[Proceeding] PIT Overload Analysis in Content Centric Networks

Original Citation:
Virgilio M., Marchetto G., Sisto R. (2013). *PIT Overload Analysis in Content Centric Networks.* In: ACM SIGCOMM Workshop on Information-Centric Networking (ICN 2013), Hong Kong, China, August 12, 2013. pp. 67-72

Availability:
This version is available at : http://porto.polito.it/2513861/ since: September 2013

Publisher:
ACM

Published version:
DOI:10.1145/2491224.2491225

PIT Overload Analysis in Content Centric Networks

Matteo Virgilio
Department of Control and
Computer Engineering
Politecnico di Torino
matteo.virgilio@polito.it

Guido Marchetto
Department of Control and
Computer Engineering
Politecnico di Torino
guido.marchetto@polito.it

Riccardo Sisto
Department of Control and
Computer Engineering
Politecnico di Torino
riccardo.sisto@polito.it

ABSTRACT

Content Centric Networking represents a paradigm shift in the evolution and definition of modern network protocols. Many research efforts have been made with the purpose of proving the feasibility and the scalability of this proposal. Our main contribution is to provide an analysis of the Pending Interest Table memory requirements in real deployment scenarios, especially considering the impact of distributed denial of service attacks. In fact, the state that the protocol maintains for each resource request makes the routers more prone to resources exhaustion issues than in traditional stateless solutions. Our results are derived by using a full custom simulator and considering the different node architectures that have been proposed as valid reference models. The main outcomes point out differentiated weaknesses in each architecture we investigated and underline the need for improvements in terms of security and scalability.

Keywords

Content Centric Networking, Security, DDoS attacks, Simulation

1. INTRODUCTION

Content Centric Networking (CCN) [7] is emerging as one of the possible dominant paradigms for the future network architecture. Starting from the valid claim that in a content distribution network it is actually important to address resources themselves, rather than their physical location, CCN aims to change the traditional network operation by making all network devices name-aware. Then, routing decisions are taken according to the name of the resource the user is requesting.

The approach is promising and may represent one of the most significant innovations in the networking field. However, its new operating mode opens the path for possible issues and threats that were not present in the traditional network. These have to be properly investigated in order to have an exhaustive evaluation of the overall CCN architecture. For example, content delivery is based on a status information (the requested URI) that has to be maintained at nodes in the so called Pending Interest Table (PIT). This may pose strict constraints in terms of reliability and scalability. In fact, this table may overflow, with consequent service disruption and possible network collapse. Furthermore, these constraints might be even exploited by attackers in order to slow down or even interrupt the normal network operation.

In this paper we deal with this specific problem by providing a performance evaluation of some possible PIT architectures in terms of resilience to (possibly malicious) overload conditions. In particular, we consider three PIT architectures, referenced in the available CCN literature: (i) a PIT storing all the bytes composing an URI, which we call SimplePIT; (ii) a PIT storing a fixed length entry for an URI, which we call HashedPIT; (iii) a PIT implemented through multiple Bloom Filters placed in each router interface, as described in [18]. This solution is known as DiPIT.

The experiments are conducted by means of an ad-hoc simulator, designed to recreate the behavior of a CCN network and to track memory usage at CCN nodes. In order to obtain significant results, the simulator implements the topology of a prominent Italian ISP. The paper has the following structure: Section 2 briefly introduces CCN and its architecture. Section 3 dives into PIT overloading issues, explaining the importance of this problem in the CCN context. Section 4 offers an overview of the related work in this field. Section 5 introduces our experimental analysis and presents the obtained results. Finally, Section 6 concludes the paper and also presents some possible future work.

2. CONTENT CENTRIC NETWORKING

Content Centric Networking (CCN) relies on two main types of packets: Interest packets and Data packets. The former are used by clients to request a (piece of) resource. They include the content name, in the form of a Uniform Resource Identifier (URI), plus a set of parameters useful for Interest processing. The latter are transmitted as responses to client requests and are used to deliver pieces of data.

