An extended Hop-by-hop Interest shaping mechanism for Content-Centric Networking

Natalya Rozhnova, Serge Fdida

An extended Hop-by-hop Interest shaping mechanism for Content-Centric Networking

Natalya Rozhnova
Université Pierre et Marie Curie (UPMC)
Sorbonne Université - CNRS
Laboratoire d'Informatique de Paris 6 (LIP6)
Paris, France
Email: natalya.rozhnova@lip6.fr

Serge Fdida
Université Pierre et Marie Curie (UPMC)
Sorbonne Université - CNRS
Laboratoire d'Informatique de Paris 6 (LIP6)
Paris, France
Email: serge.fdida@lip6.fr

***Abstract*—The evolution of the Internet has triggered a significant activity exploring new architectures among which the concept of Information-Centric-Networks (ICN) has emerged. Considering the various ICN solutions, Content-Centric Networking (CCN) is the one that received most attention. The design of CCN is progressing albeit many important issues still deserve a careful analysis and design. In this paper, we cover the important problem of congestion control in CCN and develop our hop-by-hop Interest Shaping mechanism (HoBHIS) framework. We present the foundation and general properties of our solution. We then study the behaviour of HoBHIS and demonstrate the fairness of resource sharing in presence of multiple active conversations. Further, we introduce a mechanism to control the user behavior in order to limit its ability to submit requests in excess creating a potential risk for buffer overflow. We demonstrate the effectiveness of the proposed solutions and provide a performance analysis using the implementation of our schemes in $ndnSIM$. We show that HoBHIS provides an adequate and efficient solution to the general problem of congestion control in CCN.**

I. Introduction

Content-Centric Networking (CCN) proposed by Van Jacobson et al. in [1] has been introduced to cope with the evolution of the internet usage towards massive content distribution. This architecture differs from the traditional host-based communication principle in many ways. One important change is that network addresses are replaced by content names to distribute information in CCN networks. Moreover, extensive in-network caching capabilities are introduced to benefit from the observation of the traffic flowing in the network. As a consequence, data packets can be delivered by any CCN router in addition to the source, according to various caching algorithms that are out of scope of this paper. In CCN, a request issued by a user is called $Interest$ and the part of the associated content is called $Chunk$. For convenience, we will name a stream of Interest/Chunk pairs a $Conversation$.

The key property of CCN is that one Interest retrieves at most one data packet. It enforces a flow balance in the CCN network that enables multiple Interests to be issued at once. Interest aggregation is an optimization that provides the ability to reduce the network load when multiple interests from different sources request the same content. This is achieved by aggregating them in the CCN router, forwarding a single interest upstream and carrying downstream a single copy of the associated content.

As a consequence of the CCN design principles, we have to regulate the stream of chunks as well as the stream of interests in order to avoid congestion and to improve network performance. We believe that, because of the specific design of CCN, hop-by-hop congestion control provides a suitable solution to this problem. We were first [2], to propose a hop-by-hop Interest shaping mechanism exploiting the one Interest - one Chunk rule enforced in CCN. In this scheme, each CCN router controls the future rate of data-chunks by shaping the rate of corresponding Interests it is pushing upstream. CCN routers dynamically adjust their Interest sending rate by monitoring the level of Chunks stored in their transmission buffer. The rate resulting from this operation is called the shaping rate. An important control function is the one used to derive an optimum shaping rate that maintains the queue length around an objective. Various important parameters are calibrated in order to efficiently operate the algorithm as a function of the network characteristics.

The important concern in CCN is related to the ability offered to a user to send interests without any limiting rate factor, creating a risk for buffer overflow and performance degradation. We introduce a new rate-based mechanism aiming to control the interest sending rate of a content receiver (a client or source of interest). The scheme introduces an exchange of control information between the CCN nodes and the client. The CCN nodes periodically send control packets with an explicit rate specifying the maximum sending rate for each Client on the path used by this conversation. In order to derive an optimum explicit rate, CCN routers use a control function based on the shaping rate computed by HoBHIS.

The remainder of the paper is organized as follows: Section II presents work related to congestion control in CCN. Section III provides a short introduction to CCN underlining the most meaningful properties for this work. In Section IV, we describe the hop-by-hop Interest shaping mechanism (HoBHIS) and its operating principles. We discuss in Section V, our tolerance mechanism to control the interest rate of a client when generated in excess and creating a risk for losses. We provide a performance evaluation of the solutions in Section VI. Section VII conludes the paper and proposes future work.

