Experimental Evaluation of the Impact of Mobility Management Protocols on HTTP Adaptive Streaming

Yusuf Sani[a,*], Musab Isah[a], Christopher Edwards[a], Andreas Mauthe[a]

[a]*School of Computing and Communications*
InfoLab2l, Lancaster University
Lancaster LA1 4WA, UK

Abstract

Video content is increasingly being consumed on the move using mobile devices such as smart phones and tablets. In order to deal with the challenges of heterogeneity of network access technologies and fluctuating resources, which are inherent features of mobile communication, HTTP adaptive streaming (HAS) is becoming the default technology for online video streaming. However, little research has been carried out to better understand the impact of handover schemes of the various mobility management protocols on the video quality of HAS. In this paper we present a comprehensive experimental measurement of the impact of handover on three representative HAS players. First, we implement three existing mobility management protocols, MIPv6, LISP-MN, and PMIPv6, on a network testbed. And using the fluid flow mobility model, the impact of frequent handover on the average video quality, the bandwidth utilisation and stability of the players was investigated. Our results show a degradation of all the observed parameters in all the reviewed players.

Keywords: HTTP Video Steaming, Mobility Management, Adaptive bit-rate Selection.

1. Introduction

Mobile devices have transformed not only the way video content is consumed but also how it is generated and shared. This is made possible by the combination of cheap smart phones and free video sharing sites popularised by YouTube. Cisco predicts that by 2019, global video consumption will account for 80%-90% of the entire data traffic traversing the Internet and over 61% of this traffic will come from wireless networks [1].

The burgeoning access to mobile devices and the accompanying increase in traffic have put to the fore the thought of wireless network cell miniaturisation, we now see increasing deployment of smaller cells. Perhaps, this is because other alternatives of improving wireless channel capacity are either near infeasible or at best not scalable. A candidate example is the increase of the spectrum of a mobile terminal radio, however, this is unfortunately constrained by the range of usable frequencies. Alternatively, spectral efficiency can be enhanced but it is well known fact that the current technologies are already approaching Shannon's limit [2].

When a mobile node roams across different cells, it changes its point of attachment (PoA). In fact, Gao et al. [3] have shown that 20% of mobile nodes have at least ten IP address changes per day. When this change in PoA takes place, packets need to be rerouted to the new anchor cell, and for this to happen smoothly, the old and the new cells need to efficiently coordinate the handover. However, this is seldom the case, usually transfer of control causes delay and packet loss. Various attempts have been made to design a mobility management technique, at almost all OSI layers, that guarantees the seamless movement across distinct IP wireless networks. The most attention is focused on layer three [4]. Some of the standardised solutions are: Mobile IPv6 (MIPv6) [5], Locator-Identifier Separation Protocol Mobile Node (LISP-MN) [6], and Proxy MIPv6 (PMIPv6) [7].

Concurrent to this effort, is the ongoing research on how to provide a one-system-fits-all video streaming service. HTTP Adaptive Streaming (HAS) [8] is the most recent attempt in this direction. HAS divides a video file into a number of chunks. Each chunk is encoded in multiple video rates, a client continuously monitors and estimates its capabilities. It then requests a chunk with the highest video rate that the estimated capacity can sustain. Generally, there are two distinct approaches for HAS service, i.e. throughput based and buffer based. A combination of both approaches is also being used.

A typical HAS implementation directly or indirectly equates the available bandwidth with the average TCP throughput when making rate selection decisions. However, it is a well-known fact that TCP is sensitive to packet loss, as little as 0.1% in packet loss can cause the TCP throughput to oscillate [9]. Gurius et al. [10] reported that 0.5% loss can result in an up to 25% reduction in throughput. For a mobile user, a handover can cause an amount of packet loss, which depends on the mobility protocol used. When handover is frequent, even if the loss is within what normally could have been an acceptable boundary, it be-

[*] Corresponding author
Email addresses: `y.sani@lancaster.ac.uk` (Yusuf Sani), `m.isah@lancaster.ac.uk` (Musab Isah), `c.edwards@lancaster.ac.uk` (Christopher Edwards), `a.mauthe@lancaster.ac.uk` (Andreas Mauthe)

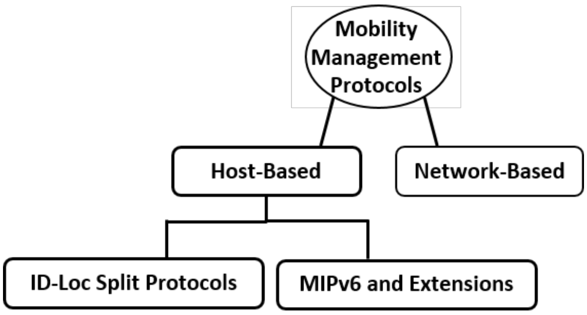


Figure 1: The taxonomy of mobility protocols.

comes a severe issue to any application that relies on accurate throughput information for decision making.