One of the key features of the CCN architecture is the caching capability. A CCN node, e.g., an edge router, can store Data packets in its local repository and autonomously serve future requests for that content, without forwarding ahead Interest packets. This possibly leads to a reduced latency time experienced by clients and to a lower bandwidth usage. In order to correctly deliver data, CCN nodes are equipped

ICN'13, August 12, 2013, Hong Kong, China.
Copyright 2013 ACM 978-1-4503-2179-2/13/08 ...$15.00.

with three data structures: (*i*) The *Content Store* (CS), which is the area where Data packets are cached. Each Interest arrival causes a Content Store lookup: if the requested name is present in the store, the response is immediately generated; (*ii*) The *Pending Interest Table* (PIT), which is the data structure where routers annotate forwarded Interests and the respective arrival interfaces. Multiple Interests for the same URI are aggregated in a single entry (Interest merging) to improve the system scalability; (*iii*) The *Forwarding Information Base* (FIB), which is the equivalent of the IP routing table. When a client generates an Interest, each router in the path towards the destination adds an entry in its PIT. In this way, devices readily forward packets to the right hosts when the response comes back. The entry remains in the PIT for a time interval called LifeTime, which is specified in the Interest itself. If the LifeTime expires and the response has not yet arrived, the memory is released. Notice that this parameter is chosen by clients. Hence, at least according to the current CCN rules, it is not under the network control.

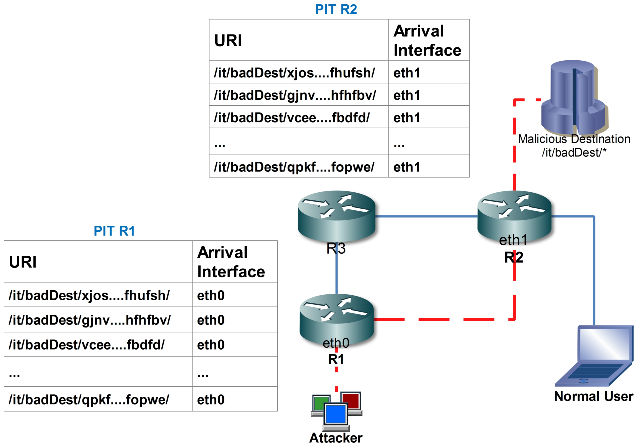
3. PROBLEM DESCRIPTION

The PIT is the fundamental structure used to maintain the state of each active flow. It grows with users sending their Interests and shrinks when Data packets arrive at the router. Considering the access speed required for such a structure and the possibilities offered in this sense by current memory technologies, the PIT size might represent a bottleneck for the entire CCN infrastructure.

The problem might be exacerbated by a massive usage of long Interest LifeTimes, which would further increase the number of simultaneous entries in the PIT. Despite the *pull* nature of CCN, this possibility cannot be excluded as it would be necessary for supporting publish/subscribe services [6] where users subscribe for a given content that will be asynchronously produced in the future. Many of these services are implemented, for example, by means of HTTP long polling strategies, which make extensive use of long (potentially unanswered) requests. [9] presents a complete discussion on the best practices about timer setting and on how 'long' these requests should be. In general, 30 s is considered a safe value, as longer timers may be undesirable for intermediate proxies placed between the server and the client. However, solutions with longer values are widely adopted in common web applications (e.g.,FaceBook and some web-based mail applications, which to our experience often use timers of more than one minute). In addition to that, we have to consider that one or more malicious users could craft artificial requests with the purpose of filling the available PIT memory on routers, thus implementing a Distributed Denial of Service (DDoS) attack.

With PIT overflow, users would see their Interests discarded by routers and, consequently, they would experience an increasing rate of retransmissions until the network completely collapses. With this problem in mind, some possible PIT architectures have been proposed with the aim of reducing the required table size. A first simple solution [7] is to realize the PIT by means of a hash-table, where each URI is encoded using a fixed number of bits. Other possible solutions deploy more complex architectures. For example, [3] proposes a tree-like structure to arrange the PIT entries in order to also ensure high lookup, insertion, and deletion speed, while [18] presents a very efficient PIT architecture based on Bloom Filters.

