]. However, if only **two** users are connected to the bridge relay at the same time, this would be classified lower, as *probable innocence*, but a defence of 'plausible deniability' could also be offered.

A 'private' bridge is a bridge relay not published by the BridgeDB. A single Tor user using a high bandwidth and unshared private bridge, acting as the first 'hop' within the Tor network, should in theory receive improved performance [23]. This is, however, only theoretical, as discussed previously in Chapter 5, the level of performance achievable depends on any bottlenecks within the circuit [116]. Therefore, any performance gain needs to be assessed against its potential effect on anonymity, hence the use of the new q-factor metric. The following key points are to be examined as part of the evaluation of the mBridge solution:

1. Improvement in client performance, from the mBridge solution, by maintaining a persistent connection to the anonymity network.

2. Any reduction of the wider impact of mobility for example, reduction in 'bad' data.

3. The effect on anonymity, from using the mBridge solution, for both possinymity and indinymity.

4. A comparison, using the q-factor approach, between the effectiveness of the mBridge and Kaplan-Meier estimator based solutions, also comparing alongside the existing default stock configuration.

6.2 Experimental Design

During the previous evaluations, the question of modelling mobility was only briefly discussed, as to remain focused on the key questions at the time: Is there an impact from mobility? Can q-factor improve network management decision making? What is the best throttling scheme for supporting mobility? This decision is often made to maintain an appropriate balance between the cost and complexity of the simulation versus the diminishing returns in the accuracy of the results, that is, to avoid Bonini's paradox [172]. As mentioned at the outset, this research is not aiming to improve mobility modelling nor through changing mobility patterns, in this case human behaviour, to 'fit' a mobility solution. In fact, the complete opposite is the case. The main aim is not to shape user behaviour, which, itself could be counterproductive, but react to the increasing mobility of users.

In earlier chapters, both trace and synthetic models are used to model mobility. During the field study, the author's walk to university generates a trace. The trace, although fairly linear in nature, is still based on a real life example. The assessment using a generic network simulator (OMNeT++) also applied linear mobility, but this time, the precise values of linear mobility are synthetically generated. The previous mathematical modelling used a Gaussian distribution. The intervals between hand-off (HOT), within a range of expected HOT timings, also based on the author's walk. These approaches provided a fair indication of impact as well as providing early comparison of different solutions. Different mobility patterns may also influence the impact of mobility and the efficiency of each of the proposed solutions, and therefore, an additional mobility model (random walk / random waypoint) is applied during this evaluation. A random walk is a mathematical formalization of a path that consists of a succession of random steps [173]. The random waypoint model is a random walk based model for the movement of mobile users [174]. The random walk / random waypoint model are the most widely used mobility models in research [175].

As previously stated, although Tor probably remains the only feasible anonymity network for browsing the World Wide Web (WWW or the Web), developers, researchers, and users are continually looking at other privacy solutions. Therefore, a decision is made to continue to step back from a Tor-specific piece of research and build upon a case study based on a hypothetical anonymity network. The anonymity network is a scaled-down version of Tor but still incorporates a conceptual implementation of the 'proven' onion routing. The goal is not only to generate discussion and drive innovation for future Tor development, but also other projects within the research area of anonymity networks, anonymous communications' systems, and privacy enhancing technologies, while supporting the increased mobility of users. Therefore, the case study described in APPENDIX A and illustrated in Figure 6.7, is used for the evaluation alongside the q-factor metric. The aim is to assess the mBridge solution for performance alongside anonymity, both possinymity and indinymity, compared to the default stock configuration and Kaplan-Meier estimator based adaptive client throttling.

The background of the case study is briefly stated for setting context. As with the Tor project's aim, the targeted primary users of the Tor project include "human rights activists use Tor to anonymously report abuses from danger zones" [176]. Therefore, rather than focusing on illegal bulk downloads of the latest Hollywood blockbuster or other nefarious uses, the case study focuses on the sharing of private messages between human rights activists and a blog run by an independent journalist. Each of the key components of the case study, *"Alice and her friends (20 in total) are human rights activists... and exchange (upload / download) messages on a website (Bob's human rights' blog)."*, define the parameters of the simulation.

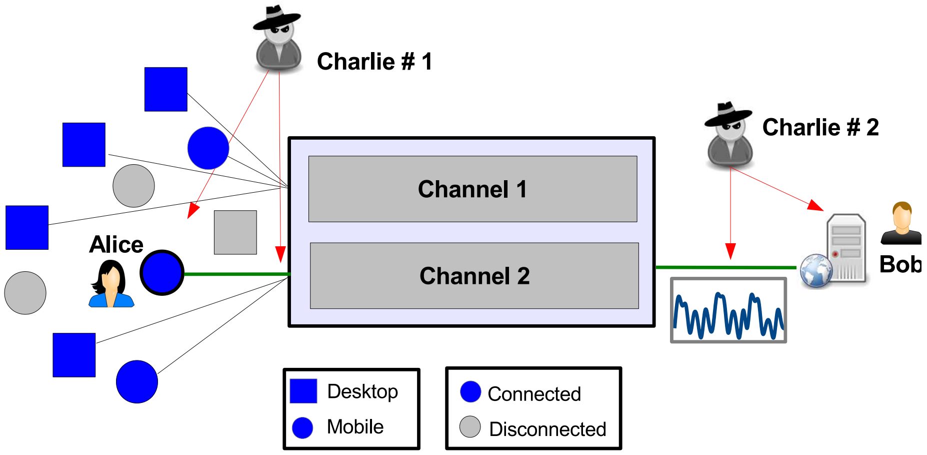


Fig. 6.7 The two-channel anonymity network used within the case study for Alice's communications with Bob, also showing Charlie's observation points

The inner workings of the anonymity network do not require over-complication to evaluate the solutions. A plethora of research has already been undertaken understanding the causes of congestion and proposing solutions for optimizing performance, such as improving circuit path building algorithms. The anonymity network uses the standard onion routing protocol, as outlined in Chapter 2, containing the two channels. A brief overview of the anonymity network and supporting mobile application is as follows:

"David has developed an anonymous communications' system, a low-latency anonymity network. The anonymity network is built on the concept of onion routing. In contrast to Tor, which uses TCP as its transport, the anonymity network has adopted a lighter transport protocol (UDP), the same approach used by I2P. Norman maintains the anonymity network. It also consists of relays supported by volunteers around the world. A number of security features have been implemented. As with Tor, the anonymity network refreshes circuits every 10 minutes to try to reduce the opportunity for traffic analysis. The anonymity network is primarily used for updating and retrieving messages on compliant blogs such as Bob's.".

The performance of the anonymity network is critically important to allow Alice and her friends to respond quickly in volatile situations, for example, if on a protest march, however, while also maintaining an acceptable degree of anonymity. To maintain this critical balance, the anonymity network operator, applies the q-factor metric to monitor the key components on the anonymity network. This generates boolean values for both anonymity (v_a) and performance (v_b) based on specified thresholds θ_a and θ_b, as $v_a \equiv a > \theta_a$ and $v_b \equiv b > \theta_b$,

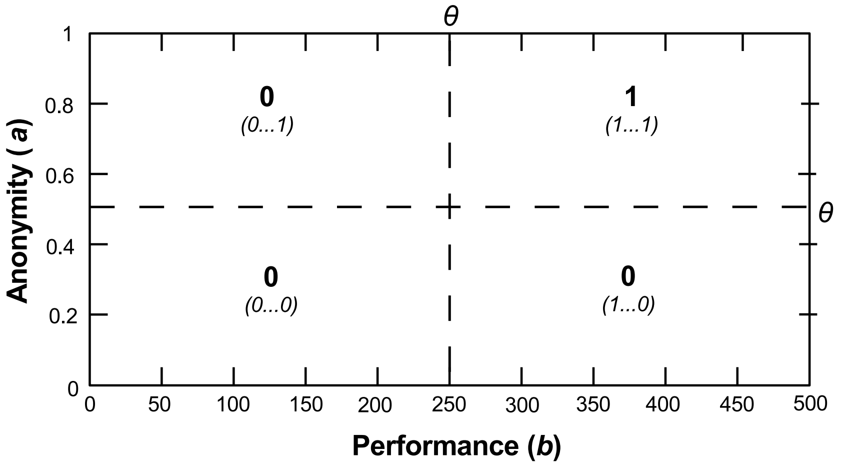


Fig. 6.8 The calculation of q-factor, as the conjunction of the resulting boolean for anonymity (θ_a) and performance (θ_b), based on whether their specific thresholds are met

for anonymity and performance respectively. As before, the q-factor is calculated as simply the conjunction of v_a and v_b, that is $q = v_a \cdot v_b$, as shown in Figure 6.8. A value of 0 indicates that intervention is required otherwise the network can continue to operate as it is. However, **no** intervention is undertaken during this chapter. This is to provide a clean assessment of the effectiveness of each of the solutions.

The software employs a number of performance and security features. For example, allowing only one active circuit per connection and limiting the size of the message to 300 kB. The file transfers are also padded to 300 kB, to reduce traffic analysis, so no one file is any more distinguishable while being transferred. An example of network traffic between Alice and Bob, showing data stream circuit windows and message blocks, is shown in Figure 6.9. Both of these security features aim to try to prevent ‘flooding’ of the anonymity network with messages to undertake congestion-based denial of service type attacks. The circuit window is fixed at a default of 1000 cells (each cell being 50 kB) with a total of 500 kB. The circuit window size can be adjusted client-side only, within the client configuration settings, if required, such as for adaptive client throttling (Kaplan-Meier).

The anonymity network also supports connections from mobile devices for which Bob has developed a separate Android application. On average, Alice and half of her friends access Bob’s website via the mobile application, the remainder from their desktops (in Internet cafés etc.), reflecting a parity of desktop and mobile Internet connections. Additionally, a number of the mobile users are roaming at any time, which includes Alice for the benefit of the case study, while the others remain static.

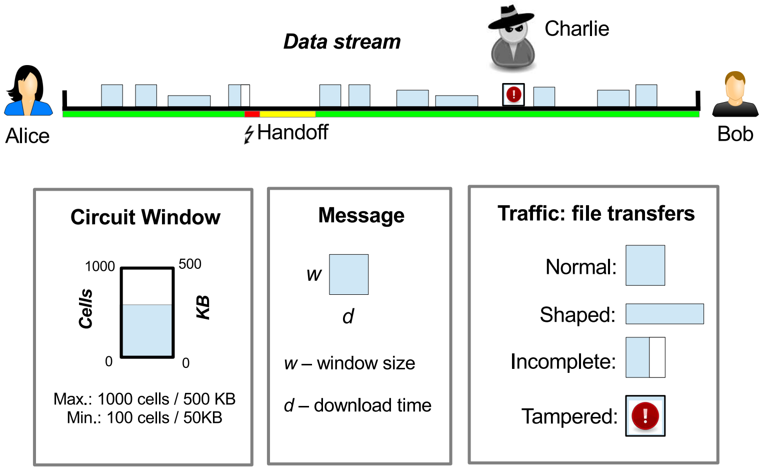
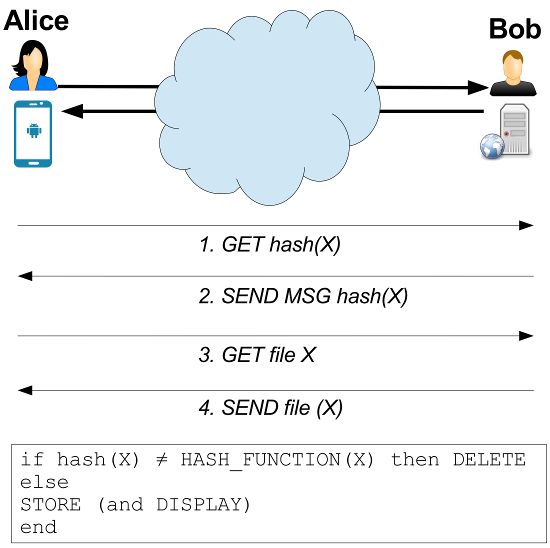


Fig. 6.9 Example network traffic between Alice and Bob showing the data stream, circuit windows, and message blocks

An existing security feature on Bob's blog is that a hash value is generated for each file. The hashing algorithm actually used is unimportant but the assumption is that it is robust. A website user can manually compare the downloaded file's hash value against the one displayed on the website. If, the two hashes match, the file has not been tampered with and is accepted. The mobile application undertakes this check automatically. Downloads are rejected if the hashes do not match as either being tampered with (fingerprint / staining attacks), corrupted in-flight, or the transfer is incomplete (a partial download), shown in Figure 6.10 and corresponding upload process in Figure 6.11.

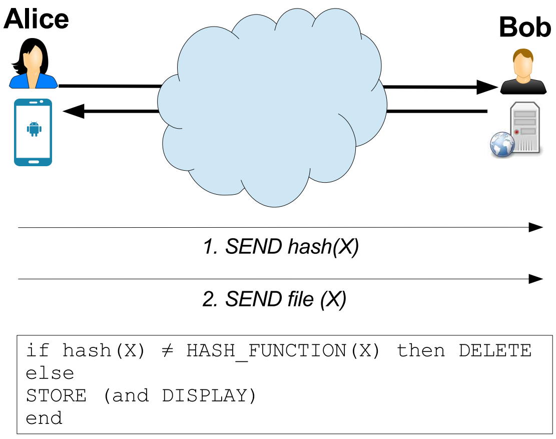
The users, excluding Alice who is ever-present, are scripted to randomly connect and disconnect during the scenario of 600 seconds. Half will be desktop users and the rest, including Alice, mobile devices. Of the mobile users, a random number will be roaming at any point in time and the remainder stationary. The schemes to be evaluated are mBridge solution, the Kaplan-Meier estimator based adaptive client throttling scheme, and finally the default stock configuration. The different entry points to the anonymity network are shown in Figure 6.12.

For Alice and the other mobile users, if the roaming mode is manually switched on their device, mBridge acts as a home agent and redirects any traffic to maintain a persistent connection to the anonymity network. The 'penalty' of using mBridge in roaming mode is an arbitrary one second. This is to update mBridge with the new external Internet Protocol (IP) address of the mobile user, which, is lower than the time rebuild the circuits. The value



```
if hash(X) ≠ HASH_FUNCTION(X) then DELETE
else
STORE (and DISPLAY)
end
```

Fig. 6.10 The file *download* process on the mobile application through the anonymity network



```
if hash(X) ≠ HASH_FUNCTION(X) then DELETE
else
STORE (and DISPLAY)
end
```

Fig. 6.11 The file *upload* process on the mobile application through the anonymity network

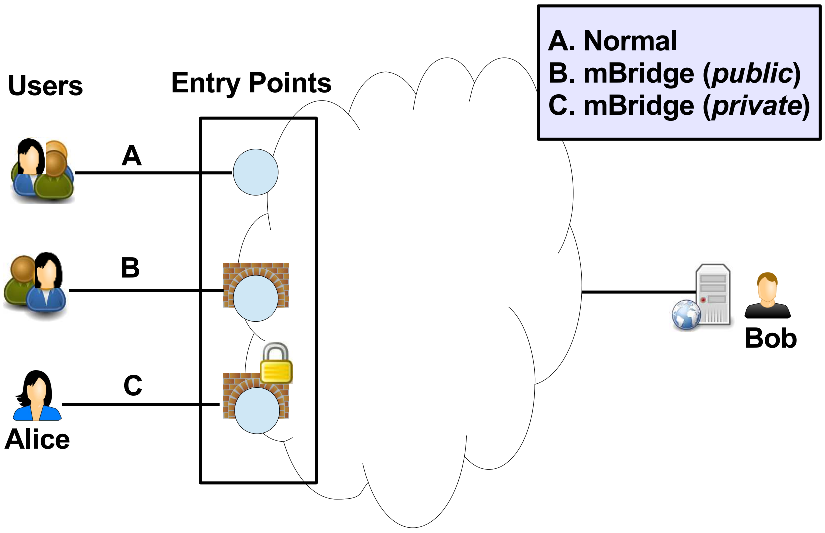


Fig. 6.12 The three types of entry points to the anonymity network: A. 'normal' entry guard relay, B. *public* mBridge, and C. *private* mBridge

of one second was chosen to reflect the acceptance that an overhead of mBridge is likely but also has to be relatively efficient otherwise it will not deliver any benefit over the faster times to recovery, specifically circuit building.

Performance is measured using average bitrate (ABR). As before, ABR is calculated as the total number of, and amount of, successfully transferred files over the course of the scenario in Kibit/s. The calculation and variables are identical to those used in Chapter 5. The initial calculation of goodput (g), including any partial downloads, is described in Equation 6.1:

$$g(i) = \frac{h-(y+k)}{r+f+(d\cdot(\frac{S}{w}))} \tag{6.1}$$

The number of successfully completed file transfers is calculated as $g \rightarrow \lfloor g \rfloor$. This is then applied to each hand-off within the array to generate the total number of successfully completed file transfers during the time frame. As the file is padded, and therefore the size is constant, the overall performance (b) is simply derived by the number of successfully complete file transfers multiplied by the request size(S). A factor (x) is applied to convert to kibibit (Kibit) and divided by t to provide the final ABR. The calculation to determine the overall performance during the scenario is described in Equation 6.2:

$$b = \frac{\sum_{i=n}(\lfloor g \rfloor(i)) \cdot S \cdot x}{t} \tag{6.2}$$

The calculation for total amount of 'bad' traffic as orphaned data (o) generated, that is incomplete file transfers and data remaining in-flight, is calculated as described in Equation 6.3:

$$o = \sum_{i=1}^{n}(g) - \sum_{i=1}^{n}(\lfloor g \rfloor(i)) \cdot S \tag{6.3}$$

Even though performance is relatively straightforward to calculate for this particular case study, Alice's requirements are more difficult to accurately predict. The values of time (r) used previously between completed transfers at every 2 seconds seems rather frequent than expected within a real-world example. A volatile situation, such as a demonstration, Alice may send and receive messages more frequently. Therefore, for consistency with previous work, the time between requests will remain, alongside a fixed (padded) file size of 300 kB, at a time to first byte of 2 seconds and average transfer time of 9 seconds (if not throttled).

In addition to client performance, resource allocation and the overall q-factor maintenance level are also recorded, as previously used in Chapter 4. The two alternative paths (channels), through the anonymity network, are implemented again, with each channel having a different fixed bandwidth capacity assigned, 2000 kB/s and 3000 kB/s. This again helps generate varying performance across the network for the benefit of positive illustration. The small set of users ($N = 20$) is partly intentional. This is to allow anonymity to fall below the required threshold during the scenario, again for positive illustration. A 'circuit window', with each user having a 500 kB/s bandwidth cap, is applied. This again allows provides a fair distribution of resources to maintain steady traffic flow across the network. Based on the combined network capacity of 5000 kB/s, the maximum number of concurrent users, circuit window size, a performance threshold of $\theta_b = 250$ kB/s is chosen for resource allocation. This is again so a range of performance levels on either side of the threshold can be observed. For *actual* performance a threshold of $\theta_b = 150$ kB/s is used to align with the 300 kB fixed file size.

The measurements of anonymity use both possinymity and indinymity to generate the anonymity (a) component of q-factor. The degree of anonymity is again required to meet a pre-set threshold. The threshold of 0.5 (a = 1 - P) is determined on ranges of probability that Alice is the sender: 0.50 to 0.74 *probable innocence*, 0.75 to 0.99 *exposed*, 1 *provably exposed*. The possinymity is derived from the connections observed and the probability that Alice is the sender as previously outlined. Indinymity is assessed as to how indistinguishable Alice's

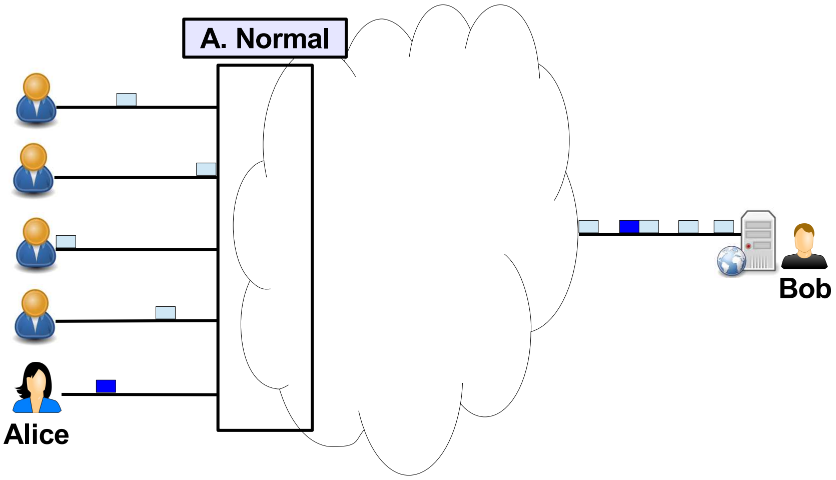


Fig. 6.13 Traffic fingerprint: A. ‘normal’ entry guard relay

messages are from the other traffic. The padding employed within the system has removed some opportunity for traffic fingerprinting. As this research is focussed on the balance between anonymity and performance, the indistinguishability is based on the probability that Alice sent the message from the time taken to transfer the file within the observed data stream. This is illustrated in Figures 6.13, 6.14, and 6.15 and in more detail later alongside the results.

The two measurements for anonymity and performance are mapped using the q-factor metric. Since 2013, a number of changes and infrastructure improvements have been implemented by the Tor project that appear to have improved the circuit build process and general performance. The most recent field-study observations, in 2015, found the circuit build process to be more reliable and quicker than the previously estimated average time of 7 seconds, at usually between 4 to 6 seconds, but even sometimes near the optimal 3 seconds. This is reflected by amending circuit build time parameter down to 5 seconds. The physical network hand-off time is reverted back from 1.0 to 1.5 seconds as again using default generated by the OMNeT++ / INET simulator.

For the mobility modelling, using only the linear mobility model is somewhat limited. Although, as previously stated, this research does not aim to study human behaviour, other mobility models should be applied for a different evaluation of the proposed solutions [177]. The random walk / waypoint(RandomWP) mobility model available within INET is applied instead, as the example illustrated in Figure 6.16. This has two purposes. Firstly, it will confirm whether the concern regarding the Kaplan-Meier estimator based algorithm may be

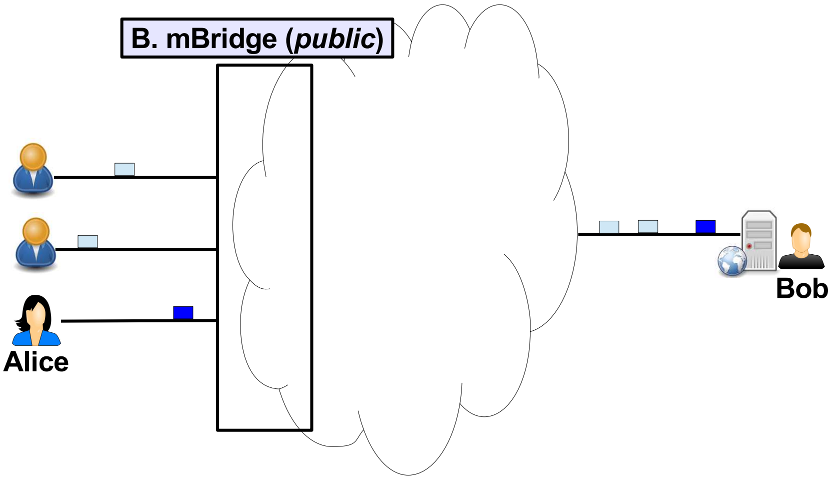


Fig. 6.14 Traffic fingerprint: B. *public* mBridge

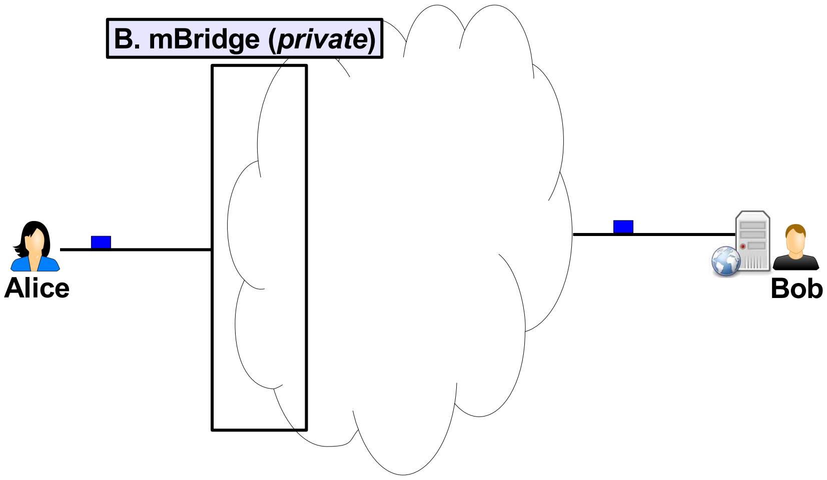


Fig. 6.15 Traffic fingerprint: C. *private* mBridge

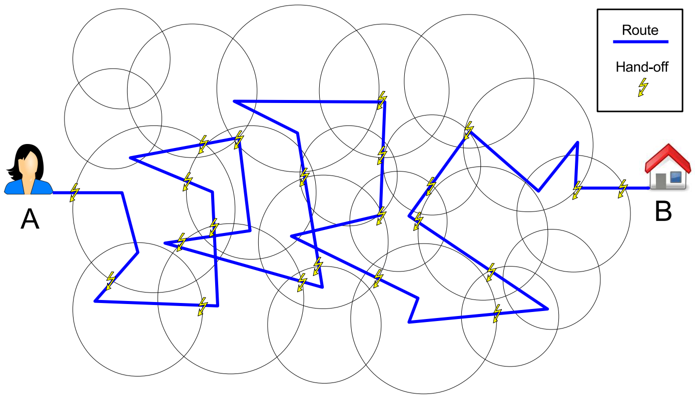


Fig. 6.16 An example route for RandomWP (random walk) mobility with indicative hand-offs

inefficient in lower levels of mobility. Secondly, it will also 'stress test' the mBridge solution at a lower level of mobility. Therefore, the predicted performance gain from the user having a persistent connection will be the most pessimistic.

