# AIMD and CCN: Past and Novel Acronyms Working Together in the Future Internet [*]

Damien Saucez
Inria
Sophia Antipolis, France
damien.saucez@inria.fr

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

Chadi Barakat
Inria
Sophia Antipolis, France
chadi.barakat@inria.fr

## ABSTRACT

Content-centric networking (CCN) is a new paradigm to better handle contents in the future Internet. Under the assumption that CCN networks will deploy a similar congestion control mechanism than in today's TCP/IP (i.e., AIMD), we can build an analytical model of the bandwidth sharing in CCN based on the "square-root formula of TCP". With this model we compare CCN download performance to what users get today. We consider different factors such as the way CCN routers are deployed, the popularity of contents, or the capacity of links and observe that when AIMD is used in a CCN network less popular content throughput is massively penalized whilst the individual gain for popular content is negligible. Finally, the main advantage of using CCN is the decrease of load at the server side. Our observations advocate the necessity to clearly define the notion of fairness in CCN and to design a proper congestion control to avoid less popular contents to become hardly accessible in tomorrow's Internet.

## Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks—*Internet (e.g., TCP/IP)*

## General Terms

Performance

## Keywords

AIMD, Caching, Content-Centric Networking, square-root formula

## 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 (e.g., Web, Email, etc). However, Internet usage undergone 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, most of the Internet traffic of our days is related to content dissemination, like file sharing and media streaming [7]. Unfortunately, the current Internet architecture does not handle the challenges of this new *content-centric* context in an effective way and thus has to be adapted to provide inherent support of content distribution.

Among the Future Internet solutions, the *information-centric networking* (ICN) approach is one of the most promising ones [1] and the *Content-Centric Networking* (CCN) [6] has the most promising ICN architecture. CCN presents the main functional blocks for efficient content support such as content protection, routing by content names, and in-network caching. In CCN, contents are independent entities that users can retrieve without having any awareness about the location of service providers. CCN communications are receiver-driven [6]. A user asks for contents by issuing Interest packets, which are routed toward the nodes possessing the required information; such nodes reply with Data packets, that are routed along the reverse path followed by the Interest packets. Every intermediate node can cache the forwarded contents, reducing network congestion and servers load, and improving 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 [9, 8, 4, 10]. 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. Carofiglio et al. propose a window-based Interest flow Control Protocol (ICP) driven by an Additive Increase Multiplicative Decrease (AIMD) mechanism to regulate the Interest rate at the downloader [3]. 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 In-

---

[*]This work was partially funded by the Apulia Region Project PS 025 (ICT supporting logistic services: a model of organized market) and the Italian National Operative Program ERMES (Enhance Risk Management through Extended Sensors).

*CSWS'12,* December 10, 2012, Nice, France.

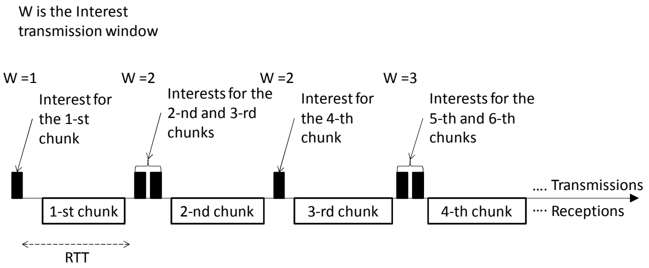


**Figure 1: AIMD probing phase applied to CCN.**

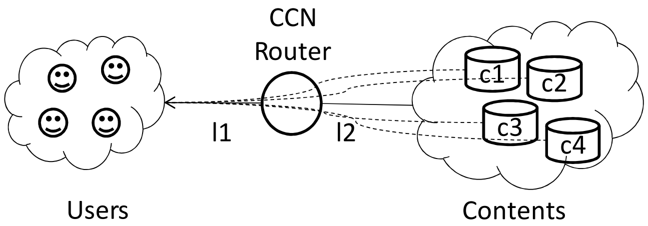


**Figure 2: Single hop scenario (H=1).**

ternet to answer the fundamental question: *Who wins and who looses with CCN?*.