While these solutions may be effective during normal operation, performance degradation might be experienced when the system is under the abovementioned DDoS attacks. This kind of attack may be implemented by distributedly generating Interest packets that contain a valid destination prefix but non-existing resource names, so that routers properly forward Interests and store new entries in their PITs, but responses never come back. The destination might possibly be colluded with the attacker and simply drop incoming packets. In order to maximize the impact of the attack, a malicious user could request always different resources thus avoiding Interest merging. Furthermore, the attacker could also select large values for the LifeTime field. Figure 1 shows a very simple network consisting of three routers, an attacker (or, equivalently, a bot net), a normal client and a bad destination, which is placed ad hoc by the attacker. The role of the bot net is that of generating Interests towards the fake destination. Routers along the dashed path (R1 and R2) are flooded with unanswered Interests, which indeed increases their memory usage.



PIT R2

URI	Arrival Interface
/it/badDest/xjos....fhufsh/	eth1
/it/badDest/gjnv....hfhfbv/	eth1
/it/badDest/vcee....fbdfd/	eth1
...	...
/it/badDest/qpkf....fopwe/	eth1

PIT R1

URI	Arrival Interface
/it/badDest/xjos....fhufsh/	eth0
/it/badDest/gjnv....hfhfbv/	eth0
/it/badDest/vcee....fbdfd/	eth0
...	...
/it/badDest/qpkf....fopwe/	eth0

Figure 1: Attack scenario

This paper presents an analysis of the PIT resilience to possible overload and consequent service disruption. A comprehensive simulation study of the possible PIT architectures is done in order to evaluate their performance from this perspective.

4. RELATED WORK

The feasibility of the CCN approach with regard to the amount of requested resources at nodes, together with a proper CCN node dimensioning, has been recently object of a wide research activity. For example, [2] presents an efficient router design and describes some possible usage scenarios. Designing an efficient forwarding plane is also the subject of [19], which identifies key issues related to the protocol fast speed implementation and establishes some principles to be observed in order to design scalable forwarding architectures. Furthermore, [11] presents a feasibility study of CCN and concludes that CCN nodes based on current technologies would still be unable to sustain requests arrival rates at the Internet scale. However, they can sustain rates at Content Distribution Network and small ISP scale. Specifically concerning the PIT dimensioning problem, two recent papers propose different architectures conceived for reducing the PIT size. As also described in the previous section, [3] proposes a tree-like PIT structure, while [18] present a PIT architecture based on Bloom Filters. However, both papers do not provide a quantitative evaluation of the PIT overload problem and do not consider possible DDoS attacks.

Concerning the security threats in a CCN network, the available literature mainly covers privacy, data authentication, and data integrity issues. For example, [15] addresses the privacy problem deriving from the stateful operation of CCN routers, while data authentication and integrity have been considered since the seminal CCN papers [7, 8]. Instead, a comprehensive study of the possible PIT overload issue deriving from a DDoS attack is missing. [4] presents a solution to mitigate these attacks, while the ongoing work by Gasti et al. [12] gives an introductive overview of the problem and of some ongoing experiments. Hence, the available information is somewhat complementary to the results of this paper, which provides a quantitative evaluation of the resilience of some proposed PIT architectures to this kind of attack.