A summary of the process is as follows. The 20 users (Alice (1), and users 2 - 20) are allocated either desktop or mobile equally. The mobility model will provide some stationary behaviour (or no hand-offs due to mobility patterns, may circle the area within the same network range for example). These remain set for the test cases to provide consistency for comparing results. A random boolean is generated using $n = randn(20,1) > 0$ to discover whether the user is connected (1) or not connected (0) at the beginning of the scenario. A random number between 1 and 600 is then generated for each user to determine the time they close their connection (for those already connected at t = 0) or the time they connect for the first time. The users that connect after the scenario commences will by default still be connected at t = 600 unless handing off. The distribution is uniform and as Alice is the main agent of interest, she is connected for the full 600 seconds, as illustrated in Figure 6.17.

The allocation of the bridge users component is a little more difficult to define. If the case study is compared to Tor, the numbers are not easily translatable (two million 'normal' connections compared to 20,000 bridge users). For the architecture, the number of public bridges total 3000 to 4000 (the number of 'indirect' connections' from private bridges is unknown by design) and 'traditional' relays between 6000 to 7000 [33]. This is a ratio of around 6 to 7 'normal' nodes compared 3 to 4 'bridge' nodes. This ratio will be adopted instead for the anonymity network. As Alice is known (by the reader only not Charlie yet) to be a mobile user, the ratio applied will be 13:7 (non-bridge / bridge), therefore Alice plus 6 other users (either desktop or mobile).

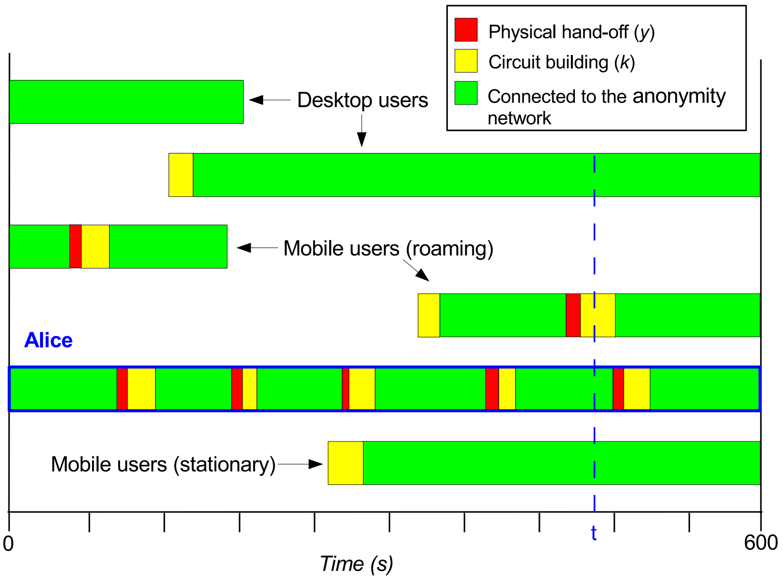


Fig. 6.17 The scenario showing Alice's connection alongside other agents, with 3 concurrent connections at $t = 460$, with another user building a connection

The generic network simulator OMNeT++ / INET) will generate, from realistic network simulation, the mobility data set for all the scenarios. These are then imported into MATLAB to apply the more complex application-level functionality, therefore a two-stage experimental environment. A total of 15,000 observations are to be made based on the different solutions for the users. Each test case will reset the random boolean, connection times, hand-off times, however the traffic load will be identical for performance comparison. A high level overview of the experimental method, a two-stage approach, used for the evaluation in shown in Figure 6.18. A table containing the key parameters can be found in Table 6.1, with the HOT_{min} and HOT_{max} values auto-generated[1] and the time to complete download (d) depending on circuit window size[2].

6.2.1 Planned attacks

As discussed in Chapter 2, there are a number attacks that can be undertaken against anonymity networks, such as Tor, to de-anonymize its users. In some circumstances merely identifying an individual is using an anonymity network, to circumvent censorship, is enough for further investigation or even criminal charges. As the anonymity network used for the case study is closely modelled on onion routing and the Tor network, many of the same attacks are relevant. To evaluate the effectiveness of the solutions, for both anonymity and

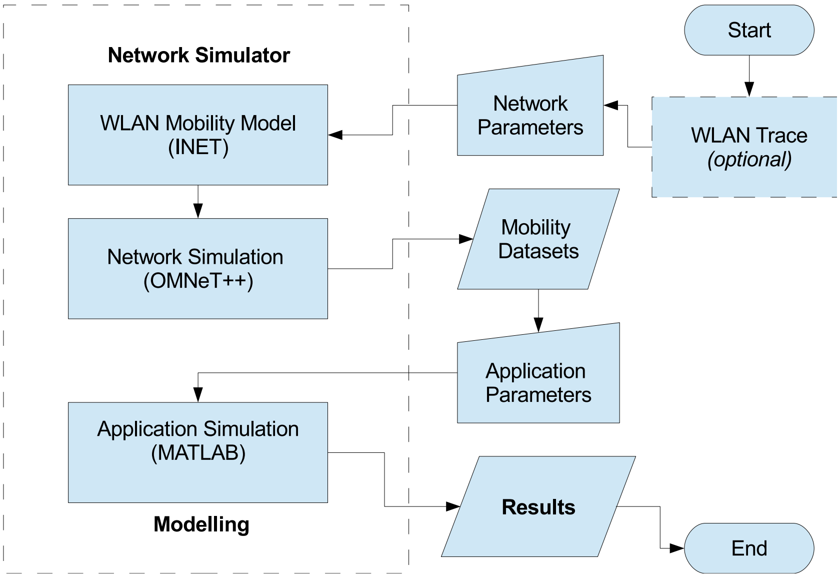


Fig. 6.18 A high level overview of the experimental method used for the evaluation in this chapter

	Experiment in §:	3.2	3.3	3.4	4.2	5.3	**6.2**
ANONYMITY:							
Possinymity:	Attack A-0						✓
	Attack A-1						✓
	Attack A-2						✓
	Attack A-3						✓
	Attack A-4						✓
	Attack A-5						✓
Indinymity:	Attack A-6						✓
	Attack A-7						✓
PERFORMANCE:							
MTTF:	Speed (mps)						1.2
	HOTs (s)						0...600
MTTR:	Physical (s)						1.5
	Circuit (s)						7‡
	Redirection (s)						1‡
Load:	File size (KiB)						300
	Wait (s)						2
	Request (s)						2

Table 6.1 The key parameters used for the evaluation of the mBridge solution. ‡The redirection overhead replaces the circuit build process for the mBridge solution.

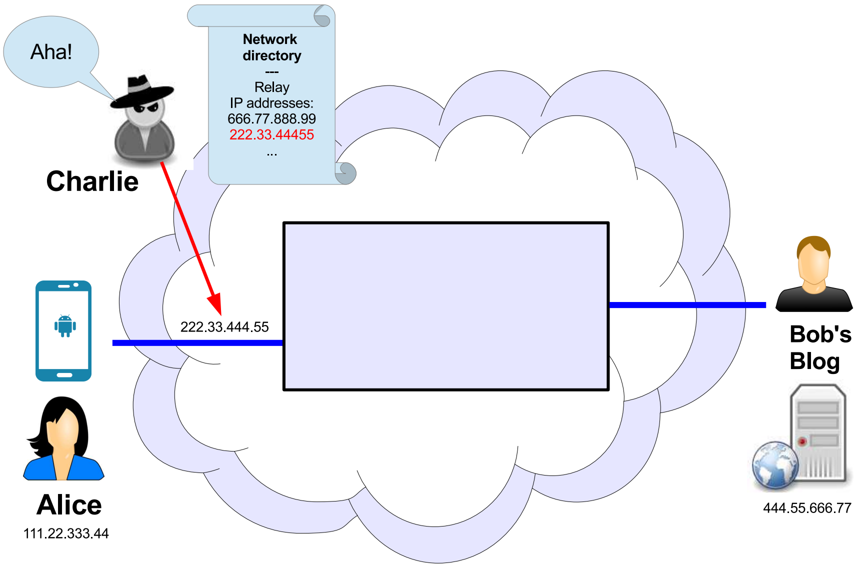


Fig. 6.19 Attack A-0: the base analysis where Charlie identifies that Alice and her friends are using the anonymity network

performance, the following attacks are hypothesized. They will be used in combination to generate a partitioning style attack where, through a sequence of attacks, Charlie builds up a profile of Alice. He then slowly partitions the anonymity network into smaller subsets with the aim to reducing anonymity until Alice is eventually exposed.

In Figure 6.19, although A-0 may not be considered a 'full-blown' attack, it is the base analysis Charlie identifies that Alice and her friends are using the anonymity network, and potentially Bob's blog. Charlie refers to the publicly published network directory of relays to the Internet Protocol (IP) addresses that Alice connects to while observing her Internet connection. At this juncture, Charlie cannot correlate any individual messages to Alice, and although it is not illegal to use anonymity network, uploading anti-government messages on Bob's blog is. Charlie decides to try to investigate Alice further.

For attack A-1, shown in Figure 6.20, Charlie determines, which channel of the anonymity network Alice is using by measuring the file download timings. In this example, channel 2, based on Charlie's previous observations (training data), is known to provide on average 50% slower performance on average when downloading a 300 kB file. Based monitoring Alice's connection over time, Charlie is sufficiently convinced that probabilistically Alice is using channel 2.

During attack A-2, Charlie can identify whether Alice is using a bridge by 'harvest-

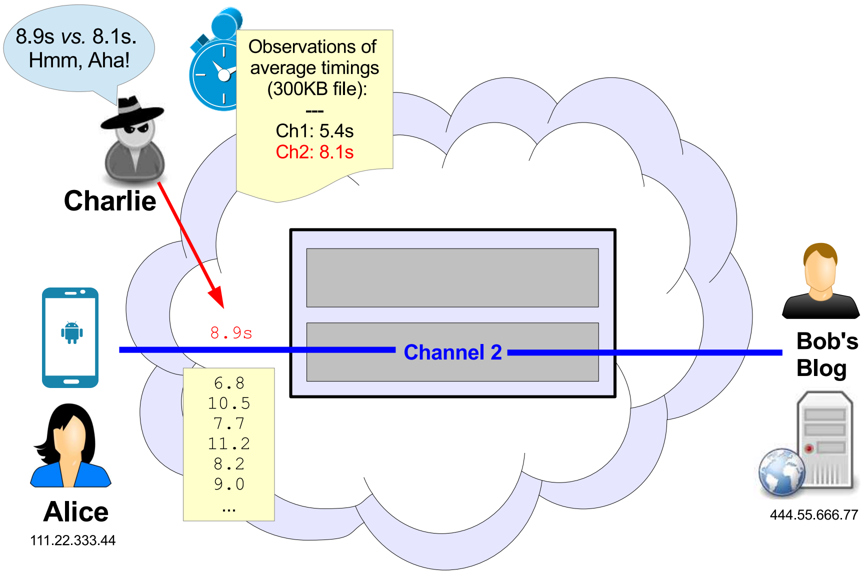


Fig. 6.20 Attack A-1: Charlie can identify, which channel Alice is using by measuring the latency / timings of file downloads based on Charlie's previous observations (training data)

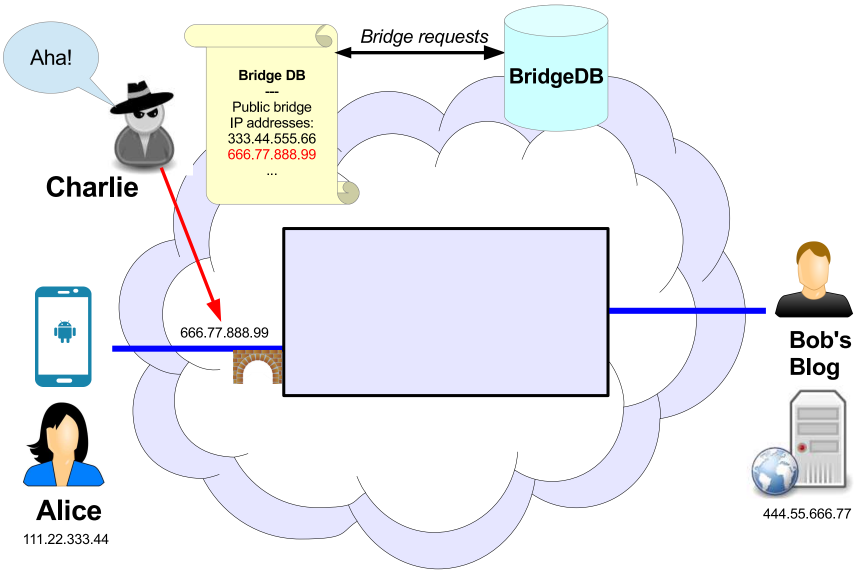


Fig. 6.21 Attack A-2: Charlie can identify whether Alice is using mBridge by 'harvesting' the *public* mBridge IP addresses from the bridge authority (BridgeDB)

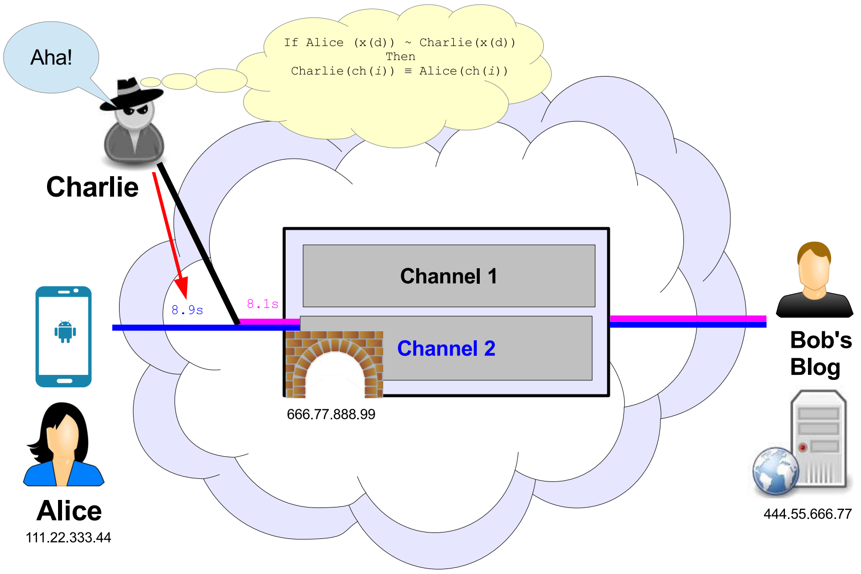


Fig. 6.22 Attack A-3: Charlie connects to the same mBridge observed for Alice and determines the channel by measuring the file download timings

ing' the public mBridge IP addresses from the bridge authority (BridgeDB), as shown in Figure 6.21.

As shown in Figure 6.22, during attack A-3, Charlie connects to the same bridge observed for Alice, determining this time, both the mBridge and channel Alice is using by measuring latency or file download timings. If these match, within a pre-set deviation, then Charlie determines that he is using the same channel as Alice, also from the same mBridge.

During the next two attacks, A-4 and A-5, Charlie identifies Alice is accessing the anonymity network from a mobile device, through a mBridge, also, which channel she is using, shown in Figures 6.23 and 6.24. How Charlie can identify whether Alice is using a mBridge and, which channel, has previously been discussed. However, there may be a number of different methods Charlie could identify Alice is using a mobile device such as network behaviour or other information leaks from her device. The signature of the data stream is covered during the next attacks.

During attack A-6, Charlie identifies the Kaplan-Meier signature while Alice uploads the fixed length files and establishes whether Alice is throttled and therefore *must* be a mobile user, as shown in Figure 6.25. In attack A-7, Charlie observes the timings (start and end times) of individual uploads and downloads to try to correlate Alice to specific entries on Bob's blog site, as shown in Figure 6.26.

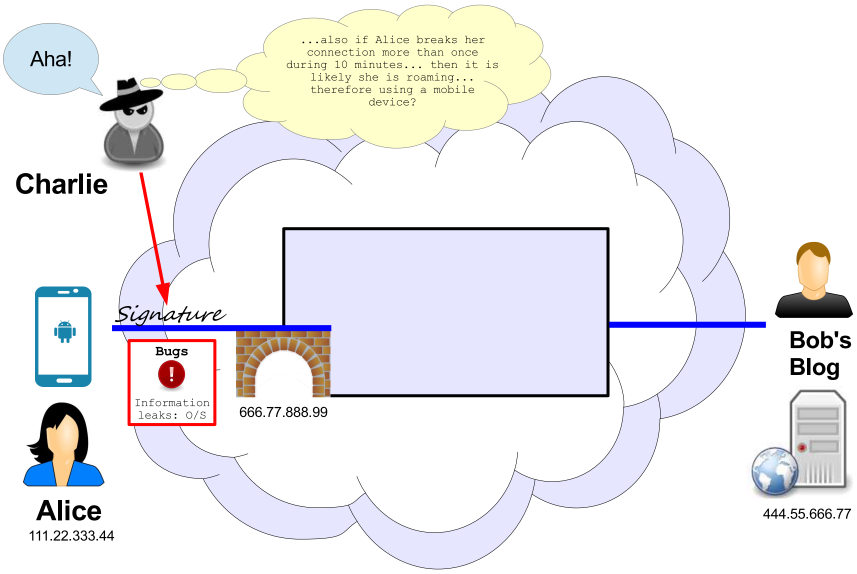

Aha!
...also if Alice breaks her connection more than once during 10 minutes... then it is likely she is roaming... therefore using a mobile device?
Charlie
Signature
Bugs
!
Information leaks: O/S
666.77.888.99
Bob's Blog
444.55.666.77
Alice
111.22.333.44

Fig. 6.23 Attack A-4: Charlie can identify that Alice is accessing the anonymity network from a mobile device via a mBridge

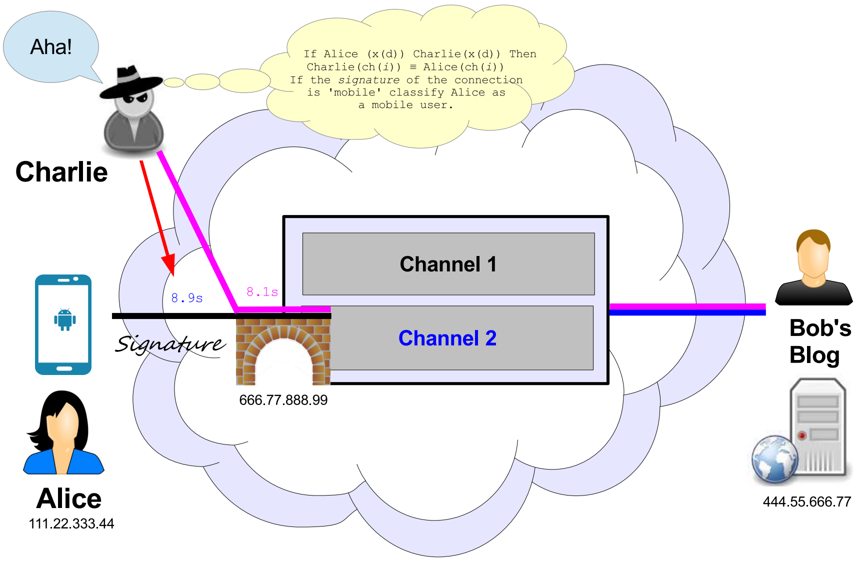

Aha!
If Alice (x(d)) Charlie(x(d)) Then Charlie(ch(i)) ≡ Alice(ch(i)) If the signature of the connection is 'mobile' classify Alice as a mobile user.
Charlie
8.9s
8.1s
Signature
Channel 1
Channel 2
666.77.888.99
Bob's Blog
444.55.666.77
Alice
111.22.333.44

Fig. 6.24 Attack A-5: Charlie can identify that Alice is accessing the anonymity network from a mobile device via a mBridge, and the channel she is using

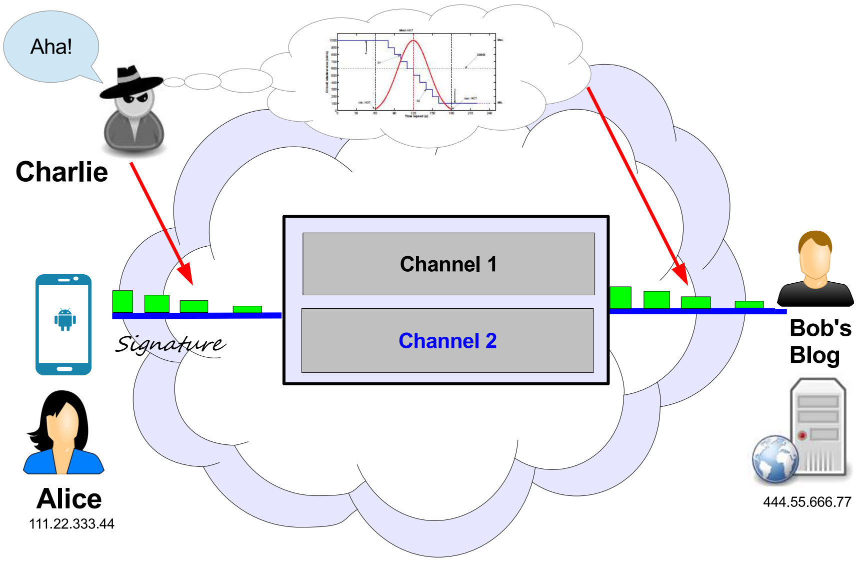


Fig. 6.25 Attack A-6: Charlie observes the time it takes Alice to transfer a file and establishes whether Alice is throttled and, therefore, must be a mobile user

In summary, the status of Alice as an active user of the anonymity network is initially established during A-0. Charlie then undertakes a number of attacks, some also used in combination, to generate an overall partitioning style attack where, through a sequence of attacks, Charlie builds up a profile of Alice. He then slowly partitions the anonymity network into smaller subsets with the aim to reducing anonymity until Alice is eventually exposed. The attacks A-1 to A-5 are used for reducing Alice's possinymity. Attacks A-6 and A-7 are assessed for indinymity, which, are also able to be combined with the previous attacks to reduce possinymity, therefore Alice's overall anonymity, and consequently the q-factor.

