Multipath Congestion Control in Content-Centric Networks

Giovanna Carofiglio*, Massimo Gallo§, Luca Muscariello§ and Michele Papalini‡
* Bell Labs, Alcatel-Lucent, § Orange Labs, France Telecom, ‡ University of Lugano, Switzerland.
giovanna.carofiglio@alcatel-lucent.com, first.last@orange.com, michele.papalini@usi.ch

***Abstract*—Data communication across the Internet has significantly changed under the pressure of massive content delivery. Content-Centric Networking (CCN) rethinks Internet communication paradigm around named data retrieval, in contrast with the host-to-host transport model of TCP/IP. Content retrieval is natively pull-based driven by user requests, point-to-multipoint and intrinsically coupled with the availability of network storage.**

By leveraging the key features of CCN transport, in this paper we propose for the first time a congestion control mechanism realizing efficient multipath communication over content-centric networks. Our proposal is based on a Remote Adaptive Active Queue Management (RAAQM) at the receiver that performs a per-route control of bottleneck queues along the paths. We analyze the stability of the proposed solution and assess its performance by means of CCN packet-level simulations under random and optimal route selection.

I. Introduction

Over the past decade, data communication across the Internet has significantly changed under the pressure of massive content delivery. Today content delivery systems (as CDNs, P2P systems, transparent caching solutions) realize such functionalities by defining overlays on top of the IP network infrastructure and integrating content-centric features in protocols like HTTP. However, the growth of the number of services and associated bandwidth requirements raises questions about the sustainability of such approach on the long term. Recently, novel future Internet proposals have emerged to rethink the network architecture and its communication primitives around a content-centric communication model [9], [16], [11].

Content-Centric Networking (CCN) advocates a name-based communication, controlled by the receiver (*pull-based*), realized via name-based routing and forwarding of user requests in a *point-to-multipoint* fashion and supported by the presence of a pervasive *network-embedded caching* infrastructure. The multipath nature of CCN transport holds considerable promises for enhanced flexibility in terms of mobility management, improved end-user performance and network resource utilization. However, the definition of multipath transport control mechanisms adapted to CCN still lacks in the literature. In this paper, we present the design of a multipath congestion control mechanism for CCN built upon ICP [4]. The contribution of the paper is twofold: (i) We present and analyze a receiver-driven congestion control mechanism based on Remote Adaptive Active Queue Management (RAAQM) and explicit route labeling in Data packet header for multipath control. (ii) We assess the performance of the proposed congestion control mechanism and provide a preliminary sensitivity analysis through packet-level simulations under random and optimal route selection. The remainder of the paper is organized as follows. In Sec.II we first surveys related work on multipath congestion control protocols. After a brief summary of system description in Sec.III, the proposed congestion control mechanism is presented and analyzed in Sec.IV, in the single path and multipath case. Preliminary results on the performance of the proposed congestion control protocol is presented in Sec.V. Finally, Sec.VI concludes the paper.

II. Related Work

There is a large body of work on *multipath congestion control* with the twofold objective to study the stability of multipath controllers ([6],[10]) and to propose transport protocol implementations based on TCP (mTCP [17], MPTCP [14]). Like most of multipath transport proposals, [17] defines uncoupled congestion control on each path, implemented through separate and independently managed connections (subflows). Each subflow maintains a congestion window as in TCP and perform its own path congestion control.

However, the main drawback of uncoupled multipath congestion control consists in inefficient/unfair control of shared bottlenecks. Some path correlation is introduced by EWTCP [8] to overcome the unfairness and inefficient control of shared bottlenecks with no need for explicit detection of shared bottlenecks. EWTCP proposes a weighted version of multipath TCP, where each subflow increases its congestion window proportionally to a weight, dynamically updated according to the resource utilization of all subflows. A theoretical analysis of the properties of the optimal multipath controller is provided in [6],[10].

Based on the analysis in [6],[10], Multipath TCP [14], as standardized by the IETF working group mptcp, improves previous designs in terms of efficient network usage utilization and of RTT unfairness compensation.

Unlike state-of-the-art multipath solutions, our approach does not require multiple connections setup in agreement with the ‘connection-less’ nature of CCN transport and it realizes separate RTT monitoring on each route distinguishing packets via a route label. Also, differently from previous multipath transport efforts, we assume a receiver-based windowed congestion control. For a presentation of receiver-driven transport advantages and vulnerabilities the interested reader is referred to [12] and to our survey in [3].