II. Related work

Hop-by-hop congestion control schemes have been widely studied in the past but only recently in the context of CCN.

Traffic control suited to CCN has been considered lately, mostly to study congestion and packet loss in CCN routers. The existing schemes can be classified into two categories: receiver-based and hop-by-hop mechanisms. The scheme introduced in [3] is one of the first window-based algorithms designed to control the Interest sending rate of a receiver in TCP-like environments. The proposed mechanism uses Chunks packets and associated timers as relevant signals to regulate the number of Interests sent by the receiver. One serious issue of this approach is to properly set the timeout value. The solution used in [3] sets the timer to the mean value of RTT, that, unfortunately, does not take into account the variability of content sources. The mechanism was later improved in [4] by adding route labels to the Chunks so that the client can identify multiple paths. CCTCP introduced in [5] also represents a receiver-driven control solution. The authors propose to list in each interest packet the subsequent Chunks the client intends to request. The network nodes indicate whether they have cached this Chunk. Thus, the receiver maintains separate congestion states for different content locations where future interest will be satisfied. A similar predictive approach has been proposed in [6] where the authors have proposed some improvements to reduce the complexity of the protocol.

As shown in the past, the receiver-driven schemes face a fairness problem in networks with heterogeneous RTTs. Some analysis and improvements can be found in [7], [8]. As caching is a key feature of CCN, the RTT related to the content sources appears as a crucial parameter for receiver-driven mechanisms. A recent work, [9], compares the performance of these schemes and concludes that the receiver-driven solutions based on timeouts are not suitable for CCN because of the unpredictability of the content locations.

A second category of congestion control mechanism uses hop-by-hop interest shaping rather than letting the receiver infer network congestion using timeouts. HoBHIS [2] was the first interest-based shaping mechanism developped for CCN. In [10] the authors present a fair sharing mechanism where Interests exceeding a Data fair rate are discarded. ICP introduced in [3] was then extended in [11] where the authors describe a joint hop-by-hop congestion control mechanism that is found similar to HoBHIS. The basic idea developped in this general approach is to shape the Interest of every conversation flowing through a CCN node to control congestion of Chunks packets and improve network performance. Other proposals based on interest shaping have been presented in [12], [13]. The proposed scheme shapes the interests so that the associated data rate is equal to some predefined ratio of the reverse link capacity. Recently, a different solution was proposed in [14]. The mechanism identifies the interdependence between the interests and its associated chunks and analyzes the fair resource allocation in presence of bidirectional traffic.

The work related to traffic engineering in CCN has developed recently. However, the various contributions are still fragmented and do not address specific important issues. The aim of this paper is to provide a comprehensive solution for congestion control in CCN.

III. Content-Centric Networking

In this section we provide a short introduction to CCN in order to highlight the most important architectural features useful in the context of our contribution.

Any content requested by a Client (or a source of Interests) can be divided into a number of data packets. In order to request a given piece of content, a Client will have to send an interest packet. This interest packet will be forwarded to a location where the content is stored. It could be the server or a CCN router where the content has been cached. As a consequence, in order to fetch a given content, the Client will have to submit as many Interest packets as the number of chunks that exist for that particular content. We observe that an Interest triggers the reception of a single chunk transmitted on the reverse path used by the one followed by the associated Interest from the Client to the location where the chunk is retrieved. The "One Interest - One Chunk" rule is an interesting property that enforces a flow balance in a CCN network.

Each CCN router supports three different types of storage. The Interests are forwarded to the data source thanks to the Forwarding Information Base (FIB). In addition, the Pending Interest Table (PIT) keeps track of the forwarded Interests so that the chunks can be returned to their requestor. Finally, a huge cache is used to store the chunk packets using any caching strategy in order to reduce the network response time for frequently asked content. As the content can be delivered by any cache in the network, the time elapsed from sending an Interest to retrieving the corresponding chunk is defined as a random variable $A(t)$. We will call this delay the $Response\ delay$. In the CCN node model we have proposed in [2] and also considered in this study, we introduced a transmission buffer associated with each face, that differs from the cache.

An important optimization in CCN is the ability to aggregate Interest packets looking for the same content when flowing through a given node. As a consequence, a single Interest copy will be sent to the network and will be used to retrieve the corresponding chunk. In this situation, the router updates an existing PIT entry adding the interfaces that have requested the same chunk and where the corresponding Interest was aggregated (namely dropped). Once a corresponding chunk is received, the node copies it to all interfaces from this entry.