6.3 Results

Firstly, it is worth returning to the issue of network churn. As shown in Figure 6.27, at the beginning of the scenario, when Alice joins the network ($t = 0$), the mean number of connections to the anonymity network (that is, both channels) is 9.92 across all the test cases. Due to the experimental set-up, this figure is as expected, being half the maximum number of users within the anonymity set of 20. The lowest number of connections at $t = 0$ is 5 and the highest 13. This reflects that the case study anonymity network uses random allocation of channels rather than any load-balancing algorithm as used in Chapter 4 and in Tor. All

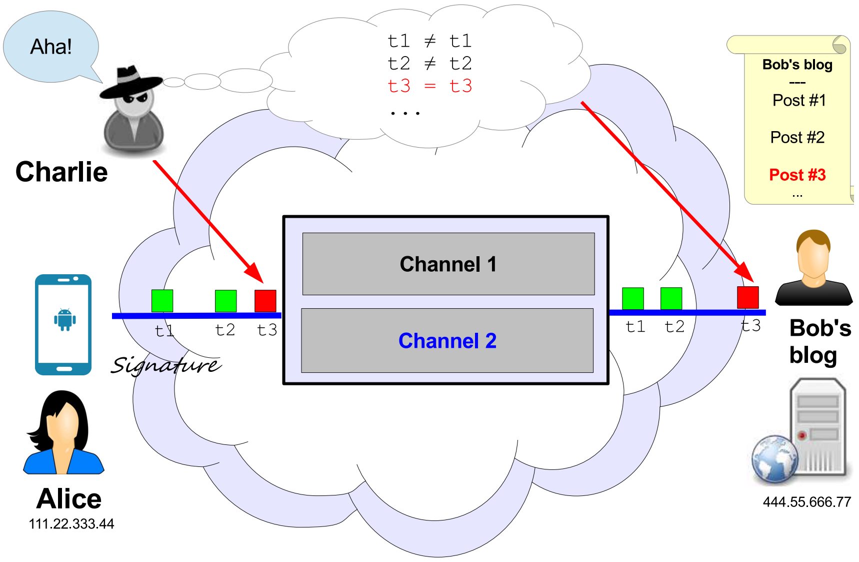


Fig. 6.26 Attack A-7: Charlie observes the start and end times of Alice's file transfer to try to correlate to specific entries on Bob's blog site

things being equal, either end of this range still provides reasonable anonymity ($a = 0.8$ and $a = 0.92$) for its users when the anonymity network is observed as a whole, that is, if Charlie can only observe how many users are connecting to the anonymity network.

Over the course of the experiment, the mean number of connections to the anonymity network is 10.76, again within the expected range. As previously discussed, using any mean measurement when trying to hypothesize anonymity is somewhat dangerous. It is possible to have a mean number of connections of 10, but at a particular point in time, only one user (namely Alice) is connected and consequently exposed. Therefore, the outer range (minimum and maximum) of the number of connections also needs to be assessed. In this case, the anonymity network still maintains acceptable anonymity, with the lowest number of connections recorded is 5 across the anonymity network at any time. Therefore, while Charlie is observing both channels Alice's anonymity is maintained.

Specifically for performance, it is also worth reflecting on the bandwidth allocation compared to the number of connections, at each observation, during the scenario. In Figure 6.28, Alice's bandwidth allocation by the anonymity network (depending on the channel) is mapped alongside her possinymity based on the anonymity set (number of concurrent connections). This shows that the anonymity network is not particularly optimized for its resources. Although, the anonymity provided is strong and the resources allocated are good, at certain

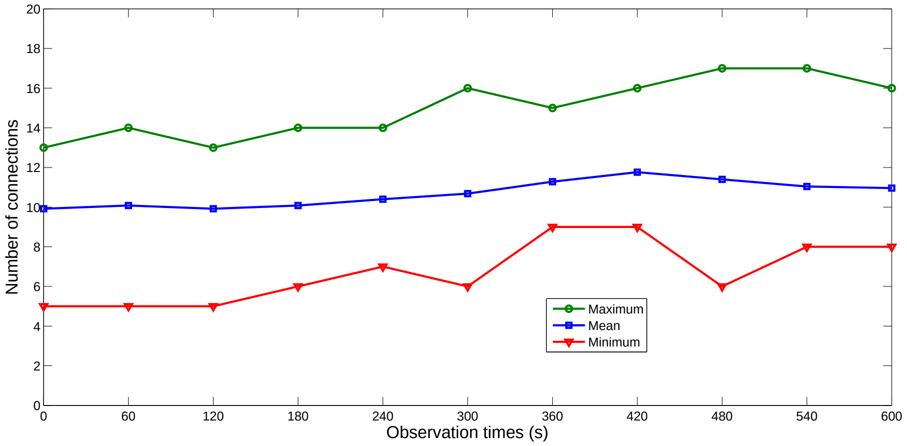


Fig. 6.27 The average number of concurrent connections to the anonymity network at each snapshot during the scenario (min. / mean / max.)

times, the resources allocated are actually too high. Also, shown in Figure 6.28, there are a number of occasions where Alice is allocated more than the maximum circuit window size. Therefore, it is impossible for Alice to actually use this surplus resource and therefore is theoretically wasted. In fact, the percentage of cases capped, potentially due to poor allocation of resources, are per scheme: Stock 1.6%, Kaplan-Meier 1.2%, mBridge 6.4%. As a reminder, Stock refers to the default configuration with no throttling applied and accessing the anonymity network through the standard entry point, which, in the case of Tor would be a ’entry guard’ node. For this case study, the capping does not cause a serious performance issue. In the worst case (mBridge) the reduction of actual performance is reported at less than 0.25%. This illustrates two key points. Firstly, to improve performance of a network service, adding additional network resources is not always necessarily required, thus avoiding Braess’s paradox [112]. Secondly, it shows again how even rudimentary load-balancing mechanism can provide better optimisation of a network service. Although, in this example the lowest bandwidth allocation (not actual performance achieved) of 214 kB/s, is unlikely to deter usage, specifically the transfer of 300 kB files, it shows once again the potential benefit of using q-factor and the opportunity to redirect resources or users to achieve the threshold for both components of the q-factor metric.

For *actual* client performance, the first comparison used is the amount of good to bad data generated. In Figure 6.29, the mean total amount of good and bad data generated by each of the solutions is shown. The mBridge solution, which provides a persistence of connection to the anonymity network achieves on average the highest number of completed downloads. This is followed by the default stock configuration, and then by the Kaplan-Meier based

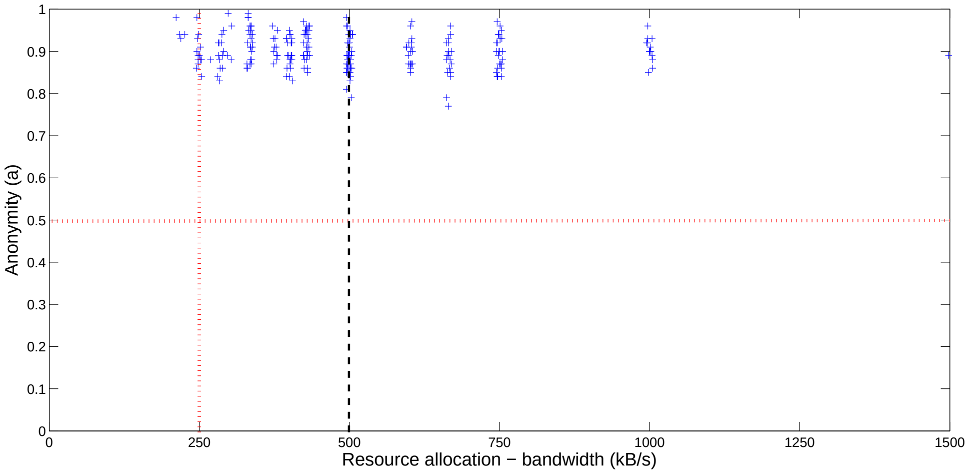


Fig. 6.28 Alice’s q-factor mappings after ‘attack’ A-0 (base analysis)

adaptive client throttling scheme. After converting to an average bitrate (ABR), the mBridge solution achieves 251.39 Kibit/s compared to 242.98 Kibit/s for Stock. This amounts for a 3.55% increase in client performance. As previously mentioned, mBridge also suffers the most from the performance cap created by the set maximum circuit window size. If the circuit window size cap is lifted, the mBridge solution would have achieved a small percentage more in this case study.

Comparing the results in Chapter 3, albeit some minor changes in the mathematical modelling, the impact of mobility at circuit build times of 3 and 7 seconds stood at a 9.45% and 14.45% reduction in the ABR respectively, than if the mobile Tor user remains stationary (0 m/s). As the mBridge provides a persistence of connection to the anonymity network, it would be expected that the potential gain in performance over Stock (5 second circuit build time) would be more reflective. This illustrates the difference the mobility model used can make. In this example, the difference between linear and random walk / waypoint (RandomWP) mobility models is probably somewhere between 6% and 11%. The key question is whether the lower end of performance gain is still significant, and more so statistically significant. Any performance gain should be deemed significant in isolation. The previous correlation of the relationship between lower circuit build times and performance is not easily argued against. However, the statistical significance to whether the performance gain is not merely by random chance should be ascertained.

To prove statistical analysis, Student’s t-tests are used to compared the mean values between the solution (Kaplan-Meier, mBridge) and the default configuration (Stock). The t-test has a number of variations. A basic unpaired test, independent two-sample t-test, compares the mean values of two equal sample sizes, as shown in Equation 6.4, where $s_{X_1X_2} = \sqrt{s_{X_1}^2 + s_{X_2}^2}$ and $s_{X_1X_2}$ is the pooled standard deviation of both samples.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_{X_1X_2} \cdot \sqrt{\frac{1}{n}}} \tag{6.4}$$

The experimental design provides the opportunity to test the data as a dependent t-test for paired samples, as shown in Equation 6.5. This is because the experiment uses repeated measures, as the same test cases are applied to each scheme. This statistical blocking reduces the effects of confounding factors and consequently provides greater validity.

$$t = \frac{\overline{X}_D - \mu_0}{\frac{S_D}{\sqrt{n}}} \tag{6.5}$$

In theory, both samples have equal variance, although when comparing mBridge to Stock, in reality, the variance of mBridge is likely to be nil. Therefore, the Welch's t-test can also be applied to provide even stronger statistical support for the unequal variance, as shown in Equation 6.6, where $s_{\overline{X}_1 - \overline{X}_2} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$.

$$t = \frac{\overline{X}_1 - \overline{X}_2}{s_{\overline{X}_1 - \overline{X}_2}} \tag{6.6}$$

The t value is determined, and the ρ-value derived, using MATLAB (*ttest* function). The threshold chosen for statistical significance is a standard $\alpha = 0.05$. A one-tailed test is initially applied as Stock should not deliver better client performance than mBridge, due to the latter providing a persistence of connection to the anonymity network. If Alice does not hand-off during the scenario then both mBridge and Stock should, in theory, generate equal client performance and therefore a two-tailed statistical test is also applied.

A list of ρ-values for the different tests of statistical significance between Stock and mBridge client performance are shown in Table 6.2. The ρ-values for both one-tailed and two-tailed for each of the statistical tests are below each respective threshold (α) and therefore considered *extremely* statistically significant. This indicates strong evidence against the null hypothesis (*H0*), so it is rejected, and *H1* accepted, that the mBridge solution has had a statistically significant effect on client performance. The result is expected as the mBridge solution, by design, should achieve the highest client performance, and consistently with little or no variance compared to Stock.

Statistical test	**One-tailed**	**Two-tailed**
Paired *t*-test	<0.0001	<0.0001
Welch's *t*-test	<0.0001	<0.0001

Table 6.2 The ρ-values from different tests of statistical significance between Stock and mBridge client performance (b) applying the random waypoint (RandomWP) mobility model over 600 seconds $\alpha = 0.05$

Statistical test	**One-tailed**	**Two-tailed**
Paired *t*-test	0.0002	0.0003

Table 6.3 The ρ-values of statistical significance between Stock and Kaplan-Meier client performance (b) applying the random waypoint (RandomWP) mobility model over 600 seconds ($\alpha = 0.05$)

The Kaplan-Meier based adaptive client throttling achieves the lowest performance at a 12.89% drop in performance compared to Stock. This is expected as the Kaplan-Meier scheme is likely to throttle the client at some point, and if a 'late miss' scenario occurs, throttle unnecessarily. A list of ρ-values for different tests statistical significance between Stock and Kaplan-Meier client performance can be found in Table 6.3. As theoretically an 'early miss' may generate equal client performance, both one-tailed and two-tailed statistical tests are applied. However, Welch's t-test is not retained as neither variances are constrained. The ρ-values for each of the statistical tests are below each respective threshold (α) and therefore again considered *extremely* statistically significant. This indicates strong evidence against the null hypothesis (*H0*), so it is rejected and *H1* accepted as having had an effect.

As previously discussed, the throttling solution's main aim is to reduce the impact of mobility, 'bad' data while maintaining reasonable client performance. The Kaplan-Meier based adaptive throttling scheme provides a 6.07% reduction in mean bad data generated compared to Stock. The list of ρ-values for different tests statistical significance between Stock and Kaplan-Meier can be found in Table 6.4. The ρ-value is above the threshold α = 0.05, and therefore, the difference is not statistically significant. This indicates weak evidence against the null hypothesis (*H0*), so fails to reject, and therefore *H1* cannot be supported to having had an effect.

This justifies the concerns raised at the end of the previous chapter that the Kaplan-Meier solution may not be as effective with lower mobility. To confirm this theory two additional data sets are generated. The first is to extend the scenario time from 600 seconds to 3600 seconds. This is to determine whether the limited number of hand-offs generated by the random waypoint mobility modelling is a possible cause. The increased number of hand-offs from running the scenario over a simulated one hour rather than 10 minutes enables

Statistical test	**One-tailed**	**Two-tailed**
Paired *t*-test	0.2682	0.5365

Table 6.4 The ρ-values of statistical significance between Stock and Kaplan-Meier reduction in impact (o) applying the random waypoint (RandomWP) mobility model over 600 seconds ($\alpha = 0.05$)

Statistical test	**One-tailed**	**Two-tailed**
Paired *t*-test	0.0128	0.0256

Table 6.5 The ρ-values of statistical significance between Stock and Kaplan-Meier reduction in impact (o) applying the random waypoint (RandomWP) mobility model over **3600** seconds ($\alpha = 0.05$)

the algorithm reduces the mean amount of bad data generated to 6.71% compared with Stock. Importantly, the denser populated dataset provides a paired t-test that is statistically significant, lower than $\alpha = 0.05$, as shown in Table 6.5. In contrast to the previous data, this indicates strong evidence against the null hypothesis (*H0*), so it is rejected and therefore *H1* accepted as having had an effect.

The same experiment is undertaken applying the HOT values generated from the previous linear mobility model. The ρ-value is below the thresholds (α), and therefore, the difference is statistically significant, as shown in Table 6.6. This indicates strong evidence against the null hypothesis (*H0*), so it is rejected and therefore *H1* accepted as having had an effect.

This confirms that the Kaplan-Meier algorithm is more effective once enough data is populated within the HOT array. In addition to the enhancements previously proposed, such as a manually switched roaming mode, geolocation/Global Positioning System (GPS) tracking, or tuning the algorithm to identify low mobility, it may be that the HOT array requires ‘priming’ before activation. It also illustrates the effect that different mobility models can have on the results. As with the undesirable effect observed with the Kaplan-Meier algorithm, if the user is rarely handing off during the scenario, then the Stock performance, without a persistence of connection to the anonymity network, is expected to achieve client performance closely matching the mBridge solution.

Statistical test	**One-tailed**	**Two-tailed**
Paired *t*-test	0.0013	0.0026

Table 6.6 The ρ-values of statistical significance between Stock and Kaplan-Meier reduction in impact (o) applying the previous **linear** mobility model over 600 seconds ($\alpha = 0.05$)

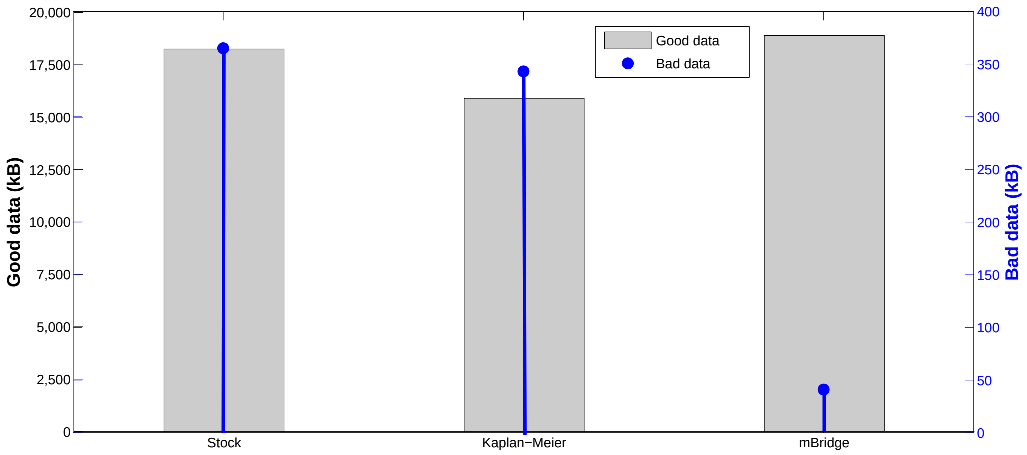


Fig. 6.29 The mean total good (g) and bad data (o) for each solution during the scenario

The amount of bad data generated is, as expected, lowest with the mBridge solution. There is likely to be bad data generated as at least one of the test case scenarios will stop during a file transfer, affecting the mean value. The good data will also be slightly impacted as Alice is required to build the initial circuits even though she maintains a persistence connection throughout the scenario. Both of these are experimental features rather than a weakness of the solution.

Another view of the effectiveness of each of the schemes is the economy achieved of good versus bad data, as illustrated in Figure 6.30. The economy achieved is the ratio of ‘good’ bits versus ‘bad’ bits transferred across the network during the scenario. Again, for clarity, this is not malformed network packets but at an application level. For this case study, the economy provided by the mBridge solution is much higher than the other schemes. This is expected as the mBridge, by design, provides the highest good and lowest bad data through the provision of a persistence connection to the anonymity network. A final point regarding economy is when comparing both good and bad data for the Kaplan-Meier scheme, it is shown than it is actually less optimized than Stock. That is, although the level of economy is similar to Stock as shown in Figure 6.30, the relative amount of bad data generated, compared to good, is greater, as illustrated previously in Figure 6.29.

Alice’s baseline performance for each of the solutions is mapped using the q-factor approach in Figure 6.31. This includes both the performance threshold of $b_\theta = 150$ (Kibit/s) but also another line marker for 250 (Kibit/s). The dual line markers will be discussed later. The q-factor mapping provides more detail for the level of optimisation from each of the solutions. Firstly, the default stock configuration performs relatively consistently over the test

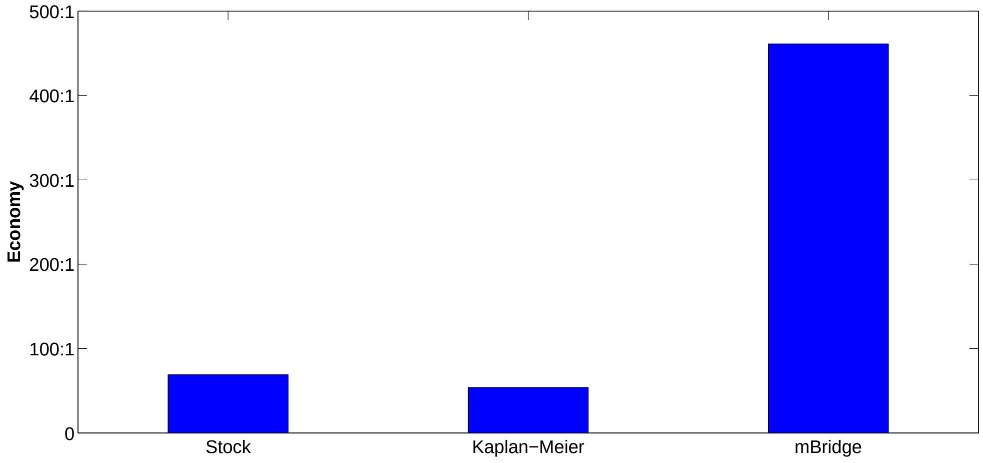


Fig. 6.30 The mean economy achieved by each solution during the scenario

cases. The small number of outliers at its lower end of performance are predominately where Alice's bandwidth is capped due to an over-utilised channel. As previously discussed, the anonymity network provides a robust anonymity solution if being observed (at A-0) without any successful partitioning of users into subsets. The default stock configuration maintains a 100% positive q-factor.

Analysis of the Kaplan-Meier based adaptive client throttling scheme illustrates the intervention of the algorithm from the wider range of mapping distribution, also showing the lower performance. There are number of cases where the performance falls below the performance threshold of 150 Kibit/s. The root cause of this issue will also be discussed later in the chapter. This generates an overall positive level of q-factor in only 92% of the cases. Although, it should also be noted that anonymity is unaffected by this issue and maintains the anonymity threshold for 100% of cases. Finally, for the mBridge solution, the results are again relatively consistent, with the same degree of anonymity but higher performance, although showing the same outliers from the occasionally capped bandwidth.

Although, the case study has been partly set up to benefit analysis, if Charlie can isolate the channels, the reduction in anonymity and consequently also a reduction in positive q-factor soon appears. The first main attack (A-1), to distinguish users from different channels, whether through latency or some other signature, the previously robust anonymity provided by the anonymity network is quickly reduced. The initial results of attack A-1 are shown in Figure 6.32. The degradation of anonymity after the first attack already has taken effect with a number of borderline cases are already generated for the anonymity component, even though resource allocation remains unaffected.