In this paper, we experimentally evaluate the performance of some carefully selected HAS players, with each player representing a particular type of HAS streaming service in both stationary and mobility contexts. Three representative mobility management protocols are implemented on a laboratory testbed for this purpose. Experimental measurements are used to answer an important question. *How much impact do the different mobility management protocols have on each player with respect to average video quality, bandwidth utilisation and stability?*

2. Background and Related Work

2.1. Mobility Management Protocols

To understand the behaviour of HAS in a mobile environment, it is necessary to consider relevant mobility management techniques representing the different approaches. Mobility Management Protocols (MMP) are designed to enable a host change its PoA, i.e. handover, from one access link to another with the goal of maintaining all ongoing data sessions with a remote device. Figure 1 presents the MMP taxonomy used in selecting the candidate protocols for the evaluation. As can be seen the MMPs are broadly classified into host-based and a network-based MMP. In the host-based techniques, all mobility signalling is handled by the mobile device, while for networked-based MMP mobility signalling is handled by network component(s), usually the access router (AR). Furthermore, the host-based mobility protocol is then divided into two groups: a MIPv6 and its variants and some Locator-Identifier (Loc-ID) Split Protocols family.

For the purpose of this work, we take a protocol from each of the (sub)categories. For the networked-based MMP the choice is straightforward since only PMIPv6 is standardised by IETF. MIPv6 is chosen because it is the baseline protocol upon which many protocols are built and adopted as the host-based mobility protocols by organisations such as 3GPP for its Evolved Packet Core [11]. LISP-MN is currently the most active implementation in the Loc-ID Split category and, to the best of our knowledge, the only protocol with a global infrastructure available to support its use.

2.1.1. Mobile IPv6

For the MIPv6[5], an MN is equipped with a permanent home address (HoA) for communicating with any correspondent node (CN) while at the home network. MN continues to use the same HoA address when it moves to a foreign network. However, the MN now configures an additional address called care-of address (CoA), which is used for routing purposes. A home agent (HA) at the home network maintains the relationship between the MNs home and care-of addresses. Whenever an MN changes its PoA it sends binding update (BU) message. On receipt of BU, the HA sends a binding acknowledgement (BA) message to the MN. All packets to and from the MN, while in foreign network, are normally sent via a bi-directional tunnel to the HA once the anchor acknowledges receipt of BU.

2.1.2. Locator Identifier Separation Mobile Node

LISP-MN [6] enables mobility through the use of a statically-configured endpoint identifier (EID) and a mutable routing locator (RLOC) both of which are IP addresses. The EID is used at the transport and upper layers of the protocol stack to identify sessions while RLOC is used for routing, and is configured every time an MN moves. A map server serves as the mobility anchor of the MN and tracks the MNs location via its registration messages sent on new RLOC configuration. Communication with non-LISP domains is achieved via a proxy ingress/egress tunnel router (or simply PxTR), which encapsulates or decapsulates packets to or from the MN to the Internet.

2.1.3. Proxy Mobile IPv6

For the PMIPv6 [7] the mobile access gateways (MAG) handles all the mobility related signalling without the participation of an MN. An MN gets the same IPv6 prefix advertised every time it moves and as such the MN maintains its current IP address in any network. The MAG sends proxy BU to the local mobility anchor (LMA) on the MNs home network once it establishes the MNs identity. The LMA replies with a proxy BA and bi-directional tunnel is created between the two network components for routing packets to and from the MN. The MAG will also send detachment messages to the LMA when the MN moves.

2.2. HTTP Adaptive Streaming

HTTP adaptive streaming (HAS) is an approach that ensures that the quality of video each user receives is commensurate to its context. The adaptation logic can be located at the server-side, the client-side, or somewhere in-between. HAS services usually divide a video file into a number of chunks of equal size with each chunk encoded in multiple bit-rates. A client progressively requests a relevant chunk; the quality of a request is based on the client's measurement of the available resources. First generation adaptive bit-rate streaming systems select a chunk with the highest video rate lower than the measured throughput [12, 13]. When the throughput changes, the buffer level is used as an indicator of either to increase, decrease or stay with the current quality level [14]. The HAS services that solely rely on throughput estimation for adaptation decision are called *throughput-based*, while those that principally

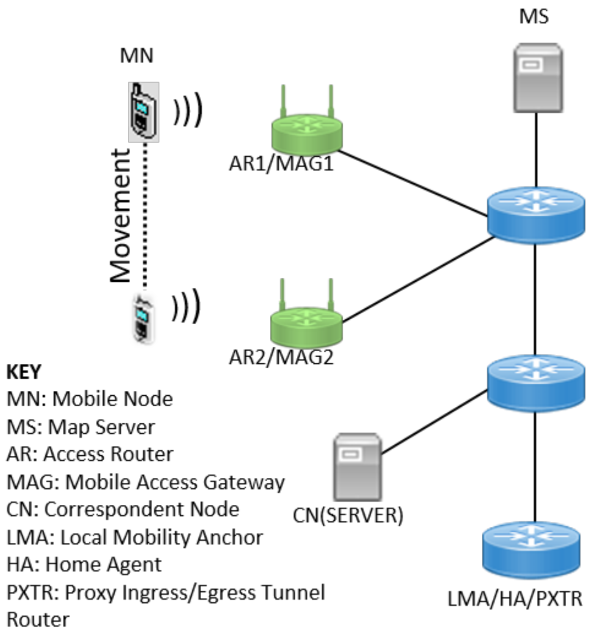


Figure 2: Testbed Topology.

rely on buffer occupancy are called *buffer-based* [15, 16]. Furthermore, in this paper, we call the HAS services that combine both throughput and buffer occupancy in video rate selection, in some form, *mixed-mode* HAS service. Since it is not possible to test every player, we elect to choose a representative player from each category. For the throughput-based player we pick the open source version of Microsoft Smooth Streamer (MSS) [17], while for the buffer-based, we chose the first purely buffer-based player proposed by Huang et al. [15]. Finally, we choose the player proposed by Miller et al. [18] as representative of mixed-mode HAS.