Pure performances are important but we must understand how everyone's performance would change if CCN were deployed tomorrow. To answer this question, we follow the assumption that CCN clients requesting contents will deploy a congestion control mechanism based on AIMD to control the rate at which they send Interest packets, as does TCP [5]. We can then use the "square-root formula of TCP" to define an analytic throughput model for CCN and 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. By coupling the AIMD throughput model with expressions for hit/miss rates in network of caches and for link loss rates, we devise a general mathematical model for AIMD in CCN and derive preliminary results by applying our model to the study of a chain topology. Provided that the underlying assumptions of the present analysis hold, our findings can be summarized as follows. CCN helps in reducing the load at the server but is biased against less popular contents. These contents will see their throughput decrease compared to toady's Internet, and the decrease factor in a chain topology can be up to the number of CCN hops over the path to the content. On the contrary, popular contents will get higher end-to-end throughput but given the bandwidth limitation inside the network, this increase is limited and might even go unnoticed. Moreover, the global server load reduction is function of the total caching capacity and is little sensitive to the number of hops in the considered chain topology. Even though this work is built around CCN, our results can be applied to any information-centric architecture presenting similar functionalities.

Our analysis confirms the benefits of CCN in terms of load reduction on the network and the servers, but points out a problem of fairness against low popularity contents. Our results advocate the necessity of clearly defining the notion of fairness in CCN in order to carefully design a congestion control mechanism, and hence a transport protocol, so as 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

**Table 1: Model Notation.**

| Symbol | Meaning |
|---|---|
| $L$ | Set of links |
| $N$ | Number 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 $l \in L$ |
| $b(l)$ | Capacity of a link $l \in L$ |
| $\Lambda(l)$ | 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 |

In this paper, we focus on the bandwidth sharing problem in Content-Centric Networking [6] and compare it with today's Internet. To that aim, we leverage the well-known results regarding the throughput of AIMD congestion control within TCP/IP, known under "the square-root formula of TCP" [2]. 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. In such an environment, a rate control mechanism is necessary to avoid congestion that would degrade performance. However, there is no standard such control in CCN so far. Therefore, we assume that CCN clients will implement a window-based *Additive Increase Multiplicative Decrease* (AIMD) congestion control, as does TCP [5]. Congestion is detected by Data packet losses and the congestion 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 does not experience losses as Interest packets are of