In the context of CCN, multipath controllers has not yet been defined. A route selection mechanism enabling multipath is defined in [15] where an adaptive request forwarding mechanism is sketched out. Interface ranking is performed at NDN routers according to a three colors scheme. Congestion control in[15] is still under definition according to the authors and envisaged via request NACKs (Negative Acknowledgements) sent back to the receiver.

III. System description

Our work primarily focuses on CCN/NDN proposal [9], [16], briefly introduced in this section, though the defined congestion control protocol have broader applicability in the context of Information-Centric Networking (ICN[1]) and systems employing a similar transport model (e.g. some HTTP-based CDNs).

Content items are split into packets, identified by a unique name, and permanently stored in one (or more) repository(ies). Users retrieve data by expressing requests (*Interests*), triggering Data packets delivery on the reverse path. A *name-based routing protocol* guarantees that requests are properly routed towards a Data repository, following one or multiple paths. Every intermediate node keeps track of outstanding Interests in data structures called PIT (Pending Interest Table), to deliver the requested data back to the receiver on the reverse path. Each entry in the PIT has an associated timer, so that all requests for the same Data, during such time interval are not forwarded upstream as long as the first query is outstanding. In addition, nodes temporarily store Data packets in a local cache, called *Content Store*.

Upon reception of an Interest packet from an input interface, a node performs a *Content Store lookup*, otherwise, in case of cache miss, a *PIT lookup*, and, in case of Interest forwarding, a *FIB lookup* returning the interface where to forward the Interest (selected among the possible ones). FIB entries are associated to name prefixes, thus FIB lookup is performed via Longest Prefix Matching (see [9]).

IV. Receiver-driven Congestion control

In this section, we introduce and analyze a receiver-driven congestion control mechanism for CCN, extending the initial design of ICP (Interest Control Protocol)[4] along two directions: (i) improving robustness to RTT estimation/heterogeneity, (ii) enabling efficient multipath control via per-route RTT monitoring. ICP is based on an Additive Increase Multiplicative Decrease (AIMD) Interest window controller and relies on round trip delay measurements that are compared to an adaptive threshold to detect congestion. From TCP literature (see e.g.[7]), it is well known that an AIMD closed-loop control is stable and efficient under proper parameters tuning. However, a simple adaptive threshold is not easy to set when RTT estimation varies significantly. In ICP, a window decrease is deterministically applied when a round trip delay measurement exceeds the threshold τ, regularly adapted through monitoring of the average round trip delay.

The simulations we carried out outline two limitations of such approach: (i) deterministic window decreases are sensitive to delay estimation errors (during a congestion phenomenon such problem is worsened by the absence of delay samples), (ii) the difficulty to adapt the threshold τ in presence of multipath communication and heterogeneous path delays. For these reasons, we introduce in the following a probabilistic window decrease mechanism aimed at controlling at the receiver bottleneck queuing delays. We denote it as Remote Adaptive Active Queue Management (RAAQM) and we further prove it to be stable and efficient in Sec.IV-C.

The rationale behind the introduction of a RAAQM-based window decrease is to *anticipate window decrease* as long as a variation of the monitored round trip delay is sensed, while realizing a *smooth decrease* of the congestion window, so limiting its oscillations.

A. AIMD Interest Control with RAAQM

Data are requested via Interests (one Interest per Data packet) in the order decided by the application, according to a window-based *AIMD* mechanism. The congestion window, W, is kept at the receiver and defines the maximum number of outstanding Interests the receiver is allowed to send. W is increased by η/W upon each Data packet reception. This corresponds to an increment of η ($= 1$ by default) each time a complete window of Interests is acknowledged by Data reception. It is worth noting that the window adjustment, hereby described in terms of a packet granularity, could more generally apply to a window expressed in bytes to manage variable packet sizes.

Remote Adaptive Active Queue Management: When an Interest is sent out, the receiver measures the instantaneous round trip delay as the time elapsed until the reception of the corresponding Data packet. The receiver estimates minimum and maximum round trip delay ($R_{\min}$ and $R_{\max}$), maintaining a history of samples (e.g. a sliding window of 30 by default in our implementation), excluding lost Interest/Data packets (and hence retransmitted) packets. A complete window of samples is required before triggering a window decrease.