There are some previous works that aimed to evaluate HAS on mobile networks. Dung et. al [19] carried out testbed experiments to evaluate the effect of segment duration to video streaming performance in mobile networks by looking at video bitrate, buffer level and end-to-end delay recorded by a client running instant and smoothed throughput-based players. They found the choice of segment duration to have profound impact on performance with longer segments resulting into higher video quality on a client whilst shorter segments prevents against underflow. Longer segments were also shown to improve network utilisation. The work of Liang et. al [20] investigated the properties of energy-efficiency and spectrum-efficiency for video streaming over mobile ad hoc networks with the aim of providing design guidelines on building an energy and spectrum efficient mobile video transmission system. Using simulation, the authors evaluated their newly developed energy-spectrum-aware scheduling scheme to show improvement in average video quality experienced by the user among other improved metrics. In comparison to our evaluation, the two works [19, 20] are limited in the number of mobility protocols and video players deployed for evaluation.

3. Methodology and Evaluation Metrics

3.1. Mobility Scenario

First we assume that the network is partitioned into a number of hexagonal-shaped cells of equal area A_i, with each cell being an IPv6 network. Within each cell there is one AR, also called MAG in PMIPv6. A pedestrian user streaming video with his MN is assumed to continuously move across the cells. MN connects to the AR at both radio and IP level using Wi-Fi while visiting a cell. A static CN acts as the server and is always connected to the network via a wired connection. Figure 2 presents the testbed set-up as used throughout the experiments. To simplify the experimentation, we assumed that MN handovers to the AR whenever it crosses a cell boundary. To model the MN's mobility pattern, we use the fluid-flow mobility model [21]. Using this model, the direction of MN's movement is uniformly distributed over $[0, 2\pi]$. The user is assumed to move with velocity $v = 5Km/h$, which is the average human speed [22]. This mobility setting corresponds to a scenario when a mobile user walks across distinct open hotspots while streaming a video.

3.2. Experimental Setup

As can be seen from Figure 2, the client (MN) connects to the network via AR1/MAG1 and AR2/MAG2 as the case may be. The two access links are foreign networks in the MIPv6's context. All the nodes run Ubuntu 14.04 Linux. For MIPv6, the UMIP code [23] is used on both the MN and the HA. In the case of LISP-MN, the LISPmob [24] implementation is used for the MN, the map-server and the PxTR. OpenAirInterface PMIPv6 developed by Eurocom is used [25] on the two mobile access gateways and the local mobility anchor for PMIPv6.

The web server used is Apache 2.4.17, which hosts the video dataset. We used the open source film Big Buck Bunny from [26]. It consists of different quality levels, from 50 to 3500 kb/s. To ensure that there is a sufficient capacity to sustain the download of the highest video rate, throughout the experimentation, we limit the maximum downstream available bandwidth to $4mbps$. Additionally, all the chunks have a segment length of two seconds unless otherwise stated.

All the players were implemented in Python. For the MSS player we use the open source version from [17]. For the buffer-based player we set the maximum buffer size to $240s$ and the reservoir to $40s$. For MSS Player we retained its default configuration, we also retained the same configuration as used in [18] for the mixed-mode player. Furthermore, the client that runs the players also hosts the following applications: Dummynet, a rate limiter used for limiting the downstream available bandwidth; and tcpdump, a packet analyser used to capture traffic for offline analysis.

3.3. Evaluation Metrics

Research in the field of quality of experience (QoE) is very much active, in this paper we limit out attention to the following metrics for performance evaluation of HAS players:

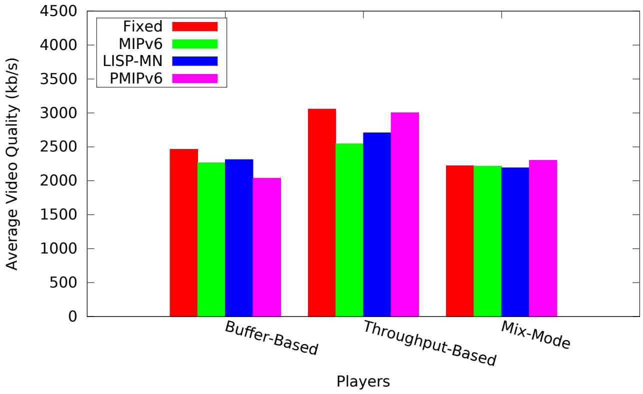


Figure 3: Average Video Quality.

- Rebuffers: this is the total number of video freeze per streaming session.
- Average video rate: is the average of video rate played, weighted by the duration each video chunk is played t_i, calculated as $\frac{t_1q_1+t_2q_2...t_nq_n}{t_n-t_1}$ and measured in kb/s [27].
- Utilisation of available network resource: is calculated by dividing of average video rate by the average network capacity [28].
- Instability: is the fraction of successive chunk requests by a player in which the requested video rate change, measured at the steady-state.