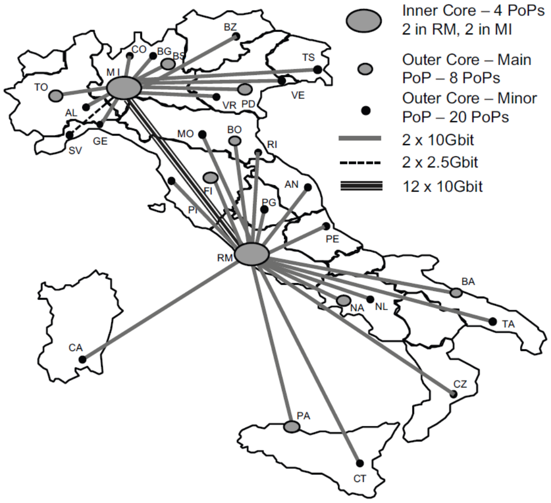


Figure 2: Telecom Italia topology

5. PIT RESILIENCE ANALYSIS

This section focuses on the evaluation of the PIT resilience to possible overload. Our analysis is performed by simulation and considers three possible PIT architectures: (*i*) a PIT storing all the bytes that compose an URI, referred to as SimplePIT; (*ii*) a PIT storing fixed length entries evaluated as hash values of the URIs, referred to as HashedPIT; (*iii*) a PIT implemented through multiple Bloom Filters placed in each router interface, as described in [18]. This last solutions is known as DiPIT. The PIT architecture presented in [3] is not explicitly investigated here as it shares with the HashedPIT the same principle of introducing fixed length entries (in [3] the numerical codes that make up the logical tree structure are reduced to fixed length).

While in the first two cases PIT overloading can be measured in terms of memory occupancy, with possible memory overflow, in the latter case this is not possible as the DiPIT is based on Counting Bloom Filters, where memory occupancy is constant. In essence, the filter fills the entire memory size and each Interest is encoded in a sequence of '1s' to be added at given positions of the filter. In this case, PIT overloading results in a high rate of the so called false positive events, namely, the node erroneously concludes that an Interest is present in the filter and either does not propagate it or erroneously removes some entries from the PIT when a chunk is received. In both cases this results in a degradation of the service perceived, as both events require one or more Interest retransmissions. [18] proposes an effective combination of filters to reduce the false positive probability during normal operation (see [18] for further details), but this may increase when the network is under attack. In order to quantitatively evaluate the effect of false positives and also enable a coherent comparison among the three considered PIT architectures, we base our analysis on the average percentage of Interest retransmissions that occur at users' machines. This is clearly proportional to the average download time they experience, and hence it is a metric of the overall network performance.

A further issue to address is the selection of a proper simulator complying with the requirements of our study. Some CCN simulators are available, but all are customized for specific analyses. For example, ccnSim [5] has been designed mainly for the evaluation of cache replacement techniques and hence is not optimized for experimenting on the PIT management. Hence, in order to focus our analysis on PIT issues and meet our specific requirements, we developed a full custom event-driven Java simulator[1].

5.1 Simulation scenario

With the main aim of obtaining significant and quantitative results, the network configuration and user behavior models are refined to faithfully recreate realistic scenarios. In particular, we adopt as a reference the network of Telecom Italia, a prominent Italian ISP. Data related to the Telecom Italia network that are of interest in this context are publicly available on the web.

First of all, the topology adopted in our simulations reproduces the real structure of the Telecom Italia network [16] (see Figure 2, represented at Point of Presence (PoP) granularity). The network is divided in two areas: the first one that covers Northern Italy and the second one that covers central and Southern Italy. PoPs are almost equally divided between the Northern and the Southern areas and in our simulator each PoP is represented by a router. This topology is typical also of many ISPs in Europe.

Concerning the network population, we adopt the number of broadband Internet subscriptions currently active in the Telecom Italia network — around 9 million — available in the ISP investor relation [17]. We also assume the topological distribution of users among the ISP PoPs to be directly proportional to the geographical distribution of the ISP customers. This allows us to accurately reproduce the download traffic pattern of the ISP users. Furthermore, access bandwidths are considered uniform for simplicity and equal to 7 Mbps (download) and 1 Mbps (upload), which are the values characterizing a large percentage of Telecom Italia users. The overall header size of the protocols underlying CCN is assumed fixed to 20 bytes.

Regarding the selection of the content to be retrieved by

[1] The NS-3 based NdnSIM CCN simulator [1] was not yet available when we started our project, but we plan to move to that environment for our future work on this topic

clients, we assume a Zipf- Mandelbrot probability distribution, which Saleh et al. [14] proved to properly model the behavior of users in a content distribution P2P network. Given the content delivery nature of a CCN network, this probability distribution could reasonably model the user behavior also in our case. According to the Zipf's Law and given the total number of resources in the network, which are uniformly distributed among per-PoP content providers, the corresponding distribution function is expressed as follows:

$$p(i) = \frac{1}{(i+q)^{\alpha}} \qquad \forall i \in [1, N]$$

where $p(i)$ is the probability of extracting the i-th content available in the network, q and α are two parameters that [14] fixed to $\alpha = 0.55$, $q = 25$ for a residential ISP (as Telecom Italia is), and N is the total amount of resources.