The measured instantaneous round trip delay $R(t)$ and the estimates of $R_{\min}$ and $R_{\max}$ concur to determine over time the probability p of a window decrease (see fig.1 for instance). p is assumed to be a monotonically increasing function of $R(t)$, going from a minimum value $p_{\min}$ (10^{-5} as default value in our implementation) up to a maximum value $p_{\max} \leq 1$ when above $R_{\max}$. In our implementation, we consider

$$p(t) = p_{\min} + \Delta p_{\max} \frac{R(t) - R_{\min}(t)}{R_{\max}(t) - R_{\min}(t)} \tag{1}$$

with $\Delta p_{\max} = p_{\max} - p_{\min}$. $R_{\min}(t), R_{\max}(t)$ indicate max and min RTT estimates at time t. For the ease of notation we will omit the time indication and use $\Delta R_{\max} = R_{\max} - R_{\min}$. Whenever $R_{\min} = R_{\max}$ in a window of samples, we take a decrease probability equal to $p_{\min}$. In case of window decrease, the window W is multiplied by a decrease factor $\beta < 1$. In this way, we realize a *remote adaptive AQM*, inspired by [2], that allows to control the Interest window at the receiver based on the bottleneck queuing delay, as inferred by the measured round trip delay over time.

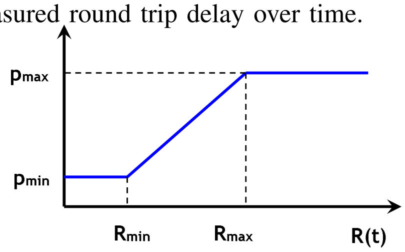


Fig. 1. Example of Window decrease probability.

B. Additional Features

1) Slow Start: To accelerate the initial growth of the Interest Window, we introduce a slow start mechanism like in TCP where the Window follows a multiplicative increase of 1 per Data packet received. Slow start phase ends when $R_{max} - R_{min}$ is above a given threshold and it is repeated each time the monitored R_{min} and R_{max} coincide in a window of samples.

2) Interest Retransmission: Interest re-expression after a time-out is necessary to recover losses, even though Data may just be delayed at the bottleneck. At the same time, retransmitted Interests sent just after a time-out can be filtered, if the PIT timer is bigger than the Interest timer at the receiver.

Indeed, the PIT timer is set to limit the number of pending Interests. The larger this value the higher the number of Interests aggregated and not forwarded upstream. On the other hand, a small PIT timer would imply a large number of unnecessary retransmissions due to delayed packets arriving after PIT time-outs and hence discarded. A trade-off between large and small PIT timers should be found in order to react fast to Interest/Data packets and aggregate a high number of Interests. However, PIT and retransmission timers' regulation is out of the scope of this paper and we consider a retransmission timer equal to the PIT timer (1s as default).

C. Single Path Analysis

In analogy with the fluid representation of ICP dynamics in [4], the modified window update can be described by the following Ordinary Differential Equation (ODE), where $W(t)$ represents the continuous version of the congestion window value.

$$\dot{W}(t) = \frac{\eta}{R(t)} - \beta W(t) p(t-R(t)) \frac{W(t-R(t))}{R(t-R(t))}, \quad (2)$$

If the additive increase term remains unchanged ($\frac{\eta}{R(t)}$), the multiplicative decrease $\beta W(t)$ occurs with dropping rate $p(t-R(t))\frac{W(t-R(t))}{R(t-R(t))}$. The rationale behind is that each incoming Data packet sent in the previous round trip time triggers a window drop with probability $p(t-R(t))$, computed based on the round trip delay estimates of the previous round trip time. A similar fluid representation of window dynamics under RED-like AQM can be found in [7].

$R(t)$ dynamics reflect the evolution of the bottleneck queue remotely controlled by the RAAQM algorithm. To describe queue and window/rate dynamics, let us focus on a single path, relying a user to a content repository through a sequence of hops and characterized by a capacity C (the bottleneck link capacity). The presence of caches at intermediate hops would make the communication naturally point-to-multipoint. We will consider such case in the next sections, while we neglect here intermediate caches and focus on the evolution of N flows (content retrievals) over a single path. Each flow is associated to a congestion window $W_i(t)$, $i = 1, \ldots, N$ and a corresponding rate $X_i(t) = W_i(t)/R(t)$, assumed to be proportional to the congestion window like for TCP by virtue of Little's law. System dynamics can be described through the following fluid ODEs:

$$\dot{X}_i(t) = \frac{\eta}{R(t)^2} - \beta X_i(t) X_i(t-R(t)) p(t-R(t)), \quad \forall i \quad (3)$$

$$\dot{Q}(t) = \sum_{i=1}^{N} X_i(t) - C, \quad R(t) = R_{\min} + Q(t)/C, \quad (4)$$

$$p(t) = p_{\min} + \Delta p_{\max} \frac{Q(t)/C}{\Delta R_{\max}}. \quad (5)$$

Note that $R_{\min}$ and $R_{\max}$ are assumed to be constant and equal to the minimum and maximum delay estimates over a history of round trip delay estimates. As in [7] we approximate $R(t)$ with a constant value, $\bar{R}$, equal to the average value. The resulting rate evolution is,

$$\dot{X}_i(t) = \frac{\eta}{\bar{R}^2} - \beta X_i(t) X_i(t-\bar{R}) p(t-\bar{R}) \quad (6)$$

