AIMD and CCN: Past and Novel Acronyms Working Together in the Future Internet

Damien Saucez, Luigi Alfredo Grieco, Chadi Barakat

AIMD and CCN: past and novel acronyms working together in the Future Internet

Luigi Alfredo Grieco
DEE - Politecnico di Bari (Italy)
a.grieco@poliba.it

Damien Saucez
INRIA (France)
damien.saucez@inria.fr

Chadi Barakat
INRIA (France)
chadi.barakat@inria.fr

ABSTRACT

Content-centric networking is a new paradigm to better handle contents in the future Internet. Routing is done on a hop-by-hop basis by content names and routers are given the possibility to cache contents for a more efficient bandwidth utilization. This new paradigm poses several challenges, among them the congestion control problem which we explore in this paper by supposing that CCN networks will deploy similar congestion control than in today TCP/IP. We analytically evaluate the way bandwidth is shared with CCN and we compare the performance of downloads to what users get today. We consider different factors such as the way CCN routers are deployed, the popularity of contents and the bandwidth of links. We make several observations and suggestions, most importantly that a CCN congestion control like TCP penalizes non-popular contents because of their low hit ratio and hence their longer network delay. A proper congestion control should in somehow compensate the decrease of network delay CCN caching provides for popular contents.

1. INTRODUCTION

The *host-centered* communication principle is at the basis of the Internet architecture and has satisfied the needs of most applications so far (Web, Email, etc). However, the way users perceive network services has undergone through a substantial evolution during last years [6]. On the one hand, services are now distributed across large platforms (such as *Content Delivery Networks* (CDN) and Clouds). On the other hand end users are now more concerned by accessing services themselves rather than caring about the host on which they are running. As a matter of fact, nowadays most of the Internet traffic is related to content dissemination, like file sharing and media streaming. This new *content-centric* context challenges the current architecture that is not able to handle it in an effective way. This demand is currently handled by application-layer solutions such as CDNs but the Internet architecture itself must evolve to provide inherent support of content distribution and this for more efficiency and better interpretability between the different services and applications.

The so called *content-centric* approach has consequently become the leading rationale of many `Future Internet` proposals.Among these solutions, *Content-Centric Networking* (CCN) [6] is considered as the most promising architecture. CCN presents the main functional blocks for an efficient content support as content protection, routing by content names, and in-network caching. In its rationale, contents become independent entities that users can retrieve without having any awareness about the location of service providers. We consider CCN as basis for this work even though it can be applied to any content-centric architecture presenting similar functionalities. *Our focus is on the way bandwidth is shared within the CCN framework compared to exiting Internet.*

CCN communications are receiver-driven [6]. A user asks for a content by issuing an Interest packet, which is routed toward the nodes possessing the required information; such nodes reply with Data packets, that are routed along the reverse paths activated by the Interest packet. Every intermediate node can cache the forwarded contents, reducing network congestion and servers load, and enhancing the end users experience. The important role of in-network caching in CCN has resulted in many research works on caching strategies and their performance in terms of hit/miss ratios [8, 7, 3, 9]. The focus on caching has deviated the attention from the congestion control problem and its relation to bandwidth sharing. This latter problem strongly influences the overall network performance and the quality of service perceived by end users, especially when popular contents are cached inside the network A first attempt in this direction has been illustrated in [2], where a window-based Interest flow Control Protocol (ICP) driven by an Additive Increase Multiplicative Decrease (AIMD) mechanism is proposed to regulate the Interest rate at the receiver. Even though ICP guarantees optimal fair and efficient bandwidth sharing we further explore the problem of congestion control and bandwidth sharing in CCN and compare it with the situation in the existing Internet and answer the fundamental question: *Who wins and who looses with CCN?.*

Pure performance are important but we must understand how the performance of everyone would change if CCN were deployed tomorrow. To answer the above question, we first assume that CCN will deploy an AIMD flow control as the one used by TCP [4]. AIMD runs at the client requesting the content, and controls the rate at which Interest packets are sent. Using this control we analytically compare the end-to-end throughput CCN would provide to content downloads with respect to the end-to-end throughput the client would realize with TCP in the current Internet. We use results on the throughput of an AIMD mechanism in TCP to devise a general mathematical model for AIMD in CCN and derive preliminary results by applying our model to the study of a chain topology.