IV. Hop-by-hop Interest shaping mechanism

In this section, we shortly introduce the control principles of HoBHIS, more detailed information can be found in [2]. This proactive mechanism is implemented in each CCN router and performs a sharing of the network capacity among different conversations. We define a congestion in CCN router as the overflow of the transmission buffer associated to an output interface. The congestion manifests by the loss of data Chunks. Interest shaping is used to control the flow of Chunks and regulate it according to the available buffering capacity in every CCN router. Every transmission buffer of a CCN router will be monitored and the Chunk traffic will be regulated in order for the data transmission buffer to converge to a given objective value r. The Chunk traffic can be regulated thanks to the "One Interest - One Chunk" rule that provides an elegant way to control the flow of information.

Let us describe the principle of HoBHIS. Figure 1 provides an illustration of the communication process when shaping is enforced. As can be seen from Figure 1, when an Interest

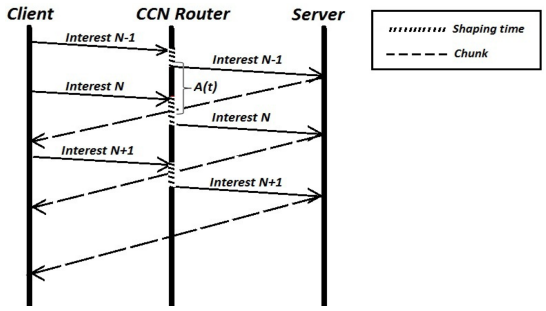


Fig. 1: Communication process when shaping is enforced

TABLE I: Notations

$C(t)$	available bandwidth to send the chunks at time t
$C_{int}(t)$	available bandwidth to send the Interests at time t
$\gamma(t)$	shaping rate at time t
$\gamma^T(t)$	tolerance rate at time t
$A(t)$	delay from Interest to the related content
$A^*(t)$	predicted delay from Interest to the related content
$e(t)$	number of queued Chunks at time t
$e^I(t)$	number of queued Interests at time t
$e_i^I(t)$	number of queued Interests of conversation i at time t
B	buffer size
B_I	Interest buffer size
r	queue threshold
h	design parameter

is received, it will be delayed in the CCN router if proved necessary in order to avoid congestion of the transmission buffer of that same node. This delay is obtained thanks to the computation of the associated shaping rate that is derived using the following formula:

$$\gamma(t) = \min[\max[C(t) + h\ \frac{r - e(t)}{A^*(t)}, 0], C_{int}(t)], \qquad (1)$$

Where $A^*(t)$ is a predicted value of the Response delay variable that corresponds to the delay of the control loop. We introduce h, a design parameter, that has an important effect on the dynamic convergence properties of our scheme towards the objective r. The other notations can be found in Table I.

The shaping rate function (1) consists of two parts. The first one corresponds to the transmission queue capacity available to send the Chunks, $C(t)$, while the second represents the additional rate to adjust, namely increase or decrease, the Interest sending rate and still be able to avoid overflow when facing a control loop of duration $A^*(t)$. Obviously, a good estimate for $A(t)$ is needed in order to reduce oscillations and cope with the potential variability of the response delay parameter.

A. Resource sharing

In this section we provide a solution for an efficient buffer sharing of the network resources, for the case when multiple conversations are active. Let suppose that there are F active conversations going through the router. In the simplest case the buffer capacity can be shared between the conversations as $\frac{r}{F}$. However, this solution does not take into account the situation where one or more conversations do not consume their share. In this case, the queue length will converge to

$$r' = \sum_{i=1}^{F} [r_i] \leq r,\ i = 1..F \qquad (2)$$

Each conversation converges to its own threshold r_i. Therefore, as can be seen from Formula (2), the total available capacity may not be completely used. In order to improve its utilization, the rest of the resources should be shared between the active conversations. Our solution to solve this issue is described below. The arrival rate for a given conversation can be estimated by the number of chunks in the queue belonging to this conversation. Thus, the rate of a conversation can be expressed as follows:

$$\rho_i(t) = \frac{e_i(t)}{\sum_{j=1}^{F}[e_j(t)]} \qquad (3)$$

Where $e_i(t)$ is the number of chunks of conversation i at time t and $\sum_{j=1}^{F}[e_j(t)]$ is the total queue length. $\rho_i(t)$ represents the ratio of the total queue length occupied by conversation i at time t. Taking into account equation (3), we modify the shaping rate formula in the case of multiple conversations as follows:

$$\gamma_i(t) = C(t) + h.\frac{r.\rho_i(t) - e_i(t)}{A_i^*(t)} \qquad (4)$$

It is important to notice here that we only need to monitor the active conversations, those that have at least one chunk in the queue and enforce accounting only for this reduced set. Note also that for the sake of simplicity, we consider Chunk packets of constant size. If it was not the case, the length of a chunk packet will be used to adjust the above equation.