The equilibrium point of Eqq.(4-5-6) is

$$\widetilde{X} = C/N, \quad \widetilde{p} = \frac{\eta N^2}{\beta \bar{R}^2 C^2}, \quad \frac{\widetilde{Q}}{C} = \frac{(\widetilde{p} - p_{\min})\Delta R_{\max}}{\Delta p_{\max}} \quad (7)$$

Single path stability: The stability analysis of AIMD congestion control with AQM has been studied in the literature [7]. Similar arguments allow to prove the stability of Eqq.(4-5-6), in absence and presence of feedback delays. The main difference is given by the form of the congestion notification and the computation of the stability region in presence of delay.

Proposition 4.1: Given the system of ODEs in Eqq.(4-5-6) in absence of delays, i.e. where the rate evolution is given by

$$\dot{X}_i(t) = \frac{\eta}{\bar{R}^2} - \beta X_i^2(t) p(t) \quad \forall i, \quad (8)$$

Eq.(7) is a global asymptotically stable equilibrium $\forall \beta > 0$, $\Delta p_{\max} > 0$. In presence of delays, where the rate evolution is described by Eq.(6), Eq.(7) is a locally asymptotically stable equilibrium. A region of attraction for the equilibrium is $\left\{(X_1(t), ..., X_N(t), p(t)) : X_i < \widetilde{X}_i \left(1 + \frac{\kappa \bar{R}\sqrt{\zeta}}{p_{min}}\right), \forall i\right\}$

Proof: The proof is reported in App.A. ∎

D. Multipath Extension

The designed congestion control mechanism enables a remote control of a bottleneck queue along the path. In presence of multiple caches over a single path and multiple paths (not necessarily disjoint), the application of RAAQM needs careful analysis as there may be multiple bottlenecks.

Let us denote with *route* any unique sequence of nodes linking a user to a potential Data source (hitting cache or repository). We consider that each CCN node takes forwarding decisions by selecting an interface on a per-packet basis, according to per prefix metric with no coordination with other nodes. Independently from the specific forwarding policy implemented by CCN nodes, let us focus on congestion control performed at the receiver when multiple routes exist.

Alternative 1. *Unique window, unique RTT estimation.* A straightforward extension of single path congestion control to the multipath case would be to keep a unique Interest window W, to gather round trip delay measurements coming from all different nodes and to exploit all RTT samples to compute a unique window decrease probability p. In this setting, the probabilistic decrease profile would refer to a global queuing delay averaged over possibly heterogeneous queuing delay samples related to different routes, while the estimated minimum and maximum round trip delay, $R_{\min}, R_{\max}$ would represent respectively the minimum and maximum over all delay routes. As confirmed by our simulations, controlling multiple bottlenecks through a unique decrease probability p results in inefficient resource control (cfr. Sec.V).

Alternative 2. *Separate windows, separate RTT estimations.* An alternative solution proposed in literature (for Multipath TCP [14]) is to control different paths through separate congestion windows, associated to independent connections, whose evolutions are coupled by a global congestion window term. However, such approach seems hardly applicable in CCN where the number of routes is not known a priori and can vary according to the popularity of the requested item, also unknown to the receiver.

The proposal. *Unique window, separate RTT estimations.* Therefore, we opt for a unique congestion window of Interests with separate per-route RAAQM state, so reducing also the flow state to keep at the receiver. This requires:
Route labeling: each Data packet carries a uniquely identified route label composed by the sequence of traversed node identifiers (i.e. MAC address or dedicated CCN node's name space), starting with the Data source. Notice that instead of route labeling, RTT samples can be separated by using spectral analysis. However, such techniques introduce further estimation errors worsening the overall performance.
Route delay monitoring: the receiver keeps separate estimates of the minimum and maximum round trip delay observed on a given route by distinguishing the samples according to their route label. Delay estimates are kept only for those routes associated to a minimum number of samples (e.g. a window of 30).
Route probabilistic decrease. Based on the route delay estimates, a per-route window decrease probability p_r, is computed. If we denote by $\mathcal{R}$ the set of monitored routes, the fluid equation of the window evolution gives:

$$\dot{W}(t) = \frac{\eta}{R(t)} - \beta \sum_{r\in\mathcal{R}} p_r(t-R(t))W(t)s_r\frac{W(t-R(t))}{R(t-R(t))}, \quad (9)$$

where $s_r W(t)$ denotes the fraction of the current window $W(t)$ of Interests routed along route r resulting from the forwarding decisions taken in a distributed way by CCN nodes. Note that the split ratios s_r of the flow over different routes do not need to be known at the receiver, who simply applies the probabilistic decrease p_r at reception of a packet identified by the label r.
Route selection and forwarding policy. The split ratios drive route selection and are determined by the forwarding policy implemented at each node. In the evaluation we will consider (i) a random policy where interface selection is performed uniformly at each node and (ii) an optimal policy where interface selection is performed via static weights computed in a centralized way to guarantee all available bandwidth to be fully exploited and fairly shared among flows in the max-min sense. Due to lack of space, we omit the problem formulation which can be found in [3].

V. Performance Evaluation

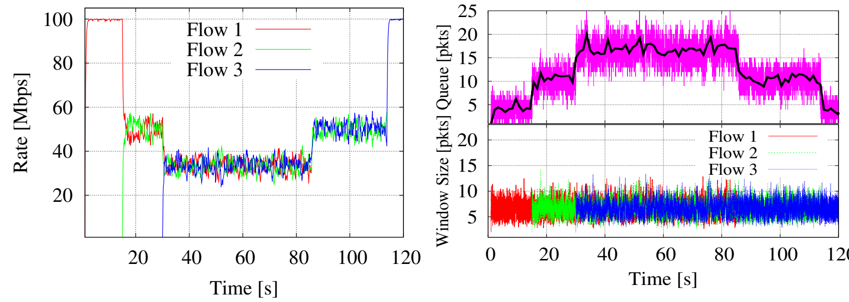
To assess the performance of the proposed RAAQM congestion control we carried out CCN simulations in different network scenarios. Specifically, we implemented the AIMD scheme with RAAQM in ccnpl-sim (https://code.google.com/p/ccnpl-sim), a C++ event driven simulator providing a complete representation of a CCN system: naming, packet-level caching, name-based forwarding and routing via CS, PIT and FIB data structures. FIB entries are pre-computed per each name prefix according to a name-based routing protocol that calculates all available paths. In this section, we gather a selected set of simulations illustrating functioning, properties and benefits of our solution.

A. Single Path and Sensitivity Analysis

We first consider a single path connecting a user to a content repository (characterized by a bottleneck capacity of 100Mbps and 1ms of propagation delay) and study performance sensitivity to congestion control parameters. The user requests 10 different content items, of the same size, namely 5000 packets of 10KB each (default content size in our simulations). The buffer size is set to a large value, in order to observe the evolution of queue dynamics under different parameters settings and in absence of packet losses. By varying $\Delta p_{\max} = p_{\max} - p_{\min}$ ($p_{\min} = 10^{-5}$ as default) and β, we observe the RAAQM stability properties. As reported in Fig.2(a), for $\beta\Delta p_{max} > 2\cdot 10^{-3}$, the queue occupancy and hence the round trip delay stabilizes, while, outside of this region the queue keeps growing over time. Such dependence of the stability from $\beta\Delta p_{\max}$ is in accordance with Prop.4.1. As predicted by Eq.(7), the average queue increases as $\beta\Delta p_{\max}$ decreases, while bandwidth is always fully utilized and rate approaches 100Mbps. Under a finite buffer size ($B = 100$pkts), instability translates into an increasing loss probability (cfr. Fig.2(a)), negatively affecting overall performance.

In Fig.2(b) and Fig.2(c) we report instantaneous throughput, as well as queue and windows time evolution. By varying the number of flows in progress N, we observe the efficiency and fairness of the protocol. The user sends out requests for

Parameters		Infinite Buffer		$B = 100$ pkts	
$\Delta p_{\max}$	β	$\widetilde{Q}$	$\widetilde{W}$	ploss (%)	$\widetilde{W}$
0.5	0.5	21.67	2.36	0	2.36
0.2	0.5	36.93	4.07	0	4.07
0.2	0.2	60.73	6.45	0	6.45
0.1	0.5	54.74	6.11	0	6.11
0.1	0.2	86.97	9.34	0.09	9.32
0.1	0.05	177.84	18.37	2.28	13.34
0.1	0.02	267.90	27.55	4.97	16.96
0.05	0.02	∞	∞	8.05	21.76



(a) (b) (c)