Our findings can be summarized as follows. CCN is a very powerful approach to reduce the load at the server but it is biased against low popularity contents. Those contents will see their throughput decrease compared to toady's Internet, and the decrease factor can be up to the number of CCN hops over the path to the server. Popular contents from their side will get higher end-to-end throughput but given the bandwidth limitation inside the network this increase is limited and might even go unnoticed. Also, the global load reduction at the server is function of the total caching capacity and little sensitive to the topology. Our analysis confirms the benefits of CCN in terms of load reduction on the network and the servers, but points out an unfairness problem and consequently to the need to carefully design congestion control so to avoid the starvation of less popular content downloads.

In Sec. 2, we present our general problem and derive the main results. In Sec. 3, we instantiate these general results to particular network scenarios. Sec. 4 follows-up with numerical results and main observations. Finally, we conclude in Sec. 5 with a summary of our contributions and perspectives on future work.

2. PROBLEM FORMULATION

We consider users downloading contents with *Content-Centric Networking* CCN [6] and focus on the problem of bandwidth sharing and comparison with toady Internet by leveraging well-known results regarding the download rate in TCP/IP with AIMD [1]. In CCN, every content is divided into chunks requested by users via *Interest* packets. Interest packets head toward the server and stop when they find a router caching copies of the requested chunks, otherwise they reach the server. The requested chunks are sent back to the requester in *Data* packets. Clearly, there is a need for a mechanism to control the rate at which Interest packets are sent, otherwise the network will be congested and everyone will suffer. For now, there is no standard congestion control mechanism for CCN, we suppose in this work

Table 1: Model Notation.

Symbol	Meaning
L	Set of links
N	Set of downloads
C	Set of downloaded contents
$L(c) \subseteq L$	Ordered set of links along the path from the requester to the server of content $c \in C$
$len(c)$	Length of path $L(c)$
$T(c)$	Throughput at the receiver for the download of content $c \in C$
$RTT(c)$	Mean Round-Trip Time during the download of content $c \in C$
$p(c)$	The probability a Data packet for a chunk of content $c \in C$ is lost due to congestion
$\omega_i(c)$	The probability that a given chunk of content $c \in C$ is available at i hops distance from the requesting client
$\Omega(c, l)$	The overall hit ratio for content $c \in C$ at all downstream routers with respect to the link $l \in L$
$b(l)$	Bitrate of a link $l \in L$ (physical capacity)
$\Lambda(l)$	Overall traffic at the link $l \in L$
$ord_l(c)$	Position of the link l with respect to the path used for the download $c \in C$: $ord_l(c) = 1$ means that l is the first link encountered by Interest packets

that CCN clients implement a window-based *Additive Increase Multiplicative Decrease* AIMD congestion control as in TCP. Congestion is inferred by Data packet losses and window is increased linearly between congestion events and is decreased multiplicatively upon congestion We further assume that congestion only occurs in the download direction which is used by Data packets. The upload direction is used by Interest packets, which given their small size, do not congest the network and hence do not experience losses. As a first effort, we assume downloaded contents to be of large and equal size S, we leave the case of small sized downloads to a future research.

General throughput expression: Denote by W the Interest transmission window, which defines the number of in-flight Interest packets still waiting for the corresponding Data packets (see Fig. 1). The AIMD for CCN controls the window W in this way:

$$W = \begin{cases} W + 1/W & \text{if a Data packet is received} \\ \alpha \cdot W & \text{otherwise} \end{cases} \quad (1)$$

If we refer to $p(c)$ as the probability that a Data packet for a chunk of content c is lost due to congestion, and to $RTT(c)$ as the mean round-trip time to retrieve chunks of content c, one can write the mean download rate $T(c)$ for content c as follows:

$$T(c) = \frac{K}{RTT(c)\sqrt{p(c)}}, \quad (2)$$

where K is a constant. This is known as the square root formula for AIMD throughput [1]. The main dif-

ference in our case is that copies of the chunks can be found over the path to the server, and so the round-trip can be smaller than in the case of TCP where the download is end-to-end. The probability to find chunks over the path depends on the hit ratio of CCN caches that depends on the popularity of the requested content c. Troughput expression is developped later in the paper.

