Pending Interest Table Sizing in Named Data Networking

Giovanna Carofiglio
Cisco Systems
gcarofig@cisco.com

Massimo Gallo
Bell Labs, Alcatel-Lucent
massimo.gallo@alcatel-lucent.com

Luca Muscariello
Orange Labs Networks
luca.muscariello@orange.com

Diego Perino
Bell Labs, Alcatel-Lucent
diego.perino@alcatel-lucent.com

ABSTRACT

Named Data Networking (NDN) has emerged as a promising candidate for shifting Internet communication model from host-centric to content-centric. A core component of NDN is its stateful forwarding plane: Content Routers keep track of pending requests (Interests) storing them in dedicated tables at routers (Pending Interest Tables). A thorough analysis of PIT scalability is fundamental for deploying NDN as a whole and questions naturally arise about memory requirements and feasibility at wire-speed. While previous works focus on data structures design under the threat of PIT state explosion, we develop for the first time an analytical model of PIT dynamics as a function of relevant system parameters. We provide a closed form characterization of average and maximum PIT size value at steady state. We build an experimental platform with high speed content router implementation to investigate PIT dynamics and to confirm the accuracy of our analytical findings. Finally, we provide guidelines on optimal PIT dimensioning and analyze the case of an ISP aggregation network with a trace-driven packet delay distribution. We conclude that, even in absence of caching and under optimal network bandwidth usage, PIT size results to be small in typical network settings.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network communications*

Keywords

Information-Centric Networking; Performance; Scalability.

1. INTRODUCTION

Internet architecture has been undergoing a tremendous transformation from a network of computers into an ubiquitous interconnection of data, things and ultimately people. Under the stress of a richer, mobile and dynamic user demand, Internet communication principles show their limitations and are today at the center of research investigations. Various future Internet proposals have emerged in the research arena, innovating the network architecture and its communication primitives around a content (or information) centric communication.

The most popular, [9, 18] CCN/NDN advocates a name-based communication, controlled by the end-user and realized via name-based routing and forwarding. Such connectionless communication model naturally takes advantage of multipath/multicast network capabilities and in-network caching. To this goal, NDN requires a stateful data plane. End -user name-based requests (Interests) are interpreted by NDN nodes and, when not satisfied by a local cache (Content Store, CS), forwarded according to the name-based FIB. NDN nodes keep track of the state of all pending Interests in dedicated data structures called PIT (Pending Interest Tables). The PIT is an essential element of NDN forwarding plane and is used to: (i) guarantee Data delivery to the requesting user(s) on Interests' reverse path, (ii) realize Interest aggregation, enabling native support for multicast and (iii) enable advanced forwarding features as hop-by-hop forwarding, loop detection, etc. The design of a router implementing the NDN data-plane (Content Router) is a challenging task. Name based FIB and CS are similar to today's router data structures (IP FIB and packet buffer), and their design and implementation is already a challenging research problem.

The PIT is a novel data structure and two main aspects require thorough investigation: (i) the feasibility of wire speed PIT implementation; (ii) the analysis of the PIT size evolution. Most of the previous works address ed feasibility issues [7, 16, 11, 17, 13], while PIT size analysis received little attention so far [7, 14]. However, none of the previous works analyze PIT dynamics as a function of content request workload and network conditions.

In this paper, we develop for the first time an analytical model of PIT dynamics under the assumption of a single PIT per node, and no a priori size or request lifetime limitations. Our traffic model assumes elastic flows only. Indeed, inelastic flows do not impact PIT dynamics when their rate is smaller than the bottleneck's fair rate, and are most likely stopped by the users otherwise. More in detail:

- We develop a fluid model of instantaneous rate, queue length and PIT size over time and derive a closed form characterization of average and maximum PIT size at steady state in a single bottleneck scenario.
- We extend our model to general network topologies and multi-bottlenecks scenarios.

ICN'15, September 30–October 2, 2015, San Francisco, CA, USA.

DOI: http://dx.doi.org/10.1145/2810156.2810167.

- We build an experimental testbed including high-speed content router and custom application layer NDN client, repository to analyze PIT dynamics and to assess model accuracy.
- We provide guidelines on optimal PIT dimensioning and analyze the case of an ISP aggregation network in presence of a realistic trace-driven packet delay distribution.

The conclusion of our study is that, even in absence of in-network caching and under optimal use of network bandwidth, PIT size results to be small in typical network settings. The remainder of the paper is organized as follow. Sec. 2 presents the related work. In Sec.3 we describe the NDN communication principles, and introduce our modeling framework. Analytical results are gathered in Sec.4. We carry out an experimental evaluation in Sec.5, with the twofold objective of analyzing PIT dynamics in realistic settings and assessing model accuracy. Sec.6 provides guidelines on PIT dimensioning based on our analytical tool. Finally, Sec.7 concludes the paper.

2. RELATED WORK

PIT feasibility has been studied for the first time in [7]. By analyzing an IP trace, authors provide an approximate quantification of PIT size, and propose Name Component Encoding as well as PIT placement on outgoing router linecards in order to reduce PIT size and accelerate management operations. Other PIT compression methods exploiting Bloom filters have been also proposed in [16, 11]. Authors of [17] provide design guidelines for high-speed PIT implementation: in order to limit memory requirements and access frequency content names are replaced with fingerprints. A performance comparison of encoding methods and hash-based methods can be found in [13]. Authors evaluate content router requirements under the assumptions of limited request lifetime and homogeneous network links with an average global RTT of 80 ms.

Motivated by the threat of PIT size explosion, authors of [12] propose a semi-stateless forwarding scheme for NDN. Their design propose to keep track of forwarded requests every d hops instead of every hop as in the original CCN/NDN proposal. PIT security analysis also received particular attention in some related works. In [15], authors derive an analytical model for denial of service attack detection. Threats to the PIT stability and security are also analyzed in [14] through analytical modeling and experimental evaluation. None of the previous work has focused on the analysis of PIT dynamics as a function of relevant system parameters. To the best of our knowledge, this is the first paper to characterize PIT dynamics and to investigate them means of experiments with a realistic platform and trace-driven packet delay distribution.

Other bodies of work in the literature are significant to this paper: buffer sizing in Internet routers and scalability of stateful schedulers. In particular dimensioning is made in [2] by using the Gaussian quantile of a core router queue occupancy created by a large number of bottlenecked flows. Flow table in [10] is sized by computing the distribution of the number of active flows in a per-flow scheduler, under dynamic workload. Our approach is similar to these works but we focus on PIT sizing instead of buffer or flow table dimensioning. We require a finer characterization of the queue evolution and its relation with the flow controller characteristics under dynamic setting which constitutes in itself a novel contribution.

3. PROBLEM DESCRIPTION

Our work builds on Named Data Networking assumptions [9, 18]. In the following we briefly revise NDN communication principles.

3.1 System description

NDN uses hierarchical names similar to URIs to address content. Content items are split into a sequence of chunks (or Data packets) uniquely identified by the content name plus the chunk identifier. Clients express per chunk requests (or Interests) to retrieve the complete content item and regulate the request rate via a window-based multipath congestion control algorithm. NDN nodes process incoming packets using the name carried in the packet header. Upon Interest packet reception, an NDN node first checks if the requested Data packet is stored in the local cache (Content Store in the NDN terminology). In this case, it sends the Data back through the requesting interface(s), satisfying the Interest. Otherwise, it checks if the PIT stores requests for the same ongoing Data request. If it is the case, it adds the requesting interface to the list of interfaces waiting for the Data packet and discards the Interest packet. Finally, if both CS and PIT lookups give negative response, the NDN node forwards the Interest to the next hop according to the name-based longest prefix match on the FIB and creates a corresponding PIT entry. If there is no match in the FIB, the Interest is discarded. When a Data packet arrives, the NDN node checks the PIT to retrieve the list of requesting interfaces and forwards the Data back to such interfaces. Also, the NDN node potentially stores the packet in the local CS for future reuse. The Data packet is discarded if there are no matching entries in the PIT.