Monitoring all conversations is far too expensive but we are concentrating on the active conversations only, those for which packets are queued in the buffer, to reduce the number of states stored in the router. It has been shown in the past (cf. [15]) that such per flow control is scalable because of the relatively small number of active flows. A conversation is considered to be active if a transmission queue of the router owns at least one packet of this conversation.

B. Dynamic adjustment of the design parameter h

In this section we analyse in details the behaviour of HoBHIS and study the dynamic adjustment of the design parameter h. This parameter is very important because it has a significant effect on the dynamic convergence properties. Let us study it with a simple analytical model.

In this model, we consider a single-conversation going through the CCN node and constant response delay $A(t) = A$. HoBHIS starts to react only after the reception of a data chunk. As during an initialisation period we do not know the size of data packets, the equation (1) can not be used to compute the shaping rate. Thus, the Interest sending rate of any CCN node during an initialisation period will be equal to its maximum possible rate to send the Interests, namely:

$$\gamma_{max,t_0} = \frac{C}{S_I} \qquad (5)$$

where S_I is the size of an Interest. The oscillations in the data queue might be large during the initialisation period due to a high Interest sending rate. Two situations are possible: case 1) an incoming Interest rate, γ_{in}, is higher or equal to γ_{max,t_0}, and case 2) γ_{in} is less than γ_{max,t_0}.

In the first situation, the router will send a huge number of Interests into the network that may lead to large oscillations in the data queue. Oscillations due to the initialisation period can not be avoided but the parameter h will provide a mean to control in order to stabilize the system as fast as possible.

For the first case where $\gamma_{in} \geq \gamma_{max,t_0}$, the solution is to quickly serve the Chunks in excess. Let's call e_0 the number of Interests forwarded by a CCN router during the initialisation period. We know that for the case where $e_0 > r$:

$$h\frac{r-e_0}{A} < 0 \tag{6}$$

Thus, to have $\gamma > 0$, we should maintain:

$$|h\frac{r-e_0}{A}| < C \tag{7}$$

Thus, we have:

$$|h| < |\frac{CA}{r-e_0}| \tag{8}$$

When a first Chunk arrives to the node, we can easily compute h using Formula (8). That allows us to make the queue converge to its objective faster and without future oscillations. The only problem we can face here is the shaping rate that immediately drops from a large value to a small one. In order to reduce this drop of the shaping rate, we can try to maintain it between $\gamma_{stable} \pm \delta$, where δ is defined from:

$$|\gamma(t) - \gamma_{stable}| \leq \delta \tag{9}$$

Where $\delta = \epsilon * stable\_value$, $\epsilon \in (0;1)$ that corresponds to $\delta = \epsilon\gamma_{stable} = \epsilon C$ in our case. Thus, to maintain γ in these bounds we can use $h = \epsilon h_{opt}$. Frequently, for control systems $\epsilon = 0.05$.

Let us study the second case where $\gamma_{in} < \gamma_{max,t_0}$. We know that:

$$h\frac{r-e(t)}{A} > 0 \tag{10}$$

It means that in practice the Interest sending rate $\gamma(t) = min\{\gamma_{max,t_0}, \gamma_{in}\} = \gamma_{in}$. When the first Chunk arrives, the shaping rate $\gamma(t)$ will be computed and expressed in number of Chunks per second and then should be applied to the Interests. Hovewer, the queue size is still less than the objective value r. Therefore, the design parameter should adjust the shaping rate $\gamma(t)$ to γ_{in} until the queue converges. We should also take into account the number of Interests that have been sent during the initialisation period and will be quickly returned as Chunks, e_0. Therefore,

$$\gamma_{in} = C + h\frac{r-e_0}{A} \tag{11}$$

Thus, the value for h can be derived from this equation and will be:

$$h = \frac{(\gamma_{in} - C)A}{r-e_0} \tag{12}$$

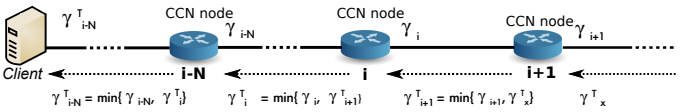


Fig. 2: Network model

V. Controlling the Interest rate

The shaping rate formula allows us to control the congestion of the transmission queue by carefully monitoring the rate of the Interest packets in the CCN nodes. However, there is no mean to prevent a Client from sending Interest at high or excessive rates in order to be privileged and retrieve its content faster. Therefore, it is important to define a tolerance mechanism that, in addition, can control the Clients as well as prevent the loss of Interest packets.