Note that if CCN routers implement explicit loss notification, $p(c)$ can model the probability that a Data packets sent back to the requester carries such a signal. Indeed, one can imagine CCN routers sending back a congestion notification when the Data traffic at their outgoing interfaces reaches some limit (i.e., the link capacity).

Throughput specification to CCN: Consider a content download c, and denote by $len(c)$ the number of links located between the requester and the original server. The calculation of the throughput requires expressions for both congestion probability $p(c)$ and mean round-trip time $RTT(c)$.

$RTT(c)$ depends on the number of CCN hops between the user requesting the content and the CCN routers providing copies of the chunks. In fact, if a content is very popular, it will very likely have cached copies of its chunks at intermediate CCN routers, so that a small $RTT(c)$ will be obtained. On the other hand, for contents with a very low popularity, chunks will be mostly retrieved from the original server so achieving a large $RTT(c)$. In other words, while $p(c)$ reflects, to some extent, the load on the physical paths of the CCN overlay, $RTT(c)$ is only coupled to the content popularity.

If we define $\omega_i(c)$ as the probability that a given chunk is available at i hops distance from the user willing to download it (the so called hit ratio of the cache of a CCN router), we can derive an expression for $RTT(c)$ as follows:

$$RTT(c) = d \sum_{i=1}^{len(c)} i\omega_i(c) \prod_{j<i}[1-\omega_j(c)], \qquad (3)$$

where d is the average hop delay and is supposed to be the same for all links for simplicity of the presentation. For very popular contents, $RTT(c)$ will be close to d and for non popular ones, it will be close to $d \cdot len(c)$, the end-to-end delay.

To determine $p(c)$ we define L as the set of links belonging to the CCN overlay and $L(c) \subseteq L$ as the ordered set of links along the path from the requester to the server of content c, $L(c) = \{l_1(c), l_2(c), \ldots, l_{len(c)}(c)\}$, with $l_j(c)$ being the j^{th} link traversed by the Interest packets issued by the requester. The congestion probability $p(c)$ can hence be written in this compact form by summing over all links that can be potentially crossed by chunk's Data packets:

$$p(c) = \sum_{i=1}^{len(c)} \omega_i(c) \prod_{j<i}[1-\omega_j(c)][1-\prod_{k\leq i}(1-\pi(l_k(c))] \quad (4)$$

$$= 1 - \sum_{i=1}^{len(c)} \omega_i(c)[1-\pi(l_i(c))] \prod_{j<i}[1-\omega_j(c)][1-\pi(l_j(c))]. \quad (5)$$

$\pi(l_i(c))$ denotes the probability that a link at i hops distance from the requester of the content sends a congestion notification.

The hit ratio $\omega_i(c)$ is a function of content popularity [7] and hence one can assume it to be input of the problem that depends on how frequently the different contents are asked and the size of CCN caches.

Closing the loop with a view at links: Whereas $\omega_i(c)$ can be seen as an input to the problem, the congestion probability at a link $l \in L$, $\pi(l)$, is a function of the traffic on this link and the available capacity. If $b(l)$ denotes the bitrate of a link (physical capacity), one can derive an expression for the data traffic at link l, and hence for the link congestion probability, by summing over the different downloads going through it. Such expressions for all links closes the loop between AIMD and network links and provides the missing system of equations to solve the entire problem. Note that in our formulation of the problem, a content download goes through a link l if this link is on the path between the requesting client and the original server. The Interest packets themselves can be satisfied by any intermediate router and might not even reach the link l.

For each content download $c \in C$, we denote by $T(c)$ its throughput (see Eq. (2)), and by $\Omega(c,l)$ the overall hit ratio for this content at all CCN routers between the requesting client and link l, i.e., $1-\prod_{j<ord_l(c)}(1-\omega_j(c))$. The load caused by this download on link l then becomes $T(c)\cdot(1-\Omega(c,l))$. We sum the load of all downloads through link l to obtain its total load, $\Lambda(l)$. This total load on link l drives its congestion probability, which can be assumed equal to the ratio of dropped Data packets, $\pi(l) = max(0,(\Lambda(l)-b(l))/\Lambda(l))$. This probability is a function of content hit ratio and congestion probabilities at other links.