3.2 Modeling framework

We model system dynamics at the timescale of content retrievals through a deterministic fluid model, where the discrete packet representation is replaced by a continuous one, either in space and in time. Similar fluid models have been adopted to analyze TCP/UDP dynamics (e.g.,[8]). Like in the TCP/UDP case, the continuous space representation captures the aggregate flow behavior neglecting microscopic packet-level dynamics. Thus, the smaller the packet size and the larger the number of multiplexed flows/packets, the more accurate is the fluid approximation of system dynamics. However, previous work neglect the fine-grained analysis of queue dynamics and its oscillations, which we provide in this paper. An accurate characterization of queue and PIT size variations is important for PIT size prediction and dimensioning. Let us summarize the main assumptions and provide the notation (Tab.1).

3.2.1 Network model

- The network is modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ with bi-directional arcs.
- Links have finite capacities $C_{ij} > 0$, $\forall (i,j) \in \mathcal{L}$. For a line topology (Fig.1) we simplify the notation, $C_i \equiv C_{i-1,i}$ (same notation for other link variables). We consider downlink bandwidth limitations only.

- A content retrieval (flow) $n \in \mathcal{N}$ is univocally associated to a user node and to a set of available sources.
- According to NDN symmetric routing principle, if Interests of flow n traverse link (i, j), corresponding Data packets are pulled down over link (j, i).
- We study PIT dynamics with no caching, no Interest aggregation nor PIT timeouts as a worst case for PIT dimensioning. Note that caching before the bottleneck will have a mitigating effect on PIT size.
- We consider infinite output buffers at downlink: Data packet losses are anticipated by request rate decrease.

3.2.2 *Source model*

- The variables X^n, $\tilde{X^n}$, respectively denote Data, Interest rate of flow n and can be specified at the receiver (index 0) or over a link. In a line network as in Fig.1, $X_i^n(t)$ denotes the downlink Data rate of flow n, at time t and at the i^{th} link, in packet/second.
- The round trip delay of flow n at time t, $R_n(t)$, is assumed to be the sum of a constant round trip propagation delay, R_{min}^n and of a variable queuing delay considering all traversed queues. In a line network with L links as in Fig.1 and for homogeneous flows, $R(t) \equiv R(t) = R_{min} + \sum_{i=1}^{L} Q_i(t)/C_i$, where $Q_i(t)$ denotes the instantaneous downlink queue length at link i at time t.
- Interests of flow n are issued by the user at a rate modulated by a Remote Adaptive Active Queue Management (RAAQM) controller, such as the one presented in ([6],[5]), where initial slow start phase and fast recovery are neglected. As described in [6], the Interest window $W(t)$ at the receiver (we omit the index 0 identifying the receiver), increases by a factor η every window of Data packets received in a round trip delay $R(t)$, and decreases by a multiplicative factor β proportionally to the Data packet rate and to a decrease probability $p(t)$. Since $\tilde{X}(t) = W(t)/R(t)$ by Little's law, the rate evolution over time at the receiver follows the DDE (Delay Differential Equation):

$$\frac{d\tilde{X}(t)}{dt} = \frac{\eta}{R(t)^2} - \beta \tilde{X}(t) X(t) p(t - R(t)) \qquad (1)$$

$$p(t) = p_{min} + \Delta p \min\left(\frac{R(t) - R_{min}}{\Delta R}, 1\right) \qquad (2)$$

where $\Delta p \equiv p_{max} - p_{min}$, $\Delta R \equiv R_{max} - R_{min}$ and $X(t)$ reflects the Interest rate $\tilde{X}(t - R(t))$ associated to the round trip delay before.

The decrease probability $p(t)$ is a RAAQM parameter: p_{min}, p_{max} are two AQM thresholds, assumed to be fixed in our analysis, while in the experiments they are adapted over time by the transport protocol according to round trip time measurements at the receiver (cfr.[6]). If not otherwise specified, $p_{min} = 0$.

$p(t)$ mirrors the delay variation due to queuing delay along the path. Note that $p(t)$ is computed over the round trip delay measurements carried by received Data packets, thus referred to the previous round trip delay. This introduces a delayed component in system DDEs which we will account for in our analysis.

OBSERVATION 3.1. *It is worth noticing that, in this work, we consider the RAAQM controller as it is, to the best of our knowledge, the unique ICN receiver-driven controller proved to be throughput optimal (cfr.[6]). However, the analysis can be easily generalized to any throughput optimal receiver-based rate controller yielding to the same optimal rate allocation.*

4. ANALYSIS OF PIT DYNAMICS

In this section we first model the PIT dynamics in presence of N homogeneous flows and over a single path composed by L cascaded links as in Fig.1 (Node 0 identifies a user node sending requests to node L, associated to the Data repository), hence considering a single bottleneck scenario. In Sec.4.2 we then extend the model to the case of variable number of flows, multi-bottlenecks, and general network topology.

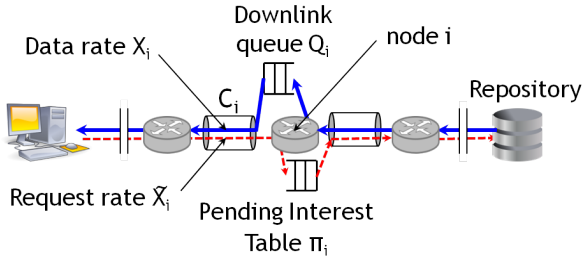

Figure 1: Line network topology.

Every node hosts a PIT keeping track of ongoing requests in uplink. Its length is denoted by $\pi_i(t)$. A downlink transmission queue serving Data packets, $Q_i(t)$, is associated to link i. Since we assume no bandwidth limitations in uplink, $\tilde{X}_i^n(t) = \tilde{X}^n(t)$, $\forall i$. The *system state* is described by $\{\tilde{X}^n(t), X_i^n(t), Q_i(t), \pi_i(t)\}$, $i = 0, \ldots, L$, respectively Interest/Data rates, downlink queue and PIT size at each link. The evolution of system dynamics is described by the following nonlinear DDEs:

$$\frac{d\tilde{X}^n(t)}{dt} = \frac{\eta}{R(t)^2} - \beta \tilde{X}^n(t) X^n(t) p(t - R(t)), \qquad (3)$$

$$p(t) = p_{\min} + \Delta p \min\left(\frac{R(t) - R_{\min}}{\Delta R}, 1\right), \qquad (4)$$

$$X_{i-1}^n(t) = X_i^n(t) \mathbb{1}_{\{Q_i(t)=0\}} + \left(C_i - \sum_{k \neq n} X_i^k(t)\right) \mathbb{1}_{\{Q_i(t)>0\}}$$

$$\text{with} i = 1, \ldots, L \text{ and } X_L^n(t) = \tilde{X}^n(t) \qquad (5)$$

$$\frac{dQ_i(t)}{dt} = \sum_n X_i^n(t) - C_i \mathbb{1}_{\{Q_i(t)>0\}}, \qquad (6)$$

$$\frac{d\pi_i(t)}{dt} = \sum_n \left(\tilde{X}^n(t) - X_{i+1}^n(t) \mathbb{1}_{\{\pi_i(t)>0\}}\right), \qquad \forall i. \qquad (7)$$

Eqq.(3)-(4) describe the RAAQM receiver-driven rate control applied to every flow n. To the ease of notation, we denote $X^n(t)$, the Data rate at the user. Queue length time evolution is described by the fluid queue equation in Eq.(6) where the input rate is the total Data rate at node i, and the output rate is the i^{th} link capacity, C_i. Finally, the PIT size follows a similar fluid queue equation with input rate, the sum of Interest rates of all flows at node i, and output rate,

$N = \|\mathcal{N}\|$	Number of flows
L	Number of links in the line topology
$\tilde{X}_{i,j}(t)$	Interest rate at link ij,
$X_{i,j}(t)$	Data rate at link ij,
$\tilde{X}_i, X_i$	Interest/Data for a line
$Q_{i,j}(t)$	Downlink Queue size at link ij, Q_i for a line
$\pi_i(t)$	PIT size at node i and time t
$C_{i,j}$	Capacity of link ij, C_i for a line
$R(t), \Delta R$	Round trip delay and variation
$p(t), \Delta p$	Packet loss probability and variation
η, β	Rate increase, decrease factor
σ_n	Mean flow size (in # of packets) of flow n.