There exist two basic methods to throttle the rate of a client: rate-based and window-based. Each of them has their own merit and both approaches are feasible in CCN. In the case of window-based, the clients behavior is strictly defined by the presence or absence of Data packets. If no feedback arrives, it will be interpreted as congestion in the network and the client will automatically stop sending. However, window-based control schemes can lead to traffic bursts. Moreover, if the window is too small, the network resources are not effectively used. An adaptation of this method is investigated in [3] and represents a variant of Additive Increase Multiplicative Decrease (AIMD) for CCN. As the authors use Data Chunks to increase/decrease the congestion window, the estimation of the response time is crucial for this type of control. In [10], the authors also choose to use AIMD in order to adjust the Interest sending rate used by clients.

Ideally, Clients should fully saturate the link, send packets and keep the Chunks arriving continuously. To achieve this, the window-based control has to estimate an ideal window size that can be computed based on the bottleneck capacity and the Response Delay. The rate-based control has to know the bottleneck capacity or should receive an explicit rate value from the network in order to adapt its sending rate. One advantage of the rate-based control is that it does not stop sending in absence of feedback, but at the same time the client has to be aware of a specific sending rate. This rate might be known thanks to the introduction of a network feedback mechanism or by using an instantaneous measured value of the input Data rate.

For a CCN client it is important to choose an initial value for the sending rate. It may be a small fixed window for window-based approach or any specific rate for rate-based schemes.

A. Tolerance rate

In this section we present the Explicit Interest Rate control mechanism that defines a tolerance with respect to the interest rate that a Client can generate. A rate-based scheme was preferred to a window-based mechanism such as AIMD because it is independent of the end users strategy. In addition, it allows a more accurate control and better network resource utilization.

We define the tolerance rate as the maximum rate that the clients are not allowed to exceed. The principle of our solution is to adjust the tolerance rate to the rate of the bottleneck node. Every CCN router will control the rate of incoming Interests by periodically sending control packets with an explicit rate field advertised back to the Clients. This explicit rate field aggregates the bottleneck rate on that path up to this router and is advertised upstream towards the source.

In our model, two complementary rates are computed for each individual conversation: the shaping rate and the tolerance rate. Thanks to HoBHIS, every CCN router controls the rate of a conversation by shaping the rate of its associated Interests. In addition, control packets will be exchanged and updated by the routers along the path followed by a conversation, to convey the tolerance rate for every single Client. For sake of simplicity and without loss of generality, we will consider a network topology illustrated in figure 2. Each $A(t)$ seconds, CCN nodes transmit control packets with an explicit rate information back to the client. Each router calculates the maximum allowed rate for the Client and updates this field if it is found smaller than the actual value. As a consequence, the control packets carry the most conservative value for the rate, namely, the maximum allowed rate of the path. It is obviously not necessary to update the explicit rate faster than the delay of the control loop $A(t)$.

B. Computation of the Tolerance rate

Let Formula 13 represent the maximum shaping rate for conversation i at time t. The Interest queue contains Interests packets from many different conversations. The control function used to compute the shaping rate of a given conversation enforces the total Chunks queue length to converge to the objective r defined as a percentage of the buffer capacity.

$$\gamma_i(t) = \min[\max[C(t) + h\,\frac{r\rho_i(t) - e(t)}{A_i^*(t)}, 0], C_{int}(t)], \quad (13)$$

As we do not know a delay between the bottleneck router and the client, we can not directly use the shaping rate formula to compute the tolerance rate because it may lead to large Interest queue oscillations. Moreover, the shaping rate formula does not depend on the Interest queue length at time t and so the level of Interests in the queue is not controlled. As a result, it would be possible to have an Interest queue always congested.

Thus, the tolerance rate formula should:1) depend on $\gamma(t)$; 2) use the filling level of Interest queue at time t.