By combining the set of expressions of $\pi(l)$ with the expressions of the throughput of the different downloads as provided in Eq. 2, we can derive a system of equations with the $\pi(l)$ as unknowns, that we can later solve for the throughput of each download. Pseudocode 1 summarizes the rationale of our approach.

3. USE CASES

The above model can be applied to any network topology and can be solved numerically for the download rate of every content, and hence for the utilization of network resources. Herein, we specify the results and pro-

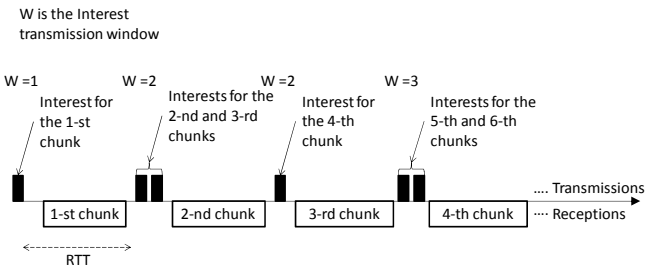


Figure 1: AIMD probing phase applied to CCN.

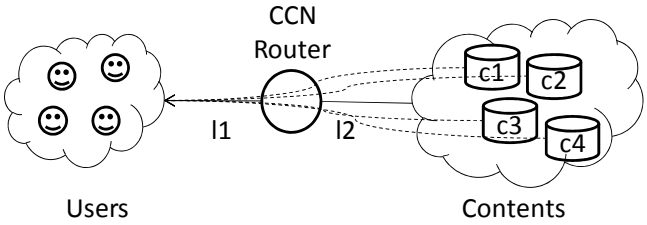


Figure 2: Single hop scenario (H=1).

Algorithm 1 CCN throughput calculation routing

for each link $l \in L$ **do** ▷ For every link l of the CCN overlay
  $\Lambda(l) \leftarrow 0$ ▷ Aggregate throughput at link l
  for each download of content $c \in C$ **do** ▷ and for every download through l
    $\Omega(c,l) \leftarrow 1 - \prod_{j<ord_l(c)}(1-\omega_j(c))$ ▷ Part of c that can be retrieved from CCN routers downstream l
    $\Lambda(l) \leftarrow \Lambda(l) + T(c)\cdot(1-\Omega(c,l))$ ▷ $T(c)$ can be evaluated from Eq. (2)
  end for ▷ T(c) is a function of $\pi(l),\ l \in L(c)$ and $\omega_i(c),\ i = 1 : len(c)$
  $\pi(l) \leftarrow max\left[0, \frac{\Lambda(l)-b(l)}{\Lambda(l)}\right]$ ▷ Congestion probability at link l
end for ▷ A system of equations having $\pi(l)$ as unknowns is now available

vide closed-form expressions for two simple but yet very useful cases, the one-hop and the multi-hop chain scenarios, with congestion at the access close to the clients. The access is often the bottleneck in today's Internet, and this very likely going to continue with CCN as the in-network caching will relax the core and put further the pressure on the access.

Our main focus is on the way CCN will affect content download throughput, the fairness among different contents, and the load at the server. For this, we introduce two metrics that provide useful insights on aforementioned issues. The first parameter $\eta(c)$, for content $c \in C$, represents the ratio among the download rate achieved by CCN (i.e., using AIMD) over the one achieved by TCP/IP over the same network topology. If $\eta(c) > 1$ (resp. $\eta(c) < 1$), it means that the CCN will improve (resp. worsen) the throughput of download c with respect to TCP/IP. The second performance indicator γ expresses the overall load reduction at the server side in a CCN network over compared to the one the server would experience with TCP/IP. The closer to one is γ the larger is the contribution of CCN in decreasing the traffic load at the server side.