4. Results

This section presents the result of the experiments conducted when the MN is completely stationary and when it is moving across different cells. For the non-stationary experiments, the MN makes six handovers every 21 seconds during the 180-second experiment. The results presented are averages of ten experimental runs.

4.1. Impact on Video Quality

In the test scenarios, the set of video representations available to the MN to choose from ranges from 50kb/s to 3500kb/s. Since the link capacity is set to 4000kb/s we expect that when the MN is not moving the players should be able to converge at the highest available video rate. However, only the throughput-based player is able to converge at the highest available video rate. While the buffer-based player converges to 3000kb/s, the mixed-mode settles at 2500kb/s. We ascribed this performance behaviour of the players to their respective algorithms (see section 5 for detail).

Figure 3 presents the average video quality level of the entire streaming sessions when the MN is both in a fixed position and when moving between the networks. As can be seen from the figure, the mixed-mode player is the least affected by the dynamics of the MMPs under consideration, achieving similar level of average video rate regardless of the mobility protocol being used. However, it achieved lowest average video rate compared to the two other players under review. Hence, as can be seen from Table 1, the mixed-mode player significantly under-utilises the available capacity of the link, using slightly above 50% of the link in all cases.

Utilisation (%)				
Players	Fixed	MIPv6	PMIPv6	LISP-MN
Buffer-Based	61.54	56.60	51.00	57.83
Throughput-Based	76.42	63.62	75.08	67.66
Mixed-Mode	55.56	55.42	58.88	54.81

Table 1: Network Utilisation.

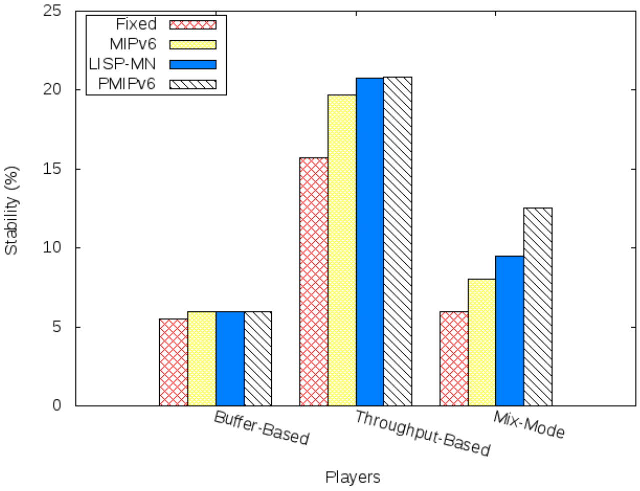


Figure 4: Stability: Percentage change in video quality.

The throughput-based player experiences the highest video quality but is the most sensitive to the change in context, with MIPv6 having highest impact on the player's average video rate, which results in the player losing 15.50% of the average video rate. But when roaming through LISP-MN only 8% drop is observed. In terms of utilisation, throughput-based player achieves the highest percentage. As can be seen from the figures, the buffer-based player is also not very sensitive to the change in MMP, nonetheless it suffers higher level of network under-subscription, utilising virtually half of the available resources when roaming.

4.2. Impact on Stability of the Players

In the event of changing condition, a HAS player is expected to adjust its video rate request. However, it is known that changing video quality level frequently generally has a detrimental impact on QoE [29]. Figure 4 shows the impact of the three mobility protocols on the stability of video quality of the players under consideration as compared to the players in stationary state. It is worth noting that the higher the value the more unstable a player is.

As can be seen the buffer-based player is the most stable in the face of mobility, in fact it achieves same stability level when

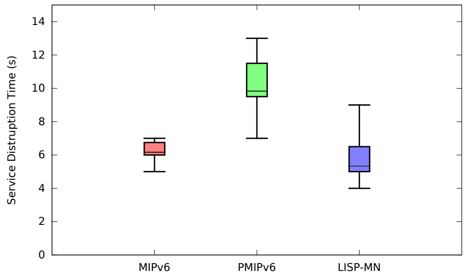


Figure 5: Service Disruption Time.

roaming within MIPv6 and LISP-MN as it does when stationary. Only 1% increase in instability is observed when PMIPv6 is used. However, the situation looks different when video is streamed using the throughput-based player. In this scenario, a significant increase in the instability is observed. For example, when the player roams within PMIPv6 context, there is an increase instability of about 6.0%. Furthermore, the throughput-based player's performance is not really sensitive to changing mobility protocols. Finally, the mixed-mode player saw a slight increase when used within MIPv6 and LISP-MN but the instability significantly increased when PMIPv6 is employed. However, as can be seen in Figure 4, it is still far less than the case of the throughput-based player.