Respecting the conditions listed above, we propose the following computation formula for the tolerance rate:

$$\gamma^T = \gamma(t).(1 - \frac{e^I(t)}{B^I}) \quad (14)$$

The factor $(1 - \frac{e_I(t)}{B_I})$ should decrease the oscillations due to feedback delay and the variations of the shaping rate. Using formula 14 to calculate the explicit rate value, allows us to maintain the total arrival Interest rate aligned with the shaping rate.

VI. Performance evaluation

The aim of this section is to analyze, through simulations, the performance of HoBHIS and Tolerance rate mechanism.

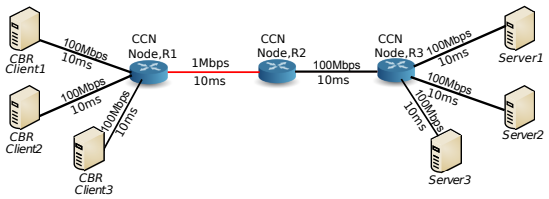


Fig. 3: Network topology to study the Tolerance rate mechanism

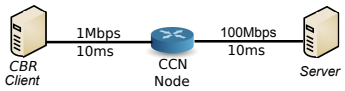


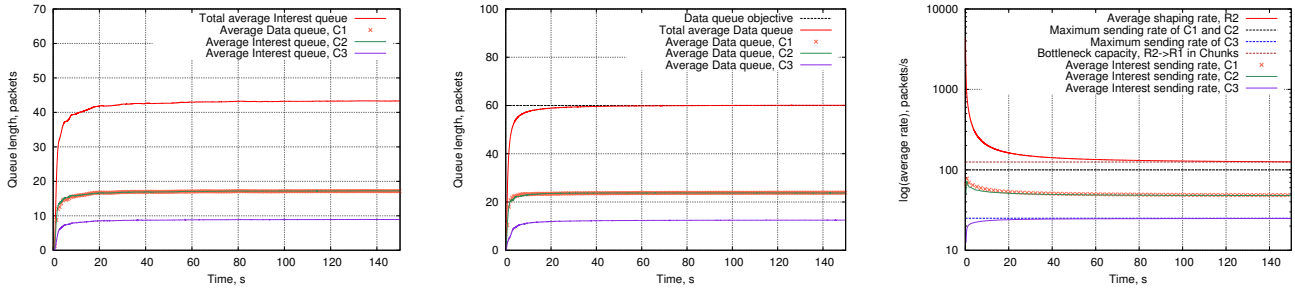
Fig. 4: Simple topology to study the influence of the design parameter

We have implemented our mechanisms in ns3-based Named Data Networking Simulator (ndnSIM), [16], that implements NDN communication model.

A. Tolerance mechanism

In this section we analyze the performance of our feedback mechanism that is used to enforce the tolerance rate. The simulation topology is presented in Figure 3. In this scenario, Clients 1, 2 and 3 are asking for the content from Server 1, 2 and 3 respectively. The shaping mechanism is implemented in each CCN node. Every CCN node has a timer set to a Response Delay value and send the control packets according to the algorithm defined in Section V. The servers represent the rest of CCN network and generate the random delay uniformly distributed from 0 to 0.1 s for every Chunk packet. All the clients are not greedy and have the maximum sending rate for 100 Interests/s for Clients 1 and 2 and 25 Interests/s for Client 3 they can not exceed. When the explicit rate value is retrieved by the Clients, they adjust their sending rate to this value. Their rates are updated every $A(t)$ seconds with the values obtained from the control packets. We are interested in observing the Interest and Chunk buffer states in bottleneck node R2. The buffer parameters are: buffer size = 100 Chunks, r = 60 Chunks. The control packets size is equal to the Interest packet size that is around 30 Bytes, the Chunk payload size is set to 1000 Bytes.

The results are shown in Figure 5(a) and 5(b). As it can be seen from the Figures, the total average Chunks queue length converges to the threshold r = 60 Chunks as expected by HoBHIS. We can see that the conversations fairly share the buffer capacity. The Clients 1 and 2 have the same percentage of buffer capacity attributed to them because they are emitting with the same rates. As the sending rate of Client 3 is slower than the rates of other two Clients, it does not need the same amount of buffer capacity that Clients 1 and 2. Thus, we see that the part of router ressources that is not used by Client 3 is attributed to Clients 1 and 2. We observe the same behavior of the Interest queue where the resources are fairly shared between the conversations. Figure 5(c) presents the client rates and the shaping rate of the bottleneck node. The rate of Client 3 achieves its maximum value at 25 Interests/s. As the Clients 1 and 2 are able to emit faster than Client 3, their tolerance rate