Fig. 2. Single Path Scenario: Sensitivity Analysis (a), Rates (b), Windows/Queues evolution (c).

three different items, at 1s, 18s, 30s, respectively for the red, green and blue curve in Fig.2(b). Link bandwidth is always fully exploited and fairly shared by ongoing flows. It is worth observing that queue occupancy reflects the number of flows in progress, since the RAAQM mechanism controls the difference between minimum and maximum RTT estimates and the monitored minimum RTT increases at the arrival of the second and third flow respectively. More precisely, as long as the maximum RTT is smaller than the value associated to a full buffer, the queue grows with the number of ongoing flows, N. When close to buffer saturation, the difference between minimum and maximum RTT diminishes leading to a larger per-flow decrease probability, more window drops and hence queue reduction. Efficiency is guaranteed by the fact that the queue never empties and link utilization is always non zero.

B. Multipath scenario

Here we analyze the multipath network scenario in Fig.3, where one user is connected to 4 repositories via 4 non disjoint paths (p_1, p_2, p_3 and p_4 in Fig.3). Each repository stores the entire content catalog and to limit the number of routes to 4 we neglect intermediate caches. The user generates 10 content requests (flows) simultaneously. Link capacities are reported in Fig.3. Note that the links between user and network, are set to 1Gbps as to avoid bottlenecks at the access that would prevent the use of multiple paths.

Route labeling, necessary, but not sufficient: In absence of route labeling, the round trip delay estimates (Fig.4(b)), computed over all undistinguished samples, captures a small minimum RTT (from the fastest route, node 7 to the user) and a very high maximum RTT (corresponding to the slowest route, node 4 to the user, whose samples dominate RTT estimation).

The large fluctuations of $RTT_{max} - RTT_{min}$ highlight the poor round trip delay control in absence of route labeling, which results in a too small global decrease probability p and hence in inefficient congestion control. This is confirmed by the oscillations of window and queue size in Fig.4(a)

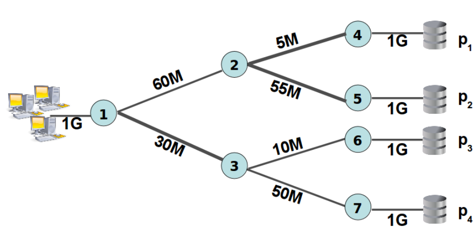


Fig. 3. Multipath network scenario.

potentially leading to packet losses whenever queue occupancy exceeds the buffer value (e.g. $B = 10$ in the figure).

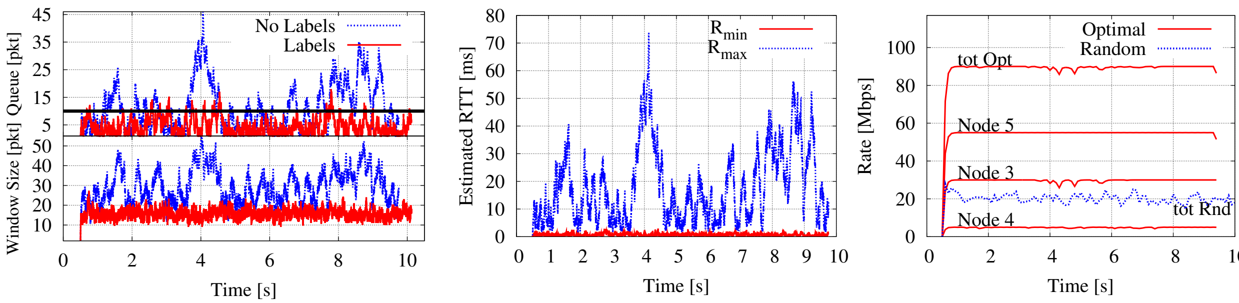
If route labels are needed to properly collect round trip delay estimates, they are not sufficient to guarantee optimal throughput in presence of *random Interest forwarding*, as illustrated in Fig.4(c). Indeed, under uniformly random interface selection, the average rate over all routes converges to the value of the route with minimum capacity (5Mbps) underutilizing the total available capacity (20Mbps instead of 90Mbps).

Instead, rates are observed to fully exploit network resources under *optimal split ratios* computed off-line and imposed via static per-node and per-interface forwarding probability. Moreover, under *optimal split ratios* the 4 paths are characterized by heterogeneous average RTTs (i.e. $RTT(p_1) =$ 30.06ms,$RTT(p_2) =$ 3.1ms,$RTT(p_3) =$ 6.8ms, $RTT(p_4) =$ 6.5ms). In this paper we do not directly analyze caching impact on the multipath congestion control protocol for lack of space. However, in-network caching will introduce high RTT's variability during the content download, and the last simulation scenario highlighted the fact that AIMD coupled with RAAQM can control multiple paths with heterogeneous RTTs. Hence, we are confident that the proposed congestion control protocol is also able to fully exploit network resources in presence of caches.