5. Discussion

5.1. Service Disruption

When handover takes place as a result of MN changing its PoA, the MN cannot exchange packets with its CN (i.e. server) and vice versa. During this period the streaming service is basically disrupted. We define the service disruption time (SDT) [30] as *time interval between when a player stops receiving packets from the server and when the first packet is received by the player as the result of a network outage*. Note that this involves more than the handover latency, which is the time interval when network layer entities reestablish connection, SDT includes the time when higher level protocols reestablish connection e.g. TCP hand shake. Figure 5 presents a box plot of all the collected data for the SDT when all the three different mobility protocols are employed. As can be observed PMIPv6 ($Mean[S]$: 9.83s, $StandardDeviation[SD]$: 2.34s) shows both the highest mean value and variability in SDT, with most of the variation towards the lower end. However, LISP-MN (M : 5.33s, SD : 1.86s) has the lowest mean value of the SDT and values that are bias towards the highest recorded. MIPv6 (M : 6.17, SD : 0.75s) shows the lowest viability but a mean value that is somewhere in-between the two.

We observed that this temporary disconnection happens either when a client is sending a request or when the server is responding. If it occurs when the MN is sending a request, then the request will certainly not go through. Therefore, the player will have to back-off and try again at a later time. However, in the event of the server responding to the MN's request, the packets dispatched by the the server will be lost, fortunately TCP will detect that some packets have been lost and the MN requests for retransmission. When the network is disrupted for this long a time, TCP enters the retransmission timeout phase, which results in a very drastic action taken. TCP assumes that there is severe congestion in the network, the window size is reduced to one unit, and then, after every RTT, the window size is increased exponentially until it reaches the slow-start threshold value. For example, in Figure 7(b), it can be observed that whenever a player hands over, the TCP throughput goes to zero (plots touching the x axis). Now we can put these facts together to explain what make the player experiences a drop in video rate and fluctuating video quality when an MN roams.

5.2. Throughput-based Player

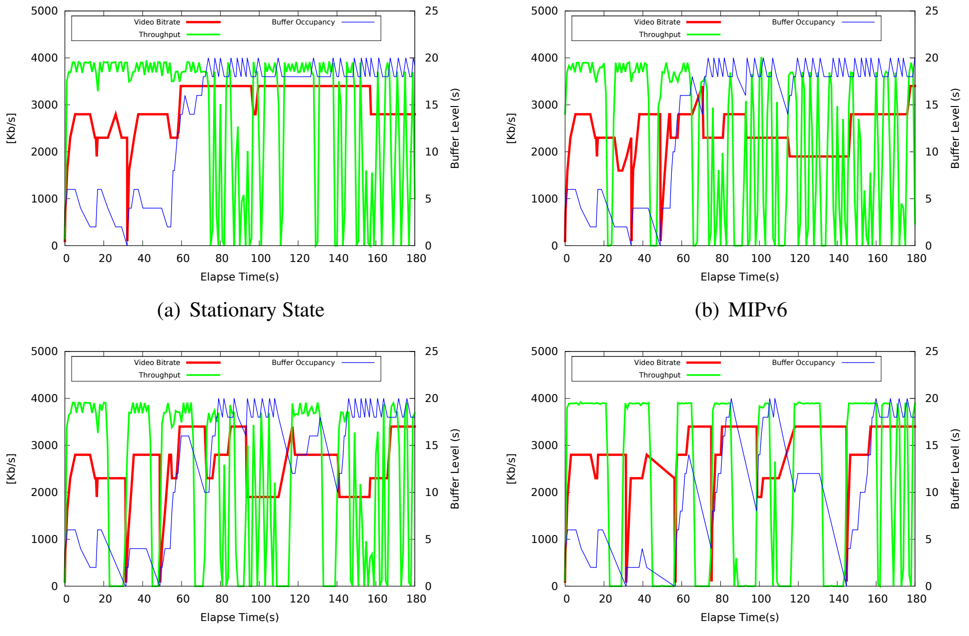
The throughput-based player primarily selects a chunk with the highest video rate lower than the measured throughput, which is derived from a running average of a number of chunks. It does not pay a deserving attention to the prevailing buffer state. Because of this lack of attention to the buffer state changes, as can be observed in Figure 6, the throughput-based player has a near instant convergence regardless of the mode of streaming used. However, the higher the bit-rate requested the slower the buffer growth. As can be noted in 6(a), with the blue plot representing the buffer changes; red plot representing the requested video rate; and the green plot showing the per-second throughput computed from the network trace, while the player is aggressively switching up its video rate, without taking into account of buffer state changes, a point is reached when the rate at which content is received becomes less than the rate at which the content is consumed by the player, hence buffer level continues to fall. This makes the throughput-based player susceptible to unnecessary rebufferings. Even whilst in a fixed position.

Furthermore, as can be seen from Figure 6(b), 6(c), and 6(d) as the service disruption time increases, both the instability of the player and the number of rebufferings rise. This happens because by relying on a historical throughput estimate, the player's responsiveness is inhibited. In other words, it takes time before the player senses that the throughput has fallen as a result of the prolonged handover, which coupled with fact the player is not tracking the buffer state changes. It becomes completely oblivious of the prevailing situation. Hence, it keeps trying to match the video bit-rate to the old pattern of throughput changes even while the buffer is fast depleting, because no content is being received from the server as a results of a handover.

This aggressive nature is why the throughput-based player has the highest average video bit-rate (see Figure 3 and throughput utilisation. However, this comes at the expense of stability (Figure 4) and increasing number rebufferings.

