Interest Packets Retransmission in Lossy CCN Networks and its Impact on Network Performance

Amuda James Abu
ajabu@cse.ust.hk

Brahim Bensaou
brahim@cse.ust.hk

Jason Min Wang
jasonwangm@cse.ust.hk

The Department of Computer Science and Engineering
The Hong Kong University of Science and Technology

ABSTRACT

When both interest packets and data chunks can be dropped on the way, due to network impairments, deciding who, the CCN router or the end-system, sets the timeout duration and retransmits the interest packet after a timeout are two key issues that affect the performance of the network. More specifically, they impact directly the occupancy of the pending interest table (PIT). The standard does not address these issues clearly and the typical CCN implementations (like CCNx) address them naively leaving room for further improvements. In lossy networks, if the router does not retransmit pending interests (no-rtx) the average PIT entry lifetime increases dramatically. Conversely, if the CCN router retransmits pending interests (rtx) periodically, it is not clear how frequently it should do so. In this paper we investigate the performance of the two types of routers in lossy networks, in the presence of both caching and interest aggregation. The study aims at shedding some light on how much performance improvement is achieved by one type of routers over the other. We also introduce a new method for estimating the PIT entry timeout that is shown to perform better than the currently used default method in CCN.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Network communications; C.2.3 [**Network Operations**]: Network management

Keywords

PIT lifetime, CCN, lossy networks, interest packet retransmission, network performance evaluation

1. INTRODUCTION

Content Centric Networking (CCN) [13] is one of the recently proposed architectures [2] for the future Internet aimed at reengineering the current Internet to support named-data communication. In CCN, contents (data) are divided into chunks and each chunk is singly identified using a hierarchical naming scheme. Communication in CCN is driven by the requesters of the data; more specifically, a receiver requests for chunks by sending interest packets, each containing the identifier of the requested chunk. This identifier is used in routing an interest packet towards the probable location of the chunk, then using the reverse path traversed by the interest packet, the data chunk is returned to the receiver when found. To improve the network performance, CCN recommends caching the data chunk at each intermediate node it traverses to possibly serve future requests for the same chunk.

The so-called Pending Interest Table (PIT) is one of the three fundamental data structures newly introduced in the CCN router design to enable a full functionality of CCN. An entry is created in the PIT for every interest packet forwarded upstream. The entry stores the incoming and outgoing interfaces for the interest packet. Having forwarded the interest upstream, the PIT entry manager waits for a period of time for the data to return. We refer in the sequel to this period of time as the PIT Entry Lifetime (*PEL*). In order to avoid transforming the PIT size into a bottleneck for the whole CCN, entries that are created are normally purged when either of the following events takes place: (i) the requested data is returned within the *PEL* and is forwarded downstream via the incoming interface(s) as indicated by the corresponding PIT entry; (ii) the *PEL* expires while the requested data has not arrived. To improve the performance of CCN, during the *PEL* of each PIT entry, subsequent interests that request for the same pending data are not forwarded but rather aggregated in the corresponding PIT entry, triggering an update of *PEL* [10].

According to the CCNx protocol technical documentation [10], *"A node MUST retransmit interest messages periodically for pending PIT entries"*. However there is no unanimous agreement on how often should this periodic retransmission be done. A common implementation is for a CCN node (router) to retransmit interest packet when it receives a new request for data chunk for which an entry exists in the PIT and the incoming interface through which the request is received also exists in the entry [22, 24]. The most obvious rationale to explain this approach is that a duplicate interest arriving on the same interface is probably a retransmitted interest that has timed out at the consumer. While this argument is fully justified for one-timers (unpopular content, requested by one user), it is however seldom the case in practice for popular content. We dub a router in which interest retransmission for active PIT entries an rtx node, and con-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ICN'14, September 24–26, 2014, Paris, France.

Copyright 2014 ACM 978-1-4503-3206-4/14/09 ...$15.00.

http://dx.doi.org/10.1145/2660129.2660149.

versely we call a router in which such interest retransmission is disabled a no-rtx node. Clearly the overall impact of using rtx routers is a higher traffic load in the network than using no-rtx routers, especially in networks where packets can be lost due to network impairments such as congestion, channel error or link failure. However, keeping network traffic load moderate is desirable without compromising the efficiency of content delivery in CCN. Early works on comparative analysis of switch-based packet-retransmission (switch-rtx) and end-host-based packet-retransmission (endhost-rtx) in lossy networks report that switch-rtx is better than endhost-rtx if the number of connected switches is large [12] while endhost-rtx is better than switch-rtx in high speed networks [4]. However, with the special features of the recently proposed CCN such as aggregation, caching and multi-path routing, it is not clear whether such conclusions are still valid. A comparative performance study of rtx and no-rtx CCN routers is still needed to shed more light on how promising are rtx and no-rtx CCN routers. This is the focus of our paper and we aim to fill this gap within the Information Centric Networking (ICN) research community.

Our contributions in this paper are twofold. First, we present a comparative study of rtx and no-rtx CCN routers in scenarios with and without packet losses. Our findings show that a rtx router achieves little or no performance improvement over no-rtx routers, except in the number of total interest satisfied. We also show that an increasing packet loss rate negatively affects both rtx and no-rtx routers. Secondly, we propose an efficient method for managing the PIT entry lifetime to avoid long reaction delays in purging stale PIT entries. Our method is based on data chunk response delays over a window of samples. Simulation results show that our method improves the performance of both rtx and no-rtx routers including the total number of interests satisfied.

The remainder of this paper is organized as follows: we present in Section 2 background information on CCN with a special focus on the Pending Interest Table management. The problem description is given in Section 3 while Section 4 describes our method for estimating the PIT entry timer. Simulation analysis and results are given in Section 5. Section 6 discusses some related work and we finally conclude the paper in Section 7.

2. BACKGROUND

In this section, we give a brief overview of Content Centric Networking, while focusing our description on the PIT issues.

2.1 Overview of Content Centric Networking

A CCN client injects an interest packet into the network to request for a portion of a content. The entire interest packet is a header including among other fields, *Identifier and interest lifetime fields.* Each intermediate CCN node on the path traversed by the interest packet checks its Content Store, if a copy of the requested data is in the cache, the data is returned along the reverse path traversed by the interest packet. Otherwise, it checks the PIT if an entry has earlier been created for the requested named data chunk. If yes, the interest is aggregated in this PIT entry. Otherwise the interest is prepared for forwarding upstream by checking the Forwarding Information Based (FIB) and if at least one outgoing interface exists in the FIB for this interest, then an entry is created in the PIT and the interest is forwarded. Otherwise, the interest is simply discarded.