Table 1: Notation

the total Data rate of such flows at time t. Indeed, ongoing requests of flow n are recorded in the PIT until reception of the corresponding Data packets, when they are eventually removed from the table.

4.1 Main results

We consider the case of a fixed number N of homogeneous flows traversing the series of L cascaded links in Fig.1. Each flow is controlled by the optimal RAAQM controller defined by Eqq.(1)-(2).With no loss of generality, we assume that the system starts empty, i.e. $\tilde{X}^n(0) = Q_i(0) = 0$, $\forall n = 1, \ldots, N, i = 1, \ldots, L$. Since flows are assumed to be homogeneous, $\tilde{X}^n(t) \equiv \tilde{X}(t)$, $X_i^n(t) \equiv X_i(t)$.

4.1.1 Scaling with the number of flows

Let us study how the system scales with the number of flows, N.

PROPOSITION 4.1. *System dynamics in presence of N flows as described by DDEs (3)-(7) are equivalent to those of a single flow, under the following linear scaling of parameters: $N\eta \leftarrow \eta$, $N\Delta R \leftarrow \Delta R$.*

PROOF. The evolution over time of the total flow rate at the receiver, $\tilde{X}^{tot}(t) \equiv N\tilde{X}(t)$, is given by

$$\frac{d\tilde{X}^{tot}(t)}{dt} = \frac{N\eta}{R(t)^2} - \beta\tilde{X}^{tot}(t)X^{tot}(t)p_N(t - R(t)), \quad (8)$$

where $p_N(t) = \frac{\Delta p}{N\Delta R}p(t)$. Queue dynamics become

$$\frac{dQ_i(t)}{dt} = X_i^{tot}(t) - C_i\mathbb{1}_{\{Q_i(t)>0\}}, \qquad \forall i \quad (9)$$

where the instantaneous Data rate at link i is given by Eq.(5), replacing the flow rate with X^{tot}. Similarly, the evolution of PIT size at different nodes still follows Eq.(7), replacing the flow rate with X^{tot}.

It follows that system dynamics in presence of N flows can be derived from those of a single flow by applying the linear scaling of parameters: $N\eta \leftarrow \eta$, $N\Delta R \leftarrow \Delta R$. □

To the ease of notation, results below are proved for $N = 1$.

4.1.2 Transient analysis

Starting from an empty system, the Interest flow rate initially grows linearly in time according to the additive increase term η/R_{min}^2, until it reaches the smallest link capacity, C_{i^*}, where $i^* = \arg\min_i C_i$. At this point in time, that we refer to as t^*, the downlink queue of the slowest link, Q_{i^*} starts filling in and the output rate is throttled to C_{i^*}. Thus, we can prove what follows.

PROPOSITION 4.2. *It exists a time instant $t^* > 0$, such that for $t > t^*$ the output rate of the bottleneck queue Q_{i^*} is equal to the bottleneck link capacity, $X_{i^*}(t) = C_{i^*}$, and downlink queues below the bottleneck remain empty, $Q_j(t) = 0$, for $j < i^*$.*

PROOF. Let us use the above definition for t^*. Proving that the output Data rate at the bottleneck remains equal to C_{i^*}, it is equivalent to show that the bottleneck queue does not remain empty for a positive interval of time. To see this, it suffices to observe that if the bottleneck queue instantaneously empties, the rate decrease term in Eq.(3) becomes zero, as well as the queue decrease term in Eq.(6). Thus, the Interest rate instantaneously increases, hence the queue derivative (Eq.(6)) and consequently the queue itself. Also, by definition of smallest link capacity, we have that $Q_j(t) = 0$, for $j < i$, $\forall t > t^*$, as the input rate C_{i^*} never exceeds C_j. □

OBSERVATION 4.3. *Round trip and bottleneck queuing delay We have observed that queues below the bottleneck remain empty after a transient phase. Above the bottleneck, queue occupancy depends on instantaneous flow rate values. In the following results, we will consider queues above the bottleneck to be empty as well, so neglecting their contribution to the round trip delay, $R(t)$. Clearly, such condition is always verified when the maximum flow rate value $\tilde{X}_{max}(t) < C_{i^{**}}$, with i^{**} identifying the second smallest capacity link after i^*, i.e. $i^{**} = \arg\min_{i\neq i^*} C_i$. The characterization of X_{max} in steady state can be derived from that of the maximum bottleneck queue/PIT value in Sec.4.1.4. One can then identify the parameters' region where the round trip delay happens to be affected mostly by the bottleneck delay. Our experiments in Sec.5 also confirm that such simplifying assumption holds in typical network settings.*

4.1.3 Bottleneck queue and PIT size equivalence

The following result allows us to derive PIT size from bottleneck queue occupancy.

PROPOSITION 4.4. *It exists a time instant $t^* > 0$, such that, for $t > t^*$, PIT sizes, $\pi_i(t)$, are empty above the bottleneck ($i \geq i^*$) and equal to the bottleneck queue length, $Q_{i^*}(t)$ below the bottleneck ($i < i^*$), that is:*

$$\pi_i(t) = \begin{cases} Q_{i^*}(t), & \forall i < i^*, \\ 0, & \forall i \geq i^*, \end{cases} \quad (10)$$

where $i^* = \arg\min_{i=1,\ldots,L} C_i$, identifies the bottleneck link.

PROOF. Let us use the above definition for t^*. For $t > t^*$ and $\forall i > i^*$, we have that $Q_i(t) = 0$, hence $X_i(t) = \tilde{X}(t)$. By invoking Eq.(7), we derive that $\pi_i(t) = 0$ $\forall i \geq i^*$. Instead, for $t > t^*$ and $\forall i < i^*$, $X_i(t) = C_{i^*}$, hence Eq.(6) and Eq.(7) coincide and $\pi_i(t) \equiv Q_{i^*}(t)$. □

4.1.4 Steady state equilibrium

Given previous results in Propp.4.1, 4.2, 4.4, we can focus on the following system of DDEs:

$$\begin{aligned} \frac{d\tilde{X}(t)}{dt} &= \frac{\eta}{R(t)^2} - \beta\tilde{X}(t)C_{i^*}p(t - R(t)), \\ p(t) &= p_{min} + \Delta p_{\min}\left(\frac{R(t) - R_{min}}{\Delta R}, 1\right), \\ \frac{d\pi_i(t)}{dt} &\equiv \frac{dQ_{i^*}(t)}{dt} = \tilde{X}(t) - C_{i^*}, \forall i < i^* \end{aligned} \quad (11)$$

and study its steady state solution. Recall that $Q_i(t) = 0$, $\forall i \neq i^*$, and $\pi_i(t) = 0$, $i \geq i^*$.