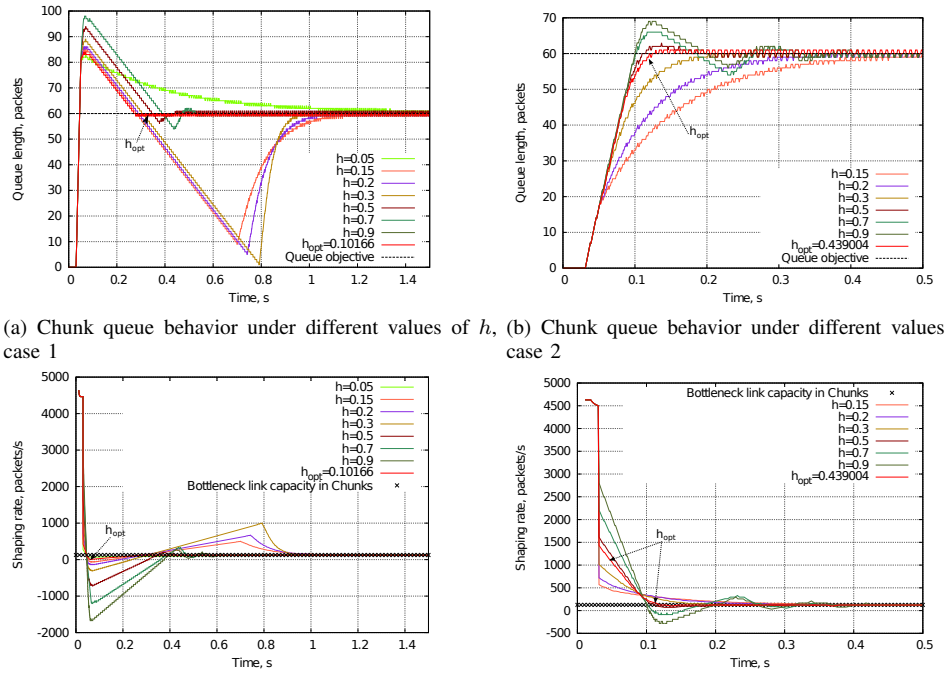


(a) Interest queue length under shaping and Interest rate control

(b) Chunk queue length under shaping and Interest rate control

(c) Conversation rates

Fig. 5: Hop-by-Hop Interest shaping and Tolerance mechanism in operation



(a) Chunk queue behavior under different values of h, case 1

(b) Chunk queue behavior under different values of h, case 2

(c) Shaping rate behavior under different values of h, case 1

(d) Shaping rate behavior under different values of h, case 2

Fig. 6: Chunk queue length and Interest shaping rate convergence speed as a function of h

is higher. The sum of Client rates is equal to the bottleneck node's shaping rate. Finally, as a result, we observe optimal resource utilization and no packet loss.

B. Dynamic adjustment of h

In this section we analyze the convergence rate of our mechanisms. We use a simple mono-conversation scenario and one-node topology presented in Figure 4. The $A(t)$ is considered to be constant for these experiments. We are interested in the convergence rate of the data queue of the bottleneck link and the shaping rate towards their objectives that are $r = 60$ and $\gamma = C$, under different values of h. We study two cases presented earlier in Section IV-B, where the first one causes the big oscillations in the data queue after an initialization period while in the second case the data queue is not sufficiently filled. To generate the oscillations in the data queue during the initialisation period we will use the Client Interest sending rate = 5.000 Interests/s that is higher than γ_{max} in order to use the maximum possible Interest sending rate of a CCN node. For the second case we reduce the Client sending rate to 1.000 Interests/s in order to have $\gamma_{in} < \gamma_{max}$.

The curves 6(a) and 6(b) show the results for the case where the data queue is overloaded due to an excess number of Interests sent during the initialization period. Tuning the value of h we observe the different behavior of the Chunk queue. It is easy to see that the optimum value of h dynamically adjusted thanks to the algorithm presented in Section IV-B provides the best convergence rate to the objective value without oscillations whilst other values of the design parameter do not provide a suitable control. For simplicity, we provide a theoretical shaping rate that may become negative.

The results for the second case are presented in Figures 6(c)

and 6(d). Again, h_{opt} provide a suitable convergence speed towards the objective. The small and large h values cause slow convergence and oscillations respectively.