5.3. Buffer-based player

The buffer-based player solely relies on information about buffer state changes for rate selection. A video rate map is used to continuously map the video bit-rate to the buffer occupancy. The player starts by separating the buffering phase (called *reservoir*), in which only the lowest available video rate is downloaded from the ramping-up period where the video quality is



(a) Stationary State

(b) MIPv6

(c) LISP-MN

(d) PMIPv6

Figure 6: Throughput-based player in both stationary and mobility contexts of the three mobility protocols.

linearly incremented. In more detail, provided the reservoir is exceeded, the current chunk quality level is maintained as long as the video rate suggested by the rate map does not exceed the next higher available video quality level, in which case the requested video is increase to the next higher available video quality level. Furthermore, the current bit-rate is only reduced to the next lower quality if the rate suggested by rate map is less than the next available quality level that is lower than the current level.

Figure 7 presents a scenario when the MN is streaming video content using the buffer-based player. Figure 7(a) shows when the player is in a fixed position from the beginning to the end of the experiment. As can be noted from the figure, the video evolution follows the pattern of buffer changes. Since buffer has slower dynamics, i.e changes occur in seconds, the player is able to resist short-term throughput fluctuations. However, this comes at the cost of very higher buffer requirement. It requires about 200s of buffer size, which for most mobile devices, this requirement borders on the infeasible. Furthermore, the player takes long to converge, wasting valuable network resource.

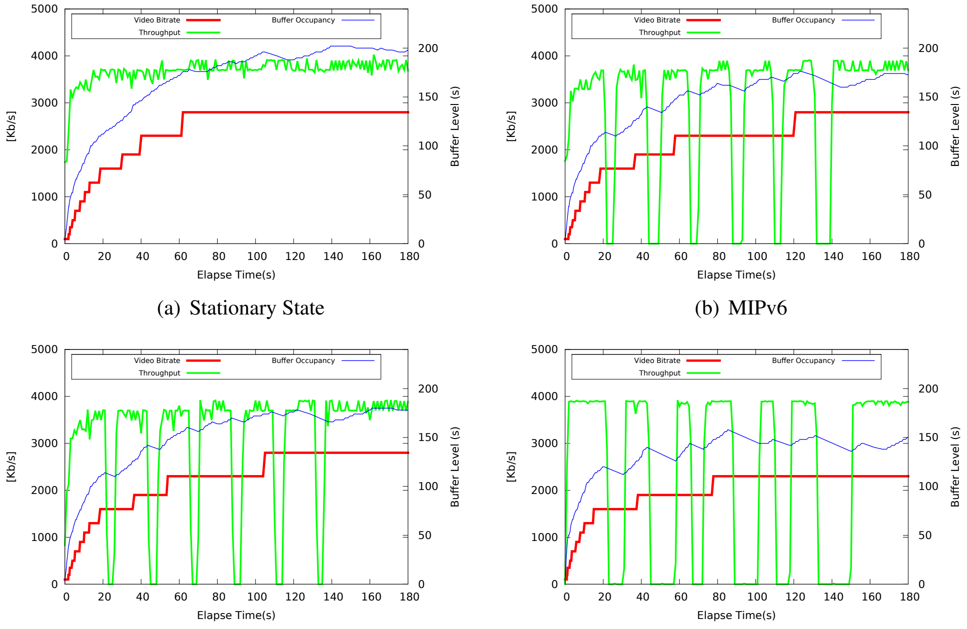
When the player roams across different MMPs, as can be seen from Figures 7(b), 7(c), and 7(d) the player retains its stability but the convergence time grows with increasing SDT. In fact, the player is not able to converge at the appropriate video rate by the time the experiment finishes running over PMIPv6 (recall it has highest mean vale of SDT). The impact of this becomes more evident if we recognise the fact that most content distribution site provide short clips with the average video clip of YouTube being 4.0s [31]. The implication of this is that a user might finish streaming the content without ever viewing the higher available video rate.

The reason for this is that whenever there is a handover, no content is received, hence the player only consumes from the prebuffered content. This results in the player buffer level falling and immediately a connection is established the buffer begins to grow again. This cycle continues resulting in the player not being able to increase its video rate even though the system capacity can sustain higher video rate. Consequently, as seen in Figure 3 and Table 1, the drop in the average video rate and significant under-utilisation of the available resources.

5.4. *Mixed-mode Player*

In contrast to the foregoing two players, the mixed-mode player tries to maintain the buffer at a target level while at the same time providing a reasonable video rate. From Figure 8(a), it is easy to see that the player is trying to maintain buffer to a certain level (about 60s). It can be observed that the player activates the ON-Off video chunk request pattern, which ensures that the buffer replenishes with similar amount of content that has been played-out since the last download. This results in a very stable player but that significantly under-utilises the available resources.