PROPOSITION 4.5. *The system of DDEs in Eqq.(11) admits a steady state solution characterized by*

$$\bar{\tilde{X}} = \bar{X} = C_{i^*}, \bar{\pi}_i = \bar{Q}_{i^*} = \left(\frac{\eta C \Delta R}{\beta \Delta p}\right)^{1/3}, \quad i < i^*. \quad (12)$$

which generalizes to

$$\bar{\tilde{X}}^n = \bar{X}^n = C_{i^*}/N, \bar{\pi}_i = \bar{Q}_{i^*} = \left(\frac{N^2 \eta C \Delta R}{\beta \Delta p}\right)^{1/3}, i < i^*. \quad (13)$$

in case of N homogeneous flows.

PROOF. By considering only the bottleneck link i^*, which is the unique non empty queue according to Prop.4.2, we compute the steady state regime by setting the derivatives to zero. We omit the link identifiers to simplify the notation. Notice that, by assuming the bottleneck queue not empty (Prop.4.2), $\bar{X} = C$. Also, $\frac{\eta C^2}{\bar{Q}^2} - \beta C \bar{\tilde{X}} \frac{\bar{Q}\Delta p}{C \Delta R} = 0$, where we considered $R_{min} << \bar{Q}/C$ to the ease of computation. From which we obtain a unique stationary point (which can be computed also in presence of R_{min}. The generalization to the case of N flows derives from Prop.4.1. □

4.1.5 Maximum PIT size in steady state

Beside the average PIT size value, it is interesting for analysis and dimensioning purposes, to compute the maximum value attained by $Q_{i^*}(t)$, thus by the PIT size. To this aim, we linearize the DDEs in Eqq.11 around the stationary point computed above and compute the maximum through the equivalence with the maximum value of the Laplace transform. We provide the two following auxiliary results.

PROPOSITION 4.6. *The Laplace transform of the queue $Q(t)$ is given by*

$$Q(s) = \frac{\bar{Q}(\kappa_2 + \kappa_3) - sC}{s^2 + \kappa_1 s + \kappa_2 + \kappa_3 e^{-sR}}$$

with $\kappa_1 = \beta \frac{\bar{Q}}{N} \frac{\Delta p}{\Delta R}$, $\kappa_2 = \frac{2\eta N C^2}{\bar{Q}^3} = 2\kappa_3$, $\kappa_3 = \beta \frac{C}{N} \frac{\Delta p}{\Delta R}$

PROOF. The following DDEs can be linearized around the stationary point by assuming that the queue never empties (Prop.4.2),

$$\frac{d\tilde{X}_t}{dt} = f(X_t, Q_t, Q_{t-R}) \qquad \frac{dQ_t}{dt} = \tilde{X}(t) - C,$$

The linearized system gives the following DDEs.

$$\frac{d\tilde{X}_t}{dt} = -\kappa_1(\tilde{X}_t - C) - \kappa_2(Q_t - \bar{Q}) - \kappa_3(Q_{t-R} - \bar{Q})$$

$\kappa_1 = -\frac{\partial f}{\partial \tilde{X}_t}$, $\kappa_2 = -\frac{\partial f}{\partial Q_t}$, $\kappa_3 = -\frac{\partial f}{\partial Q_{t-R}}$. By transforming using the Laplace operator

$$\begin{aligned} sX(s) &= -\kappa_1(X(s) - C) - \kappa_2(Q(s) - \bar{Q}) + \\ &\quad - \kappa_3(Q(s)e^{-sR} - \bar{Q}) \\ s(sQ(s) + C) &= -s\kappa_1 Q(s) - Q(s)(\kappa_2 + \kappa_3 e^{-sR}) \\ &\quad + \bar{Q}(\kappa_2 + \kappa_3) \end{aligned}$$

we obtain the Laplace transform of the linearized equation. □

Note that $R \to 0$ the Laplace transform can be easily inverted by analyzing the poles of $Q(s)$, $s_{1,2} = -\frac{1}{2}\kappa_1 \pm j\omega_0$ with $\omega_0^2 = 3\kappa_3 - \kappa_1^2/4$.

PROPOSITION 4.7. *If $R = 0$ then*

$$\max_{t>0} Q(t) \leq \bar{Q}\left(1 + 3\beta\left(\frac{T}{2\pi}\right)^2 \frac{C}{N}\frac{\Delta p}{\Delta R}\right)$$

PROOF. When $R = 0$ the Laplace transform of the queue admits poles with negative real part if

$$\beta \frac{\Delta p_{\max}}{\Delta R} < \frac{C/N}{2(\bar{Q}/N)^2}\left(1 + \sqrt{1 + \frac{2\eta}{\bar{Q}/N}}\right) \approx \frac{C/N}{(\bar{Q}/N)^2}$$

. The Laplace transform $Q(s)$ can be easily inverted to obtain the solution

$$Q(t) = Ae^{-\kappa_1 t/2}\sin(2\pi t/T + \phi)$$

$T = 2\pi/\omega_0$ and

$$A = \frac{CT}{2\pi}\sqrt{1 + \left(\frac{T\bar{Q}(\kappa_2 + \kappa_3)}{2\pi C}\right)^2} \approx \bar{Q}\left(\frac{T}{2\pi}\right)^2(\kappa_2 + \kappa_3) \quad (14)$$

$$= 3\beta\left(\frac{T}{2\pi}\right)^2 \bar{Q}\frac{C}{N}\frac{\Delta p}{\Delta R} \quad (15)$$

This is illustrated in Fig.2. Observing that $\max_{t>0} Q(t) \leq A$ we conclude the proof. □

Let us now state the main result about bottleneck queue, hence PIT size maximum value.

PROPOSITION 4.8.

$$\max_{t>0} Q(t) = \max_{t>0} \pi_i(t) \leq \bar{Q} + \frac{CR}{2\sqrt{3}} = \bar{Q}\frac{1 + 2\sqrt{3}}{2\sqrt{3}}, i < i^*$$

where $Q(t) \equiv Q_{i^*}(t)$, $C \equiv C_{i^*}$.

PROOF. The proof is based on the analysis of the modulus of the Laplace transform $|Q(s)|$. $\max_{s\in\mathbb{C}} Q(s)$ gives the value of the maximum fluctuation of $Q(t)$ from the stationary point $\bar{Q}$. In the complex plain the analysis can be easily made by focusing only on the dominant poles that can be computed by using a second order Padé approximation of the delay term

$$e^{-sR} = \frac{1 - sR/2 + s^2R^2/12}{1 + sR/2 + s^2R^2/12}$$

We have two additional dominant complex conjugate poles, $s_{1,2} = \frac{-3 \pm \sqrt{3}}{R}$ for the Laplace transform $Q(s)$, when compared to the case $R = 0$. The frequency $\frac{\sqrt{3}}{R}$ gives the fundamental frequency at which the system oscillates around its stationary value $\frac{2\pi}{\sqrt{3}}R \approx 3R$. Such frequency dominates over the other frequencies when $3/R < \kappa_1/2 = \frac{\beta}{2}\frac{\bar{Q}}{N}\frac{\Delta p}{\Delta R}$, i.e. $R > 6\frac{N}{\bar{Q}}\frac{\Delta R}{\beta \Delta p}$. In this case the maximum can be obtained by derivation of $|Q(j\omega)|$ which gives that $\max_{s\in\mathbb{C}} Q(s) = \frac{CR}{2\sqrt{3}}$ is the maximum oscillation around the stationary point. The maximum value of the queue is then obtained as

$$\bar{Q} + \frac{CR}{2\sqrt{3}} = \bar{Q}\left(\frac{1 + 2\sqrt{3}}{2\sqrt{3}}\right)$$

□

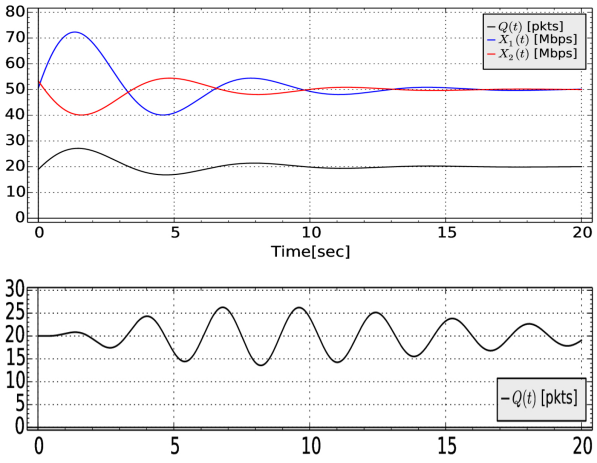

Figure 2: Flow rates and bottleneck queue time evolution, with $R = 0$ (top) and queue evolution when $R > 0$ (bottom).