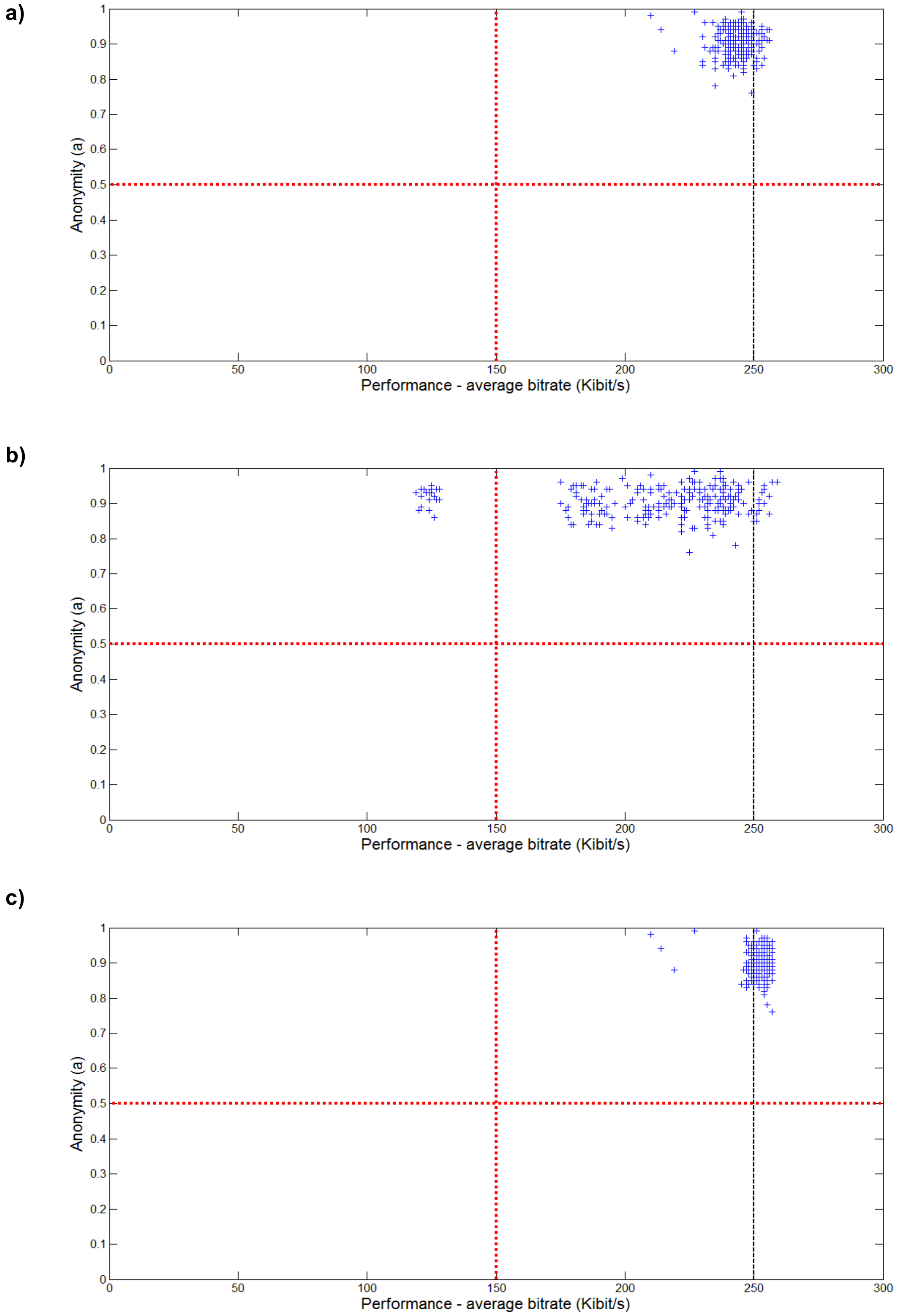


Fig. 6.31 Alice’s *q*-factor mappings after ‘attack’ A-0 (base analysis) using a) Stock b) Kaplan-Meier c) mBridge (*public*)

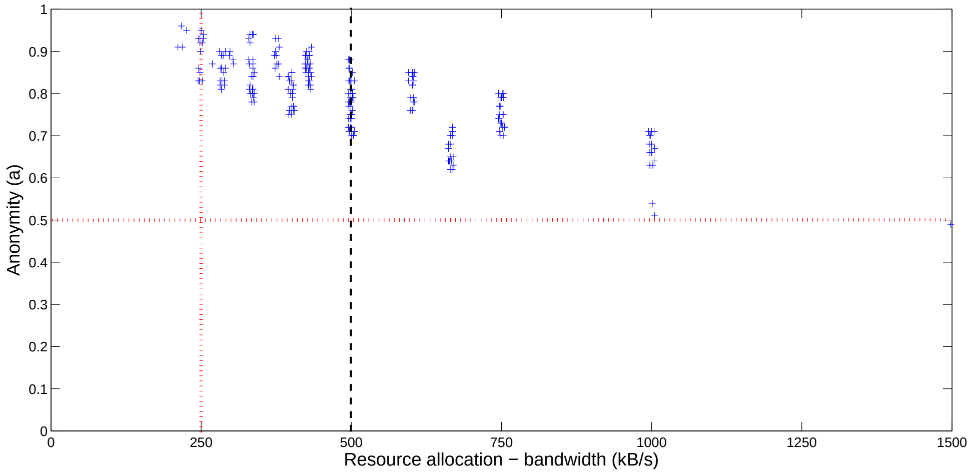


Fig. 6.32 Alice's q-factor mapping (resource allocation) after attack A-1

The effect of attack A-1 on *actual* performance for each of the solutions is shown in Figure 6.33. The identification of which channel has a relatively consistent effect across the solutions. That is, all the solutions can successfully defend against the attack but Alice starts to become vulnerable ($a = 0.5$). This illustrates again that by applying intervention alongside q-factor, borderline cases could be rectified by redirecting users from another channel.

Based on the results so far, in its present form, the Kaplan-Meier estimator adaptive client throttling, is now looking weaker as a solution for supporting mobility. This is initially based on poor performance but also as will be shown later, there are also new concerns regarding anonymity. Focusing on mBridge solution for the next set of results, from attacks A-2 to A-5, even though the mBridge solution maintains good performance, the solution may not be that effective from an overall perspective. The degree of anonymity provided by mBridge becomes more and more negatively affected after subsequent attacks and the number of positive q-factor mappings quickly starts falling. From the ratio of bridge and non-bridge users this partitioning, alongside the natural network churn, starts to provide Charlie more capability. After Charlie can identify whether Alice is using the mBridge solution, by harvesting all the public mBridge IP addresses from the bridge authority (BridgeDB), as part of attack A-2, the number of positive q-factor mappings falls, as shown in Figure 6.34. The negative impact on anonymity, and consequently q-factor, continues throughout attacks A-3 to A-5 as shown in Figures 6.35, 6.36, and 6.37 respectively.

A summary of q-factor maintenance for the mBridge (public) solution for attacks A-2 to A-5 based on information gained by Charlie can be found in Figure 6.38. As shown, once Charlie starts combining further attacks, the percentage of times where q-factor is maintained

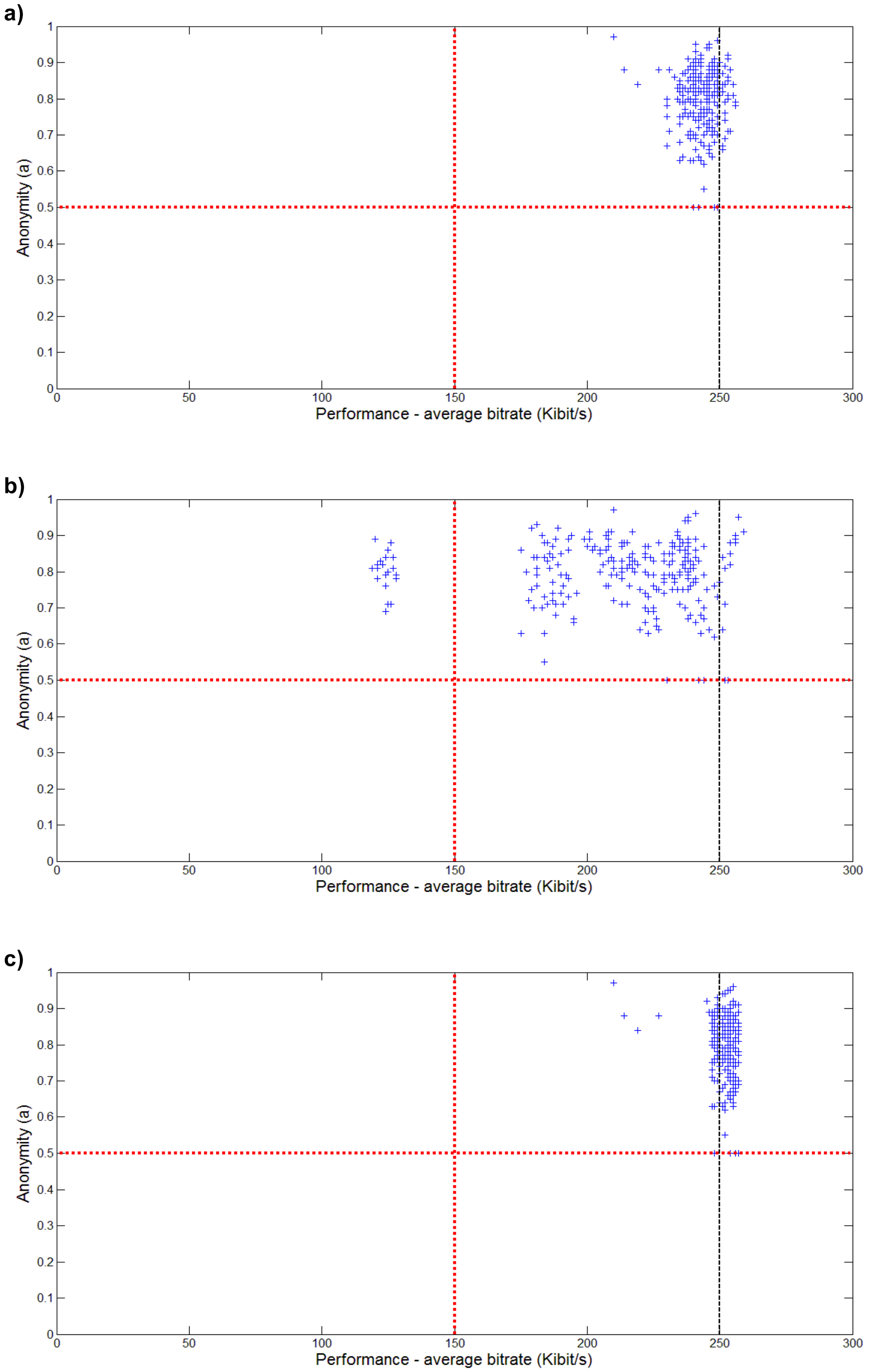


Fig. 6.33 Alice's q-factor mappings after attack A-1 for a) Stock b) Kaplan-Meier c) mBridge

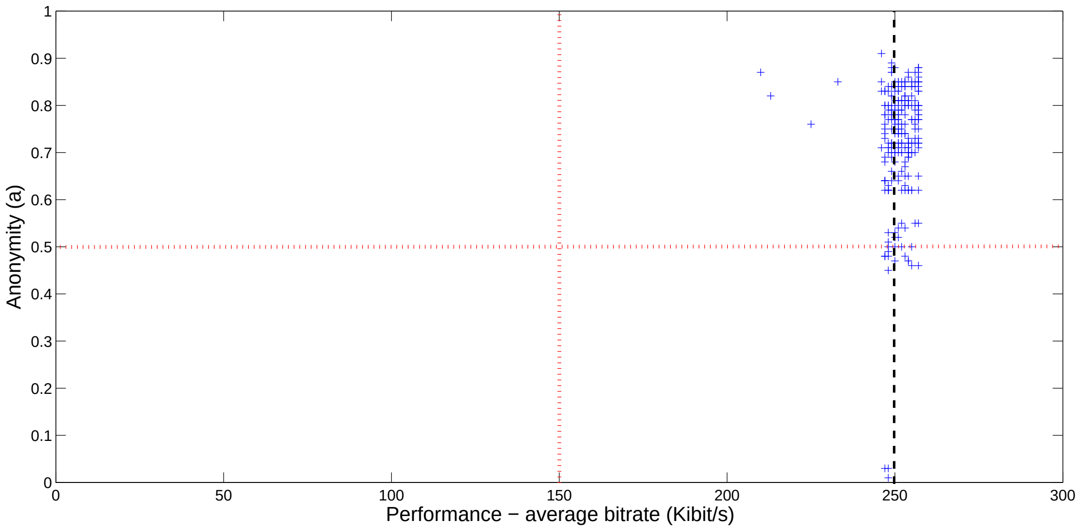


Fig. 6.34 Alice’s *q*-factor mapping after attack A-2 (mBridge)

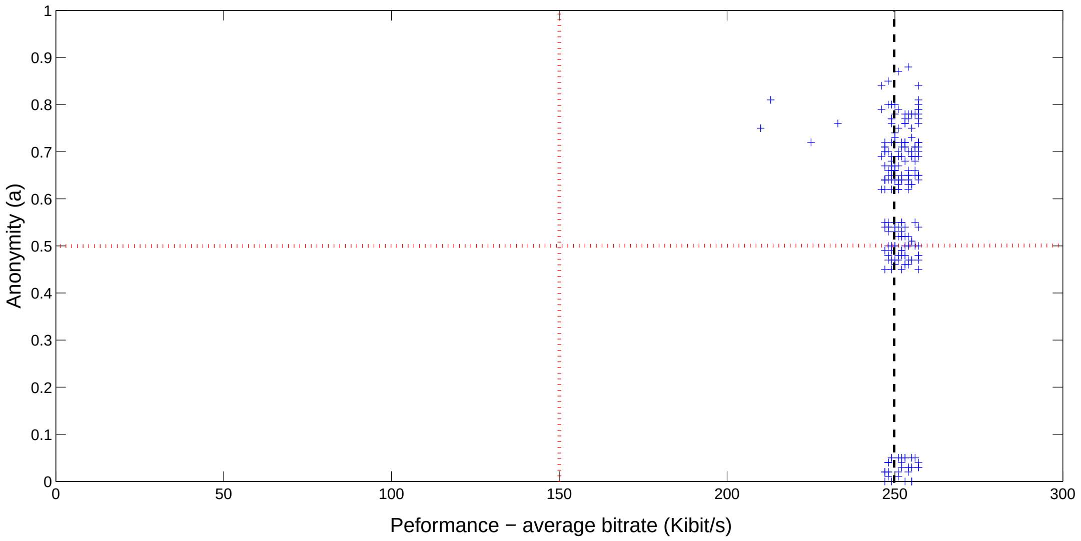


Fig. 6.35 Alice’s *q*-factor mapping after attack A-3 (mBridge)

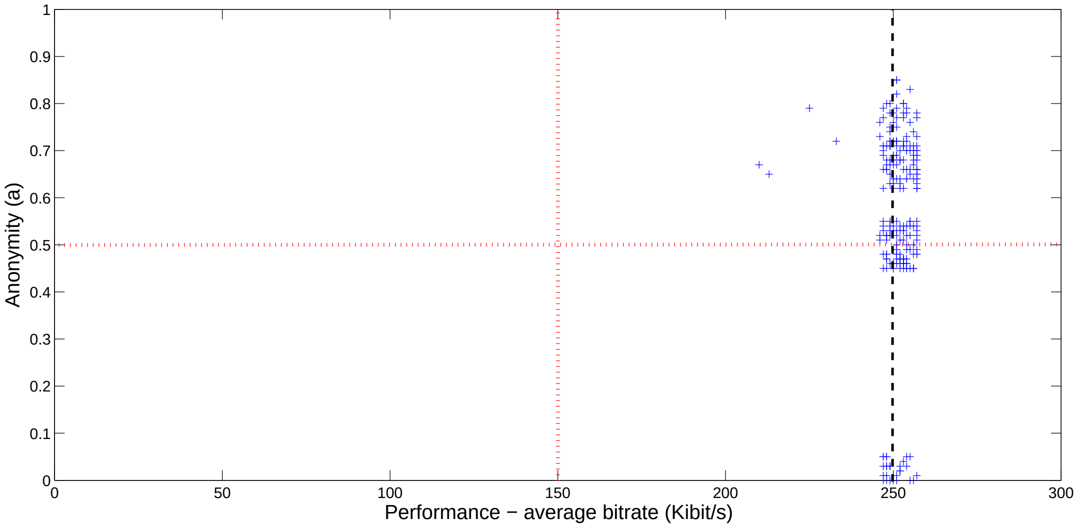


Fig. 6.36 Alice's q-factor mapping after attack A-4 (mBridge)

drops substantially, when after an attack such as A-5, just over half of the observations Alice meets the anonymity threshold, even if the performance threshold is always met.

Based on the results of the public mBridge solution, private bridge is hypothesized to fail maintaining possinymity if Alice is the only user connected to the bridge. The use of a private mBridge is therefore considered unlikely to be a viable solution to support mobility and is rejected at this juncture. However, the private mBridge solution will be re-examined alongside the Dissent protocol later on in this chapter.

As previously noted, the anonymity threshold set ($a \geq 0.50$) is not particularly high (aspirationally) and therefore would be the absolute minimum requirement, affording no more than *possible innocence* at best. If higher anonymity is required, it is shown in Figure 6.39 that unlike higher performance, the mBridge solution is unable to support the levels of anonymity required of $a = 0.67$ and $a = 0.80$, and consequently maintain a positive q-factor value, although the performance threshold of $b = 150$ is always met.

In contrast, the mBridge solution is the only scheme that can achieve a reasonable level (98.4%) of an enhanced performance threshold of 250 Kibit/s. If, a cap is not applied, the mBridge solution achieves the threshold in 100% of cases. The other two schemes, Stock and Kaplan-Meier, would not be able to support the requirements for higher performance due to a lack of persistence of connection, as shown in Figure 6.40. Therefore, if the mBridge solution could provide better possinymity, then it would become even more effective in supporting mobility.

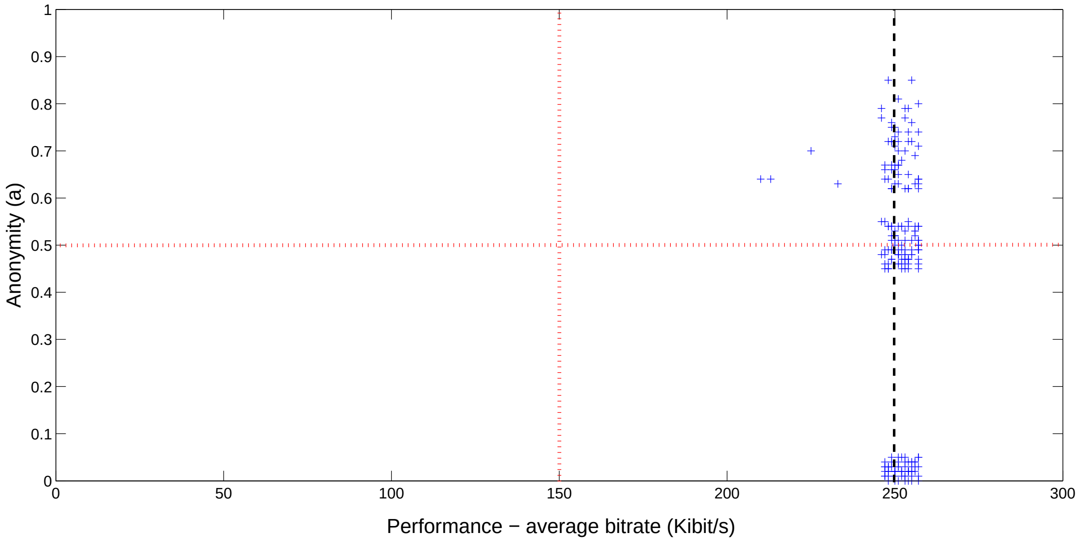


Fig. 6.37 Alice's *q*-factor mapping after attack A-5 (mBridge)

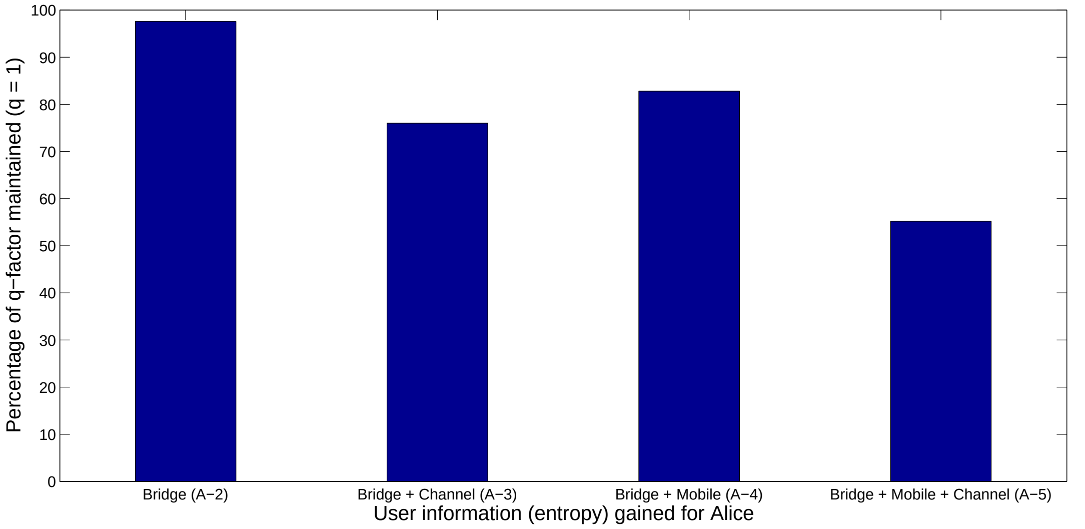


Fig. 6.38 A summary of Alice's *q*-factor maintenance percentage levels for the public mBridge solution at thresholds of $a = 0.5$ and $b = 150$ (Kibit/s)

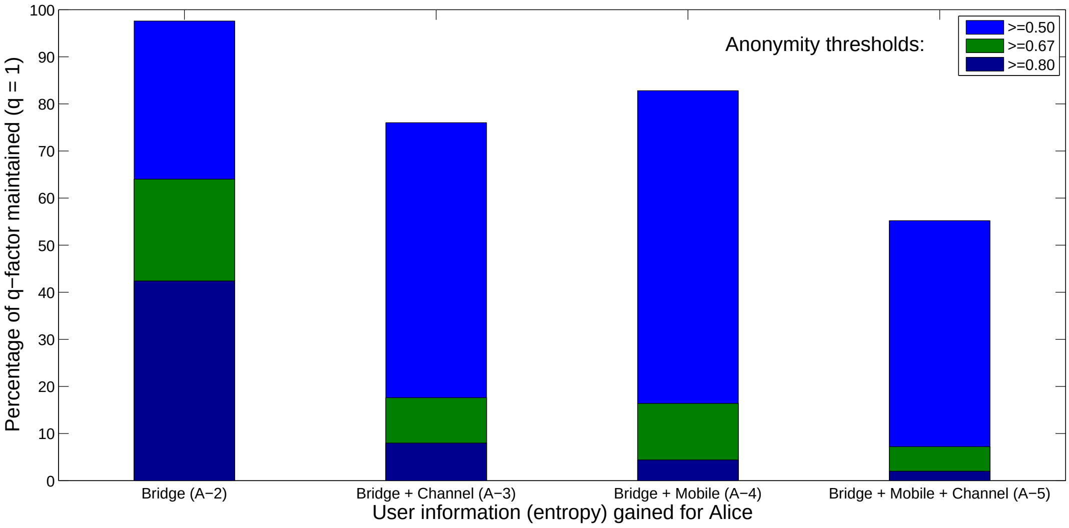


Fig. 6.39 A summary of Alice's q-factor maintenance percentage levels for the public mBridge solution at thresholds of performance b = 150 (Kibit/s) and anonymity a = 0.5, a = 0.67, a = 0.80

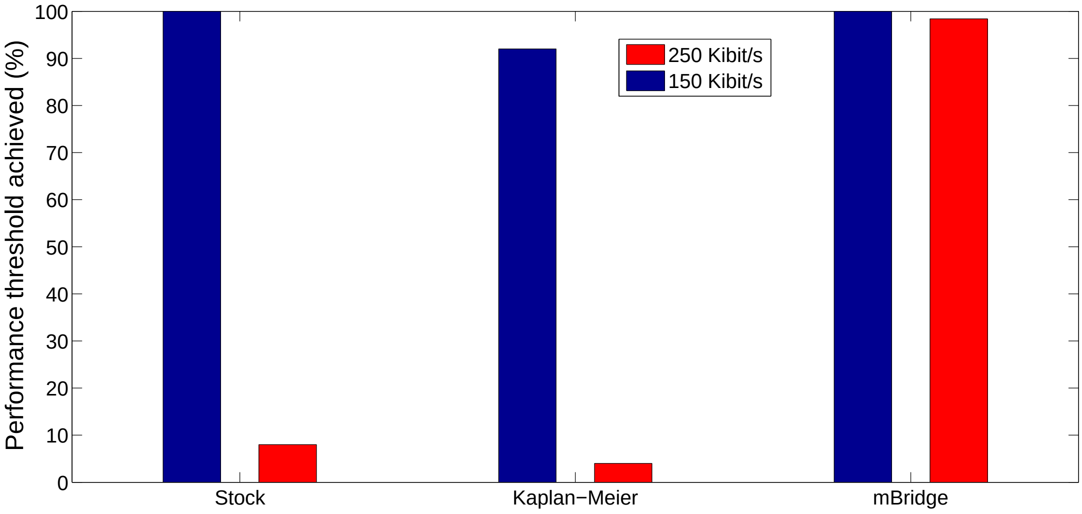


Fig. 6.40 A summary of Alice's performance (b) of 150 Kibit/s compared to a proposed enhanced performance requirement of 250 Kibit/s

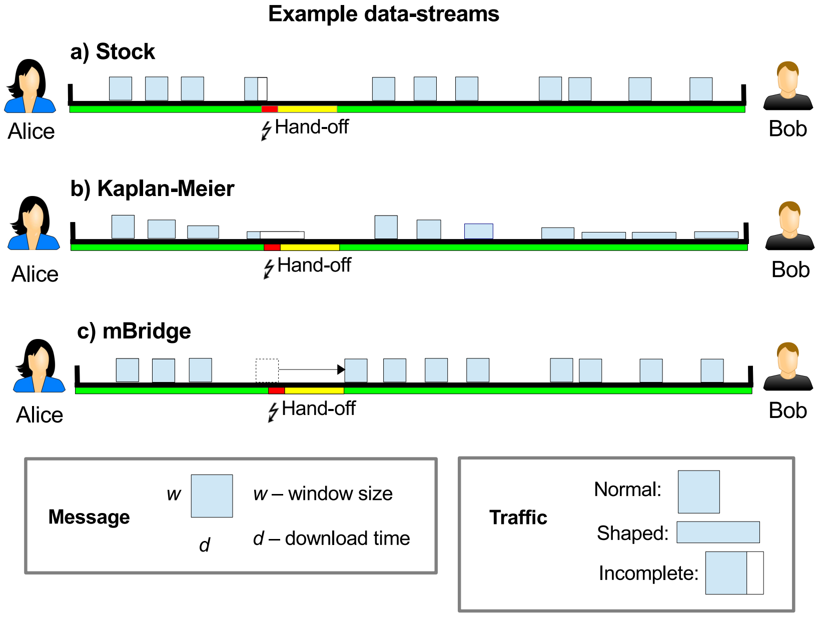
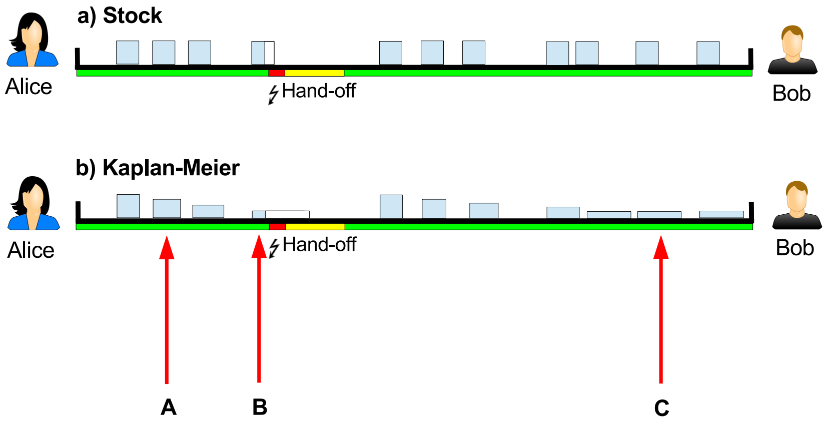


Fig. 6.41 The different schemes' data streams and distinguishing features

As previously noted, the benefit of the q-factor approach is that the underpinning measurements are pluggable. The previously used anonymity metric, possinymity, based on an anonymity set based approach, simply reflects size rather than strength. The indinymity measure is to understand the strength of the anonymity defending against attacks A-6 and A-7, that is, how distinguishable Alice's traffic for each of the solutions, as shown in Figure 6.41.