If the requested data is not cached in any of the intermediate nodes, the interest packet will eventually arrive at the content producer/custodian (the original source) of the data. The producer forwards the corresponding data downstream. When delivering the data towards the requester, each intermediate node that receives the data chunk caches it in its content store to serve future requests for the same data, if the caching policy so permits.

2.2 Pending Interest Table

The Pending Interest Table in a CCN router does not only keep track of interests forwarded upstream towards a data source but also stores the interfaces (also called faces) on which the interests have been received. An entry is created in the PIT for each interest forwarded upstream and is deleted either when the requested data chunk is returned and forwarded downstream or when the PIT entry lifetime expires. Multiple requests for the same interest name are not forwarded but rather aggregated and all the faces on which all instances of the same interest have been received are stored. The stored faces are used to drive the forwarding of the returned data towards the requesters.

Specifically, a PIT entry contains an interest name, a list of incoming faces on which all interest instances for the same name have been received and a list of outgoing faces through which the interest has been forwarded. Each incoming face j, $j = 0, ..., N$ is associated with a lifetime t_j, with N being the number of faces stored in the PIT entry. The PEL is denoted as t_{PIT}. Time t_j is refreshed whenever a new interest for the same name is received on the same face j. Denote the interest for a data with name A received on face j by I_{Aj}. If t_j expires before the requested data chunk returns, then face j is removed from the PIT entry. If this continues all faces are eventually removed, provided the requested data has not returned. The removal of the last face from the entry triggers the removal of the entry from the PIT. Therefore the lifetime of a typical PIT entry for a given data chunk A is $t_{PITj} = \max_{i=1,...,N}\{t_i\}$ where t_{PITj} is the current time plus the lifetime of the interest arriving through face j.

2.3 CCN PIT entry timer

After waiting for a reasonable amount of time, each entry created in the CCN PIT must be purged if the requested data chunk fails to return before the PEL timer expires. As such, the timer estimation remains an important factor to consider as it affects the performance of the network. Too short a timer, leads to redundant unnecessary retransmissions of interests and as a result of data. Too long a timer, makes the network response to interest or data losses sluggish and thus reduces the network throughput. In the technical documentation of CCN protocol [10], no clear directive is given as to the default setting of the PIT entry lifetime. However, in the source code of CCN protocol (lines 4025–4045 and 4176–4274) [9], a node's PIT entry inherits its lifetime from the lifetime of interest packets that arrive at the node. As suggested in [10, 3] applications using CCN protocol should be the main entity that decides the value of interest lifetime. This value is chosen depending on the properties of the application (delay sensitive or insensitive). The PIT entry is set and refreshed as specified in [10] and described in Section 2.2.

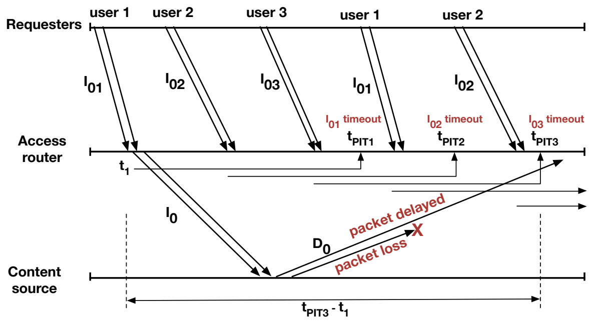


Figure 1: A scenario showing 3 CCN users sending interest packets for the same content at the rate of 2 interests per unit time to the content source via an access router, demonstrating how the PIT entry lifetime is updated in a CCN router when packets are lost or delayed

3. PROBLEM DESCRIPTION

3.1 System model

Consider the example of Figure 1 where the definitions of I_{Aj}, I_A and t_{PITj} are the same as defined in Section 2.2, D_A is the data requested by interest I_A. In Figure 1 note that there could be $m \geq 0$ intermediate nodes between the access router and the content source. With the current method used for setting and updating the PIT entry lifetime [10], the number of entries in the PIT and consequently the number of pending interests may increase dramatically, resulting in requesters waiting longer to download content, in lossy networks where both packet types or interests can be lost or delayed upstream. However enabling the periodic retransmission of interest packets by the intermediate node may eventually help retrieve the requested data chunk before the PEL expires, at the cost of increased network traffic load especially if interests are forwarded over all available output faces as suggested in CCN.

Focusing on the interest direction, assume all arriving interest packets with cache (content store) misses are processed by the PIT and there are always available outgoing interfaces in the FIB to forward interests upstream. Denote by μ the average rate of forwarding interests upstream and by τ, the average duration from the time one PIT entry is created until this PIT entry is purged. Furthermore, let the fraction of interests that are satisfied from the cache be h_{cs} and denote by h_{pit} the fraction of interests that are processed by the PIT but do not require forwarding upstream because of interest aggregation. Simply put, h_{cs} is typically the cache hit rate while h_{pit} is the PIT hit rate when looking for an entry that matches a given interest. Given the above notations, the average rate of forwarding interests upstream can be written as

$$\mu = (1 - h_{cs})(1 - h_{pit})\lambda$$

where λ is the average arrival rate of interest packet at a CCN node. Note that, due to aggregation, only the interests for which new PIT entries are created are forwarded upstream. Furthermore, a PIT entry that is created is purged on average after τ when the corresponding data has arrived (assuming no losses). Consequently the average rate at which entries are created in the PIT is also equal to μ, and the average number of entries in the PIT, can be simply obtained by Little's theorem, as

$$\Gamma = \mu \times \tau = (1 - h_{cs})(1 - h_{pit})\lambda\tau. \qquad (1)$$

From (1), we can make a few observations: Γ can be kept small if we can devise a good caching mechanism to increase h_{cs} or increase h_{pit}. However due to limited router memory, high caching dynamics and the ubiquitous caching used in CCN, cache hit rates h_{cs} in intermediate nodes (not including the first hop caching node) is known to be very small [7, 19] especially for popular contents. Furthermore, the dominance of one-timer contents in most content distribution networks implies a very small value of h_{pit} [7], resulting in an increased value of Γ. In addition, the actual number of entries in the PIT depends on several factors such as the loss rate, λ and τ. The instantaneous number of entries in the PIT oscillates around Γ. Interest packets could have been dropped due to the fluctuation of the PIT occupancy before the mean stabilized at Γ. As such, given a PIT size, P, on a 90% quantile for example, we can avoid dropping interest packets by keeping the probability that the PIT is full very small.