Download requests are modeled using a Poisson process with average rate equal to 500 requests per second. This results in an average value of around 12 million simultaneously active downloads in the steady state. We believe this network load is even higher than that observed in a 9 million users network during normal operation and hence it is significant for our study.

Another key parameter for our evaluation is the memory availability at CCN nodes. Reasonable values for the PIT size that match both the required memory access speed and current memory technologies are some hundreds of MB [18], available by means of the SRAM technology. In order to be more conservative and also with the aim of analyzing a possible near future scenario, we fix the PIT size to 1 GB. In the DiPIT case, this value refers to the overall available filters memory (the PITs plus the shared Bloom filter).

Concerning the attack parameters, we consider a maximum aggregate attack bandwidth of 4 Gb/s, which is a realistic value for a medium scale ISP such as Telecom Italia [13], and Interest LifeTimes values that vary between 4 s (the default value considered in the CCNx [10], the prototypal implementation of CCN node) and 180 s. Notice that those are reasonable LifeTime values if we think to enable complete publish/subscribe services in a CCN network, as discussed in Section 3.

5.2 Results

This section reports on the simulation results obtained for the three considered PIT architectures: the SimplePIT, the HashedPIT and the DiPIT. We consider a distributed bot net sending Interests to fake destinations connected to the Rome edge router. Hence, our analysis focuses on the Rome PoP as it is the most overloaded node. In essence, we keep track of the memory usage at the Rome PoP. Furthermore, we also measure the average percentage of retransmissions experienced by users, as depicted above.

5.2.1 I scenario - SimplePIT

In the first scenario, we consider to deploy a SimplePIT at each router. This PIT implementation stores the entire URI in the memory so it is the simplest architecture we can think of. In order to maximize transmission efficiency, and consequently maximize attack impact, it is important for the attackers to craft very long URIs. We selected 1000 bytes in our simulation. In particular, each malicious URI has a valid 13 bytes prefix (e.g. **/it/badPubRm/** required for reaching the destination) and a non-existing resource name whose length is set to a randomly selected string of 1000 bytes.

Simulation results concerning the memory occupancy at the Rome PoP and the related retransmission rate are shown in Table 1 for several values of the overall bot net bandwidth and of the fake Interest LifeTimes (retransmissions are referred only to finished downloads). Users do not experience a considerable retransmission rate until the PIT is completely full. However, by increasing either the overall attack bandwidth or the Interest LifeTime, routers start to be unable to correctly handle incoming traffic and all the connections are significantly slowed down.

These results can be analytically justified. Let us consider the following case as an example (we recall that the overall headers of underlying protocols is fixed to 20 bytes in our simulations):

$$B_{attackers} = 2\,Gbps,\ \ LifeTime = 4sec \tag{1}$$

$$|Interest|_{attackers} = 1033\,bytes(\overbrace{1013}^{\text{URI}} + \overbrace{20}^{\text{HEADER}}) \tag{2}$$

The number of Interests per second generated by the bot net can be calculated in the following manner:

$$\frac{(2*10^9)\ bps}{(1033*8)\ bits} \equiv 242013\ \frac{Interest}{sec} \tag{3}$$

Considering a 4 s LifeTime, we obtain:

$$\approx 242013*4 = 968.052\ entries \Rightarrow \approx 980MB\ occupied \tag{4}$$

which is consistent with our simulation result.