3.1 A single hop scenario

PROPOSITION 1. *In a single hop scenario (see also Fig. 2) composed of: (i) a network of users that trigger N downloads, i.e., $c_1 \dots c_N$; (ii) 2 links, i.e., l_1 and l_2; and (iii) one CCN router, assuming only l_1 is congested, the $\eta(c_i)$ metric for the ith download can be approximated as:*

$$\eta(c_i) \approx \frac{1/RTT(c_i)}{\frac{\sum_{j=1}^{N} 1/RTT(c_j)}{N}}. \quad (6)$$

PROOF. If we consider the download c_i, according to Eqs. (3) and (4), we can derive both RTT and p:

$$RTT(c_i) = d \cdot [2 - \omega_1(c_i)], \quad (7)$$

$$p(c_i) = \pi(l_1) + \pi(l_2)[1-\omega_1(c_i)][1-\pi(l_1)], \quad (8)$$

which, combined with Eq. (2), give the following download rate:

$$T(c_i) = \frac{K}{d \cdot [2-\omega_1(c_i)]\sqrt{p(c_i)}}. \quad (9)$$

Since it has been assumed that only link l_1 is congested, it follows:

$$\begin{cases} b(l_1) = [1-\pi(l_1)] \sum_{i=1}^{N} \frac{K}{d\cdot[2-\omega_1(c_i)]\sqrt{\pi(l_1)}}, \\ \pi(l_2) = 0. \end{cases} \quad (10)$$

By assuming this approximation holds for moderate congestion: $\frac{[1-\pi(l_1)]}{\sqrt{\pi(l_1)}} \approx \frac{1}{\sqrt{\pi(l_1)}}$, we can now derive $\sqrt{\pi(l_1)}$:

$$\sqrt{\pi(l_1)} \approx \frac{1}{b(l_1)} \sum_{i=1}^{N} \frac{K}{d\cdot[2-\omega_1(c_i)]}. \quad (11)$$

The expression for $T(c_i)$ follows by inserting this expression into Eq. (9). We move to the calculation of the download rate with TCP/IP under the same assumptions on the network. The main difference with CCN is that routers do not cache data and so content

is only served by the server. The TCP/IP throughput for download c_i hence becomes [1]:

$$\hat{T}(c_i) = \frac{K}{2d\sqrt{\hat{\pi}(l_1) + \hat{\pi}(l_2) - \hat{\pi}(l_1)\hat{\pi}(l_2)}}. \quad (12)$$

When only link l_1 is congested, we get:

$$\begin{cases} b(l_1) = [1 - \pi(\hat{l_1})] \sum_{i=1}^{N} \frac{K}{2d\sqrt{\pi(\hat{l_1})}}, \\ \pi(l_2) = 0, \end{cases} \quad (13)$$

which gives for the loss rate at link l_1:

$$\sqrt{\pi(\hat{l_1})} \approx \frac{1}{b(l_1)} \sum_{i=1}^{N} \frac{K}{2d}. \quad (14)$$

Using the above expressions, the throughput gain for the ith download $\eta(c_i) = \frac{T(c_i)}{\hat{T}(c_i)}$ after deploying CCN can be expressed us:

$$\eta(c_i) = \frac{2d}{RTT(c_i)} \sqrt{\frac{\pi(\hat{l_1})}{\pi(l_1)}}. \quad (15)$$

The proof follows by exploiting results in (11) and (14). □

Proposition 1 means that the throughput gain using CCN is inversely proportional to the mean round-trip time. It is also proportional to the harmonic mean of round-trip time over all downloads (having different popularity). Since the mean round-trip time decreases with popularity because of a higher hit rate (7), popular contents will get higher throughput than non-popular ones. In our case, the difference can be up to a factor of two between the most popular content and the least popular one (a chain of two links of equal delay). As we will see later, popular contents will not realize high gain with respect to TCP/IP though since they drive the harmonic mean of round-trip with their popularity. The non popular contents from their side can lose up to a factor of two with respect to TCP/IP.