It was observed in Chapter 5 that the Kaplan-Meier based throttling scheme provides the best anonymity, or least distinguishable traffic, over the other throttling solutions. This probability is based on comparing download times with a cumulative distribution function (CDF) of training data. The training data used by Charlie is also based on empirical data from previous related research on Tor client performance. For example, the 'heavy' (and static) throttling scheme not only provide the worst client performance but also leaves a significant signature. That is, not only are download times are substantially worse than the other schemes applied, the traffic is shaped so that download times appear more uniform with the traffic not peaking above the 50 kB (100 cells) level. Therefore, this is poor solution for both performance and anonymity, specifically indinymity. The additive increase multiplicative decrease (AIMD) fares slightly better but again the algorithm tended to restrict the level of circuit window size to the middle to lower levels therefore providing some intelligence to an observer that the user is throttled. As shown in Figure 6.42, even with the Kaplan-Meier



Distinguishing features:

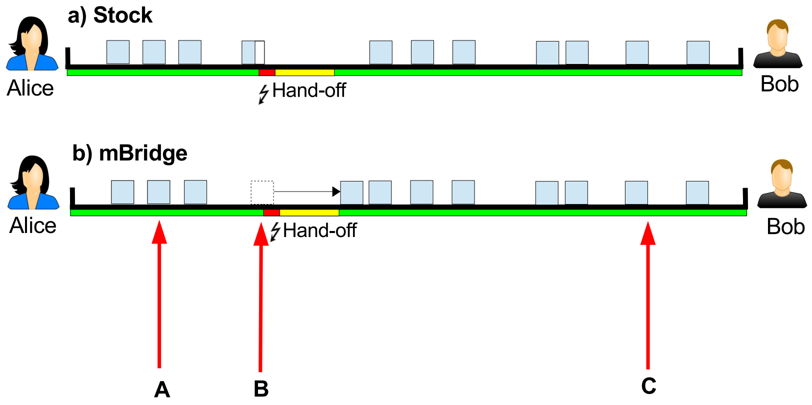
A. Observed tapering off in performance between hand-offs

B. Consistently less data left at the entry point buffers

C. Overall longer mean download times than stock (and mBridge)

Fig. 6.42 An example traffic flow pattern showing a *higher* level of distinguishing features from the Kaplan-Meier solution compared to Stock

estimator, there still remains the opportunity to observe whether Alice, as part of attack A-6, is being throttled and therefore must be a mobile user. The distinguishing features may include, a consistent tapering of performance levels between hand-offs, mean reduction in bad data left at the buffers on recycling the connection, and longer download times than expected, compared to other users. The mBridge solution, in theory, does not have the same issues as above. As shown in Figure 6.43, it could be argued that the features are far less defined and could easily be lost in the latency introduced by onion routing. If the bridge relay does not at any time set the circuit window size different to a desktop connection then this particular issue is not applicable. Secondly, download times, as part of the overall client performance from the mBridge solution are less distinct. This lower level of deviation would be less distinguishable again within normal deviations of network latency, which are higher on low-latency anonymity networks due to the architecture and multilayered encryption. That is, if the small delay in redirecting the traffic to Alice after hand-off is approximately one second, this may not distinct enough, for certain types of attack such as end-to-end traffic correlation (confirmation) using packet counting and timing analysis. Indeed, if Charlie is only observing Bob's connection and has limited insight to Alice's network connection, Charlie may not even ascertain that Alice is indeed a mobile user, therefore potentially obfuscating Alice's connection to being a desktop, or unclassified, user.



Features:

A. No data left at the entry point buffers.

B. Comparable (with stock) mean download times..

C: Similar traffic patterns.

Fig. 6.43 An example traffic flow pattern showing less distinguishing features from the mBridge solution, compared to Stock, than the Kaplan-Meier solution.

6.4 Discussion

The mBridge solution, by providing a persistence to the anonymity network, appears to be the most logical approach. The solution, however, does raise some specific concerns regarding anonymity. As previously discussed in Chapter 2, maintaining anonymity on the Internet is not trivial. Even an established, popular implementation such as Tor, although appears to be effective against certain attacks, is still vulnerable to attacks for which no solution is known, for instance, packet counting and timing. The Tor project publicly states the risks have to be accepted by the user, especially a global adversary specifically targeting their connection [67]. As a reminder, some example attacks are as follows:

1. Exit node / point ‘sniffing’ of unencrypted traffic for example, using the insecure Hyper Text Transfer Protocol (HTTP).

2. Traffic analysis, end-to-end correlation (confirmation), through passive (fingerprinting), active (staining) of traffic signatures through packet counting and timing.

3. Intersection attacks, by rapidly narrowing the anonymity of a target via linkable actions across time.

4. Active attacks for example, congestion-based, false-advertising, Sybil.

5. Poor design, development, and testing. Software exploits, also external vulnerabilities for example, Firefox, Heartbleed.

6. Operator and user error.

The mBridge solution's weakness to indinymity appears to be no worse than the default stock configuration, and better than the adaptive client throttling. For maintaining possinymity, however, mBridge solution appears weaker. The main issue, also reflecting on the operation of Tor bridge relays, is that identifying a public bridge relay user is relatively simple. This, in conjunction with the lower recorded number of bridge compared to direct connections, means that from an anonymity set perspective, there is already a reduction in the potential anonymity provided [178]. As with the previous example in Chapter 4, how many Tor users are from Cyprus? How many of these are direct connections or use bridge relays? How long would it take Charlie to harvest the Internet Protocol (IP) addresses of Cypriot users connecting to public bridge relays, and a Cypriot student from Northumbria University at Newcastle's external IP address? If this is also considered alongside other attacks, such as Charlie is able to monitor the connection from Alice's exit node and Bob's blog, this could also be supplemented by reveal entropy bits of user-agent information, such as language settings of *EN-CY* if unencrypted. Add this to intersection attacks, software exploits, and finally user error, then the task becomes more difficult.

Although, a private bridge potentially mitigates the bridge IP address harvesting, Deep Packet Inspection (DPI) may also identify Alice as a user of the anonymity network. If checked against the directory of relays and list of harvested public bridge details, it is possible to determine that Alice is operating a private bridge within relatively short time. If this is the case, then traffic analysis and active attacks could become highly effective if there are only few concurrent user connections, or in the worse case scenario, if Alice is the only user. Therefore, Charlie, specifically targeting Alice's connection, could apply the logic in Figure 6.44, to expose her. The mBridge proposal is not the *perfect* solution to supporting mobility but a contribution by:

1. Persistence of connection: providing better than Stock client performance while mobile.

2. Reduced impact: no loss of data while mobile.

3. Trust: a *trusted* agent to redirect traffic and support mobility.

4. Resistance to traffic analysis: no worse indinymity than Stock.

The key outstanding issue is the effect on possinymity. There are, however, a number of potential approaches to prevent a mBridge user becoming more vulnerable:

1. Increase the pool of public mBridge relays.
2. Rotate public mBridge connections.
3. Recycle mBridge connections automatically after 10 minutes.
4. Attract more users to mBridge relays to generate more concurrent users per bridge.
5. Implement q-factor to identify 'weak' bridges and undertake the appropriate intervention to help maintain possinymity.
6. Obfuscation of mBridge traffic and usage.
7. Further resistance to traffic analysis (indinymity).

The issues 1 to 4 are self-explanatory and attempt to improve anonymity through increasing the set size and reducing the time spent on each bridge to make traffic analysis more difficult [178] [179]. The implementation of q-factor (issue 5) has previously been covered in depth in Chapter 4. The final two proposals (6 and 7) are related to the previous example of Charlie's logic in Figure 6.44. As shown, Charlie first needs to identify whether Alice is using the anonymity network. As previously discussed, this may be relatively simple by observing the 'known' signature of anonymity network traffic in the same way Tor traffic has a distinct signature. It is then a case of eliminating connection types and selecting the most appropriate attack to follow-up with. There has already been development to obfuscate Tor as Skype and other traffic types [109]. This makes sense, if Alice is not identified as using the anonymity network, then she may not be targeted for further investigation. In this example, the obfuscation approach has the inherent weakness. If Charlie also blocks the Internet Protocol (IP) addresses of the publicly listed relays and bridges, then it is still relatively straightforward to identify Alice as a user of the anonymity network. In theory, the obfuscation solution is only robust when used in conjunction with a private bridge. Also, any weakness in the obfuscation, which with networks forensics is feasible, this potentially has a catastrophic outcome. Alice may become overly dependent and over-trusting, and only allow her connection to the private bridge, which a failure of the obfuscation, Alice will become more easily exposed. If the technology of which the obfuscation is using is also banned (or even itself subject to greater scrutiny), such as Skype (SkypeMorph for Tor), then this solution may not be viable, or in fact undesirable.

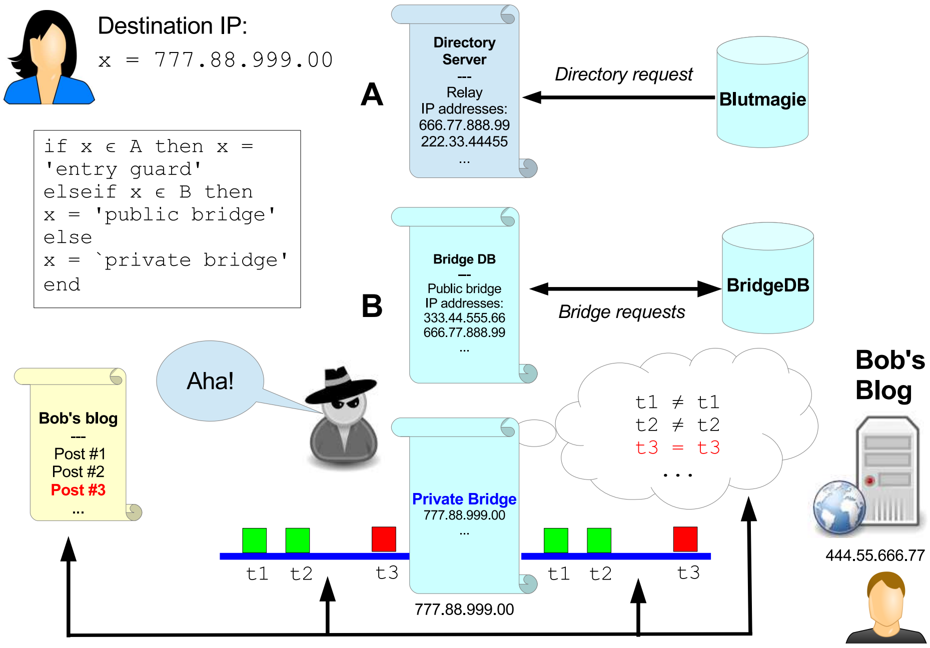


Fig. 6.44 Charlie's logic in determining Alice is using a *private* bridge

An alternative approach to obfuscation, is *pseudo-obfuscation*, by allowing the connection to the anonymity network to be one-step removed, that is, via an additional proxy. However, with a step removed, the proxy must also be trusted and additionally mix the traffic to such a degree it is more difficult to undertake traffic analysis. The outcome would be Charlie is unable to identify that Alice is connecting to the anonymity network. Also, the signature to Bob's blog is mixed to provide resistance to traffic analysis. To achieve this for a low-latency anonymity network alongside high network churn is challenging. In the Further Work section of the next chapter, the Dissent protocol is examined to how it may be able to enhance the current mBridge solution, but also in turn the mBridge solution could help Dissent in supporting mobility.

6.5 Summary

In this chapter, the mBridge solution is evaluated to whether it is able to mitigate the impact of mobility. The results show that the mBridge solution reduces the negative performance impact of mobility to the user, and potentially the anonymity network. Depending on the level of redirection overhead, this negative impact of mobility can almost be eradicated. The

	a (*possinymity*)	a (*indinymity*)
Kaplan-Meier	🟢	🟢
mBridge	🟡	🟢

Anonymity

	a *(combined)*	b *(performance)*
Kaplan-Meier	🟢	🟡
mBridge	🟡	🟢

Anonymity & Performance

Fig. 6.45 A summary of the findings for the mBridge and Kaplan-Meier solutions compared to Stock; green = 'no concerns', amber = 'some concerns'

balance between anonymity and performance is a critical, and at times, complex. The use of bridge relays, especially a connection to a private mBridge, raises concern regarding its potential negative effect on anonymity. As even shown with public bridges, the identification of a bridge user can start reducing anonymity. Furthermore, if used as part of a combinatorial attack, this can quickly reduce anonymity compared to an 'organic' environment, such as the default stock configuration. That aside, the proposed mBridge solution offers greater potential than the Kaplan-Meier estimator based adaptive client throttling in supporting mobility. An overall summary of the findings is presented in Figure 6.45. Each of proposed solutions for supporting mobility have their own weaknesses in maintaining anonymity or performance. However, existing implementations, such as Tor, have been increasingly attacked over the last few years and whether a clean slate approach, as suggested by the Dissent project, is required strategically to support both anonymity and mobility. This may seem far away aspirationally, however, as an interim, a hybrid mBridge and Dissent solution (mDissent) may offer a more robust alternative.

Chapter 7

Conclusions and Further Work

7.1 Conclusions

Mobility, and its potential impact on anonymity and performance, for low-latency anonymity networks such as Tor network has not been re-examined since 2007. The development of smartphone technology and subsequent increase in mobile Internet usage generated a number of research questions:

1. What is the impact of mobility on performance using Tor while roaming?
2. What is the effect of mobility, and increased network churn, on anonymity?
3. What is the best approach to mitigate the impact of mobility while maintaining acceptable anonymity and performance?
4. How should anonymity and performance be best measured in dynamic environments?

The predicted parity of the number of mobile and desktop Internet connections meant this research is timely. Each of the research questions were addressed as follows:

1. What is the impact of mobility on performance using Tor while roaming?

In Chapter 3, by applying a range of approaches: field experimentation, network simulation, and mathematical modelling; while a mobile Tor user is roaming, the time required to rebuild the Tor circuits after each hand-off, negatively impacts client performance. This research is the first to highlight empirically the negative impact of mobility and enough to warrant further investigation.

2. What is the effect of mobility, and increased network churn, on anonymity?

In Chapter 4, while demonstrating the new q-factor metric, it is shown that the increased network churn can greatly impact anonymity. By using the q-factor metric, not only is any performance gain measured but also anonymity is considered when measuring the effectiveness of the proposed solutions. The new q-factor metric is proven to be more effective at monitoring low-latency anonymity networks by identifying critical events, in particular for anonymity, from the increased network churn.

3. What is the best approach to mitigate the impact of mobility while maintaining acceptable anonymity and performance?

In chapters 5 and 6 the effectiveness of the two proposed solutions for supporting mobility on low-latency anonymity networks are evaluated using the q-factor metric to measure. The mBridge solution is found to provide the best overall performance. However, the use of bridge relays, especially 'private' bridges, to support mobility raises anonymity concerns. The Kaplan-Meier estimator based algorithm is found to provide the most effective adaptive throttling scheme for supporting mobility, although, it may require further enhancements to manage more efficiently scenarios with 'limited' mobility. The proposed migration of Tor from Transmission Control Protocol (TCP) to a lighter User Datagram Protocol (UDP) based transport, (μTP), has stalled. Dissent, a clean slate development of a practical, provable, anonymity system, still remains an experimental prototype not yet ready for widespread deployment to 'normal' users. The Dissent project however remains particularly interesting for its potential symbiosis with future work to enhance the mBridge solution, while also mBridge providing mobility support for Dissent. The 'optimisation' of low-latency anonymous communications, if not undertaken with care, can become a self-defeating paradox. As previously discussed, the balance between anonymity and performance is critical for maintaining effective low-latency anonymity networks. By applying Anderson's rule, too strong security (anonymity) can make a system unusable, deterring usage, and eventually lowering anonymity itself. Is an 'optimum' solution even achievable? There are also many uncontrolled external variables, such as the physical network performance, of which the anonymity network operator has no control. Even then, the level of service does not necessarily correlate to user requirements (and expectations). This is particularly difficult for anonymity networks where user opinion and technical error reporting may not be as forthcoming as 'normal' Internet services. Therefore, it has to be accepted there are limitations to how far a service can be optimized. A more appropriate statement may be to add: '. . . where the level of optimisation equals the quality achieved at the cost of service', see APPENDIX B for further details.

4. How should anonymity and performance be best measured in dynamic environments?

As onion routing, and consequently Tor, were originally designed for static wired Internet connections, the anonymity metrics have evolved accordingly. On reviewing the existing approaches to the measurement of anonymity and performance, weaknesses in the metrics used for low-latency anonymity networks were identified. A new metric (q-factor) is presented to measure the trade-off between anonymity and performance over the duration of a communication session. A range of solutions, including the new metric, are constructed based on this analysis to be evaluated in the subsequent chapters. In chapters 5 and 6 the use of the q-factor metric was successfully proven.

The consensus is that onion routing, based on Chaum's original mix network design, remains secure. Tor itself, however, is becoming more vulnerable. This is not a criticism of Tor but an unfortunate consequence of its own achievements. Due to its popularity, with over two million daily users worldwide, Tor has increasingly become the focus of attacks. However, it is not only adversaries that Tor needs to adapt to but also its own users. Mobile Tor users can negatively impact both anonymity and performance. The key points identified from this research are summarized as follows:

- The increase in mobile Internet usage is changing the dynamics in low-latency anonymous communications.

- Anonymity and performance are intrinsically linked in low-latency anonymous communications' systems such as the Tor anonymity network.

- Increased mobility generates a *direct* negative impact on client performance for a mobile Tor user while roaming.

- Data remaining in-flight at the break in connection, and any resubmitted failed requests, generates additional congestion on the Tor network. This may cause a negative impact on performance for all users, whether mobile or not.

- The additional network churn from mobile users impacts anonymity increasing the effectiveness of certain types of attack.

- A lack of persistence of connection to the Tor network, when breaking connections to the Internet, and consequently the time required to rebuild the circuits, suggests the existing design of Tor, and its underpinning onion routing, cannot support mobility efficiently.

- Metrics used for measuring the effectiveness of anonymity networks treat anonymity and performance separately. Although the relationship between anonymity and performance suggests this is still a valid approach, a combinatorial metric would be more suitable.

To conclude, a low-latency anonymity networks, such as Tor, need to review its current design to support the increasing mobility of Internet users. A strength of both of the solutions presented in this research, adaptive client throttling and Mobile IP / bridge relays, is that changes are restricted to client devices; the existing algorithms and protocols of the interior Tor network are unaffected. A bespoke anonymity and mobility protocol may, however, be required as part of a longer-term strategy. Reflecting on Anderson's rule, users also need to accept that there will always be a reduction in usability to support security. As illustrated with the Dissent protocol, the shuffle, as with *time* itself, always goes forwards and therefore a performance overhead (latency) introduced. The best strategy is not trying to achieve the strongest anonymity or highest performance but finding the most appropriate balance between the two. This this research makes the following novel and original contributions:

1. The first study to highlight, and quantify, the negative impact of mobility on low-latency anonymity networks, such as Tor.

2. A new, and more effective, metric (q-factor) to measure the trade-off between anonymity and performance in dynamic network topologies.

3. An original application of a Kaplan-Meier estimator used in adaptive client throttling for supporting mobility.

4. A novel application (mBridge) of an existing solution, bridge relays, used for circumventing the blocking of Tor connections, to also support mobility.

7.2 Further work

During the last ten years Tor has had to constantly adapt to changes in technology and user requirements (and expectations). It is difficult to predict what the next 10 years will bring, but based on this research, there are some general points to consider for future work:

1. Mobile devices will likely remain popular, evolve further, and potentially become an all-in-one device, that is, both a mobile and 'desktop' [180].

2. Mobile users will increasingly demand anonymous interactive, real-time services while roaming including mobile Voice-over-IP (mVoIP) [181].

3. Governments, law enforcement, and commercial organizations will continue to undertake monitoring and surveillance of an individual's communications.

4. The 'Snowden effect' will continue to raise the awareness of the human right to privacy of communications.

5. There will always be both 'good' and 'bad' individuals but this should not prevent research on the human right to privacy of communications.

The wider issue of managing horizontal and vertical hand-offs seamlessly, that is, seamless enough not affect real-time services, is still ongoing with projects such as IEEE 802.21 [182]. Without this, Tor and the underpinning onion routing, cannot support mobility efficiently. This research project generates a number of areas for further work. The work can be divided into two main categories: internal and external. Internal builds upon the current work of this research by evaluating the proposed enhancements previously discussed within the relevant chapters:

1. Compare the effectiveness of the single and multilayered q-factor metric in various implementations.

2. Evaluate the proposed enhancements of the Kaplan-Meier estimator based solution and identify other potential applications.