3.2 PIT sizing problem

As new interests for data chunks arrive at a CCN router, entries are created in the PIT for all interest packets forwarded upstream. When the PIT is saturated, i.e., when new PIT entries cannot be created as the PIT memory is full, newly arrived interests at a router are not inserted into the PIT but discarded, which may degrade the performance of the network. To avoid such PIT congestion, several approaches are worth investigating; the most obvious being to conduct a PIT dimensioning study via queueing theory. A good PIT size is one that almost always (e.g., with probability 0.99) has room for new entries. This implies that the probability that the PIT is full when a new request arrives at a CCN router should be kept small. To achieve this a fully fledged queueing model has to be designed. As such, one needs to characterise accurately λ, τ, h_{cs} and h_{pit}, which in turn depend on several non-trivial unknowns and complicated factors such as the caching/replacement policy, the traffic spatial and temporal distribution, the routing, the filtering effect due to interest aggregation, and so on. The work in [21] provides a rough estimate of a typical PIT size but the analysis therein does not consider requests aggregation (i.e., assumes $h_{pit} = 0$) among others.

Heuristic approaches may be adopted to avoid the complication of the modelling. For example a router may apply cache replacement policies to the PIT as well, by removing the longest-lived entry from the PIT to accommodate the new one. Such approach needs further investigation.

3.3 Interest packet retransmission: Frequency and timer estimation

Now consider the case where requested data never returns due to packet loss or link failure along the path towards the data source[1]: in this case, $\tau \rightarrow PEL$, and the requesters retransmits interests after the expiration of a given timer. No recommended method for computing the timer has yet been proposed, but [22] for example adopts a TCP-like RTO

---

[1] Here we define a data source as either the data producer/custodian or the node that has a cached copy of the data in its Content Store.

while [3] uses the interest lifetime. To make a CCN requester proactive we follow the method used in [22].

A CCN node must periodically retransmit interest packets for data chunks it has not yet received. The frequency of the interest retransmission is not specified in [10] at the time of writing of this paper, and a candidate approach is to retransmit only when the node receives a retransmitted interest from an Interface that is already stored in the PIT entry for the same data name. A problem with this strategy is an increased traffic load in the network especially when the requested data chunk experiences some delay (queuing and/or propagation) and the FIB entries are associated with multiple output interfaces per entry. Another problem is the unnecessary retransmissions caused by the arrival of interests for popular contents, thus making a misbehaving/misconfigured CCN consumer stage a DoS attack of interest-request retransmissions. This is in contrast to one advantage of built-in DoS resilience (DoS-ed interests are not always forwarded) claimed for CCN.

In addition we consider an alternative strategy where an intermediate CCN node does not retransmit interest packet even when it receives retransmitted interests from the requester. It rather waits for a period of *PEL* for the requested data to arrive. If the requested data is not received before PEL expires the PIT entry is purged. As the requester always retransmits interests for the data chunks that it has not received within a timer, the intermediate node will eventually receive the retransmitted interests, create new PIT entries for these interests and forward them upstream. In contrast, with this strategy, the traffic load in the network is expected to be moderate.

How these two strategies perform comparatively in CCN is not yet reported, and in this paper we use a packet-level simulation to study the two strategies, highlighting their performance merits and demerits.

4. A DYNAMIC PIT ENTRY TIMER

Given the existence of caching in CCN, the rate at which entries are purged from the PIT depends on where the requests are being satisfied. It also depends on whether or not packets experience additional delay (queueing and processing), as well as the presence of multiple paths for forwarding interests. Thus a good value of the PIT entry lifetime should take into account the aforementioned factors. To this end, we consider an alternative approach for setting the PIT entry lifetime.

Denote the time from which a given node forwards an interest upstream to the time the node receives the requested data as the response delay *RD*. Ideally the PIT entry lifetime should commensurate with the maximum response delay observed for a content over all the requests sent for the same content from a router to all the nodes from which the content is retrieved (including in-network caching). In view of this, we propose to set the PEL based on the maximum response delay over all samples of data packets received in a given window of samples. See Figure 2. We avoid any fluctuation in the measured response delay due to in-network caching by taking the maximum from all the values observed. Although a router R_1 estimates *RD* for each of the returned data chunk, only the maximum of all the *RD*s is stored implying that a router keeps only the *RD* of the content producer if the content is a one-timer, or that of the farthest caching node if the content is popular. The entry lifetime for a given content C is set as follows: Initially $\tau^C = 1$s. After receiving the first data chunk for C, $\tau^C = RD_1$. For subsequent data chunk i for the same content C received within an interval of time γ, $\tau^C = RD_i$ if $\tau^C < RD_i$ where $i > 1$. We use $\gamma = 60$s in our experiments. To ensure the freshness of τ^C, we set $\tau^C = RD_{min} + (RD_{max} - RD_{min})\delta$ every 60s, $\delta = 0.5$. In the next interval, τ^C is compared with subsequent RD_i and updated accordingly. τ^C is discarded after an idle period of τ^C. The idle period for a given content is the period in which no request for the content is received. The values of γ and δ used in our experiment are the same as those used in [17] and [15], respectively. However, we make use of the values of γ and δ at the routers while the works in [17] and [15] use them at the end-hosts.

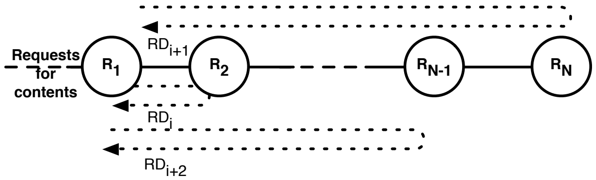


Figure 2: Estimating PIT entry lifetime from all requests for a given content

The resulting PIT entry lifetimes adapt well to the dynamics of the network upstream. For scalability reason, τ^C is maintained per content as opposed to per interest packet.

5. PERFORMANCE EVALUATION

We discuss in this section the simulation scenario under consideration in our study including simulation setup in ns-3. Simulation results showing the advantages and disadvantages of an interest-retransmission enabled/disabled CCN router in both lossy and lossless scenarios are presented.

5.1 Scenario description

We consider a parking-lot topology as shown in Figure 3. Access routers $R1$ and $R2$ receive requests from many users. Users' requests follow a Zipf probability distribution with Zipf's skewness parameter α. In the event that requests cannot be satisfied by in-network caches, all requests from access router $R1$ can only be satisfied by content producer $P1$ while all requests from $R2$ can only be satisfied by producer $P2$. To introduce traffic mixing and caching dynamics, we add CCN cross traffic sources and receivers. We believe that this topology is sufficient to capture the impact of caching and aggregation on the performance of the network. This is because the arrivals of traffic (interest packets) in one direction are capable of filling up the entries in the PIT. As such we do not consider complex topologies and bidirectional flows of interest packets.