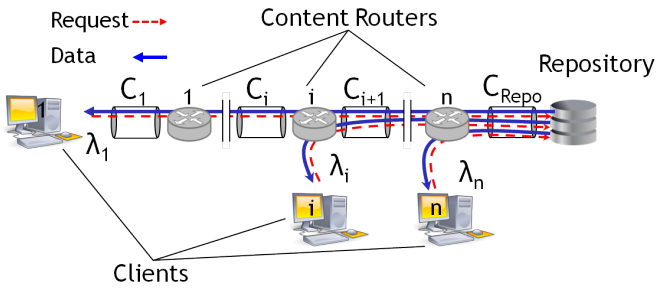

Figure 3: Test topology.

4.2 Model extensions

In this section we generalize our PIT dynamics analysis to the case of variable number of flows, multiple bottlenecks and general network topology. Eqq.(3-7) hold pathwise, where downlink queues and PITs occupancy results of the contribution of all traversing classes of flows. To compute the average PIT size in general network setting, let us recall that the network topology is represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ composed of a set of nodes $\mathcal{V}$ and links $\mathcal{L}$. We group flows according to classes of equal routing, referred to by the index r. Flows belonging to a given class r, share the same sequence of directed links from a user node to a repository node and, hence, the same bottleneck link ij_r^*. We introduce the additional notation $\mathcal{R}_{ij}$ for the set of routes crossing link ij and $\mathcal{L}_v$ for the set of links having node v as egress node. We further distinguish routes bottleneck upstream and downstream within $\mathcal{R}_{ij}$, respectively $\mathcal{R}_{ij}^u$ and $\mathcal{R}_{ij}^d$. Clearly, according to the definition $\mathcal{R}_{ij}^u \cup \mathcal{R}_{ij}^d = \mathcal{R}_{ij}$, $\mathcal{R}_{ij}^u \cap \mathcal{R}_{ij}^d = \emptyset$.

The number of ongoing flows associated to route r, is N_r. We assume flows arrival rate λ_r and average file size of σ_r packets. Let $\rho_{ij} = \sum_{r\in\mathcal{R}_{ij}} \lambda_r\sigma_r/C_{ij}$ denote the load offered to link ij. In case of congestion, link capacity is shared according to max-min fairness and each flow in class r gets the fair share X_r, i.e. $\sum_{r\in\mathcal{R}_{ij}} X_r N_r \leq C_{ij}, \forall ij$ (See [3] for an example of max-min fairness applied to ICN). Thus, $\forall r$ there exists at least one bottleneck link such that

$$\sum_{r\in\mathcal{R}_{ij}} X_r N_r = C_{ij} \text{ and } X_r = \max_{r'\in\mathcal{R}_{ij}} X_{r'}.$$

Such conditions uniquely define the max-min shares X_r across route r. N_r is a Markov process positive recurrent under the stability condition $\rho_{ij} < 1$ for each link on route r.

According to the model presented in Sec.4.1 the queue occupancy at every bottleneck link ij, Q_{ij} is caused only by the flows in progress bottlenecked at link ij. Also

$$Q_{ij}\left(\sum_{r\in\mathcal{R}_{ij}} N_r\right) = Q_{ij}(1) \sum_{r\in\mathcal{R}_{ij}} N_r \tag{16}$$

We compute the average PIT size in steady state at node v as:

$$\mathbb{E}[\pi_v] = \mathbb{E}\left[\sum_{ij\in\mathcal{L}_v}\sum_{r\in\mathcal{R}_{ij}} Q_r\right] = \sum_{ij\in\mathcal{L}_v}\sum_{r\in\mathcal{R}_{ij}} \mathbb{E}[Q_r] \tag{17}$$

$$= \sum_{ij\in\mathcal{L}_v}\left(\sum_{r\in\mathcal{R}_{ij}^u} \mathbb{E}[N_r]\mathbb{E}[Q_r|N_r=1] + \sum_{r\in\mathcal{R}_{ij}^d} \mathbb{E}[N_r]\mathbb{E}[Q_r|N_r=1]\right)$$

$$= \sum_{ij\in\mathcal{L}_v}\sum_{r\in\mathcal{R}_{ij}^u} \mathbb{E}[N_r]\mathbb{E}[Q_r|N_r=1]$$

$$= \sum_{ij\in\mathcal{L}_v} Q_{ij}(1) \sum_{r\in\mathcal{R}_{ij}^u} \mathbb{E}[N_r] \leq \sum_{ij\in\mathcal{L}_v} Q_{ij}(1) \sum_{r\in\mathcal{R}_{ij}^u} \frac{\rho_r}{1-\rho_r}.$$

The PIT occupancy at node v is decomposed into the sum of the contributions associated to each incoming link $ij \in \mathcal{L}_v$ and for each of them into the contribution of each class of flows $r \in \mathcal{R}_{ij}$. Proposition 4.2 allows us to exclude the flows bottlenecked downstream as they do not contribute to the PIT. The linear dependence of the bottleneck queue, hence of PIT contributions, from the number of active flows allows to isolate in the computation the expected number of flows, $\mathbb{E}[N_r]$. According to the max min fairness model, $\mathbb{E}[N_r]$ can be upper bounded by the average number of packets in a M/M/1 queue, $\mathbb{E}[N_r] \leq \frac{\rho_r}{1-\rho_r}$, as in [4].

5. EXPERIMENTAL EVALUATION

In this section we experimentally evaluate PIT dynamics and assess our analytical model accuracy. We consider different scenarios on a parking lot topology with fix/variable number of parallel downloads (e.g., flows) and single/multiple bandwidth bottlenecks. Finally, we analyze the realistic case of an ISP aggregation network.

5.1 Testbed

Our experimental platform is composed of: i) a content router testbed, with a set of hardware nodes running the NDN data plane described in Sec. 3.1; ii) a set of general purpose servers running a custom application layer NDN client/repository implementation. Servers are connected to the content router testbed via optical fibers; iii) a controller, running on a general purpose server to orchestrate the experimental platform and collect statistics.

Content Router – The content router consists of a micro telecommunications computing architecture (μTCA) chassis for advanced mezzanine cards (AMCs). Four slots are occupied by AMC boards equipped with a network processor unit (NPU), 4GB off-chip DRAM, a set of 10GbE interfaces. Each NPU has a 12 cores 800 MHz 64-bits MIPS processor with 16-KB L1 cache per core, and 2MB L2 shared cache. An Ethernet switch enables communications between the different slots of the chassis. In the testbed, every board represents a single content router implementing NDN data plane and network topologies are configured by means of hardware traffic shapers and L2 tunnels among boards. The PIT is implemented using an optimized open-addressed hash table and hardware timers for managing PIT timeouts as in [13]. A detailed description of all data structures (including FIB and CS) is omitted for lack of space. Finally, content router software is instrumented to sample PIT and Queue size values. Samples are collected by an platform controller that computes average and maximum values. We do not process samples inside the board in order to minimize the impact of measurements on processing performance.