5.2.2 II scenario - HashedPIT

In the second scenario, we consider the HashedPIT, namely, a centralized hash table storing a fixed length entry for each URI in transit. We exploit the SHA-1 hashing algorithm in order to ensure a negligible collision rate. In this context, the best size for an attacker's URI is 20 bytes, according to the SHA-1 output digest (160 bits). Longer URIs are useless as would be reduced to 20 byte strings by CCN nodes. In this case the resultant attackers transmission efficiency, here intended as the amount of transmitted data that will be stored in the PIT with respect to the total traffic generated by the host, is around 50% due to the non-negligible underlying header size (20 byte headers in our simulations lead to 20 / (20+20) = 50 % transmission efficiency). Since in the previous case the efficiency was about 98% (1013 byte long URIs lead to 1013 / (1013 + 20) ≈ 98%), one could infer that an attacker has simply to double its bandwidth to get the same impact of the SimplePIT case. This is only partially confirmed by simulation results presented in Table 1. We can observe how, as expected, the memory occupancy is halved at equal simulation conditions and, consequently, the retransmission rate grows slowly with the increase of the attack intensity. However, it is worth noticing the greater attack effectiveness when the PIT is almost completely full with respect to the previous case. Such a situation here prevents users from finishing any downloads and the number of retransmissions tends to infinite, namely, no useful data are received by clients, who continue to retransmit Interests: the network is completely unusable. The infinite value in Table 1 just represents the situation in which no stable value is reached for the number of retransmitted packet from the client point of view. This is due to the fact that the first

Attack settings	**SimplePIT**		**HashedPIT**		**DiPIT**	
	Retransmissions	RAM Usage	Retransmissions	RAM Usage	Retransmissions	RAM Usage
Band = 100 Mbps *LifeTime = 4 sec*	0	≈ 49 MB	0	≈ 25 MB %	≈ 0.01 %	1 GB
Band = 500 Mbps *LifeTime = 4 sec*	0	≈ 245 MB	0	≈ 125 MB	≈ 2.42 %	1 GB
Band = 2 Gbps *LifeTime = 4 sec*	0	≈ 980 MB	0	≈ 500 MB	≈ 87.6 %	1 GB
Band = 4 Gbps *LifeTime = 4 sec*	≈ 15 %	≈ FULL	≈ 83 %	≈ FULL	≈ 90 %	1 GB
Band = 100 Mbps *LifeTime = 60 sec*	0	≈ 735 MB	0	≈ 375 MB	≈ 21 %	1 GB
Band = 100 Mbps *LifeTime = 120 sec*	≈ 37 %	≈ FULL	0	≈ 750 MB	≈ 86 %	1 GB
Band = 100 Mbps *LifeTime = 180 sec*	≈ 52 %	≈ FULL	∞	≈ FULL	≈ 88 %	1 GB

Table 1: PITs performance evaluation

data structure is more complex to overfill as stored Interests are of highly variable length and attackers do not have exact information of the amount of memory that is still available at routers. Hence, when the PIT is almost full many attackers' fake Interests are discarded as they are too large with respect to the available space. Some shorter users' Interest can instead be accommodated and hence the network is still able to operate, although with a significantly reduced efficiency. In the HashedPIT case, instead, 20 bytes is always the best choice for URI length so the attack is more effective and destructive. This clearly does not mean that the SimplePIT can be considered resilient to DDoS attacks as specific attacker techniques may be adopted to possibly saturate the PIT. This analysis is left for future work.

5.2.3 III scenario - DiPIT

The last scenario refers to a very different PIT and router architecture. The central PIT is split into multiple smaller per-interface PITs, each implemented by a Counting Bloom Filter data structure. The specific description of this proposal and the algorithm adopted is available in [18]. The retransmissions observed in this case are due to the Bloom Filter false positive events, as described above in this section. The probability for a Bloom Filter to return a false positive may be approximately evaluated[2] as $(1 - e^{-\frac{k*n}{m}})^k$, where k is the number of hash functions deployed to code a given element, n is the number of elements currently in the filter, and m is the total size of the filter. In our simulator each Bloom Filter is modeled by means of this probability function. Furthermore, the specific architecture based on different filters and the insertion/deletion algorithms presented in [18] are reproduced in order to obtain consistent results.