**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)
                                                                    ▷ T(c) is a function of $\pi(l),\ l \in L(c)$ and $\omega_i(c),\ i = 1 : len(c)$
    end for
    $\pi(l) \leftarrow max\left[0, \frac{\Lambda(l)-b(l)}{\Lambda(l)}\right]$   ▷ Congestion probability at link $l$
end for                                                             ▷ A system of equations having $\pi(l)$ as unknowns is now available
```

small size compared to Data packets and hence do not congest the network. 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 inflight Interest packets waiting for their corresponding Data packets (see Fig. 1). The AIMD for CCN controls the window $W$ in this way is:

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

If $p(c)$ is the probability that a Data packet for a chunk of content $c$ is lost due to congestion[1], and $RTT(c)$ is the mean round-trip time to retrieve chunks of content $c$, the mean download rate $T(c)$ for content $c$ is:

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

where $K$ is a constant. This is known as the square-root formula for AIMD throughput [2]. The main difference in our case is that copies of the chunks can be found on the path to the server, and so the round-trip can be shorter 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, which in its turn depends on the popularity of the requested content $c$.

**Throughput specification to CCN:** We require expressions for both congestion probability $p(c)$ and mean round-trip time $RTT(c)$ to determine the throughput of a content download $c$.

$RTT(c)$ depends on the number of CCN hops between the node that requests the content and the CCN routers providing copies of the chunks. In fact, if a content is very popular, it is likely to be cached at intermediate CCN routers, so that a small $RTT(c)$ is be obtained. On the contrary, contents with a very low popularity are 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 related to the content popularity.

If $\omega_i(c)$ is the probability that a given chunk is available at $i$ hops distance from the node 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:

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

where $d$ is the average link delay, which we suppose to be the same for all links for simplicity of the presentation, and $len(c)$ is the number of links located on the path between the requester node and the original server. For very popular contents, $RTT(c)$ will be close to $d$ and for less popular ones, it will be close to the end-to-end delay $d \cdot len(c)$.

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 content $c$'s server, $L(c) = \{l_1(c), l_2(c), \ldots, l_{len(c)}(c)\}$, with $l_j(c)$ being the $j^{\text{th}}$ 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 Data packets:

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

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

$\pi(l_i(c))$ denotes the probability of a congestion event at the link at $i$ hops distance from the requester. It is function of the load on links and is to be determined.

The hit ratio $\omega_i(c)$ is a function of content popularity [8] and hence we assume it to be the input of the problem that depends on how frequently the different contents are requested and the size of CCN caches.

**Closing the loop with a view at links:** Whereas $\omega_i(c)$ is 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 capacity of link $l$, we 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 provide the missing system of equations to solve the entire problem. Note that in our problem formulation a content download goes through a link $l$ if this link is on the path between the requesting client and the original server. 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

---

[1] Losses of Data packets, $p(c)$, can be learned either explicitly with intermediate routers sending congestion notification, or implicitly with timeouts and sequence number holes detection as in current TCP.

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 network resource utilization. In this section, we specify the results and provide 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 client. The access is often the bottleneck in today's Internet, and this is 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 would affect content download throughput, the fairness among different contents, and the load at servers. For this, we introduce two metrics that provide insights on the aforementioned issues. On the one hand, for content $c \in C$, $\eta(c)$ models the ratio of the download rate achieved by CCN (i.e., using AIMD) and the one achieved by TCP/IP over the same network topology. $\eta(c) > 1$ (resp. $\eta(c) < 1$) means that CCN will improve (resp. worsen) the throughput of download $c$ with respect to TCP/IP. On the other hand, $\gamma$ compares the overall load at the server side in a CCN network with the one the server would experience with TCP/IP. The closer to zero is $\gamma$ the larger is the contribution of CCN in decreasing the traffic load at the server side.