Client/repository application – Client/repository applications are written in C and run on general purpose Linux servers. The repository consists of one or multiple UDP/IP socket(s) listening for incoming requests (Interests). Whenever the repository receives an Interest packet, it replies with a fixed size Data packet with the requested chunk name in the header and a dummy payload. The client consists in a local UDP/IP socket sending Interests in order to retrieve a named file. The Interest rate is controlled by the RAAQM mechanism described in Sec. 3.2. Unless otherwise specified we set the controller parameters to $\eta = 1$, $\beta = 0.4$, $\Delta p = 0.02$. The client application is instrumented to report measured round trip delay for each chunk (i.e. the time between a chunk request and the reception of the corresponding data). We use such measures to compute ΔR value used by our model to estimate queue an PIT size (See Prop. 4.5, 4.8.)

Repository and client applications use UDP/IP stack to directly send/receive packets to/from the first directly connected content router. Indeed, the NDN header carries the name of the requested chunk encapsulated in a UDP/IP packet. Content items (each one composed of σ chunks), are requested by clients according to a specific workload (*i.e.,* Poisson arrival). Data and Interest packets are 1,422 Bytes and 92 Bytes respectively including Ethernet, IP, UDP and NDN.

Platform controller – The experimental platform is controlled by a centralized application running on a general purpose server. The controller is in charge of configuring the content router testbed, run client/server application and collect statistics. It is also in charge of the post-processing of the collected data to gather the desired statistics.

5.2 Static case

We consider a static scenario with N parallel flows (*i.e.,* file downloads) and single bottleneck link. We use the network topology in Fig. 3 with $L = 4$, $C_3 = 100$ Mbps, $C_1, C_2, C_{Repo} = 5$ Gbps, we disable caching, and set the maximum queue size of every link to 1,000 packets. Clients run on server 1 connected to CR_1, and retrieve N content items from the repository using route 1 (*i.e.,* the one between server 1 and the repository). Every item is composed of $\sigma = 1$ million Data packets emulating long lived data transfers.

Fig. 4(a) reports the average/maximum PIT and downlink queue sizes at Content Router j (π_j and Q_j respectively) measured from the test and estimated analytically (computed according to Prop. 4.5, 4.8 respectively), as a function of $N = [1 : 30]$. ΔR is set according to the monitored round trip delay variability at the receiver. Note that for each of the following experiments the congestion controller is observed to achieve a fully efficient and fair rate equilibrium as expected (cfr. [6]). Also note that there are not timer expiration: data packet losses are anticipated by a request rate decrease of the controller before the queue of any link gets full.

Fig. 4(a) shows four main results. (i) The queue size of the bottlenecked link (e.g., link 3 in our experiment) is equivalent to the PIT size in the content routers below the bottleneck (e.g., CR 1,2 in our experiment). (ii) The downlink queue and PIT sizes on the nodes that are above the bottlenecked (e.g., CR 3 in our experiment) link are approximately zero. (iii) The PIT size linearly increases with the number of parallel flows N. (iv) Results confirm the trends predicted by the analytical model and the average/maximum PIT size prediction derived in Prop. 4.5, 4.8 appear to be very accurate.

In Fig. 4(b) we plot the average experimental and analytical PIT size at Content Router 2 as a function of the bottleneck capacity, $C_3 = [100 : 1000]$ Mbps, with $N = [10, 20, 30]$ parallel flows. The most striking result is that the PIT size does not depend on the the bottleneck capacity but only on the number of parallel flows N. While Prop. 4.5 highlights the PIT size is related to the ratio $C/\Delta R$, we observe ΔR decreases proportionally with C: it follows the ratio $C/\Delta R$ remains constant for all values of C and the PIT size only depends on N. Also, the figure shows the PIT size prediction of Prop. 4.5 remains very accurate for all bottleneck capacities.

Finally, Fig.4(c) reports the PIT size at Content Router 2 with different congestion control algorithm parameters (namely Δp and β) with $N = 10$ and $C_3 = 100$ Mbps. Two main observations can be made. First, a proper setting of the congestion control parameters can reduce the PIT size while fully utilizing the available capacity. Second, the analytical model is able to correctly predict the PIT size independently from the parameters used by the congestion control algorithm.

5.3 Dynamic case

In this section we consider dynamic scenarios where the number of parallel flows (*i.e.,* file downloads) varies during the experiment. Content requests are generated at servers according to a Poisson process of rate λ and retrieve items from the content repository. Three routes are available to retrieve items from servers 1,2,3 to the repository and are identified as route 1,2,3 respectively. Requested content items belong to a fixed size catalog of 10000 items, divided in $\sigma = 15000$ Data packets each, while content popularity is Zipf distributed with shape parameter $\alpha = 0.8$. It is worth observing that as cache size is set to zero, file popularity does not influence PIT size evolution. We also disable the PIT request aggregation mechanism.

We start the analysis of the dynamic case with a single bottleneck scenario in which content requests are generated

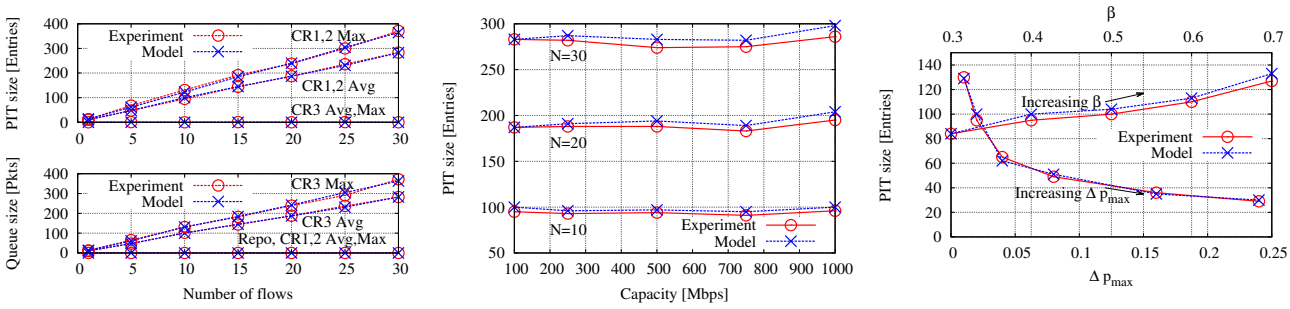

Figure 4: Static scenarios.(a) Average PIT and queue size as function of the Number of flows $N \in [1:30]$, (b) Average PIT size as function of Link capacity $C_3 \in [100:1000]$ Mbps, (c) Average PIT size as function of $p_{max} \in [0.01, 0.24]$ and $\beta \in [0.3, 0.7]$.

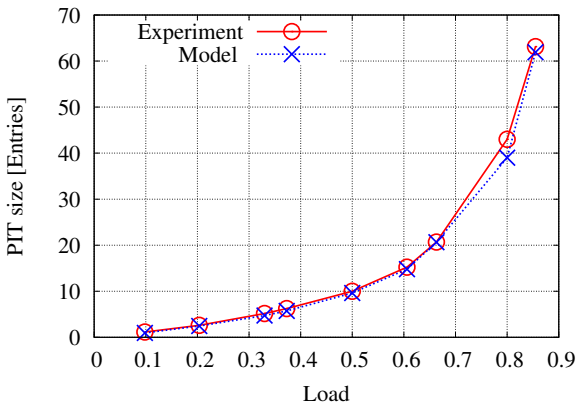

Figure 5: Dynamic scenario, single bottlenecked link: average PIT size as function of the link load $\rho \in [0.1:0.9]$.