Proposition 2. *Under the same hypotheses of Proposition 1, the server load reduction* $\gamma = 1 - \frac{\Lambda(l_2)}{\Lambda(\hat{l_2})}$ *can be approximated as follows:*

$$\gamma \approx 1 - \frac{\sum_{i=1}^{N} \frac{1-\omega_1(c_i)}{2-\omega_1(c_i)}}{\sum_{i=1}^{N} \frac{1}{2-\omega_1(c_i)}}. \quad (16)$$

Proof. With CCN, the load a content puts on the server is equal to $T(c_i).[1 - \omega_1]$, whereas with TCP/IP it is simply equal to $\hat{T}(c_i)$. By summing load over all contents, we get:

$$\gamma = 1 - \frac{\sum_{i=1}^{N} \frac{K \cdot [1-\omega_1(c_i)]}{d \cdot [2-\omega_1(c_i)]}}{N \cdot k/2d} \sqrt{\frac{\pi(\hat{l_1})}{\pi(l_1)}}. \quad (17)$$

[1]In this analysis, all variables will reported with a *hat*, i.e., $\hat{x}$, to refer to the value of variable x one would observe in a classical TCP/IP network.

This expression, if combined with (11) and (14) provides the proof. □

3.2 H-hops chain topology

In a more general chain topology:

$$\eta(c_i) = \frac{(H+1) \cdot d}{RTT(c_i)} \sqrt{\frac{\pi(\hat{l_1})}{\pi(l_1)}}. \quad (18)$$

Following similar arguments as for (11) and (14), we can write:

$$\sqrt{\frac{\pi(\hat{l_1})}{\pi(l_1)}} \approx \frac{N}{\sum_{j=1}^{N} \frac{(H+1) \cdot d}{RTT(c_j)}}, \quad (19)$$

which, combined to (18) provides this result:

$$\eta(c_i) = \frac{1/RTT(c_i)}{\frac{\sum_{j=1}^{N} 1/RTT(c_j)}{N}}. \quad (20)$$

This is the same result as in the simple 1-hop scenario. The difference is in the values taken by $RTT(c_i)$ which now ranges from d for very popular contents (served by the first router) to $H \cdot d$ for non popular ones (served by the server). It follows that the difference in download rates between popular and non popular contents can be up to a factor of $H + 1$, H being the number of hops.

For what concerns the reduction of load on the server due to CCN:

$$\gamma = 1 - \frac{\Lambda(l_{H+1})}{\hat{\Lambda(l_{H+1})}} = \frac{\sum_{i=1}^{N} \frac{K \cdot \prod_{j=1}^{H} [1-\omega_j(c_i)]}{RTT(c_i)}}{N \cdot k/[(H+1)d]} \sqrt{\frac{\pi(\hat{l_1})}{\pi(l_1)}}. \quad (21)$$

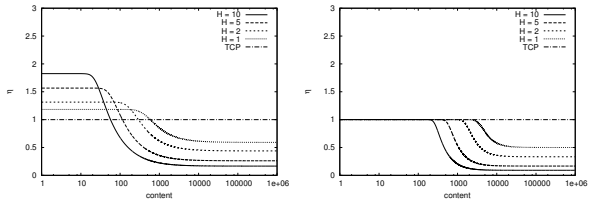
By considering (19), we get this expression:

$$\gamma = 1 - \frac{\sum_{i=1}^{N} \frac{\prod_{j=1}^{H} [1-\omega_j(c_i)]}{RTT(c_i)}}{\sum_{j=1}^{N} \frac{1}{RTT(c_j)}}. \quad (22)$$

4. EVALUATION

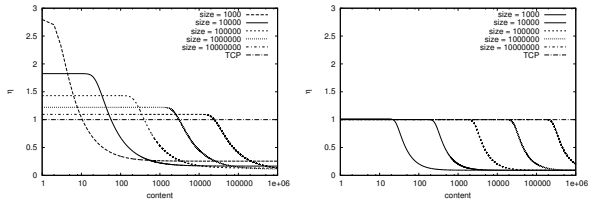
In this section, we compare the performances of CCN with AIMD with those with actual TCP/IP. To that aim, we first determine the gain in terms of download rates and then the impact of caching on the fairness. For this analytic evaluation, we consider different network configurations and two traffic patterns. The first traffic pattern, popularity difference between contents is small while the in the other some contents are much more popular than any other contents. In both cases, we use a zipf distribution for content popularity together with $1,000,000$ contents (forming the set C) and $N = 100,000,000$ downloads. The zipf α parameter of the first pattern is set to 1.1 and 2 for the second pattern.