To evaluate $\eta$ and $\gamma$ we have to introduce the new variables $\hat{T}(c)$ and $\hat{\Lambda}(l)$, defining the download throughput for content $c$ and the load at link $l$, respectively, if classic TCP is used instead of CCN.

Proposition 1. *In a chain topology with $H$ hops (see also Fig. 2) composed of $(i)$ a network of users that trigger $N$ downloads, i.e., $c_1 \dots c_N$; $(ii)$ $H+1$ links, i.e., $l_1 \dots l_{H+1}$; and $(iii)$ $H$ CCN routers, assuming only $l_1$ is congested, the $\eta(c_i)$ metric for the ith download can be approximated as:*

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

The Proof is reported due to lack of space in [11].

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 at first routers, popular contents get higher throughput than non-popular ones. In our case, the difference can be up to a factor of $H+1$ between the most popular contents and the less popular ones (a chain of two links of equal delay). However, as we show later in the text, popular contents do not realize high gain with respect to TCP/IP though since they drive the harmonic mean of round-trip time with their popularity. The download of less popular contents, instead, can significantly be impaired with respect to today's TCP/IP.

Proposition 2. *Under hypotheses of Proposition 1, the server load ratio $\gamma = \frac{\Lambda(l_{H+1})}{\hat{\Lambda}(l_{H+1})}$ can be approximated as follows:*

$$\gamma = \frac{\Lambda(l_{H+1})}{\hat{\Lambda}(l_{H+1})} = \frac{\sum_{i=1}^{N} \frac{\prod_{j=1}^{H}[1-\omega_j(c_i)]}{RTT(c_i)}}{\sum_{j=1}^{N} \frac{1}{RTT(c_j)}}. \tag{7}$$

Again, we report the Proof in [11].

# 4. NUMERICAL EVALUATION

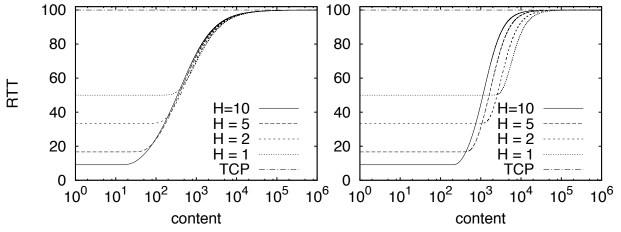
In this section, we compare the performance of the combination of CCN and AIMD with those of current TCP/IP for different traffic patterns and chain topologies. To that aim, we first determine the gain in terms of download rate and then the impact of caching on the fairness.

In our evaluation, we always consider $1,000,000$ contents forming the set $C$ and a total of $N = 100,000,000$ downloads with a content popularity distribution following a Zipf law. We change the way traffic behaves by modifying the steepness of the popularity distribution. We take two extreme cases; on the one hand, the difference of popularity between contents is low which is materialized with an *alpha* Zipf's parameter of 1.1. On the other hand, the difference of popularity between contents is high with *alpha* equals to 2.

In this paper, we consider networks composed of a chain of LRU caches. For the sake of the evaluation, we fix the end-to-end delay (i.e., the delay between the two ends of the chain) to 100 ms but change the number of hops $H$ (i.e., caches) in the network and adapt the hop delay so that the end-to-end delay remains the same. Our evaluation considers $H \in \{1, 2, 5, 10\}$ with a delay at each hop equal to $\frac{100}{H+1}$ ms. Analytic expressions for the hit rates of caches in chain topologies are given in [8] which capture the influence of cache sizes and network traffic load. We simulate different caching capacity in the network and evaluate the impact of different memory allocation policies among caches. 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 caches.

## 4.1 RTT behavior

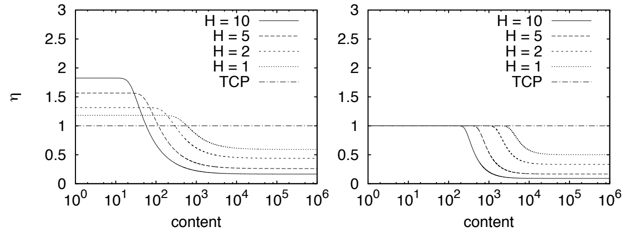
The mean round-trip time has a direct impact on the download rate as can be seen in Eq. (6). From the literature of TCP/IP and AIMD [2], it is known that the download rate for long transfers is inversely proportional to the mean round-trip time. Fig. 3 gives the mean RTT defined in Eq. 3, for each content in $C$, ordered by its popularity, for the two popularity distributions and for different values of $H$. The total caching capacity is equal to $10,000$ and the `equal` memory allocation policy is used. As one can expect, the popularity of content has a direct impact on its mean delay when CCN is used, which is not the case in TCP/IP where all contents are served by the server. 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. As a result, the most popular contents have a delay close to $100/H$ ms while the least popular contents have a 100 ms delay. In addition, we observe that if the popularity distribution is steep, caching is more selective leading to more contents served with low delay.



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

**Figure 3: Impact of the chain length on the mean RTT.**

## 4.2 Throughput gain

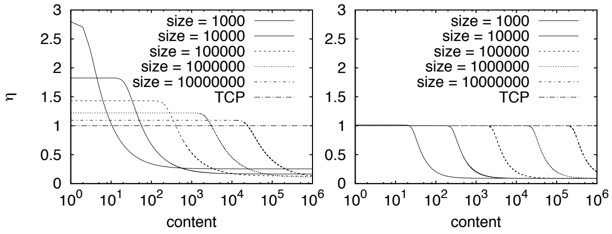


(a) $\alpha = 1.1$ (b) $\alpha = 2$