We consider lossy scenarios where packets are dropped randomly on the link marked L_D in Figure 3. L_D actually represents multiple links connecting the content producers to the rest of the network. So randomly dropping packets on L_D does not necessarily mean that we drop a packet on the link directly connecting the content producer. For performance comparison we also consider lossless scenarios where no packet is dropped in the network. $R1$, $R2$, C_R1, C_R2 and C_R3 have relatively small cache sizes and thus requests are not satisfied by these nodes. Requests are forwarded us-

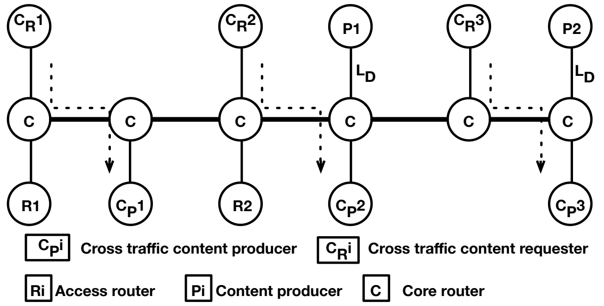


Figure 3: A parking-lot simulation model showing the access routers, content producers, cross traffic sources and receivers

ing the default CCN flooding and content request arrivals follow a Poisson process. All requests for the same content have the same interest lifetime. Consumers do not need to receive data for interest packets in flight before sending the next interest packets, but can retransmit interests for which data have not been received, after a timer expiration.

There are 2 groups of user requests and each group is associated with a content producer. For each group, including cross traffic, we use proWGen [5] to generate workloads for our experiments. Details are given in Section 5.2.

Performance metrics considered in this study include: the number of entries in the PIT (i.e., the number of pending interest entries in the PIT), the PIT entry actual lifetime, the fraction of requests satisfied by in-network caching including the content producer, the total number of interests satisfied, the data chunk response delay and the network traffic load.

5.2 Simulation setup

The scenario described in Section 5.1 was simulated in ns-3 network simulator using the ndn/ccn ns-3 modules [1]. We developed a custom application module that can use the workload generated by ProwGen in ns-3. Our algorithm for estimating PIT entry lifetime was also implemented in ndn/ccn ns-3.

All link delays are set to 20ms while the capacities of all thin links in Figure 3 are set to 200Mbps unless otherwise stated. Each thick link as shown in Figure 3 has a capacity of 500Mbps. Cross traffic sources and receivers run for the entire duration of the simulation. Each returned data chunk is 1500 bytes. For the characteristics of our workload, we use $\alpha = 0.96$, respectively 0.76, for user groups $R1$, respectively $R2$. For the cross traffic, α is in the range [0.6, 0.8]. The values of α used are within the values recommended in the literature. There are 60% one-timers and 40% unique contents. The mean file size for contents is 14KB. For both group, we generated 100,000 content requests (not at the chunk level). Note that the number of data chunks per content depends on the file size of the content. We enable in-network caching with LRU replacement policy using the default ubiquitous caching policy of CCN. We set the interest lifetime to 4s, the default value in CCNx.

We refer to a network with packet loss as a lossy scenario while a network without packet loss as lossless scenario. In each of the two scenarios considered in our simulations, we use rtx routers in one simulation set and no-rtx routers in another set. We observed similar results for access routers $R1$ and $R2$. Thus we report simulation results for $R1$ in the following sections. The default values for the parameters that we vary in our experiments are the average request rate (5000 interest/s), the packet loss rate (0.01), and the cache size per router (500 chunks). Next, we present simulation results with 95-percent confidence intervals.

5.3 PIT occupancy and entry actual lifetime

Figure 4 shows the CDFs of the number of entries in the PIT for different loss rates using rtx and no-rtx. It can be observed in Figure 4a that most of the time the number of entries in the PIT remains at 900, 1100, 1750 and 2700 for 0.001, 0.01, 0.05 and 0.1 loss rates respectively. This large variation in the PIT size is a consequence of the entries staying longer than necessary in the PIT as corroborated by the PIT entries lifetime CDF shown in Figure 5a. More importantly, the figures clearly show that rtx does not offer much improvement over no-rtx. In contrast if we replace the PEL calculation algorithm with our approach, as shown in Figure 4b, about 99% of the time the number of PIT entries is less than 1000 in all the cases considered, regardless of the loss rate. This shows that using our method for estimating *PEL* prevents the size of the PIT from becoming a bottleneck in a CCN infrastructure as more packets are dropped.

Figure 5 shows the CDFs of the PIT entry actual lifetime for rtx and no-rtx using different packet loss rates. The actual lifetime is the time between the entry creation and its deletion (because of timeout or interest satisfaction by a data chunk). Specifically, the data in Figure 5a indicates that about 30%, 50%, 80% and 98% of the PIT entries stay longer than 200ms[2] in the PIT for 0.1, 0.05, 0.01 and 0.001 packet loss rates respectively. Note that about 10% of the entries stay more than 4s for 0.1 loss rate. These are entries that contain aggregated interests for the same name resulting in the *PEL* being refreshed for every aggregation. Similarly, rtx's improvement over no-rtx's remains insignificant. However, with our method for calculating the *PEL*, most of the entries' actual lifetimes are within the maximum response delay from the content source as shown in Figure 5b. Due to aggregation only a few entries stay beyond the maximum response delay.

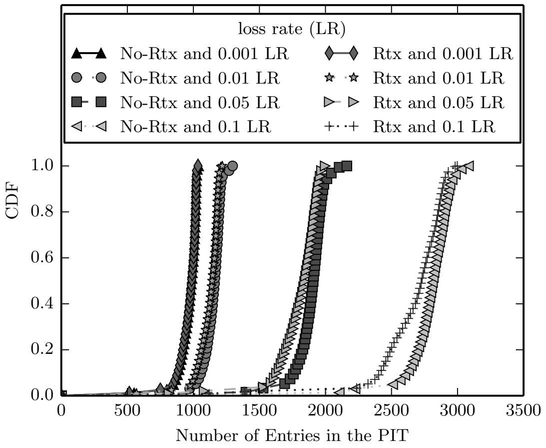
We also present in Figure 7 the average number of entries in the PIT for different cache sizes and request rates. Figures 6a and 6b show the average number of entries in the PIT with varying average request rate in lossy (0.01 loss rate) and lossless scenarios. Both figures show that the average number of entries grows as we increase the average request rate. However, with our method for estimating the *PEL* it can be observed in Figure 6b that we can achieve a similar performance for lossy and lossless scenarios. Caching indeed plays a key role in CCN. Figures 7a and 7b further assert this as the average number of entries in the PIT decreases with increasing cache size. In lossy scenario rtx and no-rtx differ in performance with cache size up to 5000 chunks but converge at relatively large cache sizes (this represents the case where there is always room for caching data chunk, see Figure 7a). Similarly Figure 7b shows the same performance for lossy and lossless scenarios as well as for rtx and no-rtx routers.