In order to challenge our mechanisms we should compare them with other hop-by-hop solutions presented earlier in the literature, i.e. [10], [11], but unfortunately the corresponding source codes are not publicly available.

VII. Conclusion

We presented a first comprehensive solution for congestion control in CCN networks. This problem is of utmost importance and has not been globally addressed in the past. Our framework is grounded on our original HoBHIS mechanism that was the first introduced to provide a hop-by-hop shaping mechanism. It nicely exploits the flow balance enforced in CCN between Interest and Chunk packets. It mostly consists in monitoring active conversations sharing the transmission buffer of a CCN node face in order to dynamically adjust their Interest sending rate and enforce the Chunk queue length to converge to a defined objective. This mechanism is implemented in each CCN node. We extended the design of HoBHIS in order to address the important concerns that might occur in CCN. We first demonstrated the fairness of resource sharing among competing conversations. Second, we introduced an explicit Interest rate feedback mechanism designed to control the Client behavior and prevent a potential risk of network congestion. Each node will compute a Client tolerance rate that is returned upstream towards the source in order to collect the bottleneck rate of the path. A thorough evaluation was conducted using different simulation scenarios. We observed that the results fully satisfy the design objectives and we can conclude that HoBHIS is an efficient and operational solution to the problem of congestion control in CCN. In future work we would certainly like to consider different types of traffic and services and analyze if differentiating between them will provide additional value.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *ACM CoNEXT*, Dec. 2009.

[2] N. Rozhnova and S. Fdida, "An Effective Hop-by-hop Interest Shaping Mechanism for CCN Communications," in *IEEE NOMEN'12*, Mar. 2012.

[3] G. Carofiglio, M. Gallo, and L. Muscariello, "ICP: Design and Evaluation of an Interest Control Protocol for Content-Centric Networking," in *IEEE NOMEN'12*, Mar. 2012.

[4] G. Carofiglio, M. Gallo, L. Muscariello, and M. Papalini, "Multipath Congestion Control in Content-Centric Networks," in *IEEE NOMEN'13*, Apr. 2013.

[5] L. Saino, C. Cocora, and G. Pavlou, "CCTCP: A Scalable Receiver-driven Congestion Control Protocol for Content Centric Networking," in *IEEE ICC 2013 - Next-Generation Networking Symposium (ICC'13 NGN)*, Jun. 2013.

[6] S. Braun, M. Monti, M. Sifalakis, and C. Tschudin, "CCN and TCP coexistence in the Future Internet: Should CCN be compatible to TCP?" in *The Fifth IFIP/IEEE International Workshop on Management of the Future Internet*, May 2013.

[7] M. Vojnovic, J.-Y. Le Boudec, and C. Boutremans, "Global fairness of additive-increase and multiplicative-decrease with heterogeneous round-trip times," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2000, pp. 1303–1312 vol.3.

[8] T. Henderson, E. Sahouria, S. McCanne, and R. Katz, "On improving the fairness of tcp congestion avoidance," in *Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE*, vol. 1, 1998, pp. 539–544 vol.1.

[9] S. Braun, M. Monti, M. Sifalakis, and C. Tschudin, "An Empirical Study of Receiver-based AIMD Flow-Control Strategies for CCN," in *IEEE ICCCN'13*, Jul. 2013.

[10] S. Oueslati, J. Roberts, and N. Sbihi, "Flow-aware Traffic Control for a Content-Centric Network," in *IEEE INFOCOM*, Mar. 2012.

[11] G. Carofiglio, M. Gallo, and L. Muscariello, "Joint Hop-by-hop and Receiver-driven Interest Control Protocol for Content-Centric Networks," in *ACM ICN'12*, Aug. 2012.

[12] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive Forwarding in Named Data Networking," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 3, pp. 62–67, Jun. 2012.

[13] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A Case for Stateful Forwarding Plane," Tech. Rep. NDN-0002, Jul. 2012.

[14] Y. Wang, N. Rozhnova, A. Narayanan, D. Oran, and I. Rhee, "An Improved Hop-by-hop Interest Shaper for Congestion Control in Named Data Networking," in *ACM ICN'13*, Aug. 2013.

[15] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts, "Evaluating the number of active flows in a scheduler realizing fair statistical bandwidth sharing," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 217–228, Jun. 2005.

[16] "NS-3 based Named Data Networking simulator," http://ndnsim.net.