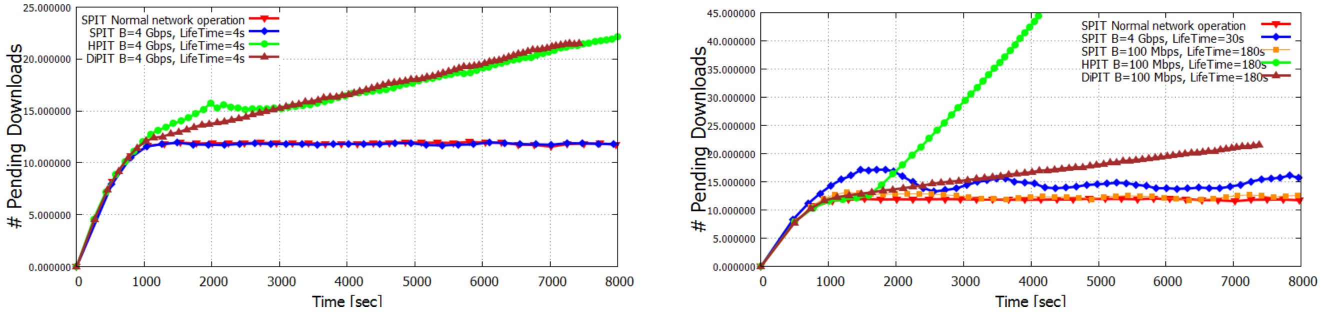
In [18], the authors also suggest possible values for k, which vary according to the Interest arrival rate that the router has to support. We set this value to 4 hash functions because larger values are not suitable for high-end routers. For the Bloom Filter, we assume for simplicity 8 bit counters and no counter overflow. Simulation results are presented in Table 1. It is worth noticing how although the DiPIT does not suffer from memory overflow (neglecting counter overflow), false positive events become a truly limiting factor when the network is under attack and many entries are inserted into the filter. In fact, considerable retransmission rates are observed also with non-huge attack intensities.

5.3 Discussion

These simulation results lead us to a twofold conclusion. First, none of the analyzed PIT architectures is overloaded during normal operation in the considered network scenario. Even with a low intensity attack, memory usage is reasonable and no retransmissions are observed. This in some way confirms that even a traditional SimplePIT-based solution might be currently deployed at ISP scale, as also concluded by the analysis in [11]. However, second, there are significant weaknesses in all the architectures when the attack intensity grows.

To summarize our results, Figure 3 plots the number of pending downloads in the network over time for some reference values of attack bandwidth B and fake Interest LifeTime. In Figure 3(a), the considered attack intensity leads this number to slowly diverge with a similar trend for both the HashedPIT and DiPIT based nodes. For the SimplePIT we do not observe a significant gap with respect to the system behavior during normal operation. Figure 3(b) considers a more critical scenario: B=100 Mbps, LifeTime=180 s and also B=4 Gbps, Lifetime=30 s for the SimplePIT. We can observe how the SimplePIT is only partially affected in this second case, while the system fast exits the steady state due to retransmission rate divergence when the HashedPIT is deployed. We can conclude that the HashedPIT is the architecture most affected by the considered attack, while the SimplePIT is the architecture most resilient for the reasons explained above. The DiPIT has an intermediate behavior. Clearly, these results hold in our attack scenario. Starting from this point, one could design other specific attacker behaviors that even worsen the perceived service, especially for the SimplePIT. For example, an attacker may: (i) combine broad bandwidth and higher LifeTime to increase attack effectiveness; (ii) distribute more zombies around the network to avoid attack source detection; (iii) exploit more bad prefixes in order to make any countermeasures even more complex to deploy. This is a non-exhaustive list that further motivates the real need for proper countermeasures to DDoS in a CCN network.

[2] Assuming independence for the probabilities of each bit being set.