5.4 Network traffic load

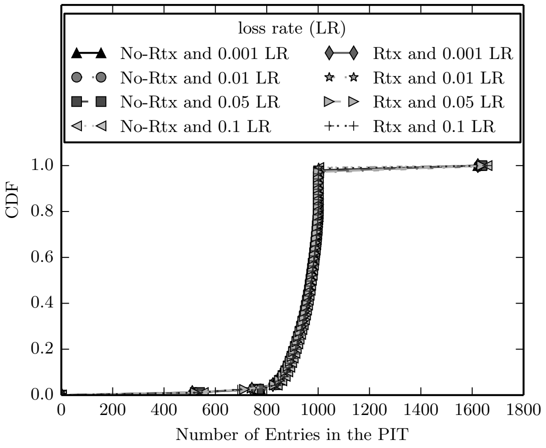
In this section we study the total traffic load injected into the network for different loss rates and average request rates

---

[2] 200ms is the end-to-end RTT in our network.

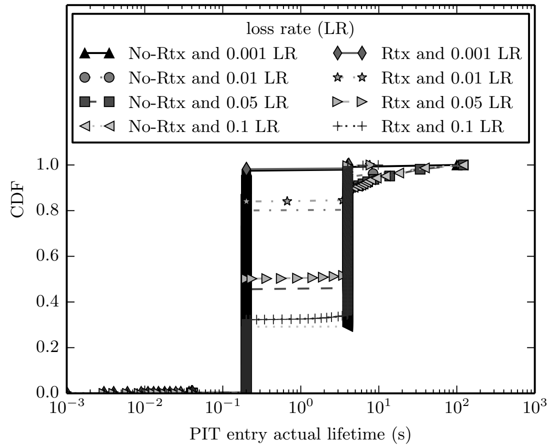


(a) Using a fixed 4s *PEL*

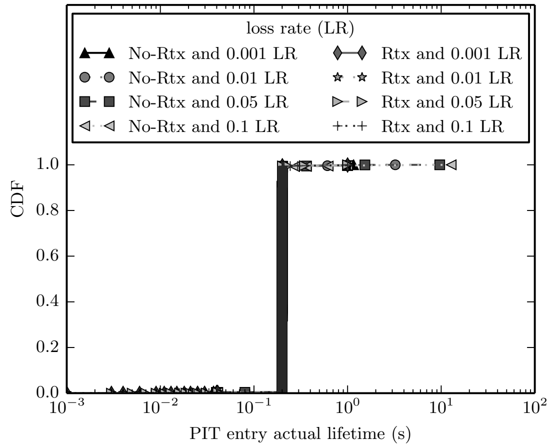


(b) Alternative method for estimating *PEL*

Figure 4: CDFs of the number of entries in the PIT



(a) Using a fixed 4s *PEL*



(b) Alternative method for estimating *PEL*

Figure 5: CDFs of the PIT entry actual lifetime (x-axes in log scale)

in Figure 8a and 8b respectively. Although both rtx and no-rtx increase the total network traffic load as we increase the packet loss rate, they differ in that rtx incurs about 3×10^4 additional traffic load. That is, not only rtx is shown in the previous section to not improve the PIT occupancy much, it also increases the traffic load. In Figure 8b both rtx and no-rtx show nearly similar performance up to 5000 interest/s average request rate, but diverge for rates greater than 5000 interest/s. Ideally such divergence should not be present when there is no packet loss. However, note that the retransmission timer estimated by the requester may expires prematurely triggering a retransmission for data chunks that are still in transit. The results described in Figures 8a and 8b show that no-rtx is better than rtx in terms of traffic load.

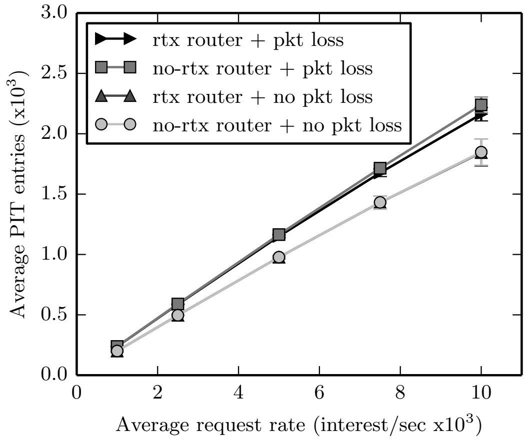
5.5 Interests satisfied

In Figure 9, we show the total number of interests satisfied for different packet loss rates using rtx and no-rtx routers. The total number of interests satisfied can be observed to decrease as we increase the loss rate. Despite the no-rtx method achieving less traffic load compared to rtx, the former suffers from a lower throughput (number of satisfied interests). Note that there is not much performance improvement achieved by rtx over the performance of no-rtx in the presence of little or no packet losses. The improvement becomes significant as the rate of packets drop increases. More importantly, using our method for calculating the *PEL* we can improve the performance of no-rtx making it nearly the same as rtx as shown in Figure 9b.

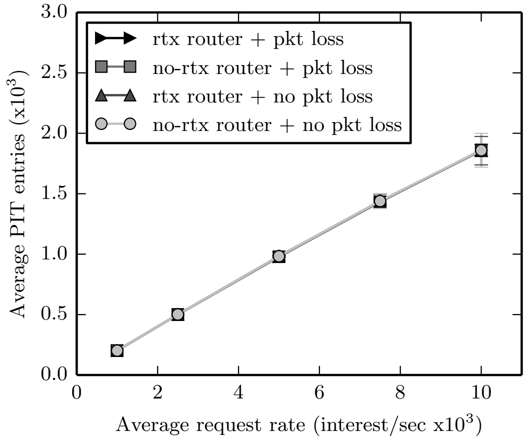
5.6 Data chunk response delay and hop count for satisfied interests

Figure 10 further demonstrates that rtx offers no much improvement over no-rtx in terms of data chunk response delay for different packet loss rates. Both rtx and no-rtx are affected by increasing loss rates. When using our method for estimating the *PEL* the performance of rtx and no-rtx becomes similar as shown in Figure 10b.