Scenario	PIT size [Entries] / Queue size [Pkt]		
bottlenecks	π_1/Q_2	π_2/Q_3	π_3/Q_{repo}
(Link 3)	32.48/0	63.92/0	93.14/91.2
(Links 1,3)	15.69/12.1	42.61/0	78.7/75.6

Table 2: Dynamic scenario, single and multiple bottleneck links. Experimental average PIT size/queue size.

at server 1 only. We set $C_2 = 100$ Mbps, $C_1, C_3, C_{Repo} = 5$ Gbps, *i.e.,* link 2 is the bottleneck, and we vary λ_1 in the range [0.058:0.52] downloads/second in order to vary the link load ρ_2. Fig.5 shows PIT size at Content Router 1 as a function of link load ρ_2; the analytical value is computed with Eq.17. We observe the PIT size increases with the load and the model is very accurate in predicting the measured value.

We now consider a second dynamic scenario with a single bottleneck link and multiple sets of content requests Poisson-generated by all servers according to the rate $\lambda_1 = \lambda_2 = \lambda_3 = 1.75$ requests/second. Link capacities are set to $C_1, C_2, C_3, C_{Repo} = 5$ Gbps, with link towards the Repo representing the bottleneck for all routes. Tab.2 (top row) reports the experimental average PIT size at content routers $j = 1, 2, 3$, and the average downlink queue size at content router $j = 2, 3$ and at the repository. As predicted by our analytical model, the PIT size after the bottlenecked link π_3 is equal to the sum of the PIT entries generated by route 1,2,3 that are bottlenecked on link 3. We also observe that all packets are queued at the node before the bottleneck (*i.e.,* at the repository), and the other queues are empty (as expected from Prop.4.2).

Finally, we consider a third scenario with two bottleneck links and requests Poisson-generated by all servers according the rates $\lambda_1 = 0.35$, $\lambda_2 = 2.63$, $\lambda_3 = 2.63$ requests/second. Link capacities are set to $C_2 = 100$ Mbps, $C_1, C_3, C_{Repo} = 5$ Gbps, with link 1 and 3 representing the bottlenecks for route 1 and route 2,3 respectively. Tab.2 (bottom row) reports the results for this scenario. Unlike the previous scenario, route 1 is bottleneck on link 1 and π_1 does not contribute to π_3. We also observe that packets following route 1 are mostly queued before its bottleneck (*i.e.,* at CR_2), while packets following route 2,3 are queued at the repository.

5.4 ISP network case

In this section we consider a realistic scenario to show how our analytical model can be used as a tool providing useful estimations of the PIT size for different network equipments. We focus on three types of equipments: an OLT (optical line terminator) in the access network, an IP edge router in the backhaul and a backbone router interconnecting the backhaul to the transit network. These three equipments have very different user fan out, starting from a typical 1024 for an OLT for 1Gbps GPON (Gigabit Passive Optical Network) to about 50 000 for IP edge routers in backhaul ring deployments. A router interconnecting the backhaul to the transit network may achieve a user fan out around 5 millions users. The network scenario is depicted in Fig.6(a).

Using our experimental platform we reproduce the upstream link of these network elements serving two classes of users: flows belonging to the first group are bottlenecked at user access gateways, while flows belonging to the second group are bottlenecked outside the aggregation network, *i.e.,* after the backbone router. We assume the load between the OLT and the backhaul network to be equal to $\rho = 0.65$ which is a typical maximum load in operations. The fraction of users belonging to each class is varied from 0% to 100%. The experiments are run using packet delays derived from real traces collected at an OLT of a major European ISP.

Fig.6(b) shows the model prediction and the experimental measurements of the PIT size at the three considered network elements. We observe that the size of the OLT PIT depends on the percentage of flows bottlenecked up-

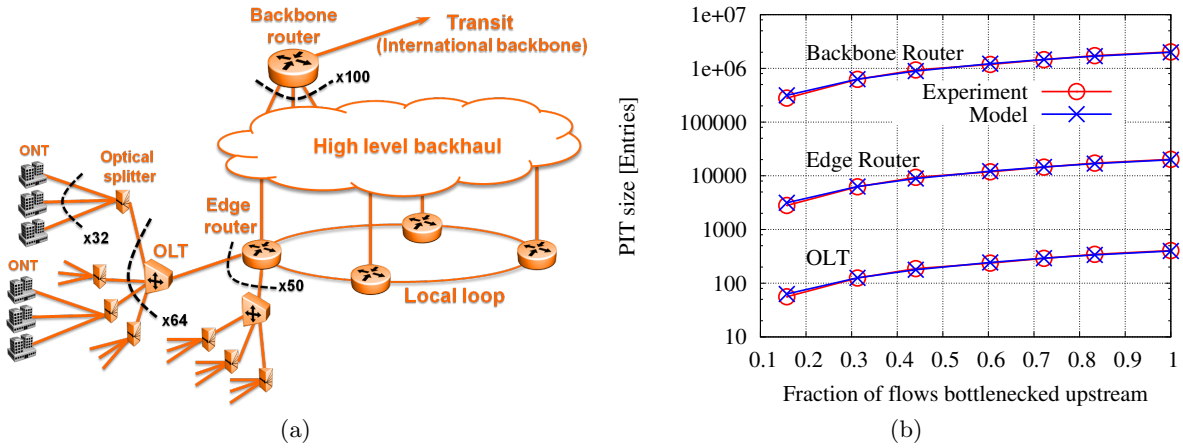

Figure 6: (a) Realistic case topology, (b) Realistic case: PIT size as a function of the percentage of upstream bottlenecked flows.

stream (e.g., peering link between a small ISP and a content provider) and remains very small (about 200 entries) when the percentage of the flows bottlenecked upstream is less than 30%. Even when all flows are bottlenecked upstream the OLT the PIT size does not exceed the 1,000 entries.

The PIT size at backbone routers does not exceed 2 millions entries even in the worst case when all traffic is bottlenecked outside the ISP network. Assuming content names of 40B on average, the memory required to store that many entries is about 400MB in our implementation: this amount of memory is available in today's platforms and allows to perform PIT operations at wire speed up to 40Gbps with current technology.

In addition, we observe that the PIT size is dramatically reduced when part of the traffic is bottlenecked within the backhaul, the access or in the home network. This is typically the case today because content servers are rarely bottlenecks, due to the massive usage of CDNs for popular web applications. On the one hand, a small portion of today's customers have access to very high speed rate fiber at 500 Mbps, capable to create congestion in the backhaul network. On the other hand, we expect user available rates to significantly increase in the following five years: (i) in wireless network thanks to the massive deployment of 4G that will replace all previous data mobile technologies;(ii) in wired access due to the growth of fiber-to-the-home especially in Europe. Finally, we observe that the PIT size at an edge router never rises feasibility issues.