As the SDT increases from MIPv6 to PMIPv6 (see Figures 8(b), 8(c), and 8(d)), the buffer becomes more and more difficult to manage, because for every handover the buffer state fluctuates, which makes the player very sensitive to change in MMP in terms of buffer management. Understandably, the instability of the player increases with rise in the SDT (see Figure 4). This eventually led to buffer under-run in the case of PMIPv6. Another important observation that can be made is, in an attempt to manage the buffer, the player becomes very conservative in its download, resulting in low average video rate and network utilisation. However, it should be observed that change in mobility



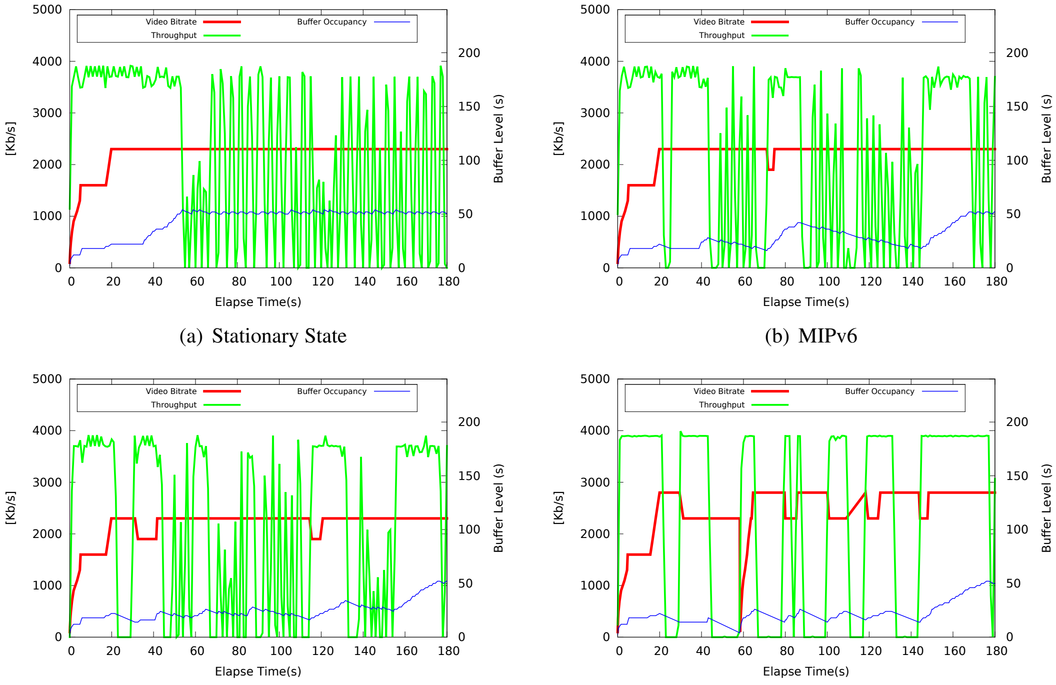
(a) Stationary State

(b) MIPv6

(c) LISP-MN

(d) PMIPv6

Figure 7: Buffer-based player in both stationary and mobility context of the three mobility protocols.



(a) Stationary State

(b) MIPv6

(c) LISP-MN

(d) PMIPv6

Figure 8: Mixed-mode player in both stationary and mobility context of the three mobility protocols.

protocol does not affect the quality of video downloaded. Infact the player's performance in the video quality aspect whilst running on top of the three mobility protocols is almost similar to stationary scenario, as discussed in section 4.1.

6. Conclusion

By the year 2019, more than 50% of Internet traffic will be video content streamed over wireless networks owing to the astronomical rise in the use of mobile devices. HTTP Adaptive Streaming (HAS) is currently the popular method of providing an efficient video streaming service. Hence, understanding the impact of mobility on HAS is critical for providing a reasonable user experience. Using a laboratory testbed, we experimentally evaluate the performance of three selected HAS players: a buffer-based, a throughput-based and a mixed-mode player; running over three different mobility protocols: LISP-MN, MIPv6, and PMIPv6. And using a fluid-flow mobility model, we measured the impact of the mobility protocols on buffer dynamics, video quality and stability of the players.

We found the mixed-mode player to be the most sensitive to change in mobility protocol, and experienced both the lowest average video quality and network utilisation. In contrast, the throughput-based player has the highest video quality that can be achieved and seems the least affected by the impact of the various mobility protocols but at the expense of stability. The player also suffers an increase in the number of rebuffers as it continues to match the video quality level to the throughput even while the buffer is fast depleting. The buffer-based player is the most stable in the face of mobility showing low percentage change in the quality of video requested but unable to reach the maximum quality level during the mobile device's movement. We also found that PMIPv6 induces the highest instability across all the players followed by LISP-MN with MIPv6 having the least impact. Mobility, generally, affects the performance of the players and causes degradation of all the observed parameters. In future, we intend to improve one of the mobility management protocols, specifically LISP-MN with capability to cache incoming packets close to the MN's new location and forwarding them to the MN on handover completion. This we believe will go a long way in ameliorating the observed issues.

7. Acknowledgement

This research is funded by Petroleum Technology Development Fund (PTDF), Nigeria.

References

[1] Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021 white paper (April 2016).

[2] H. Ali-Ahmad, C. Cicconetti, A. La Oliva, M. Draxler, R. Gupta, V. Mancuso, L. Roullet, V. Sciancalepore, Crowd: An sdn approach for densenets, in: Software Defined Networks (EWSDN), 2013 Second European Workshop on, IEEE, 2013, pp. 25–31.

[3] Z. Gao, A. Venkataramani, J. Kurose, S. Heimlicher, Towards a quantitative comparison of the cost-benefit trade-offs of location-independent network architectures, Tech. rep., Technical report, School of Computer Science, University of Masachusetts, Amherst MA 01003 (2014).