(a) (b)

Figure 3: Network performance evaluation

6. CONCLUSION AND FUTURE WORK

This paper compares the existing PIT architectures with the aim of providing a quantitative evaluation of their resilience to overload conditions, especially when these are due to DDoS attacks. Our results show how all architectures are affected by these attacks and hence further studies are needed to figure out possible countermeasures. One can be, for example, the introduction of smarter algorithms for LifeTime management at content routers, which adapt LifeTimes as a function of the network load: a router can grant larger LifeTime values in case of low traffic congestion and, conversely, implement a sort of LifeTime shaping when the load increases. With such a mechanism, we would break down the hypothesis that intermediate nodes do not manage LifeTime values as well as PIT entries removal (for example by discarding the ones which have a too long expiration time). Other mechanisms could be based on an Interest RED (Random Early Discard) strategy based on the amount of occupied memory. The higher the congestion, the larger is the probability of discarding an incoming Interest. This would call for a complete performance comparison with existing mechanisms that, in the context of IP networks, drop packets to handle traffic congestion. Last but not least, it is necessary to figure out possible strategies to detect and drop malicious Interests. These and other possible mechanisms will be subject of our future work. We will perform a quantitative evaluation of all of them, also considering their complexity in terms of additional computing and memory resource needs.

7. ACKNOWLEDGMENTS

The authors would like to thank Prof. Luigi Ciminiera, who had a fundamental role in starting up this work.

8. REFERENCES

[1] A. Afanasyev, I. Moiseenko, and L. Zhang. ndnsim: Ndn simulator for ns-3. *Technical Report, NDN-0005*, Oct. 2012.

[2] S. Arianfar, P. Nikander, and J. Ott. On content-centric router design and implications. In *ReARCH '10*, Philadelphia, PA, USA, Nov. 2010.

[3] H. Dai, B. Liu, Y. Chen, and Y. Wang. On pending interest table in named data networking. In *ANCS '12*, Austin, TX, USA, Oct. 2012.

[4] Huichen Dai, Yi Wang, Jindou Fan, and Bin Liu. Mitigate ddos attacks in ndn by interest traceback. In *NOMEN '13*, Turin, Italy, Apr. 2013.

[5] D.Rossi and G. Rossini. Caching performance of content centric networks under multi-path routing (and more). *Technical Report, Telecom Paris Tech*, 2011.

[6] P.Th. Eugster, P. A. Felber, R. Guerraoui, and A. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys*, 35:114–131, 2003.

[7] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard. Networking named content. In *CoNEXT '09*, Rome, Italy, Dec. 2009.

[8] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard. Voccn: Voice-over content-centric networks. In *ReArch '09*, Rome, Italy, Dec. 2009.

[9] S. Loreto, P. Saint-Andre, S. Salsano, and G. Wilkins. Known issues and best practices for the use of long polling and streaming in bidirectional http. *RFC 6202*, Apr. 2011.

[10] PARC. Ccnx technical documentation. *http://www.ccnx.org/*.

[11] D. Perino and M. Varvello. A reality check for content centric networking. In *ICN '11*, Toronto, Canada, Aug. 2011.

[12] P.Gasti, G. Tsudik, E. Uzun, and L. Zhang. Dos & ddos in named-data networking. *arXiv:1208.0952*, Aug. 2012.

[13] Radware. Global application & network security report. *Annual Report*, 2011.

[14] O. Saleh and M. Hefeeda. Modeling and caching of peer-to-peer traffic. In *ICNP '06*, Santa Barbara, CA, USA, Nov. 2006.

[15] S.Arianfar, T. Koponen, B. Raghavan, and S. Shenker. On preserving privacy in content-oriented networks. In *ICN '11*, Toronto, Canada, Aug. 2011.

[16] A. Soldati. Telecom italia ip backbone and peering policies. In *Italian Peering Forum (PFI 2008)*, 2008.

[17] Telecom Italia S.p.A. Resoconto intermedio di gestione al 31 marzo 2012 (in italian). *http://1q2012.telecomitalia.com/attachments/telecomitalia2012q1it.pdf*, 2012.

[18] W. You, B. Mathieu, P. Truong, J. Peltier, and G. Simon. Dipit: a distributed bloom-filter based pit table for ccn nodes. In *ICCCN '12*, Munich, Germany, July 2012.

[19] Haowei Yuan, Tian Song, and Patrick Crowley. Scalable ndn forwarding: Concepts, issues and principles. In *ICCCN '12*, Munich, Germany, July 2012.