**Figure 4: Impact of the chain length on the throughput gain.**

As said before, popularity leads to a smaller round-trip time in CCN which is not the case in TCP/IP where all contents are served on an end-to-end basis. We know from the square-root formula that the throughput of AIMD is inversely proportional to the mean round-trip time [2]. In CCN, the round-trip time is influenced by the hit rate (i.e., Eq. (2) ) which is a function of the popularity, thus one should expect different throughput for different popularity. We thus evaluate $\eta(c)$, the throughput gain for content $c$ observed by the client, based on Eq. (6). Fig. 4 shows $\eta(c)$ for every content $c$ with a total of 10,000 caching entries allocated among caches with the `equal` 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 (e.g., $alpha = 2$), the throughput gain for contents having a high popularity is close to 1 and close to $\frac{1}{H+1}$ for less popular ones. 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. (6), the gain for them is hence almost equal to one, whereas the throughput for less popular contents decreases by a factor of $H + 1$ compared to standard TCP/IP. A practical explanation for this behavior is that contents having a high popularity saturate the access link in both TCP/IP and CCN. Since they realize equal throughput in both cases, they do not 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 reduces the hit rate of popular contents. The harmonic mean of the RTT hence increases and the gain for very popular contents can reach larger values than 1 as expressed in Eq. (6), whereas for others it drops below one. In this case, the access is always saturated by popular contents, but given they have different throughput; they behave differently compared to standard TCP/IP. Concerning less 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.

## 4.3 Caching capacity and throughput gain

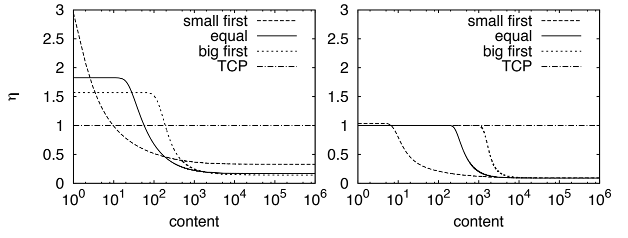


(a) $\alpha = 1.1$ (b) $\alpha = 2$

**Figure 5: Impact of the total caching capacity on throughput gain.**

Fig. 5 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 very 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 fewer contents, leading to the observed reduction in the throughput of less popular contents. Fig. 6 complements the study by showing the impact of the cache placement policy. For the sake of readability, Fig. 6 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 increases the number of content with a short RTT. As a result more contents have a gain, but the less popular ones (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 less popular contents.

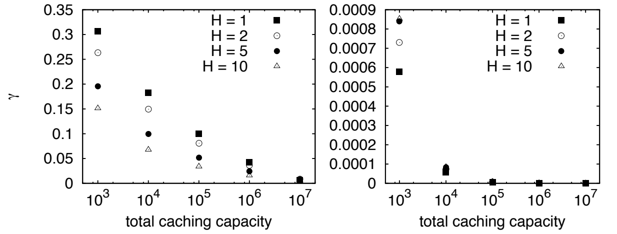
On the contrary, a smaller harmonic mean of RTT (e.g., with `big first`) shortens the paths but degrades more the throughput of less popular contents.



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

**Figure 6: Impact of the cache placement on the throughput gain.**

## 4.4 Server load



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

**Figure 7: Impact of the total cache capacity on the server load.**

Fig. 7 shows the impact of the total caching capacity and the chain topology (i.e., number of routers) on the server load. The figure plots $\gamma$, where $\gamma$ is defined in Eq. (7) as being the server load with CCN compared to TCP/IP. As one can expect, there is a considerable reduction of server load with CCN and the load consistently decreases 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 than how memory is distributed in the topology. The figure also shows that the more the total capacity, the less the topology impacts this gain.

# 5. CONCLUSION

In this work, we have modeled and studied the performance of the CCN architecture adopting classic AIMD congestion control. Our work focuses on performance gain/loss every user would experience based on the popularity of contents she retrieves from the network with respect to the current Internet. Our results clearly point to a fairness issue if AIMD is used with CCN. Indeed, combining blindly AIMD and CCN can severely worsen the download throughput of less popular contents with respect to the today's Internet due to subtle interactions with in-network caching strategies. The way cache memories are distributed within chain topologies 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 independently of the cache allocation strategy. Our findings advocate the urge of clearly defining the notion of fairness in CCN and designing congestion control algorithms able to limit the unfairness observed between contents of different popularities. In addition, for the sake of generality, our model must be extended to complex topologies and validated with experiments.

# 6. REFERENCES

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of information-centric networking. *Communications Magazine, IEEE*, 50(7):26 –36, july 2012.
[2] 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.
[3] 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.
[4] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino. Modeling data transfer in content-centric networking. In *Int. Teletraffic Congress, (ITC)*, 2011.
[5] 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.
[6] 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.
[7] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, F. Jahanian, and M. Karir. ATLAS Internet Observatory 2009 Annual Report. Technical report, Arbor Networks, the University of Michigan and Merit Network, 2009.
[8] 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.
[9] 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.
[10] D. Rossi and G. Rossini. On sizing CCN content stores by exploiting topological information. In *IEEE INFOCOM, NOMEN Worshop*, 2012.
[11] D. Saucez, L. A. Grieco, and C. Barakat. AIMD and CCN: Past and Novel Acronyms Working Together in the Future Internet. http://hal.inria.fr/hal-00719793.