6. PIT SIZING

In this section we present a simple dimensioning tool based on the 95% quantile estimation of the PIT size derived from the analysis in Sec.4. We recall that the number of entries in the PIT at node v is the aggregate sum of the number of packets in routes bottlenecked upstream node v. The number of packets queued at the bottleneck has been computed in Sec.4 and has average $\bar{Q}^3 = N\frac{\eta C\Delta R}{\beta\Delta p}$. If the parameter ΔR of the flow controller is optimally set, i.e $\Delta R = NQ_{\max}/C = N\bar{Q}(1+1/2\sqrt{3})/C$. Hence $\bar{Q}^3 = N^2\frac{\eta CQ_{\max}}{C\beta\Delta p} = N^2\frac{\eta\bar{Q}(1+1/2\sqrt{3})}{\beta\Delta p}$, where we set ΔR equal to N times the value for $N=1$ (see Prop.4.1). By replacing ΔR with $Q_{\max}/C$ (neglecting R_{min}) and using Prop.4.7, we get $\bar{Q}_N = N\sqrt{\frac{\eta}{\beta\Delta p}(1+2\sqrt{3})}$. By taking the expectation of N (Sec.5.3) we obtain an average PIT size

$$\mu = \mathbb{E}[N]\bar{Q}_1 = \mathbb{E}\left[\sum_{r=1}^{M} N_r\right]\bar{Q}_1 = \sum_{r=1}^{M}\frac{\rho_r}{1-\rho_r}\bar{Q}_1$$

The standard deviation σ can be computed approximating the queue size distribution with a uniform random variable with range $[0, Q_{\max}]$. The variance is then $\frac{Q_{\max}^2}{12}$, hence

$$\sigma^2 = \mathbb{E}[N^2]\frac{Q_{\max}^2}{12} = \sum_{r=1}^{M}\frac{\rho_r(1+\rho_r)}{(1-\rho_r)^2}\frac{\eta(1+2\sqrt{3})^3}{12^2\beta\Delta p}$$

Once mean and variance known as a function of the loads, we can use the central limit theorem to estimate the 95% percentile of the Gaussian distribution, with mean μ and variance σ^2 and set the PIT size to $\mu + \sigma z_\alpha$, with $\alpha = 0.05$ being z_α the $1-\alpha$ quantile of the standard Gaussian distribution. Let us consider a simple case when all M routes traversing node v have the same load $\rho_r = \rho$ then

$$\mu = M\frac{\rho}{1-\rho}\sqrt{\frac{\eta}{\beta\Delta p}(1+2\sqrt{3})}$$

and

$$\sigma = \mu(1+1/2\sqrt{3})\sqrt{1+1/\rho}/\sqrt{M}$$

The PIT size can be set to

$$\mu(1+z_\alpha c_v)$$

being $c_v = (1+1/2\sqrt{3})\sqrt{1+1/\rho}/\sqrt{M}$ the coefficient of variation.

The estimation of the PIT size is reported in Fig.6 as a function of the load of M different routes bottlenecked upstream the node under consideration. We note that in practice it is unlikely to observe many routes bottlenecked upstream, on a backbone router for instance, because networks are operated to have bottlenecks in the user's access. A backbone router also is usually not operated above 60% load so that the PIT size would be counted in hundreds of entries or a few thousands at normal loads.

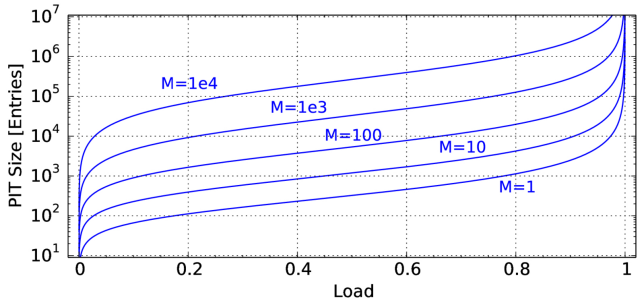

Figure 7: PIT sizing at different loads for a variable number of routes M bottlenecked upstream with respect to the considered router ($\beta = 0.4$, $\Delta p = 0.02$, $\eta = 1$).

7. CONCLUSIONS

The PIT is a key element of NDN design and its sizing is a critical factor for the feasibility of NDN deployment as a whole. In this paper we carry out, to the best of our knowledge, the first analytical investigation on PIT dynamics, deriving average and maximum stationary values. The closed-form characterization of stationary PIT size allows us to provide guidelines on optimal PIT dimensioning. We also build an experimental high speed platform allowing us to assess the accuracy of our model in presence of synthetic traffic workload and trace-driven packet delay distribution. Concerns about PIT state explosion expressed by previous works on the basis of approximate estimations are disproved by our study. We show PIT size is small in typical network settings even without in-network caching and under efficient use of network bandwidth.

In this work we have assumed clients express requests that are flow controlled by a throughput optimal congestion controller that we take from [6]. By considering controlled requesters we neglect congestion collapse scenarios or denial of service attacks that might happen in practice and should be faced by using active PIT management mechanisms (e.g. [1]). In such cases, PIT size minimization can be performed by explicit flow management at a finer-grained scale. Flow drops can be orchestrated via PIT entry removal on a per application basis and for different purposes,e.g.,admission control or overload control in a hop-by-hop fashion. Such decisions, leveraging PIT state over time may enable adaptive re-routing of flows or fast reaction to congestion phenomena by in-network congestion control. Further research is required to design and analyze such PIT management schemes.

Acknowledgments

This research work has been partially funded by the Technological Research Institute SystemX, within the project "Network Architectures" hosted at LINCS.

8. REFERENCES

[1] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang. Interest flooding attack and countermeasures in named data networking. In *IFIP Networking Conference, 2013*, pages 1–9, May 2013.

[2] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing Router Buffers. In *Proc. of ACM SIGCOMM*, 2004.

[3] G. Carofiglio, M. Gallo, and L. Muscariello. Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks. In *proc. of ACM Sigcomm ICN workshop*.

[4] G. Carofiglio, M. Gallo, and L. Muscariello. Bandwidth and Storage Sharing Performance in Information Centric Networking. *Elsevier Science, Computer Networks Journal, Vol.57, Issue 17*, 2013.

[5] G. Carofiglio, M. Gallo, L. Muscariello, and L. Papalini. Multipath congestion control in content-centric networks. In *Proc. of IEEE INFOCOM NOMEN*, 2013.

[6] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang. Optimal Multipath Congestion Control and Request Forwarding in Information-Centric Networks. In *Proc. of IEEE ICNP*, 2013.

[7] H. Dai, B. Liu, Y. Chen, and Y. Wang. On Pending Interest Table in Named Data Networking. In *Proc. of ACM ANCS 2012*.

[8] C. Hollot, V. Misra, D. Towsley, and W. Gong. Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Transactions on Automatic Control*, 47(6):945 –959, 2002.

[9] V. Jacobson, D. Smetters, J. Thornton, and al. Networking named content. In *Proc. of ACM CoNEXT*, 2009.

[10] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts. Evaluating the Number of Active Flows in a Scheduler Realizing Fair Statistical Bandwidth Sharing. In *Proc. of ACM SIGMETRICS*, 2005.

[11] Z. Li, J. Bi, S. Wang, and X. Jiang. Compression of Pending Interest Table with Bloom Filter in Content Centric Network. In *Proc. of ACM CFI 2012*, Seoul, Korea.

[12] C. Tsilopoulos, G. Xylomenos, and Y. Thomas. Reducing Forwarding State in Content-Centric Networks with Semi-Stateless Forwarding. In *Proc. of IEEE INFOCOM*, 2014.

[13] M. Varvello, D. Perino, and L. Linguaglossa. On the Design and Implementation of a wire-speed Pending Interest Table. In *Proc. of IEEE INFOCOM NOMEN workshop*, 2013.

[14] M. Wählisch, T. C. Schmidt, and M. Vahlenkamp. Backscatter from the Data Plane – Threats to Stability and Security in Information-Centric Network Infrastructure. *Computer Networks Journal*, Nov. 2013.

[15] K. Wang, J. Chen, H. Zhou, Y. Qin, and H. Zhang. Modeling denial-of-service against pending interest table in named data networking. *International Journal of Communication Systems*, 2013.

[16] W. You, B. Mathieu, P. Truong, J. Peltier, and G. Simon. DiPIT: A Distributed Bloom-Filter Based PIT Table for CCN Nodes. In *Proc. of IEEE ICCCN*, 2012.

[17] H. Yuan and P. Crowley. Scalable Pending Interest Table Design: From Principles to Practice. In *Proc. of IEEE INFOCOM*, 2014.

[18] L. Zhang and al. Named Data Networking (NDN) Project, 2010. `http://named-data.net/ndn-proj.pdf`.