Figure 11 shows the fraction of requests for popular contents served by in-network caches including the content producer for different cache sizes. Similarly, rtx does not achieve

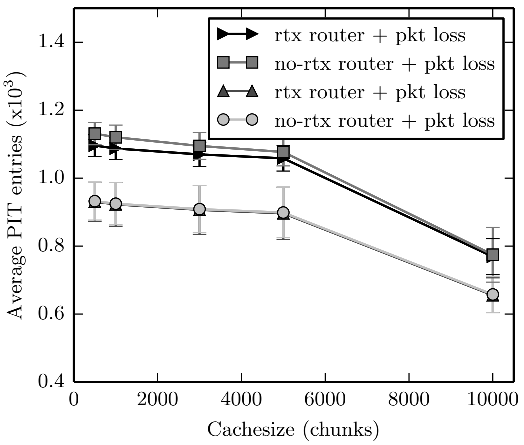


(a) Using a fixed 4s *PEL*

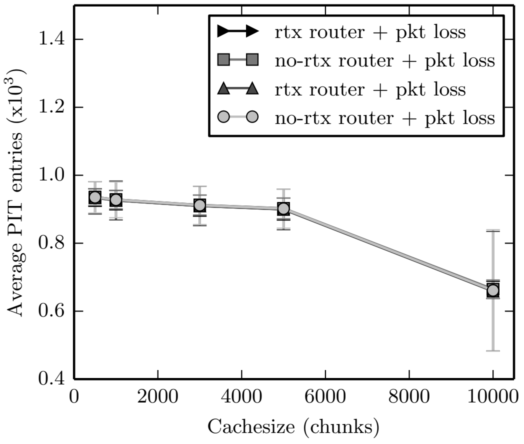


(b) Alternative method for estimating *PEL*

Figure 6: Average number of entries in the PIT with different average request rates



(a) Using a fixed 4s *PEL*



(b) Alternative method for estimating *PEL*

Figure 7: Average number of entries in the PIT with different cache sizes

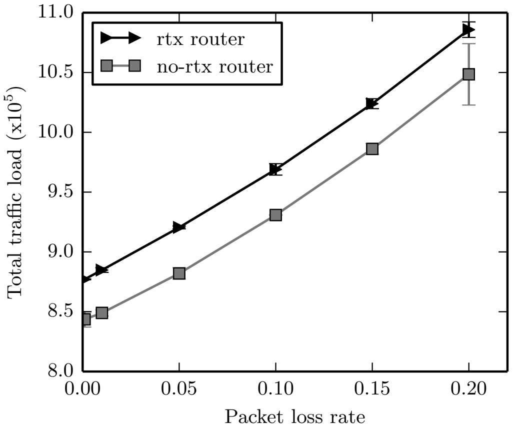
much performance improvement over no-rtx in terms of bringing the contents closer to the requesters. A better performance can be observed with relatively large cache size than small cache size. Both rtx and no-rtx benefit from this. Using our method for calculating *PEL* achieves similar performance with using a fixed *PEL* as shown in Figures 11b .

5.7 Additional remarks

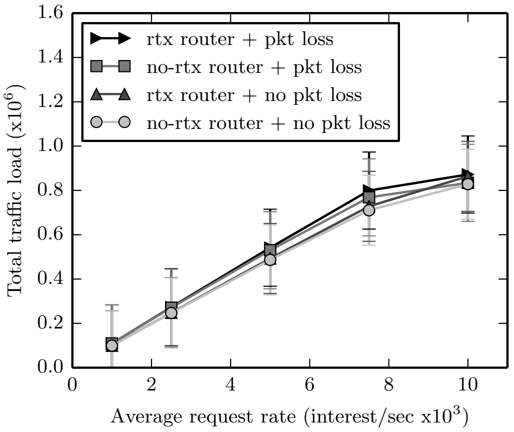
In addition to preventing the size of the PIT from becoming a bottleneck in CCN, our proposed method for PIT entry management does not require the exchange of control message(s) such as Interest NACKs between neighbour nodes to trigger the purging of a PIT entry if upstream nodes fail to deliver the requested data downstream. Similar to [8] we believe that this is desirable as exchanging control messages between neighbour nodes may require message prioritization and incur additional overhead. Besides, the method has the virtue of being very simple to implement. A potential distributed denial of service (DDoS) attack has been demonstrated by Virgilio *et al* [18] where artificial interests with relatively large lifetimes are crafted by malicious users with the goal of occupying the available PIT memory in CCN routers. Our proposed method for estimating the PIT entry timer is capable of minimizing the impact of such PIT overloading as it is independent of the interest lifetime.

The performance study of rtx and no-rtx presented in this paper has been carried out using a parking-lot topology. Although the insights gained from the study are of great importance in understanding how rtx and no-rtx perform in CCN, there still remains many other aspects to consider such as other network topologies as well as the impact of different content sizes, transport strategies and distance between a CCN consumer and the content provider.

Our proposed method can help reduce the PIT occupancy for rtx and no-rtx in CCN by measuring the response delay for every data received per content. Yet there are other mechanisms that can achieve the same goal. One approach is to make one-timer content requests bypass both the content store and the PIT while popular content requests pass



(a) Different packet loss rates



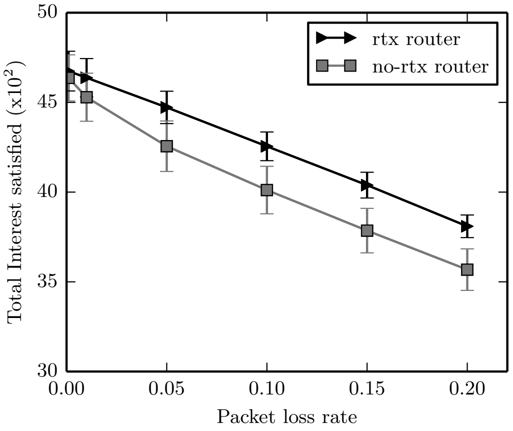
(b) Different average request rates

Figure 8: Total traffic load from $R1$

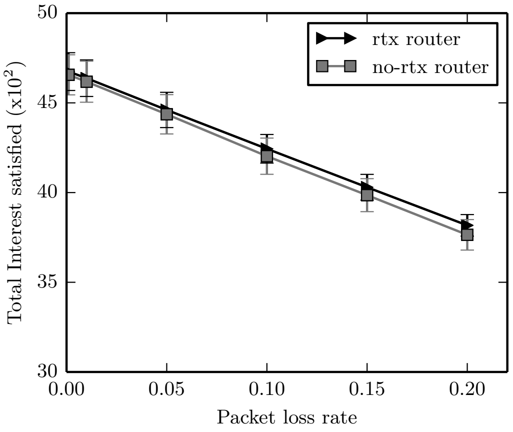
through the content store and the PIT [16]. With the dominance of one-timer contents in most content distribution networks, the load on the PIT as well as the content store can be greatly reduced, thus reducing the PIT occupancy. Nevertheless this method requires the classification of traffic into one-timer and popular content which is a characteristic that changes in both time, space and may be different from one router to another (due to the well known filtering effect in CCN caused by Interest aggregation). Another interesting approach to explore is regulating the rate of interest packets transmission via flow control.