3. Extend investigations into a hybrid mBridge and Dissent solution, namely mDissent.

A decade after the public release of Tor, and with the recent well-publicized attacks, some researchers have began advocating clean slate Internet anonymity designs [183]. Mobile devices are expected to remain popular and may even become an 'all-in-one' technology, replacing traditional desktop use altogether [180]. Therefore, it seems timely to reassess how mobility is supported within low-latency anonymity networks, and anonymous communications' systems as a whole.

Dissent is a clean slate development with the primary aim to provide accountable, provable, anonymous communications [144] [184] [185]. Dissent returns to Herbivore's first attempt to build provable anonymity guarantees into a practical system that addresses the dining cryptographers problem, through a DC-net, see APPENDIX C.i [186] [187]. The design of Dissent uses the core concepts of both DC-nets *and* verifiable shuffles in a client / server architecture, as described in more detail in APPENDIX C.ii. In contrast to Tor, Dissent has built-in 'provably guaranteed' anonymity, with communications shuffled to maintain a defence against traffic analysis, as illustrated in Figure 7.1. How Dissent approaches both active and passive traffic analysis, compared to the Tor network, is illustrated in Figures 7.2

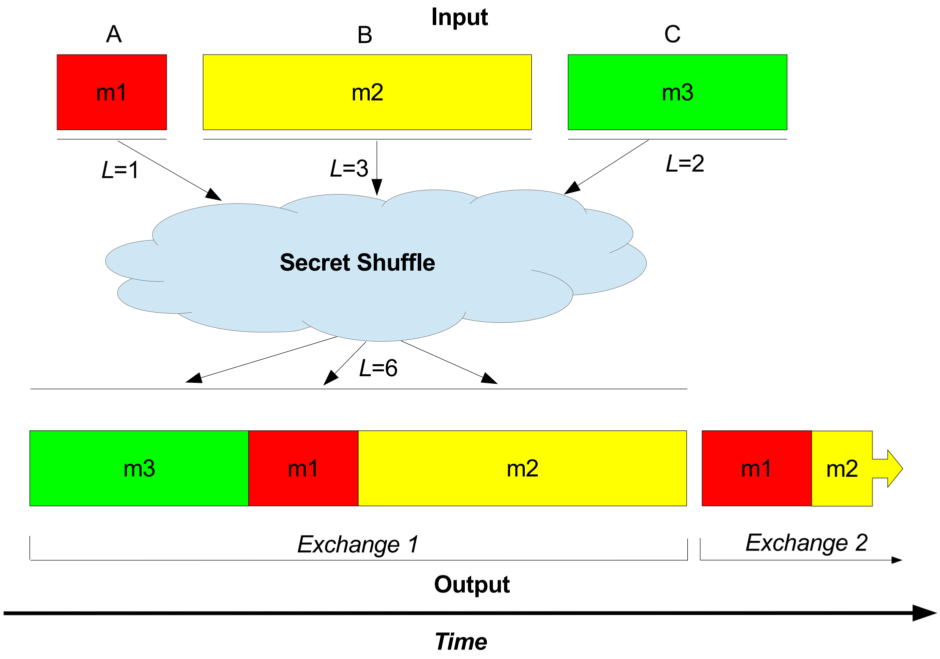


Fig. 7.1 The dining cryptographers' secret shuffle

and 7.3. As Tor handles each client's circuit individually, this generally preserves the timing pattern (although with some (variable) latency) as it passes through the relays. Charlie may be able to distinguish Alice's timing pattern at the exit point, as shown as (a) in Figure 7.2. If Charlie is able to actively stain Alice's traffic pattern before it enters the Tor network, this potentially increases the ability of Charlie to distinguish and therefore confirm it is Alice's traffic, as shown as (a) in Figure 7.3. In contrast, the Dissent approach is less vulnerable to both active and passive traffic analysis as the secret shuffle should not leave a traceable fingerprint or stain even under active attacks, as shown as (b) in Figures 7.2 and 7.3. Although, much less efficient or scalable than onion routing (Tor) for point-to-point communications, Dissent appears to provide a more robust defence against traffic analysis.

At the time of writing, Dissent is an experimental prototype and not yet ready for widespread deployment to 'normal' users. An early study on the performance of Dissent, the results shown in APPENDIX C.iii, show the performance achieved via Tor, Dissent, Dissent and Tor [188]. As expected, 'no anonymity' provides the best client performance due to its minimal security overhead. Tor performs the next best, then the most 'defensive' Dissent. The lowest performance achieved is using Dissent and Tor together although at the same

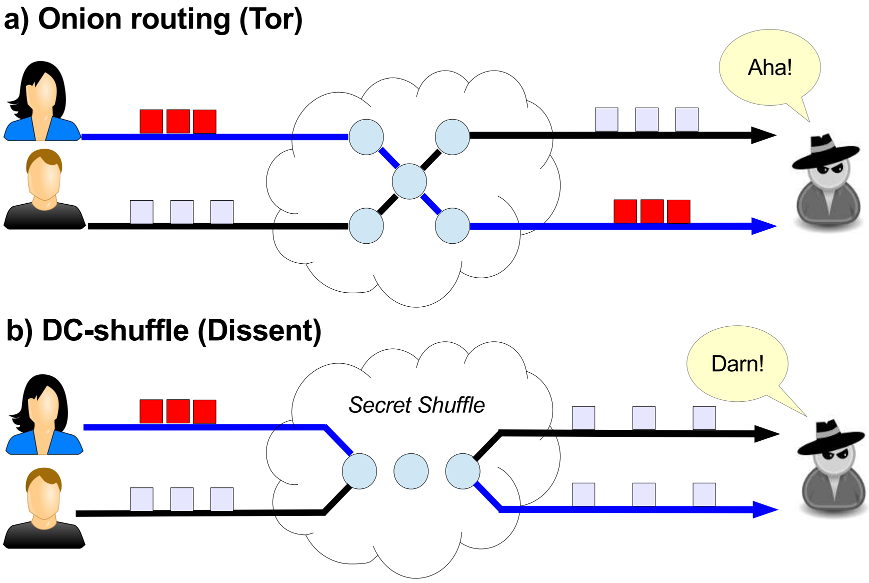


Fig. 7.2 Dissent's defence against *passive* traffic fingerprinting

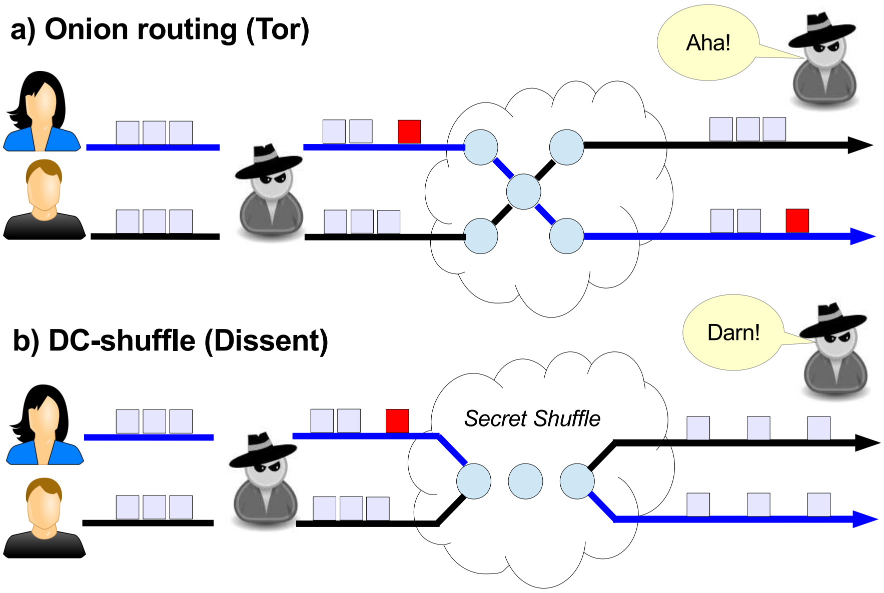


Fig. 7.3 Dissent's defence against *active* traffic marking / staining

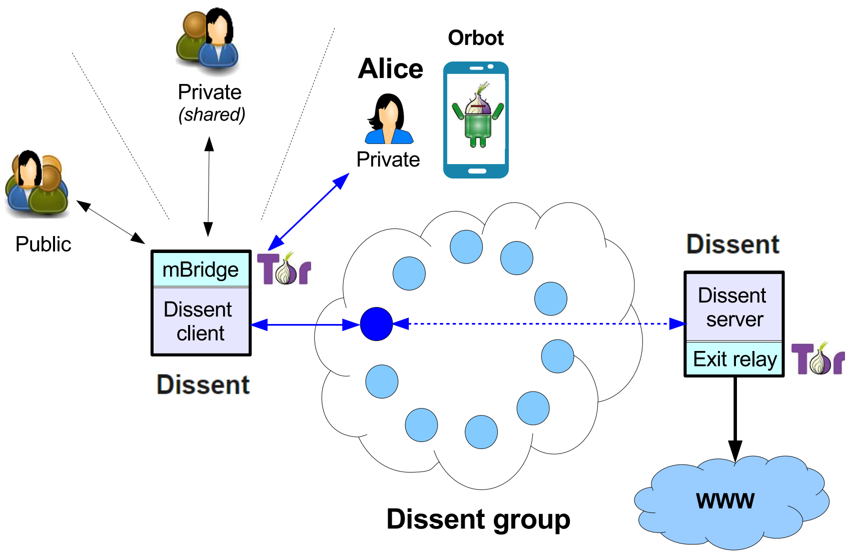


Fig. 7.4 A proposed hybrid mBridge and Dissent solution, named mDissent

time stated to be the most resilient to attacks on anonymity of the all the solutions. There are two key remaining challenges before Dissent is ready for widespread deployment:

1. Dissent needs better scalability (hundreds of thousands of users or more), balancing performance against anonymity guarantees, adequate for interactive web browsing.

2. While Dissent can measure vulnerability to an intersection attack and control anonymity loss, it cannot also ensure availability if users exhibit high churn and individualistic behaviour.

Reflecting on the mBridge solution, a combination of mBridge and Dissent, alongside the implementation of the q-factor metric, may form the template to achieving the elusive goal of provable anonymity with mobility, as shown in Figure 7.4, with further details in APPENDIX C.iv. The potential synergy is due to Dissent unable to support mobility (network churn) effectively and still has some scalability issues, yet has more robust defences, while in contrast, Tor also cannot support mobility efficiently and has inherent weakness against traffic analysis. An early discussion of the key points can be found in APPENDIX C.v.

It is uncertain what the difference in performance, or indeed the effect on anonymity, is between these 'organic' and 'non-organic' approaches with the proposed mDissent approach. It may be dependent on other external factors such as physical network conditions, waiting on other Dissent group clients, and so on. Either way, there needs to be careful consideration of the design as any wrong choice could be counterproductive. It may be that the design needs to be flexible and be able to manage both scenarios. An acceptable 'waiting' period is unlikely to be the average time it takes to rebuild Alice's Tor circuits (3, 5, 7, ... seconds), so either solution will offer performance benefit. Of course, as already outlined, the Dissent solution, with or without the mBridge components provides benefit for anonymity, and therefore a more optimized anonymity and mobility solution. External work relates to factors such as changes in society, technology and the relationship between the two. Not all the proposed work is achievable alone, however, it may generate interest from other researchers for potential collaboration in the future. Each of these items generate their own challenges and technical considerations that need to considered for mobility. For example, any future anonymous VoIP implementation will require a review of the persistence issue, such as Mobile VoIP (mVoIP) support, if to be used efficiently while mobile and maintain anonymity and therefore privacy:

1. Development of anonymous Voice over IP (VoIP) including via Tor [189].

2. HTML5 support from YouTube to replace 'unsafe' Flash, meaning Tor users can watch / listen to live news from censored media sources while mobile [190].

3. Other factors including changes to legislation. For example, the Government of the United Kingdom is planning to repeal the current Human Rights Act and replaced it with a 'bill of rights' [191]. How will this affect an individual's right to privacy?

Document prepared using CUED LaTeX PhD Thesis Template - v2.0 [192]

References

[1] United Nations. The Universal Declaration of Human Rights. Website, 1948. http://www.un.org/en/universal-declaration-human-rights/ [Accessed 05 July 2016].

[2] Council of Europe. Convention for the Protection of Human Rights and Fundamental Freedoms. Website, 1950. http://conventions.coe.int/treaty/en/Treaties/Html/005.htm [Accessed 05 July 2016].

[3] Council of Europe. Declaration on freedom of communication on the Internet. Website, 2003. https://wcd.coe.int/ViewDoc.jsp?id=37031 [Accessed 05 July 2016].

[4] Great Britain. Human Rights Act. Website, 1998. http://www.legislation.gov.uk/ukpga/1998/42/contents [Accessed 05 July 2016].

[5] The Guardian. The NSA files. Website, 2016. http://www.theguardian.com/us-news/the-nsa-files [Accessed 05 July 2016].

[6] Daniel J. Solove. "I've Got Nothing to Hide" and Other Misunderstandings of Privacy. *San Diego Law Review, Vol. 44, 2007*, 2007.

[7] Jeremy Clark. Privacy Enhancing Technologies Symposium (PETS). Website, 2016. https://petsymposium.org/2016/index.php [Accessed 05 July 2016].

[8] Peng Zhong. PRISM BREAK - Opt out of global data surveillance programs like PRISM, XKeyscore and Tempora. Website, 2016. https://prism-break.org/en/ [Accessed 05 July 2016].

[9] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[10] Europol. Press Release: Global Action Against Dark Markets on Tor Network. Website, 2014. https://www.europol.europa.eu/content/global-action-against-dark-markets-tor-network [Accessed 05 July 2016].

[11] Federal Bureau of Investigation. Press release: Operator of Silk Road 2.0 Website Charged in Manhattan Federal Court. Website, 2014. http://www.fbi.gov/newyork/press-releases/2014/operator-of-silk-road-2.0-website-charged-in-manhattan-federal-court [Accessed 05 July 2016].

[12] Sean Gallagher. Law Enforcement Seized Tor Nodes and May Have Run Some of Its Own. Website, 2014. http://arstechnica.com/security/2014/11/law-enforcement-seized-tor-nodes-and-may-have-run-some-of-its-own/ [Accessed 05 July 2016].

[13] Kashmir Hill. How did the FBI break Tor? Website, 2014. http://www.forbes.com/sites/kashmirhill/2014/11/07/how-did-law-enforcement-break-tor/ [Accessed 05 July 2016].

[14] The Tor Project. Thoughts and Concerns about Operation Onymous. Website, 2014. https://blog.torproject.org/blog/thoughts-and-concerns-about-operation-onymous [Accessed 05 July 2016].

[15] David Gilbert. Iraq Crisis: Government Blocks Access to Tor Project Following Isis Insurgency. Website, 2014. http://www.ibtimes.co.uk/iraq-crisis-gvoernment-blocks-access-tor-project-following-isis-insurgency-1452879 [Accessed 05 July 2016].

[16] Keith D Watson. The Tor Network: A Global Inquiry into the Legal Status of Anonymity Networks. *Wash. U. Global Stud. L. Rev.*, 11:715, 2012.

[17] Philipp Winter and Stefan Lindskog. How the Great Firewall of China is blocking Tor. In *Proceedings of the USENIX Workshop on Free and Open Communications on the Internet (FOCI 2012)*, August 2012.

[18] Sean Gallagher. Russia publicly joins war on Tor privacy with $111,000 bounty. Website, 2014. http://arstechnica.com/security/2014/07/russia-publicly-joins-war-on-tor-privacy-with-111000-bounty/ [Accessed 05 July 2016].

[19] James Ball, Bruce Schneier, and Glenn Greenwald. NSA and GCHQ Target Tor Network That Protects Anonymity of Web Users. Website, 2013. http://www.theguardian.com/world/2013/oct/04/nsa-gchq-attack-tor-network-encryption [Accessed 05 July 2016].

[20] Pierluigi Paganini. Hacking Tor and Online Anonymity. Website, 2014. http://resources.infosecinstitute.com/hacking-tor-online-anonymity/ [Accessed 05 July 2016].

[21] Kevin Poulsen. FBI Admits It Controlled Tor Servers Behind Mass Malware Attack. Website, 2013. http://www.wired.com/2013/09/freedom-hosting-fbi/ [Accessed 05 July 2016].

[22] Nikita Borisov, George Danezis, Prateek Mittal, and Parisa Tabriz. Denial of service or denial of security? How attacks on reliability can compromise anonymity. In *Proceedings of CCS 2007*, October 2007.

[23] Kevin Bauer, Joshua Juen, Nikita Borisov, Dirk Grunwald, Douglas Sicker, and Damon McCoy. On the optimal path length for Tor. *HotPets in conjunction with Tenth International Symposium on Privacy Enhancing Technologies (PETS 2010), Berlin, Germany*, 2010.

[24] Tao Wang, Kevin Bauer, Clara Forero, and Ian Goldberg. Congestion-aware Path Selection for Tor. In *Proceedings of Financial Cryptography and Data Security (FC'12)*, February 2012.

[25] BT Wi-fi. Find a hotspot. Website, 2016. http://www.btwifi.co.uk/find/ [Accessed 05 July 2016].

[26] Stephen Doswell and Sean Callard. Private correspondence with Sean Callard (Technical Manager - BT Wi-Fi), 2012.

[27] Ashutosh Dutta. RFC 6252 - A Framework of Media-Independent Pre-Authentication (MPA) for Inter-Domain Handover Optimization. 2011.

[28] Christer Andersson and Andriy Panchenko. Practical anonymous communication on the mobile internet using Tor. *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, pages 39–48, 2007.

[29] Rungrat Wiangsripanawan, Willy Susilo, and Rei Safavi-Naini. Achieving Mobility and Anonymity in IP-Based Networks. In Feng Bao, San Ling, Tatsuaki Okamoto, Huaxiong Wang, and Chaoping Xing, editors, *Cryptology and Network Security*, volume 4856 of *Lecture Notes in Computer Science*, pages 60–79. Springer Berlin Heidelberg, 2007.

[30] Google. Google Play - Orbot: Proxy with Tor. Website, 2016. https://play.google.com/store/apps/details?id=org.torproject.android [Accessed 05 July 2016].

[31] The Guardian Project. The Guardian Project: People, Apps and Code You Can Trust. Website, 2016. https://guardianproject.info/ [Accessed 05 July 2016].

[32] The Guardian Project. Orfox: Aspiring to bring Tor Browser to Android. Website, 2015. https://guardianproject.info/2015/06/30/orfox-aspiring-to-bring-tor-browser-to-android/ [Accessed 05 July 2016].

[33] The Tor Project. Tor Metrics. Website, 2016. https://metrics.torproject.org/ [Accessed 05 July 2016].

[34] Facebook, Inc. Making Connections to Facebook more Secure. Website, 2014. https://www.facebook.com/notes/protect-the-graph/making-connections-to-facebook-more-secure/1526085754298237/ [Accessed 05 July 2016].

[35] Facebook, Inc. Facebook Reports First Quarter 2015 Results. Website, 2015. https://investor.fb.com/investor-news/press-release-details/2015/Facebook-to-Announce-First-Quarter-2015-Results/default.aspx [Accessed 5 July 2016].

[36] The Guardian. Facebook adds Android app support for anonymity service Tor. Website, 2016. http://www.theguardian.com/technology/2016/jan/19/facebook-android-app-anonymity-service-tor [Accessed 05 July 2016].

[37] Andriy Panchenko, Fabian Lanze, and Thomas Engel. Improving performance and anonymity in the Tor network. In *Proceedings of the 31st IEEE International Performance Computing and Communications Conference (IPCCC 2012)*, December 2012.

[38] Eric Rescorla. RFC 2818 - HTTP Over TLS. 2000.

[39] Marc Shapiro. Structure and Encapsulation in Distributed Systems: The Proxy Principle. In *ICDCS*, pages 198–204. IEEE Computer Society, 1986.

[40] Hal Roberts, Ethan Zuckerman, Jillian York, Robert Faris, and John Palfrey. 2010 circumvention tool usage report. *Harvard University - The Berkman Center for Internet & Society*, 2010.

[41] Anonymouse. Anonymization since 1997. Website, 2016. http://anonymouse.org [Accessed 05 July 2016].

[42] Proxy List. Proxy list - only working proxies. Website, 2016. https://proxy-list.org/english/index.php [Accessed 05 July 2016].

[43] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones. SOCKS Protocol Version 5. RFC 1928 (Proposed Standard), March 1996.

[44] Open VPN. Open VPN - your private path to access network resources and services securely. Website, 2016. https://openvpn.net/ [Accessed 05 July 2016].

[45] Matthew Edman and Bülent Yener. On anonymity in an electronic society: A survey of anonymous communication systems. *ACM Computing Surveys*, 42(1):1–35, 2009.

[46] Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In Ira S. Moskowitz, editor, *Proceedings of Information Hiding Workshop (IH 2001)*, pages 245–257. Springer-Verlag, LNCS 2137, April 2001.

[47] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), February 1981.

[48] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pages 2–15, May 2003.

[49] Ceki Gülcü and Gene Tsudik. Mixing E-mail with Babel. In *Proceedings of the Network and Distributed Security Symposium - NDSS '96*, pages 2–16. IEEE, February 1996.

[50] Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster Protocol — Version 2. IETF Internet Draft, July 2003.

[51] Marc Rennhard and Bernhard Plattner. Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002)*, November 2002.

[52] Marc Rennhard and Bernhard Plattner. Practical anonymity for the masses with morphmix. In Ari Juels, editor, *Proceedings of Financial Cryptography (FC '04)*, pages 233–250. Springer-Verlag, LNCS 3110, February 2004.

[53] Parisa Tabriz and Nikita Borisov. Breaking the collusion detection mechanism of morphmix. In George Danezis and Philippe Golle, editors, *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006)*, pages 368–384. Springer, June 2006.

[54] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding Routing Information. In R. Anderson, editor, *Proceedings of Information Hiding: First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174, May 1996.

[55] Electronic Frontier Foundation. Defending your Rights in the Digital World. Website, 2016. https://www.eff.org/ [Accessed 05 July 2016].

[56] The Tor Project. Core Tor People. Website, 2016. https://www.torproject.org/about/corepeople.html.en [Accessed 05 July 2016].

[57] The Tor Project. Tor: Sponsors. Website, 2016. https://www.torproject.org/about/sponsors.html.en [Accessed 05 July 2016].

[58] The Tor Project. Tor: Anonymity online. Website, 2016. https://www.torproject.org/ [Accessed 05 July 2016].

[59] The Tor Project. Tor Browser - What is the Tor Browser? Website, 2016. https://www.torproject.org/projects/torbrowser.html.en [Accessed 05 July 2016].

[60] The Tor Project. Tor Network Status. Website, 2016. https://torstatus.blutmagie.de/index.php [Accessed 05 July 2016].

[61] R. Snader and N. Borisov. Improving security and performance in the Tor network through tunable path selection. *Dependable and Secure Computing, IEEE Transactions on*, 8(5):728–741, Sept 2011.

[62] The Tor Project. Tor's protocol specifications. Website, 2016. https://gitweb.torproject.org/torspec.git/ [Accessed 05 July 2016].

[63] Lasse Øverlier and Paul Syverson. Improving efficiency and simplicity of Tor circuit establishment and hidden services. In Nikita Borisov and Philippe Golle, editors, *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*. Springer, June 2007.

[64] The Tor Project. Tor: Hidden Service Protocol. Website, 2016. https://www.torproject.org/docs/hidden-services.html [Accessed 05 July 2016].

[65] Alex Biryukov, Ivan Pustogarov, and Ralf-Philipp Weinmann. Trawling for Tor hidden services: Detection, measurement, deanonymization. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, May 2013.

[66] Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an Analysis of Onion Routing Security. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 96–114. Springer-Verlag, LNCS 2009, July 2000.

[67] The Tor Project. attacks | Possible upcoming attempts to disable the Tor network. Website, 2014. https://blog.torproject.org/category/tags/attacks [Accessed 05 July 2016].