[4] A. Gladisch, R. Daher, D. Tavangarian, Survey on mobility and multihoming in future internet, Wireless personal communications 74 (1) (2014) 45–81.

[5] C. Perkins, D. Johnson, J. Arkko, Rfc 6275: mobility support in ipv6, Internet Engineering Task Force (IETF).

[6] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, Rfc 6830: The locator, ID Separation Protocol (LISP).

[7] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, B. patil," proxy mobile ipv6, Tech. rep., RFC 5213, August (2008).

[8] Mpeg dash specification (iso/iec 23009-1:2012) dynamic adaptive streaming over http (dash) part 1: Media presentation description and segment formats, Tech. rep. (2012).

[9] A. Biernacki, K. Tutschku, Performance of http video streaming under different network conditions, Multimedia Tools and Applications (2013) 1–24.

[10] M. Gorius, Y. Shuai, T. Herfet, Dynamic media streaming over wireless and mobile ip networks, in: Consumer Electronics - Berlin (ICCE-Berlin), 2012 IEEE International Conference on, 2012, pp. 158–162. doi:10.1109/ICCE-Berlin.2012.6336461.

[11] E. U. T. R. Access, and evolved universal terrestrial radio access network (e-utran), Overall description 126.

[12] S. Akhshabi, A. C. Begen, C. Dovrolis, An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http, in: MMSys '11, MMSys '11, ACM, New York, NY, USA, 2011, pp. 157–168. doi:10.1145/1943552.1943574.
URL http://doi.acm.org/10.1145/1943552.1943574

[13] T. C. Thang, Q.-D. Ho, J. W. Kang, A. T. Pham, Adaptive streaming of audiovisual content using mpeg dash, Consumer Electronics, IEEE Transactions on 58 (1) (2012) 78–85.

[14] S. Akhshabi, L. Anantakrishnan, A. C. Begen, C. Dovrolis, What happens when http adaptive streaming players compete for bandwidth?, in: NOSSDAV '12, ACM, 2012, pp. 9–14.

[15] T.-Y. Huang, R. Johari, N. McKeown, Downton abbey without the hiccups: Buffer-based rate adaptation for http video streaming, in: Proc. of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking, ACM, 2013, pp. 9–14.

[16] Y. Sani, A. Mauthe, C. Edwards, Modelling video rate evolution in adaptive bitrate selection, in: The IEEE International Symposium on Multimedia (ISM 2015), 2015, pp. 89–94.

[17] Slextensions adaptive streaming, https://slextensions.svn.codeplex.com/svn/trunk/SLExtensions/AdaptiveStreaming (March 2014).

[18] K. Miller, E. Quacchio, G. Gennari, A. Wolisz, Adaptation algorithm for adaptive streaming over http, in: Packet Video Workshop (PV), 2012 19th International, IEEE, 2012, pp. 173–178.

[19] D. M. Nguyen, L. B. Tran, H. T. Le, N. P. Ngoc, T. C. Thang, An evaluation of segment duration effects in http adaptive streaming over mobile networks, in: 2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS), 2015, pp. 248–253.

[20] L. Zhou, R. Q. Hu, Y. Qian, H. H. Chen, Energy-spectrum efficiency tradeoff for video streaming over mobile ad hoc networks, IEEE Journal on Selected Areas in Communications 31 (5) (2013) 981–991.

[21] I. F. Akyildiz, W. Wang, A dynamic location management scheme for next-generation multitier pcs systems, IEEE Transactions on Wireless Communications 1 (1) (2002) 178–189. doi:10.1109/7693.975456.

[22] How do i know how fast i am walking?, http://walking.about.com/od/measure/f/howfastwalking.htm (January 2014).

[23] Mobile ipv6 and nemo basic support implementation for linux, http://www.umip.org/, accessed: 2014-06-12.

[24] An open-source lisp implemantation for linux, android and openwrt, http://lispmob.org, accessed: 2015-05-31.

[25] Openairinterface proxy mobile ipv6, http://www.umip.org/contrib/umip-oai-pmipv6.html (December 2015).

[26] S. Lederer, C. Muller, C. Timmerer, Dynamic adaptive streaming over http dataset, in: Proc. of the 3rd Multimedia Systems Conference, ACM, 2012, pp. 89–94.

[27] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, H. Zhang, Understanding the impact of video quality on user engagement, ACM SIGCOMM Computer Communication Review 41 (4) (2011) 362–373.

[28] G. Tian, Y. Liu, Towards agile and smooth video adaptation in dynamic http streaming, in: Proc. of the 8th International Conference on Emerging Networking Experiments and Technologies, ACM, 2012, pp. 109–120.

[29] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, P. Tran-Gia, A survey on quality of experience of http adaptive streaming, Communications Surveys Tutorials, IEEE 17 (1) (2015) 469–492.

[30] M. Isah, C. Edwards, Inter-domain mobility with lisp-mn–a performance comparison with mipv6, in: 2015 8th IFIP Wireless and Mobile Networking Conference (WMNC), IEEE, 2015, pp. 80–87.

[31] comscore releases january 2014 u.s. online video rankings, http://www.comscore.com/Insights/Press-Releases/2014/2/comScore-Releases-January 2014-US-Online-Video-Rankings (June 2016).