6. RELATED WORK

CCN has recently drawn a significant attention from the networking community. In particular, efforts have been directed towards exhaustive performance evaluation of CCN caching [19, 20] (Content Store), routing and forwarding (FIB and PIT), security (data integrity and confidentiality) and transport (congestion control) under different network conditions. For example, how to efficiently manage the PIT entry lifetime and the size of the PIT is still an ongoing research issue.



(a) Using a fixed 4s *PEL*

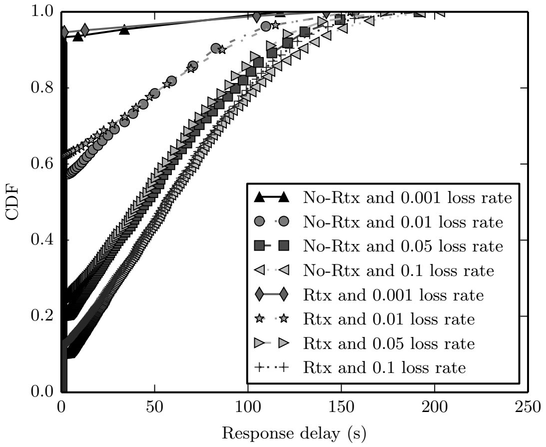


(b) Alternative method for estimating *PEL*

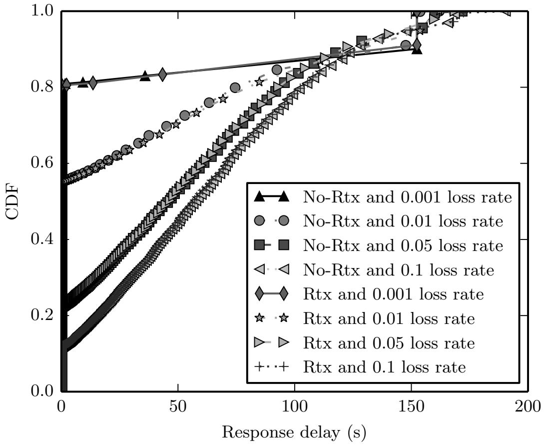
Figure 9: Total number of interests satisfied with different packet loss rates

The work by You *et. al.* [23] proposes a new implementation of the PIT based on Bloom Filter to reduce the memory space required to implement the PIT. Evaluation results show that the proposed PIT architecture achieves a significant reduction of the memory space. A tree-like implementation of the PIT is also proposed in [11] to shrink the size of the PIT. Another proposed solution to reduce the size of the PIT employs the idea of traffic differentiation where non-shareable traffic (one-timer content) bypasses both the content store and the PIT while shareable traffic (always cached content) follows the conventional CCN processing [16]. None of these approaches has addressed the impact of the PIT entry lifetime on the PIT size (also known as the number of pending downloads in [18]). A more recent work by Virgilio *et. al.* [18] compares via simulation the existing PIT architectures (SimplePIT, HashedPIT and DiPIT) under heavy traffic load. The analysis results show that all three architectures are adversely affected, making thus the case for the need of a better PIT entry lifetime management.

The PIT entry lifetime determines when the entry can be purged in the event the data takes too long to return (queue-



(a) Using a fixed 4s *PEL*

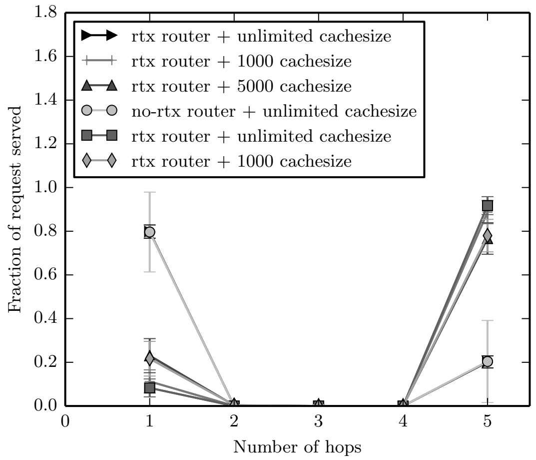


(b) Alternative method for estimating *PEL*

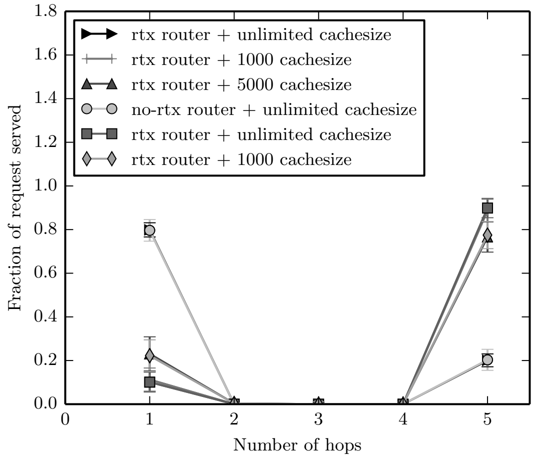
Figure 10: CDFs of the data chunk response delay

ing delay) or never returns (packet loss) due to congestion in the network. In view of this, Yi *et. al.* [22] proposed a mechanism where intermediate nodes can send a NACK packet to downstream neighbour nodes if an interest cannot be satisfied. On receiving the NACK packet, the node either retransmits the request via another outgoing interface if one exists. Otherwise it purges the corresponding PIT entry. In addition to the drawbacks reported in [8], potential issues with this method are threefold: first upstream nodes's *PEL* may take a long time to expire or never expire when each of the intermediate nodes or the last hop node always receive multiple requests for the same yet-to-return data, resulting in the *PEL* being refreshed/extended [10]. Second, the NACK packet may itself be lost due to congestion. Third, consumers may no longer be interested in receiving an earlier requested data resulting in an inefficient usage of network resources.