[68] Anupam Das, Nikita Borisov, Prateek Mittal, and Matthew Caesar. *Re*^{3}: Relay Reliability Reputation for Anonymity Systems. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2014)*, June 2014.

[69] Codenomicon. The Heartbleed Bug. Website, 2016. http://heartbleed.com/ [Accessed 05 July 2016].

[70] Common Vulnerabilities and Exposures (CVE). CVE - celebrating 15 years: CVE-2014-0160. Website, 2016. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160 [Accessed 05 July 2016].

[71] The Guardian. Tor may be forced to cut back capacity after Heartbleed bug. Website, 2014. http://www.theguardian.com/technology/2014/apr/17/tor-heartbleed-bug-vulnerable-servers [Accessed 05 July 2016].

[72] Ball, James. Cameron wants to ban encryption he can say goodbye to digital Britain. Website, 2015. http://www.theguardian.com/commentisfree/2015/jan/13/cameron-ban-encryption-digital-britain-online-shopping-banking-messaging-terror [Accessed 05 July 2016].

[73] The Guardian. "Tor Stinks" Presentation. Website, 2013. http://www.theguardian.com/world/interactive/2013/oct/04/tor-stinks-nsa-presentation-document [Accessed 05 July 2016].

[74] Roger Dingledine. Trip Report: Tor Trainings for the Dutch and Belgian Police. Website, 2013. https://blog.torproject.org/blog/trip-report-tor-trainings-dutch-and-belgian-police [Accessed 05 July 2016].

[75] Joe Mullinn. Silk Road 2.0, infiltrated from the start, sold $8M per month in drugs. Website, 2014. http://arstechnica.com/tech-policy/2014/11/silk-road-2-0-infiltrated-from-the-start-sold-8m-per-month-in-drugs/ [Accessed 05 July 2016].

[76] D. Kelly, R. Raines, R. Baldwin, M. Grimaila, and B. Mullins. Exploring extant and emerging issues in anonymous networks: A taxonomy and survey of protocols and metrics. *Communications Surveys Tutorials, IEEE*, 14(2):579–606, Second 2012.

[77] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Roger Dingledine and Paul Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.

[78] Andreas Pfitzmann and Marit Hansen. Anonymity, unobservability, and pseudonymity: A consolidated proposal for terminology. Draft, July 2000.

[79] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[80] Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Low-resource routing attacks against Tor. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2007)*, October 2007.

[81] Steven J. Murdoch and Piotr Zieliński. Sampled traffic analysis by Internet-exchange-level adversaries. In Nikita Borisov and Philippe Golle, editors, *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*. Springer, June 2007.

[82] Juha Salo. Recent Attacks On Tor. Website, 2010. www.cse.hut.fi/en/publications/B/11/papers/salo.pdf [Accessed 05 July 2016].

[83] Joan Feigenbaum, Aaron Johnson, and Paul Syverson. A probabilistic analysis of onion routing in a black-box model. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2007)*, October 2007.

[84] John Douceur. The Sybil Attack. In *Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002)*, March 2002.

[85] Nathan Evans, Roger Dingledine, and Christian Grothoff. A practical congestion attack on Tor using long paths. In *Proceedings of the 18th USENIX Security Symposium*, August 2009.

[86] Nick Mathewson and Roger Dingledine. Practical traffic analysis: Extending and resisting statistical disclosure. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, volume 3424 of *LNCS*, pages 17–34, May 2004.

[87] Andrei Serjantov and Peter Sewell. Passive attack analysis for connection-based anonymity systems. In *Proceedings of ESORICS 2003*, October 2003.

[88] Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew K. Wright. Timing attacks in low-latency mix-based systems. In Ari Juels, editor, *Proceedings of Financial Cryptography (FC '04)*, pages 251–265. Springer-Verlag, LNCS 3110, February 2004.

[89] George Danezis and Andrei Serjantov. Statistical disclosure or intersection attacks on anonymity systems. In *Proceedings of 6th Information Hiding Workshop (IH 2004)*, LNCS, May 2004.

[90] George Danezis. Statistical disclosure attacks: Traffic confirmation in open environments. In Gritzalis, Vimercati, Samarati, and Katsikas, editors, *Proceedings of Security and Privacy in the Age of Uncertainty, (SEC2003)*, pages 421–426. IFIP TC11, Kluwer, May 2003.

[91] Roger Dingledine. Tor security advisory: relay early traffic confirmation attack. Website, 2014. https://blog.torproject.org/blog/tor-security-advisory-relay-early-traffic-confirmation-attack [Accessed 05 July 2016].

[92] Sambuddho Chakravarty. *Traffic Analysis Attacks and Defenses in Low Latency Anonymous Communication*. PhD thesis, University of Columbia, 2014.

[93] The Tor Project. Tor Trac: Custom Query. Website, 2016. https://trac.torproject.org/projects/tor/query [Accessed 05 July 2016].

[94] Shaun Nichols. Tor Project claims 'fake' Tor Browser sat in iOS App Store for months. Website, 2014. http://tinyurl.com/lsr43bu [Accessed 05 July 2016].

[95] The Tor Project. 10549: torbrowser in the apple app store is fake. Website, 2013. https://trac.torproject.org/projects/tor/ticket/10549 [Accessed 05 July 2016].

[96] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. Users get routed: Traffic correlation on Tor by realistic adversaries. In *Proceedings of the 20th ACM conference on Computer and Communications Security (CCS 2013)*, November 2013.

[97] Reporters Without Borders. Government steps up control of news and information. Website, 2012. http://en.rsf.org/ethiopia-government-steps-up-control-of-07-06-2012,42735.html [Accessed 05 July 2016].

[98] The Tor Project. The Design and Implementation of the Tor Browser [DRAFT]. Website, 2015. https://www.torproject.org/projects/torbrowser/design/ [Accessed 05 July 2016].

[99] Tim Wilde. Great Firewall Tor Probing. Website, 2012. https://gist.github.com/twilde/da3c7a9af01d74cd7de7.

[100] Linus Larsson and Daniel Goldberg. 100 Embassies and governments hacked in global security breach. Website, 2007. http://computersweden.idg.se/2.2683/1.118684 [Accessed 05 July 2016].

[101] Stevens Le Blond, Pere Manils, Abdelberi Chaabane, Mohamed Ali Kaafar, Claude Castelluccia, Arnaud Legout, and Walid Dabbous. One bad apple spoils the bunch: Exploiting p2p applications to trace and profile Tor users. In *Proceedings of the 4th USENIX conference on Large-scale exploits and emergent threats*, LEET'11. USENIX Association, 2011.

[102] Electronic Frontier Foundation. HTTPS Everywhere. Website, 2016. https://www.eff.org/https-everywhere [Accessed 05 July 2016].

[103] N. Al Barghouthy, A. Marrington, and I. Baggili. The forensic investigation of Android private browsing sessions using Orweb. In *Computer Science and Information Technology (CSIT), 2013 5th International Conference on*, pages 33–37, March 2013.

[104] Runa A. Sandvik. Forensic analysis of the Tor Browser Bundle on OS X, Linux, and Windows. Technical Report 2013-06-001, The Tor Project, June 2013. https://research.torproject.org/techreports/tbb-forensic-analysis-2013-06-28.pdf [Accessed 05 July 2016].

[105] The Tails Project. Tails: Privacy for anyone anywhere. Website, 2016. https://tails.boum.org/index.en.html [Accessed 05 July 2016].

[106] George Kadianakis. Packet size pluggable transport and traffic morphing. Technical Report 2012-03-004, The Tor Project, March 2012. https://research.torproject.org/techreports/morpher-2012-03-13.pdf [Accessed 05 July 2016].

[107] Steven J. Murdoch and George Kadianakis. Pluggable transports roadmap. Technical Report 2012-03-003, The Tor Project, March 2012. https://research.torproject.org/techreports/pluggable-roadmap-2012-03-17.pdf [Accessed 05 July 2016].

[108] The Tor Project. Tor: Pluggable transports. Website, 2016. https://www.torproject.org/docs/pluggable-transports.html [Accessed 05 July 2016].

[109] Hooman Mohajeri Moghaddam, Baiyu Li, Mohammad Derakhshani, and Ian Goldberg. Skypemorph: Protocol obfuscation for Tor bridges. In *Proceedings of the 19th ACM conference on Computer and Communications Security (CCS 2012)*, October 2012.

[110] Amir Houmansadr, Chad Brubaker, and Vitaly Shmatikov. The parrot is dead: Observing unobservable network communications. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, May 2013.

[111] Roger Dingledine and Steven J. Murdoch. Performance improvements on Tor or, why Tor is slow and what we're going to do about it. Technical Report 2009-11-001, The Tor Project, November 2009. https://research.torproject.org/techreports/performance-2009-11-09.pdf [Accessed 05 July 2016].

[112] Dietrich Braess, Anna Nagurney, and Tina Wakolbinger. On a Paradox of Traffic Planning (Translation of original paper in 1969). *Transportation Science*, 39(4):446–450, 2005.

[113] Radhika Mittal, Justine Sherry, Sylvia Ratnasamy, and Scott Shenker. How to improve your network performance by asking your provider for worse service. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, page 25. ACM, 2013.

[114] Ross Anderson. The Eternity Service. In *Proceedings of Pragocrypt '96*, 1996.

[115] Roger Dingledine and Nick Mathewson. Anonymity loves company: Usability and the network effect. In Ross Anderson, editor, *Proceedings of the Fifth Workshop on the Economics of Information Security (WEIS 2006)*, June 2006.

[116] Mashael AlSabah, Kevin Bauer, Ian Goldberg, Dirk Grunwald, Damon McCoy, Stefan Savage, and Geoffrey Voelker. Defenestrator: Throwing out windows in Tor. In *Proceedings of the 11th Privacy Enhancing Technologies Symposium (PETS 2011)*, July 2011.

[117] Karsten Loesing, Steven J. Murdoch, and Rob Jansen. Evaluation of a libutp-based Tor datagram implementation. Technical Report 2013-10-001, The Tor Project, October 2013. https://research.torproject.org/techreports/libutp-2013-10-30.pdf [Accessed 05 July 2016].

[118] W. Brad Moore, Chris Wacek, and Micah Sherr. Exploring the potential benefits of expanded rate limiting in Tor: Slow and steady wins the race with tortoise. In *Proceedings of 2011 Annual Computer Security Applications Conference (ACSAC'11), Orlando, FL, USA*, December 2011.

[119] Steven J. Murdoch. Comparison of Tor datagram designs. Technical Report 2011-11-001, The Tor Project, November 2011. https://research.torproject.org/techreports/datagram-comparison-2011-11-07.pdf [Accessed 05 July 2016].

[120] Steven J. Murdoch. Datagram testing plan. Technical Report 2012-03-002, The Tor Project, March 2012. https://research.torproject.org/techreports/datagram-testing-plan-2012-03-16.pdf [Accessed 05 July 2016].

[121] Can Tang and Ian Goldberg. An improved algorithm for Tor circuit scheduling. In Angelos D. Keromytis and Vitaly Shmatikov, editors, *Proceedings of the 2010 ACM Conference on Computer and Communications Security (CCS 2010)*. ACM, October 2010.

[122] Ye Tian, K. Xu, and N. Ansari. TCP in wireless environments: problems and solutions. *Communications Magazine, IEEE*, 43(3):S27–S32, March 2005.

[123] Jon McLachlan and Nicholas Hopper. Don’t clog the queue: Circuit clogging and mitigation in P2P anonymity schemes. In *Proceedings of Financial Cryptography (FC ’08)*, January 2008.

[124] Richard Clayton, George Danezis, and Markus G. Kuhn. Real world patterns of failure in anonymity systems. In Ira S. Moskowitz, editor, *Proceedings of Information Hiding Workshop (IH 2001)*, pages 230–244. Springer-Verlag, LNCS 2137, April 2001.

[125] Sebastian Hahn and Karsten Loesing. Privacy-preserving ways to estimate the number of Tor users. Technical Report 2010-11-001, The Tor Project, November 2010. https://research.torproject.org/techreports/countingusers-2010-11-30.pdf [Accessed 05 July 2016].

[126] Robin Snader and Nikita Borisov. A tune-up for Tor: Improving security and performance in the Tor network. In *Proceedings of the Network and Distributed Security Symposium - NDSS ’08*. Internet Society, February 2008.

[127] Mike Perry. TorFlow: Tor network analysis. Technical Report 2009-08-003, The Tor Project, August 2009. https://research.torproject.org/techreports/torflow-2009-08-07.pdf [Accessed 05 July 2016].

[128] Steven J. Murdoch and Robert N. M. Watson. Metrics for security and performance in low-latency anonymity networks. In Nikita Borisov and Ian Goldberg, editors, *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, pages 115–132. Springer, July 2008.

[129] Michael Backes and Aniket Kate. (nothing else) mator(s): Monitoring the anonymity of tors path selection, 2014.

[130] Joel Reardon and Ian Goldberg. Improving Tor using a TCP-over-DTLS Tunnel. In *Presented as part of the 18th USENIX Security Symposium (USENIX Security 09)*, Montreal, Canada, 2009. USENIX.

[131] Aaron Johnson, Paul Syverson, Roger Dingledine, and Nick Mathewson. Trust-based anonymous communication: Adversary models and routing algorithms. In *Proceedings of the 18th ACM conference on Computer and Communications Security (CCS 2011)*, October 2011.

[132] Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), June 1998.

[133] Angus (ed.) Stevenson. *Oxford Dictionary of English (3 ed.)*. Oxford University Press, 2010.

[134] Peter Eckersley. How unique is your web browser? In *Proceedings of the 10th Privacy Enhancing Technologies Symposium*, pages 1–18, July 2010.

[135] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.

[136] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, and Pierangela Samarati. k-anonymity. Chapter in Security in Decentralized Data Management, (T. Yu and S. Jajodia editors), Springer, 2007.

[137] Reza Shokri, Carmela Troncoso, Claudia Diaz, Julien Freudiger, and Jean-Pierre Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2010)*. ACM, October 2010.

[138] George Danezis. Measuring anonymity: a few thoughts and a differentially private bound. Website, 2016. http://www0.cs.ucl.ac.uk/staff/G.Danezis/papers/Danezis-MeasuringThoughts.pdf [Accessed 05 July 2016].

[139] Paul Syverson. Why I'm not an entropist. In Bruce Christianson, James A. Malcolm, Vashek Matyáš, and Michael Roe, editors, *Proceedings of Security Protocols XVII: 17th International Workshop, April 2009, Revised Selected Papers*, pages 231–239. Springer-Verlag, LNCS 7028, 2013.

[140] Marie Elisabeth Gaup Moe. Quantification of anonymity for mobile ad hoc networks. In *Proceedings of the 4th International Workshop on Security and Trust Management (STM 08)*, pages 25–36, June 2008.

[141] Rajiv Bagai, Huabo Lu, Rong Li, and Bin Tang. An accurate system-wide anonymity metric for probabilistic attacks. In *Proceedings of the 11th Privacy Enhancing Technologies Symposium (PETS 2011)*, July 2011.

[142] M. Edman, F. Sivrikaya, and B. Yener. A combinatorial approach to measuring anonymity. *Intelligence and Security Informatics, 2007 IEEE*, pages 356–363, May 2007.

[143] Gergely Tóth and Zoltán Hornák. Measuring anonymity in a non-adaptive, real-time system. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, volume 3424 of *Springer-Verlag, LNCS*, pages 226–241, 2004.

[144] Decentralized and Distributed Systems Research at Yale. Dissent accountable anonymous group communication. Website, 2011. http://dedis.cs.yale.edu/dissent/ [Accessed 05 July 2016].

[145] Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, 7(1), 2008.

[146] David Wolinsky, Ewa Syta, and Bryan Ford. Hang with your buddies to resist intersection attacks. In *Proceedings of the 20th ACM conference on Computer and Communications Security (CCS 2013)*, November 2013.

[147] walkit.com. The urban walking route planner. Website, 2016. http://walkit.com/ [Accessed 05 July 2016].

[148] Auri Aittokallio. Merger of O2 and 3 not an instant fix, say analysts. Website, 2015. http://telecoms.com/391612/merger-of-o2-and-3-not-an-instant-fix-say-analysts/ [Accessed 05 July 2016].

[149] Hutchison 3G. ThreeinTouch. Website, 2016. http://www.three.co.uk/discover/threeintouch [Accessed 05 July 2016].

[150] XIA Weiwei and SHEN Lianfeng. Modeling and analysis of hybrid cellular/wlan systems with integrated service-based vertical handoff schemes. *IEICE transactions on communications*, 92(6):2032–2043, 2009.

[151] The Guardian Project. Bug 1900: Orbot seems unable to cope with roaming Wi-Fi. Website, 2013. https://dev.guardianproject.info/issues/1900 [Accessed 05 July 2016].

[152] Rob Jansen, Kevin Bauer, Nicholas Hopper, and Roger Dingledine. Methodically Modeling the Tor Network. In *Proceedings of the USENIX Workshop on Cyber Security Experimentation and Test (CSET 2012)*, August 2012.

[153] Kevin Bauer, Micah Sherr, Damon McCoy, and Dirk Grunwald. Experimentor: A testbed for safe and realistic Tor experimentation. In *Proceedings of the USENIX Workshop on Cyber Security Experimentation and Test (CSET 2011)*, August 2011.

[154] Rob Jansen and Nicholas Hopper. Shadow: Running Tor in a Box for Accurate and Efficient Experimentation. In *Proceedings of the Network and Distributed System Security Symposium - NDSS'12*. Internet Society, February 2012.

[155] OpenSim Ltd. OMNeT++ Discrete Event Simulator. Website, 2015. http://www.omnetpp.org/ [Accessed 05 July 2016].

[156] INET Framework. INET Framework. Website, 2016. http://inet.omnetpp.org/ [Accessed 05 July 2016].

[157] Bushra Naeem and Able Nyamapfene. Seamless vertical handover in wifi and wimax networks using rss and motion detection: An investigation. *The Pacific Journal of Science and Technology*, 12(1):298–304, 2011.

[158] Mashael AlSabah, Kevin Bauer, Tariq Elahi, and Ian Goldberg. The path less travelled: Overcoming Tor's bottlenecks with traffic splitting. In *Proceedings of the 13th Privacy Enhancing Technologies Symposium (PETS 2013)*, July 2013.

[159] Rob Jansen, Paul Syverson, and Nicholas Hopper. Throttling Tor Bandwidth Parasites. In *Proceedings of the 21st USENIX Security Symposium*, August 2012.

[160] Christopher Wacek, Henry Tan, Kevin Bauer, and Micah Sherr. An Empirical Evaluation of Relay Selection in Tor. In *Proceedings of the Network and Distributed System Security Symposium - NDSS'13*. Internet Society, February 2013.

[161] Roger Dingledine. Tor development roadmap, 2008-2011. Technical report, The Tor Project, 2008. https://www.torproject.org/press/presskit/2008-12-19-roadmap-full.pdf [Accessed 05 July 2016].

[162] Mathworks. Matlab: The language of technical computing. Website, 2016. http://uk.mathworks.com/products/matlab/index.html [Accessed 05 July 2016].

[163] Karsten Loesing. Reducing the Circuit Window Size in Tor. Technical Report 2009-09-002, The Tor Project, September 2009. https://research.torproject.org/techreports/circwindow-2009-09-20.pdf [Accessed 05 July 2016].

[164] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Shining light in dark places: Understanding the Tor network. In Nikita Borisov and Ian Goldberg, editors, *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, pages 63–76. Springer, July 2008.

[165] Nicholas Hopper. Challenges in protecting Tor hidden services from botnet abuse. In *Proceedings of Financial Cryptography and Data Security (FC'14)*, March 2014.

[166] Man Ho Au, Patrick P. Tsang, and Apu Kapadia. PEREA: Practical TTP-free revocation of repeatedly misbehaving anonymous users. *ACM Transactions on Information and System Security (ACM TISSEC)*, 14:29:1–29:34, December 2011.

[167] Patrick P. Tsang, Apu Kapadia, Cory Cornelius, and Sean W. Smith. Nymble: Blocking misbehaving users in anonymizing networks. *IEEE Transactions on Dependable and Secure Computing*, 8(2):256–269, March–April 2011.

[168] Roger Dingledine. Adaptive throttling of Tor clients by entry guards. Technical Report 2010-09-001, The Tor Project, September 2010. https://research.torproject.org/techreports/adaptive-throttling-tor-clients-entry-guards-2010-09-19.pdf [Accessed 05 July 2016].

[169] E. L. Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958.

[170] Ling-Jyh Chen, Yung-Chin Chen, Tony Sun, P. Sreedevi, Kuan-Ta Chen, Chen-Hung Yu, and Hao-Hua Chu. Finding Self-Similarities in Opportunistic People Networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2286–2290, May 2007.

[171] Nicholas Hopper, Eugene Y. Vasserman, and Eric Chan-Tin. How much anonymity does network latency leak? In *Proceedings of CCS 2007*, October 2007.

[172] C.P. Bonini. *Simulation of information and decision systems in the firm.* Ford Foundation doctoral dissertation series. Prentice-Hall, 1963.

[173] Karl Pearson. The problem of the random walk. *Nature*, 72:342, 1905.

[174] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pages 153–181. Springer, 1996.

[175] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2:483–502, 2002.

[176] The Tor Project. Who Uses Tor? Website, 2016. https://www.torproject.org/about/torusers.html [Accessed 05 July 2016].

[177] O. Helgason, S.T. Kouyoumdjieva, and G. Karlsson. Opportunistic communication and human mobility. *Mobile Computing, IEEE Transactions on*, 13(7):1597–1610, July 2014.

[178] Roger Dingledine. Strategies for getting more bridges. Technical Report 2011-05-001, The Tor Project, May 2011. https://research.torproject.org/techreports/strategies-getting-more-bridge-addresses-2011-05-13.pdf [Accessed 05 July 2016].

[179] Tariq Elahi, Kevin Bauer, Mashael AlSabah, Roger Dingledine, and Ian Goldberg. Changing of the guards: A framework for understanding and improving entry guard selection in Tor. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2012)*. ACM, October 2012.

[180] Microsoft. Continuum - Microsoft Display Dock. Website, 2016. https://www.microsoft.com/en-gb/mobile/accessory/hd-500/ [Accessed 05 July 2016].

[181] Amir Houmansadr, Thomas Riedl, Nikita Borisov, and Andrew Singer. I Want my Voice to be Heard: IP over Voice-over-IP for Unobservable Censorship Circumvention. In *Proceedings of the Network and Distributed System Security Symposium - NDSS'13*. Internet Society, February 2013.

[182] Institute of Electrical and Electronics Engineers (IEEE). 802.21-2008 - IEEE Standard for Local and metropolitan area networks - Media Independent Handover Services. Website, 2008. https://standards.ieee.org/findstds/standard/802.21-2008.html [Accessed 05 July 2016].

[183] Jody Sankey and Matthew Wright. Dovetail: Stronger anonymity in next-generation internet routing. In *Proceedings of the 14th Privacy Enhancing Technologies Symposium (PETS 2014)*, July 2014.

[184] Henry Corrigan-Gibbs and Bryan Ford. Dissent: Accountable Anonymous Group Messaging. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 340–350, New York, NY, USA, 2010. ACM.

[185] Ewa Syta, Henry Corrigan-Gibbs, Shu-Chun Weng, David Wolinsky, Bryan Ford, and Aaron Johnson. Security Analysis of Accountable Anonymity in Dissent. *ACM Trans. Inf. Syst. Secur.*, 17(1):4:1–4:35, August 2014.

[186] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[187] Sharad Goel, Mark Robson, Milo Polte, and Emin Gun Sirer. Herbivore: A Scalable and Efficient Protocol for Anonymous Communication. Technical Report 2003-1890, Cornell University, Ithaca, NY, February 2003.

[188] David Isaac Wolinsky, Henry Corrigan-Gibbs, Bryan Ford, and Aaron Johnson. Dissent in Numbers: Making Strong Anonymity Scale. In *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, OSDI'12, pages 179–192, Berkeley, CA, USA, 2012. USENIX Association.

[189] The Guardian Project. Voice over Tor? Website, 2012. https://guardianproject.info/2012/12/10/voice-over-tor/ [Accessed 05 July 2016].