We consider a network composed of a chain of LRU caches in our simulation, but vary the number of hops H (i.e., caches) in the network $H \in \{1, 2, 5, 10\}$, and fix the end-to-end delay to allow fair comparison between the different CCN configurations and standard



(a) $\alpha = 1.1$ (b) $\alpha = 2$

Figure 3: Impact of the popularity on the throughput gain.



(a) $\alpha = 1.1$ (b) $\alpha = 2$

Figure 4: Impact of the total caching capacity on the throughput gain.

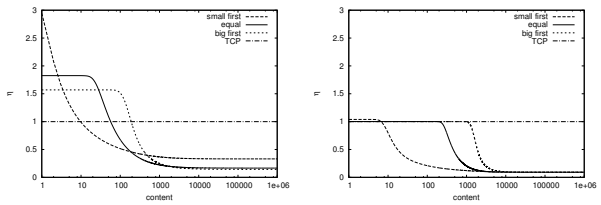
TCP/IP. Without loss of generality, this delay is taken equal to 100 ms (or $\frac{100}{H+1}$ ms at each hop). Analytic expressions for the hit rates on each cache in such networks is given in [7]. The size of the caches has an impact on the hit rates. Therefore, we simulate different overall capacity space in the network and evaluate the impact of different memory allocation policies among the CCN routers. More precisely, we use three different allocation policies. The `small first` policy is such that the size of the memory allocated to each router exponentially increases with the distance from the client, whereas the `big first` allocation policy behaves in a contrary way. Finally, the `equal` allocation policy uniformly distributes memory among available routers.

4.1 Throughput gain

In CCN, popular contents tend to be cached close to the clients and least popular contents are cached close to the server or not cached at all. On the contrary the popularity has no impact on the delay of a content in the case of TCP/IP since contents are always provided directly by the server. However, the download rate is inversely proportional to the mean round-trip time for long transfers in TCP/IP [1] and in CCN, the throughput is also influenced by the hit rate Eq. (2) which is a function of the popularity. We thus evaluate $\eta(c)$, the throughput gain for content c observed by the client. Fig. 3 shows $\eta(c)$ for every content c using 10,000 total caching capacity and the `equal` memory allocation policy. It is not surprising to observe that the more the content is popular (i.e., the lower its delay) the higher is its throughput gain. However, we can see that when the popularity distribution is steep, the throughput gain for the most popular contents is close to 1 and close to $\frac{1}{H+1}$ for the least popular contents. Indeed, when popular contents are much more requested than the other contents, the caching for them at the first hop is very efficient, leading to almost equal delay and equal throughput for all of them. Given their popularity, they drive the harmonic mean of the RTT over all contents and make it equal to their own RTT. According to the expression of $\eta(c)$ in Eq. (20), the gain for them is hence almost equal to one, whereas for least popular contents their throughput decreases by a factor of $H + 1$ compared to standard TCP/IP. A practical explanation for this behavior is that most popular contents saturate the access link in both TCP/IP and CCN. Since they realize equal throughput in both cases, they don't notice an increase in their throughput when deploying CCN with steep content popularity demand. This is clearly not the case when the popularity distribution is gradual where caching is less efficient and worsen the hit rate of popular contents. The harmonic mean of the RTT hence increases and the gain for most popular contents can reach larger values than 1 as expressed in Eq. (20), whereas for others it drops below one. In this case, the access is always saturated by most popular contents, but given they have different throughput, they behave differently compared to standard TCP/IP; their throughput in this latter case is the same. Concerning non popular contents, they keep loosing compared to standard TCP/IP even if they realize a slightly better throughput than in the case of steep popularity because of a larger harmonic mean of RTT.