VI. Conclusion

In the paper, we present for the first time the design of a congestion control mechanism realizing fair and efficient multipath communication in CCN. By leveraging the key features of CCN transport, we propose a Remote Adaptive Active Queue Management (RAAQM) scheme for window control at the receiver, fed by per-route round trip delay estimates. Unlike traditional multipath approaches exploiting separate flow control over different paths, we apply a per-route RAAQM control to a unique congestion window regulating the total number of Interests for a given content retrieval.

Initial results are encouraging and outline that our solution applied on labeled routes can provide efficient and fair resource utilization, when coupled with an optimal Interest forwarding strategy. We leave for future work the definition of distributed and dynamic forwarding algorithms approaching optimal flow allocation. Also, we envisage to extend our simulation campaign and proceed with a real implementation in a CCNx testbed.



(a) (b) (c)

Fig. 4. Multipath Scenario: Packet Loss (a), RTT estimation without route labels (b), Rate comparison (c).

Acknowledgements

This work has been partially funded by the French national research agency (ANR), CONNECT project, under grant number ANR-10-VERS-001.

References

[1] Information Centric Network projects. http://anr-connect.org/index.php?p=1_5_Related-Projects.

[2] D. Ardelean, E. Blanton, and M. Martynov. Remote active queue management. In *Proc. of ACM NOSSDAV*, 2008.

[3] G. Carofiglio, M. Gallo, E. Gourdin, L. Muscariello, and M. Papalini. Technical report, 2012. http://perso.rd.francetelecom.fr/muscariello/report-multipath.pdf.

[4] G. Carofiglio, M. Gallo, and L. Muscariello. Icp: Design and evaluation of an interest control protocol for content-centric networking. In *Proc. of IEEE INFOCOM NOMEN*, 2012.

[5] J. K. Hale and S. M. Lunel. *Nonlinear differential equations and dynamical systems*. Springer-Verlag New York, 1990.

[6] H. Han, S. Shakkottai, and e. a. Hollot. Multi-path tcp: a joint congestion control and routing scheme to exploit path diversity in the internet. *IEEE/ACM Trans. Netw.*, 14(6):1260–1271, Dec 2006.

[7] C. Hollot and Y. Chait. Nonlinear stability analysis for a class of tcp/aqm networks. In *In Prof of IEEE Conference on Decision and Control*, 2001.

[8] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, and H. Tokuda. Multipath congestion control for shared bottleneck. In *Proc. of PFLDNeT Workshop*, 2009.

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

[10] F. Kelly and T. Voice. Stability of end-to-end algorithms for joint routing and rate control. *SIGCOMM CCR*, 35(2):5–12, 2005.

[11] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *Proc. of ACM SIGCOMM*, 2007.

[12] A. Kuzmanovic and E. W. Knightly. Receiver-centric congestion control with a misbehaving receiver: Vulnerabilities and end-point solutions. *Elsevier Computer Networks*, 2007.

[13] F. Verhulst. *Introduction to Functional Differential Equations*. Springer-Verlag New York, 1993.

[14] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath tcp. In *Proc. of NSDI*, 2011.

[15] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang. Adaptive forwarding in named data networking. *SIGCOMM CCR*, 42(3), 2012.

[16] L. Zhang and al. Named data networking (ndn) project, 2010. http://named-data.net/ndn-proj.pdf.

[17] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang. A transport layer approach for improving end-to-end performance and robustness using redundant paths. In *Proc. of Usenix ATEC*, 2004.

Appendix A

Proof: Let us denote with $(\widetilde{\mathbf{X}}, \widetilde{p})$ the equilibrium point of Eqq.(4-5-6), where $\mathbf{X}(t) = (X_1(t), \dots, X_N(t))$. Let us consider the following Lyapunov function

$$V(\mathbf{X}(t), p) = \frac{1}{2}\sum_{i=1}^{N} \Delta X_i(t)^2 + \frac{1}{2}\zeta \Delta p(t)^2 \qquad (10)$$

where $\Delta X_i(t) = X_i(t) - \widetilde{X}_i$, $\Delta p(t) = p(t) - \widetilde{p}_i$ and $\zeta = \frac{\Delta R_{\max}\beta C \widetilde{X}_i^2}{\Delta p_{\max}}$. Clearly, $V(\widetilde{\mathbf{X}}, \widetilde{p}) = 0$, $V(\mathbf{X}, p) \geq 0$ and