Unlike existing works such as [18, 6] that use fixed value for the *PEL*, Kazi and Badr propose a novel method for estimating the *PEL* at routers and the interest packet timeout at receivers [14]. The estimates depend on the queue size



(a) A fixed 4s *PEL*



(b) Alternative method for estimating *PEL*

Figure 11: Fraction of requests satisfied by content sources including in network caches

at the most congested node in the network, the processing and propagation delays and the network diameter. However, this approach assumes both interest and data traverse the full diameter of the network. This assumption is indeed not realistic as content sources may be in practice anywhere in the network, requiring actually a more robust and adaptive method to estimate the timer.

7. DISCUSSION AND CONCLUSION

This paper presented a performance study of CCN with rtx and no-rtx routers, shedding more light on how much performance improvement is achieved by one approach over the other.

Our findings suggest that interest retransmission at a CCN router does not appear to be a good design idea. While it has the virtue of speeding up the recovery of delayed or lost packets in the network, it turned out to increase the network traffic load without reducing the number of entries in the PIT. This may eventually result into the PIT memory becoming a performance bottleneck. Since CCN does not specify any method of retransmitting lost packets we have considered in this paper one approach that intuitively leads to a small additional overhead: an rtx router retransmits an interest only when it receives a retransmitted interest from an Interface that is already stored in the PIT entry for the same data name. Other frequencies of retransmission are expected to yield similar or worse results.

In addition, results from our study reveal that the default method used for setting the PIT entry timer is not a good design choice as it is oblivious of the network conditions such as packet loss and delay and can be subject to malicious attacks. Notably, it has the potential of bloating the PIT making it a performance bottleneck. To address this issue we introduced a novel adaptive method to estimate the PIT entry timer that relies on the data chunk response delays observed over a window of samples. Simulation results show that our approach outperforms the currently used method especially in lossy networks. Nevertheless, our proposed solution still needs additional performance analysis considering the impact of different factors such as the number of hops between a content requester and producer, different network topologies, and so on. We leave this as future work.

8. ACKNOWLEDGMENTS

This work was in part supported by HKPFS (PF11-02174) and UGC grant FS-GRF14EG24.

9. REFERENCES

[1] A. Afanasyev, I. Moiseenko, and L. Zhang. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN, October 2012.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of information-centric networking. *IEEE Comm. Mag.*, 50(7):26–36, 2012.

[3] S. Arianfar, P. Sarolahti, and J. Ott. Deadline-based resource management for information-centric networks. In *Proceedings of the 3rd ACM SIGCOMM workshop on ICN*, pages 49–54, 2013.

[4] A. Bhargava, J. F. Kurose, D. Towsley, and G. Vanleemput. Performance comparison of error control schemes in high-speed computer communication networks. *IEEE Journal on Selected Areas in Communications*, 6(9):1565–1575, Dec 1988.

[5] M. Busari and C. Williamson. Prowgen: a synthetic workload generation tool for simulation evaluation of web proxy caches. *Comp. Net.*, 38(6):779 – 794, 2002.

[6] G. Carofiglio, M. Gallo, and L. Muscariello. Icp: Design and evaluation of an interest control protocol for content-centric networking. In *Proceedings of the 2012 IEEE INFOCOM Workshops*, pages 304–309, Orlando, FL, Mar. 2012.

[7] G. Carofiglio, M. Gallo, and L. Muscariello. On the performance of bandwidth and storage sharing in information-centric networks. *Computer Networks*, 57(17):3743 – 3758, 2013.

[8] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang. Optimal multipath congestion control and request forwarding in information-centric networks. In *Proceedings of the 21st IEEE ICNP*, Gottingen, Germany, Oct. 2013.

[9] CCNx. Source code for ccnx 0.8.2. `http://www.ccnx.org/software-download-information-request/download-releases/`, Accessed on 1st May 2014, Apr. 2014.

[10] CCNx Protocol. Technical documentation for ccnx 0.8.2. `http://www.ccnx.org/releases/ccnx-0.8.2/doc/technical/CCNxProtocol.html`, Accessed on 30th May 2014, July 2013.

[11] H. Dai, B. Liu, Y. Chen, and Y. Wang. On pending interest table in named data networking. In *Proceedings of the Eighth ACM/IEEE ANCS*, pages 211–222, New York, NY, USA, 2012.

[12] M. Irland and G. Pujolle. Comparison of two packet-retransmission techniques (corresp.). *IEEE Trans. Inf. Theory*, 26(1):92–97, Jan 1980.

[13] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proceedings of CoNEXT '09*, pages 1–12, New York, NY, USA, 2009.

[14] A. Kazi and H. Badr. Some observations on the performance of ccn-flooding. In *Proceedings of ICNC*, pages 334–340, Honolulu, HI, Feb. 2014.

[15] A. Kuzmanovic and E. W. Knightly. TCP-LP: Low-priority service via end-point congestion control. *IEEE/ACM Trans. Netw.*, 14(4):739–752, Aug. 2006.

[16] R. Ravindran, G. Wang, X. Zhang, and A. Chakraborti. Supporting dual-mode forwarding in content-centric network. In *Proceedings of the 2012 IEEE ANTS*, pages 55–60, 2012.

[17] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind. Low Extra Delay Background Transport (LEDBAT). IETF RFC 6817, Dec. 2012. `http://www.rfc-base.org/txt/rfc-6817.txt`.

[18] M. Virgilio, G. Marchetto, and R. Sisto. Pit overload analysis in content centric networks. In *Proceedings of the 3rd ACM SIGCOMM Workshop on ICN*, pages 67–72, New York, NY, USA, 2013.

[19] J. Wang and B. Bensaou. Progressive caching in ccn. In *Proceedings of the 2012 IEEE GLOBECOM*, pages 2727–2732, Anaheim, CA, Dec. 2012.

[20] J. M. Wang, J. Zhang, and B. Bensaou. Intra-as cooperative caching for content-centric networks. In *Proceedings of the 3rd ACM SIGCOMM Workshop on ICN*, pages 61–66, 2013.

[21] M. WÃd'hlisch, T. C. Schmidt, and M. Vahlenkamp. Backscatter from the data plane âĂŞ threats to stability and security in information-centric network infrastructure. *Computer Networks*, 57(16):3192–3206, 2013.

[22] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang. A case for stateful forwarding plane. *Computer Communications*, 36(7):779–791, 2013.

[23] W. You, B. Mathieu, P. Truong, J. Peltier, and G. Simon. Dipit: A distributed bloom-filter based pit table for ccn nodes. In *Proceedings of the 21st ICCCN*, pages 1–7, 2012.

[24] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang. Named data networking. Technical Report NDN-0019, NDN project, Apr. 2014.