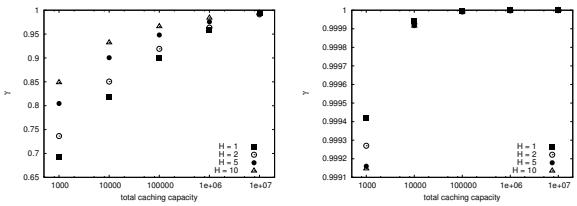
4.2 Caching capacity and throughput gain

Fig. 4 determines the impact of the total caching capacity on the throughput gain, for the `equal` memory allocation policy. As expected, the less the caching capacity of the network, the higher the throughput gain for the most popular contents since they are fewer and fewer to leverage the small RTT provided by caching and hence the better throughput of AIMD. The harmonic mean of RTT increases when the network caches less contents, leading to the observed degradation in the throughput of non popular contents. Fig. 5 complements the study by showing the impact of the cache placement policy. For the sake of readability, Fig. 5 only shows the results for a total of 10,000 caching entries and $H = 10$ hops. On the one hand, when using the `small first` cache sizing policy, a very limited set of popular contents is able to reach a remarkable gain with respect to TCP, whereas remaining ones will experience less throughput than in a TCP/IP network. On the other hand, pushing memory towards client (i.e., `big first`) reduces the harmonic mean of RTT and



(a) $\alpha = 1.1$ (b) $\alpha = 2$

Figure 5: Impact of the cache placement on the throughput gain.



(a) $\alpha = 1.1$ (b) $\alpha = 2$

Figure 6: Impact of the total cache capacity on the server load.

increases the number of content with a short RTT. As a result more contents have a gain, but the non popular contents (i.e., those with a longer RTT) suffer from a higher throughput degradation. A large harmonic mean of RTT (e.g., with `small first`) implies longer paths but limits the throughput degradation for non popular contents. On the contrary, a shorter harmonic mean of RTT (e.g., with `big first`) shortens the paths but degrades more the throughput of non popular contents.

4.3 Load on server

Fig. 6 shows the impact of the total caching capacity and the CCN topology (i.e., number of routers) on the server load. The figure plots γ, where γ is defined in Eq. (22) as being the server load reduction with CCN compared to TCP/IP. As one can expect, there is a considerable reduction of server load with CCN and this reduction consistently increases with the overall caching capacity. Indeed, the more the caching capacity in the network, the more likely contents are cached and server does not intervene. The gain from increasing the total capacity has more impact on the server load reduction than the CCN topology itself (i.e., how this total capacity is placed). The figure also shows that the more the total capacity, the less the topology impacts this gain.

5. CONCLUSION AND FUTURE RESEARCH

In this work the performance the CCN architecture adopting classic AIMD congestion control has been modeled and studied from a novel point of view. Our work is mainly centered on the performance gain/loss every user would experience based on the popularity of contents it retrieves from the network with respect to the current Internet. Results clearly indicate that plain AIMD plus CCN can severely worsen the download throughput of not popular contents due to subtle interactions with in-network caching strategies. The way cache memories are distributed within the topology has been investigated too, showing that for small and heterogeneous cache spaces, placing the biggest caches close to clients improves performance due to a smaller RTT on average. On the other hand, CCN can significantly reduce the load at the server side whatever the cache allocation strategy. We plan to extend this study to more complex topologies and to design a congestion control scheme able to limit the download throughput unfairness between contents with different popularities.

6. REFERENCES

[1] E. Altman, K. Avrachenkov, and C. Barakat. A stochastic model of tcp/ip with stationary random losses. *IEEE/ACM Trans. Netw.*, 13(2), Apr. 2005.

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

[3] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino. Modeling data transfer in content-centric networking. In *Int. Teletraffic Congress, (ITC)*, 2011.

[4] D. M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Comput. Netw. ISDN Syst.*, 17(1), June 1989.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *ACM CoNEXT '09*, 2009.

[6] L. Muscariello, G. Carofiglio, and M. Gallo. Bandwidth and storage sharing performance in information centric networking. In *ACM SIGCOMM workshop on Information-centric networking (ICN '11)*, 2011.

[7] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou. Modelling and evaluation of ccn-caching trees. In *10th Int. IFIP TC 6 conference on Networking*, 2011.

[8] D. Rossi and G. Rossini. On sizing CCN content stores by exploiting topological information. In *IEEE INFOCOM, NOMEN Worshop*, 2012.