[190] YouTube. YouTube HTML5 Video Player. Website, 2016. https://www.youtube.com/html5?gl=GB [Accessed 05 July 2016].

[191] Prime Minister's Press Office. Queen's Speech 2015. Website, 2015. https://www.gov.uk/government/topical-events/queens-speech-2015 [Accessed 05 July 2016].

[192] University of Cambridge. PhD/MPhil Thesis - a LaTeX Template. Website, 2016. http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/ThesisStyle/ [Accessed 05 July 2016].

[193] The Freenet Project Inc. Freenet. Website, 2015. https://freenetproject.org/index.html [Accessed 05 July 2016].

[194] What does I2P do for you? The Invisible Internet Project. Website, 2016. https://geti2p.net/ [Accessed 05 July 2016].

Appendix A

Case study - Alice in Philistia

Alice and her friends are human rights activists. They all live in a country (Philistia) that its government oppresses free speech. The government monitor all Internet connections on the state run telecommunications company. Bob is an independent journalist and lives in a 'free' country (Utopia). He helps by allowing Alice and her friends exchange (upload / download) messages on a website (Bob's human rights' blog). Several news agencies monitor Bob's blog to report what is happening in Philistia. Unfortunately, the blog, being public, is also monitored by Charlie, a secret agent for the Philistine government. David has developed an anonymous communications' system, a low-latency anonymity network. The anonymity network is built upon the concept of onion routing. In contrast to Tor, which uses TCP as its transport, the anonymity network has adopted a lighter transport protocol (UDP), the same approach used by I2P. Norman maintains the anonymity network, which consists of relays supported by volunteers around the world. A number of security features have been implemented in the anonymity network. As with Tor, the anonymity network refreshes circuits every 10 minutes to try to reduce the opportunity for traffic analysis. The anonymity network is primarily used for updating and retrieving messages on compliant blogs like Bob's. The performance of the anonymity network is important to allow activists to respond quickly, for example, if on a protest march, while also maintaining anonymity. To maintain this critical balance, Norman applies the q-factor model to monitor the key components on the anonymity network. To help Norman maintain an efficient network, David's software employs a number of features. The software only allows one active circuit per user on the network. The anonymity network also restricts the size of blog entries to only allow a photograph and short description to a maximum file transfer size of 300 KB. These features deter active attacks such as congestion-based attacks by 'flooding'. To prevent passive attacks, such as traffic analysis, any files less than 300 KB are padded so no one file is any more distinguishable while being transferred. The circuit window is fixed at a default of 1000 *cells (each cell being 512 bytes), at a total of approximately 500 KB, although the circuit window size can be adjusted client-side, within the client configuration settings, if required. The anonymity network also supports connections from mobile devices for which Bob has developed an Android application. On average, Alice and half of her friends access Bob's website via the mobile application, the remainder from their desktops (in Internet cafés etc.), reflecting the parity between desktop and mobile Internet connections. Additionally, of those mobile users, approximately half are roaming at any time (usually including Alice), while the others remain static. The mobile application also has enhanced functionality. An existing security feature on Bob's blog is that a hash value is displayed for each file. A website user, once the file is downloaded, can manually compare the file's hash value against the one displayed on the website. If a match then it is assumed OK, if not, then the file may have been tampered with, potentially containing malware or a tracking 'tag', inserted by Charlie. For the mobile application this check is undertaken automatically. Downloads are discarded if either being tampered with, corrupted in-flight, or the transfer is incomplete (a partial download). Recently, Norman has observed that the number of connections from within Philistia has fallen yet the performance on anonymity network has got worse. On further investigation Norman identified that the Philistine government sometimes blocks access to the anonymity network at highly sensitive times, especially during demonstrations. He has also observed a large amount of congestion on the first in, first out (FIFO) message queue buffers at the entry points to the anonymity network. He concludes the congestion is from mobile users using the service while roaming, with the entry node, as designed, retries / waits 3 times to send the message to the anonymity network user. This process unfortunately takes 30 seconds in total causing other messages to be delayed. Norman called a meeting with David to discuss the issues. The key points arising from the discussion are as follows. David and Norman pondered the challenges in supporting the increasing mobility of users such as Alice and her friends. That is, roaming across Philistia's networks, in addition to the issue of the Philistine government sometimes blocking connections to the anonymity network. They all agree in the benefit of supporting a 'real-time' service for roaming mobile users and the added security features of the Android application. The current entry point used to access the anonymity network cannot support roaming efficiently and consequently there is a negative impact on client performance. Any data left in-flight and repeated transfer requests causes additional congestion on the anonymity network for ALL users. They agree, for maintaining anonymity, the existing architecture is likely to be the best solution. This is because it is more likely there is enough concurrent connections to maintain the required anonymity threshold. However, to circumvent the blocking of connections to the anonymity network, they also agree to implement a bridge relay solution similar to that found on Tor. Entry points to the* *anonymity network now come in three types: entry guard (public), bridge relay (public), and bridge relay (private). The existing entry guards (as the first hop) allow anyone to connect to the anonymity network and send a message to Bob's blog. The bridge relay function allows a user to either configure their connection to act as a public bridge relay or connect to another public bridge, or connect to their own private bridge. Connection details of public bridge relays are only available by request from Norman. A private bridge relay is run and only used by the activist themselves. Bob agrees to use the bridges to also support mobility and implements a 'roaming mode' within the mobile application. If the roaming mode is switched on, the bridge acts as a home agent and will redirect the bridge owner's traffic to maintain a persistent connection to the anonymity network. The 'penalty' of using bridge relays in roaming mode is an overhead of approximately 1 second to update the home agent with the new external Internet Protocol (IP) address of the mobile user. However, this is lower than the time to take to rebuild the circuits as well as reducing congestion on the anonymity network. David suggests whether client throttling should also be evaluated for mitigating the impact of mobility, as previously proposed for bulk downloads on the Tor network. If it can be predicted when Alice may hand-off and throttling can be employed beforehand, then this may also mitigate any negative impact for both Alice and the anonymity network? Bob also agrees to implement a configurable circuit window size setting within the mobile application and allow different algorithms to be tested. David and Norman ask Stephen to independently evaluate the use of client throttling and a bridge relay solution, for supporting mobility, compared to the existing design, from both an anonymity and performance perspective by applying the q-factor model.*

Appendix B

Measuring optimisation using q-factor

The level of optimisation equals the quality achieved at the 'cost of service'. This cost of service can be applied to q-factor as the level of utilization (Z). A 'perfect' anonymity solution with high performance may be not economically viable, especially if the system relies on volunteers. Therefore, the optimisation of an anonymity network could additionally be measured in different ways, depending of the requirements:

- $Z = b / c$ ('actual' performance / capacity)
- $Z = r / c$ (resource allocation / capacity)
- $Z = b / r$ ('actual' performance / resource allocation)

The standard optimisation cycle of network management should be applied to try to maintain a reliable service, as shown in Figure B.2.

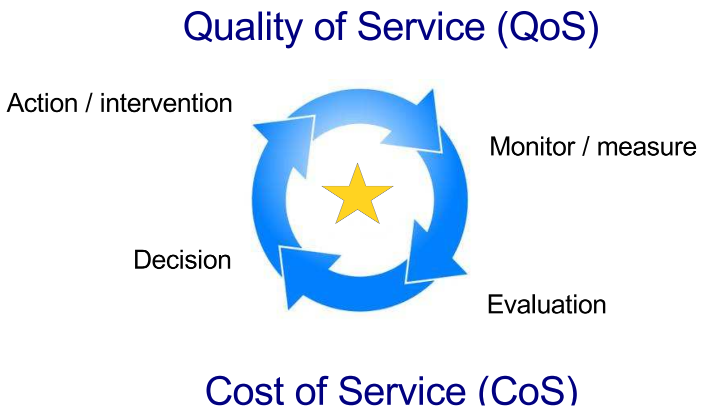


Fig. B.1 An optimisation cycle of anonymity network management

Some of these measurements may able to be taken at the design / testing stage by comparing to other proposed solutions, i.e., 'what happens to r if c is reduced for N users?' Once these measurements are taken, what could different levels of optimisation mean?

1. A 'high' quality of service, for example, a q-factor of ≈ 0.9, but a low utilization of $Z = 0.5$. This could mean the need to optimize further by providing a higher resource allocation, i.e., set the circuit window size higher.

2. A 'low' quality of service (a q-factor of ≈ 0.4), but with a high utilization of $Z = 0.9$. This could mean congestion is high and resource allocation should be lowered.

Appendix C

Dissent

i. The Dining cryptographers problem:

As shown in Figure C.1, in the dining cryptographers problem, if none of the cryptographers paid, then Alice (A) would announce $1 \oplus 0 = 1$, B would announce $1 \oplus 1 = 0$, and C would announce $0 \oplus 1 = 1$. Yet, if Alice (A) had paid, she would announce $\neg\ (1 \oplus 0) = 0$. Dissent aims to address the inherent vulnerability to traffic analysis in low-latency anonymity networks, such as Tor.

ii. Dissent design:

In their basic form, DC-nets are only practical when high latencies are tolerable, such as for anonymous re-mailers like Mixminion, which were popular before onion routing (and Tor) was successfully implemented [48]. One of the main issues with a DC-net is that it is difficult to scale up for practical use due to the $N \times N$ coin-sharing problem. For example, for 20 users to 'dine' (cryptographers Alice and her 19 friends), there are 20 different possibilities for 1st, 19 different possibilities for 2nd, 18 different possibilities for 3rd, etc. This generates a total of 2.4^{1018} possible combinations. If one of Alice's friends drops their connection during the coin-sharing process, then the process has to start again. Also, if Charlie (as a spy) manages to infiltrate the group, he is able to jam the coin-sharing with random bits. A summary of the limitations of DC-nets are as follows:

1. Collision: two cryptographers paid resulting in a group-wide XOR of 0.

2. Disruption (jamming): a malicious diner can send the random bits rather than the pairwise XOR output rendering the group-wide XOR incorrect.

3. Complexity: the DC-net protocol is 'unconditionally secure' but depends on the assumption that unconditionally secure channels also exist between the participants.

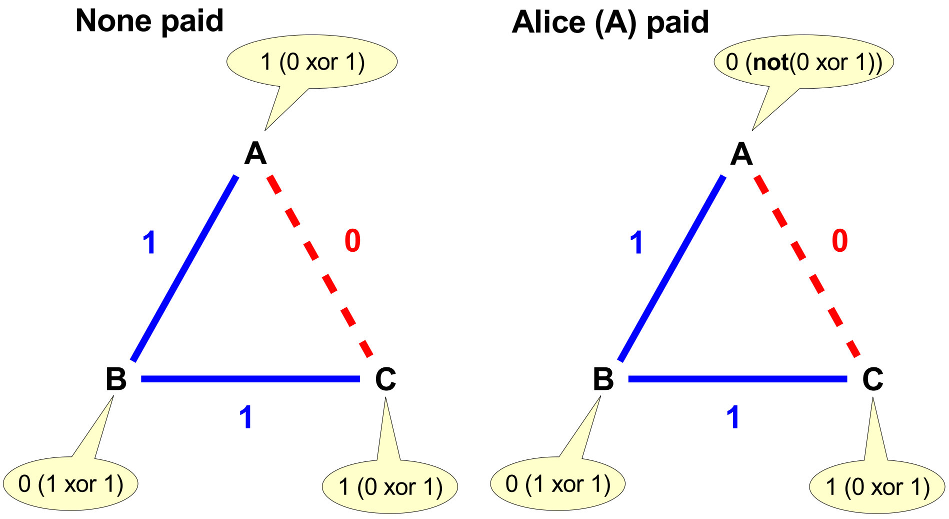


Fig. C.1 The dining cryptographers problem

Herbivore originally aimed to provide a solution for two of these limitations: 1. disruption (jamming) 2. complexity (and consequently scalability). For example, Herbivore addresses the $N \times N$ complexity problem via a star topology, in which a designated member of each group collects the other members' bits (or cipher-texts), XORs them together, and broadcasts the results to all members. However, without a solution to network churn, Herbivore and early versions of the Dissent protocol are limited in practice to small anonymity sets.

Dissent addressed the issue of scalability and network churn, as the project evolved, when migrating from a 'closed' network to a more 'populist' implementation, that is, on a larger scale with an Internet interface†. Dissent now adopts a client to multi-server trust model, as shown in Figure C.2. The rule is that no single server is trusted and Dissent can maintain anonymity unless all the servers maliciously collude with each other. Alice does need not know (or guess) which server to trust, but only trust that at least one trustworthy server exists. The Dissent servers have a similar role to relays on the Tor network, and like Tor relays, the Dissent servers are selected from a public directory of available servers to help balance network load.

†*This is probably due to the realization that the original version of Dissent did not provide anything different to existing implementations of closed file-sharing networks such as Freenet and I2P [193] [194].*

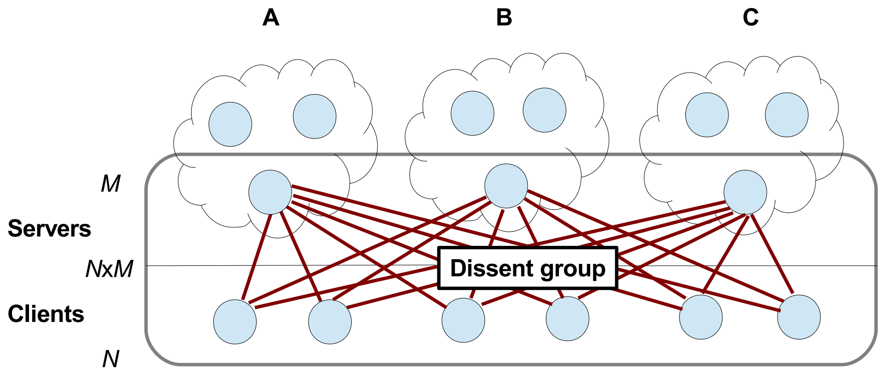


Fig. C.2 The Dissent architecture (client/server model)

iii. Dissent performance:

Wolinsky et al. analysis of Dissent's performance overhead on the Internet, specifically World Wide Web (WWW) performance in a range of collaborative scenarios, in Figures C.3 and C.4 [188]. Clearly, Dissent introduces a performance overhead, in addition to the existing Tor overhead, when browsing the WWW. However, in scenarios that require a high degree of anonymity, this degraded performance may be acceptable.

iv. Dissent and q-factor

The q-factor approach appears to fit with Dissent. This leads to whether q-factor can support future development of Dissent alongside mobility. One of the benefits of the q-factor approach is the anonymity metrics used by Dissent (possinymity an indinymity) are easily 'plugged-in' to the anonymity component, alongside the chosen performance metric. In theory, both possinymity an indinymity could be applied combinatorially as the anonymity metric, if required. This could be approached in a number of ways. The network operator may decide that one of the metrics is of higher priority than the other, and therefore only one is required to meet its threshold to return $a = 1$. Alternatively, the operator may decided that both possinymity an indinymity are required to meet the threshold. In addition to be able to facilitate a choice of underpinning anonymity metrics, q-factor also appears to support Dissent's own network management process. For example, how q-factor could support each of Dissent's actions in maximizing anonymity is as follows:

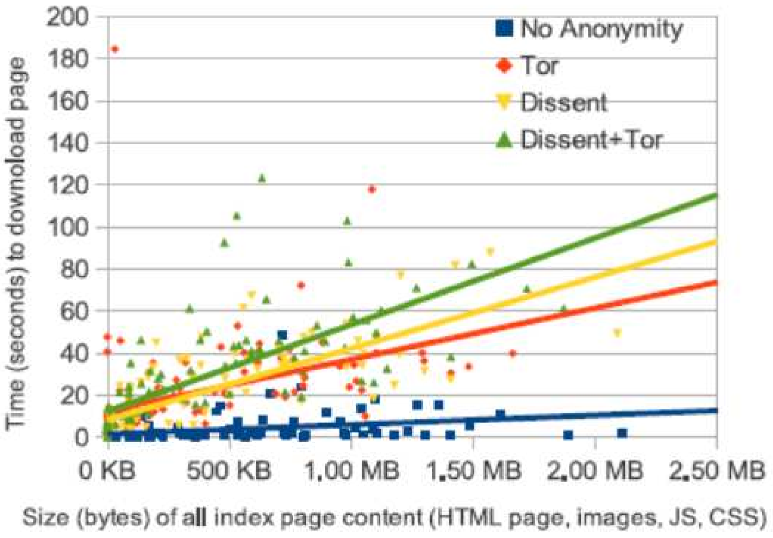


Fig. C.3 Download times (actual) for the top 100 Alexa home pages over a combination of Dissent and Tor [188]

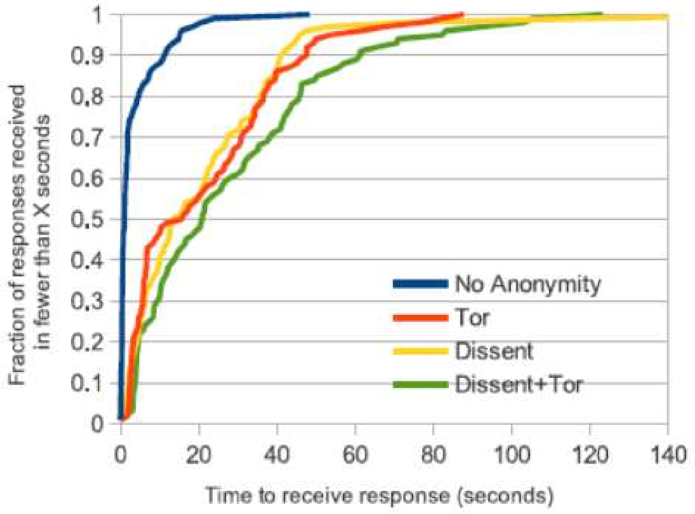


Fig. C.4 Download times (CDF) for the top 100 Alexa home pages over a combination of Dissent and Tor [188]

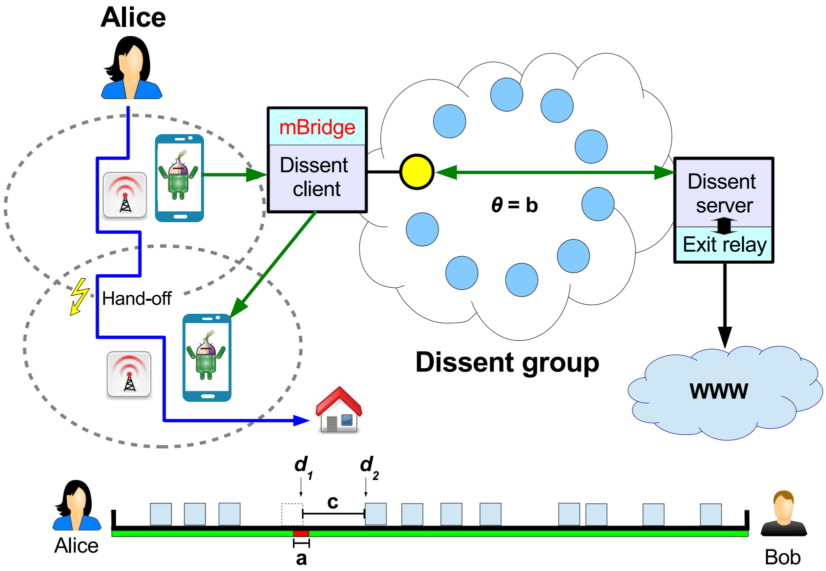


Fig. C.5 The physical network hand-off time is less than Dissent's tolerance level and therefore mBridge is not required to maintain persistence to Dissent (and Tor)

1. Maintaining possinymity: q-factor can continually monitor / measure possinymity to identify critical events.

2. Limiting possinymity loss rate: from the above monitoring, critical events are identified, for which an appropriate intervention can be undertaken.

3. 'Users worth waiting for': the double-layered version (v2) of q-factor can support the Dissent 'tolerance' period while a mobile user is reconnecting.

As discussed, there appears to be a wider benefit and potential use of q-factor. Not only for the case study, but also Dissent, Tor, other low-latency anonymous communications' systems, that operate within a dynamic network environment.

v. Dissent and mBridge

Adding a bridge relay, such as the mBridge solution, into the Dissent architecture may be of mutual benefit. Firstly, the mBridge solution would not only provides a persistence of connection to the Tor network but consequently also the Dissent group. This reduces the likelihood, and therefore removing the overhead, of having to rebuild the Dissent group. As the Dissent group effectively acts as the middle Tor relay, this has the benefit of mitigating network churn. Therefore, a reduction in the level of instability within the Dissent group

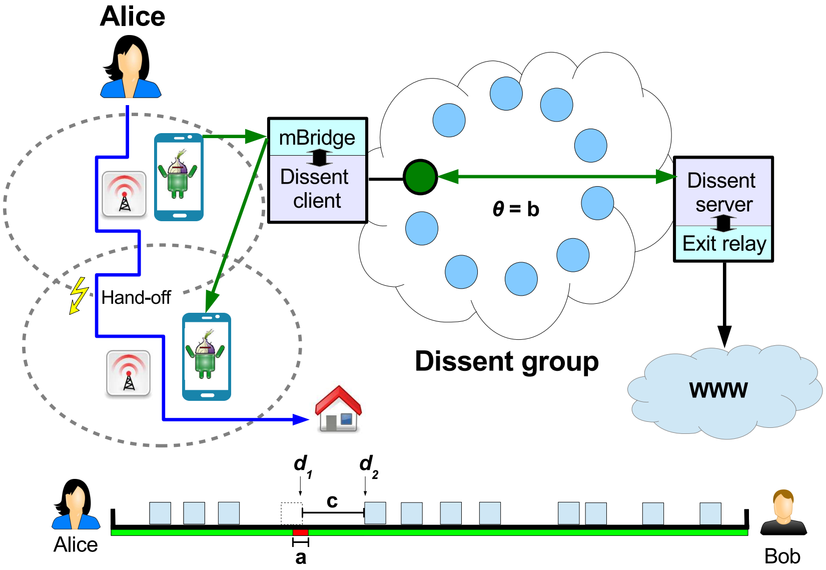


Fig. C.6 The physical network hand-off time exceeds Dissent's tolerance level and therefore mBridge is required to maintain persistence to Dissent (and Tor)

should improve overall performance, especially if Dissent begins to be impacted by mobile usage.

Dissent also enhances the mBridge solution so that Alice's connection becomes 'one step removed' from the Tor network, especially if Tor users are already targeted for surveillance. It could be argued this protection may be enough to also allow the possibility of using the private mBridge instead of a public one. As Dissent can be used for both non-Tor and Tor traffic and therefore this could certainly be beneficial in enhancing anonymity. By maintaining client performance, limiting possinymity loss rate, and providing robust protection against traffic analysis (indinymity), a hybrid mBridge / Dissent solution could enhance Dissent. This would not only mitigate Dissent's concerns over network churn (mobility) but also support a more feasible implementation for mBridge.

A final option is a more speculative system. Although, the Dissent project has already raised concerns of the potential negative effect of network churn (from mobility) on its protocol, the proposed refinements allow users to leave and rejoin the group, even if within a 'short' amount of time, may *unintentionally* support mobility. The Dissent project does not explicitly state how long this time should be. If the time threshold is, for example, two seconds, this will likely be able to negotiate a subscribed network hand-off, that is, if the mobile user's credentials are already stored. In this case, Alice's home Wi-Fi network, the walk to university using a Wi-Fi hotspot service and finally the Wi-Fi network in the laboratory, are all managed efficiently by Dissent. Therefore, the mBridge solution is considered unnecessary, as shown in Figure C.5. However, if the time to complete a hand-off (a) is less than the tolerance (b), then consequently membership of the dissent group is maintained, and downloads in progress will be delayed by c, and received at d_2 instead of d_1. Therefore, the impact of mobility on performance for this particular download can be accurately quantified by d_2 - d_1. If, however, the time allowed for users to rejoin is less than an average hand-off time (for pre-negotiated connections), say less than one second, then it is likely a bridge may still be required, as shown in Figure C.6.