$$\begin{aligned}
\dot{V} &= \sum_{i=1}^{N} \Delta X_i \dot{X}_i + \zeta \Delta p\ \dot{p} = \\
&= \sum_{i=1}^{N} \Delta X_i \left(\frac{\eta}{\bar{R}^2} - \beta X_i^2 p \right) + \frac{\beta C^2 \Delta p}{N^2} \sum_{i=1}^{N} \Delta X_i \\
&= \sum_{i=1}^{N} \Delta X_i \left(\frac{\eta}{\bar{R}^2} - \beta X_i^2 p + \frac{\beta C^2 p}{N^2} - \frac{\beta C^2 \eta N^2}{\bar{R}^2 \beta N^2 C^2} \right) \\
&= -\beta p \sum_{i=1}^{N} \Delta X_i (X_i^2 - \widetilde{X}_i^2) = -\beta p \sum_{i=1}^{N} \Delta {X_i}^2 (X_i + \widetilde{X}_i) < 0
\end{aligned}$$

where we omitted the indication of the time variable t. Therefore $\dot{V}(\mathbf{X}, p)$ is negatively semi-definite for any ball including the equilibrium point. This proves that $(\widetilde{\mathbf{X}}, \widetilde{p})$ is a globally stable equilibrium as per [13]. Let us consider the case with delays and start from the same Lyapunov function.

$$\begin{aligned}
\dot{V} &= \sum_{i=1}^{N} \Delta X_i \dot{X}_i + \zeta \Delta p \dot{p} = -\beta \sum_{i=1}^{N} \Delta X_i (X_i^2 p(t-R) - \widetilde{X}_i^2 p(t)) \\
&= -\beta \sum_{i=1}^{N} \Delta X_i \left(X_i^2 p - \widetilde{X}_i^2 p - X_i^2 \int_{t-R}^{t} \dot{p}(u) du \right) \\
&\leq -\beta \sum_{i=1}^{N} \Delta X_i \left(X_i^2 p - \widetilde{X}_i^2 p - \frac{X_i^2 \Delta p_{\max}}{C \Delta R_{\max}} \int_{t-R}^{t} \Delta X_i(u) du \right)
\end{aligned}$$

We apply the Lyapunov-Razumikhin theorem [5] to compute a stability region for the set of parameters. Hence, by assuming $V(\mathbf{X}(u), p(u)) \leq V(\mathbf{X}(t), p(t))$ as $u \in [t-R, t]$ we need to show that $\dot{V}(\mathbf{X}(t), p(t)) < 0$ in a non empty region including the equilibrium.

$$\dot{V}(\mathbf{X}(t), p(t)) \leq -\beta \sum_{i=1}^{N} \Delta X_i \left(p(X_i^2 - \widetilde{X}_i^2) - X_i^2 \kappa R \sqrt{\zeta + \Delta X_i^2} \right)$$

with $\kappa = (\Delta p_{\max})/(C \Delta R_{\max})$. It is easy to see that $\dot{V}(\mathbf{X}(t), p(t)) < 0$ as long as $X_i < \widetilde{X}_i$ for all i. Note that by assuming $X_i > \widetilde{X}_i$ for all i we are computing a smaller stability region, as in general $X_i < \widetilde{X}_i$ may hold for some i only. To have $\dot{V}_t < 0$, when $X_i > \widetilde{X}_i$ $\forall i$ it is sufficient to impose $p(1 - \widetilde{X}_i^2/X_i^2) > \kappa R \sqrt{\zeta + \Delta X_i^2}$. The exact X_i range where the inequality holds can be numerically computed. However, since we are interested in a region of attraction including the equilibrium point, we can provide a smaller region of attraction for $X_i \approx \widetilde{X}_i$. In this case, $\sqrt{\zeta + \Delta X_i^2} \approx \sqrt{\zeta}$ and $X_i + \widetilde{X}_i \approx 2\widetilde{X}_i$. Thus, a region of attraction for the considered equilibrium point is $\left\{ (\mathbf{X}(t), p(t)) : X_i < \widetilde{X}_i \left(1 + \frac{\kappa \bar{R} \sqrt{\zeta}}{p_{min}}\right), \forall i \right\}$ Remark that $k\sqrt{\zeta} = \widetilde{X}_i \sqrt{\frac{\Delta p_{max} \beta}{\Delta R_{max}}} \propto \sqrt{(p_{max} - p_{min})\beta}$. ■