An effective hop-by-hop Interest shaping mechanism for CCN communications

Natalya Rozhnova, Serge Fdida

To cite this version:

Natalya Rozhnova, Serge Fdida. An effective hop-by-hop Interest shaping mechanism for CCN communications. INFOCOM NOMEN workshop, Mar 2012, Orlando, Florida, United States. pp.322 - 327, 2012, <10.1109/INFCOMW.2012.6193514>. <hal-00997088>

HAL Id: hal-00997088
https://hal.sorbonne-universite.fr/hal-00997088
Submitted on 28 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
An effective hop-by-hop Interest shaping mechanism for CCN communications

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—We introduce a rate-based congestion control mechanism for Content-Centric Networking (CCN). It builds on the fact that one Interest retrieves at most one Data packet. Congestion can occur when aggregate conversations arrive in excess and fill up the transmission queue of a CCN router. We compute the available capacity of each CCN router in a distributed way in order to shape their conversations Interest rate and therefore, adjust dynamically their Data rate and transmission buffer occupancy. We demonstrate the convergence properties of this Hop-by-hop Interest Shaping mechanism (HoBHIS) and provide a performance analysis based on various scenarios using our $ns_2$ simulation environment.

I. INTRODUCTION

The increasing importance of content has triggered a tremendous interest from the networking research community, moving away from packets and getting closer to the service delivered to the user. In particular, the architecture of Content-Centric Networking (CCN) proposed by Van Jacobson and al. in [1] appears as the most popular one. At this stage, CCN defines design principles and has developed a CCNx Open Source Platform. However, many functionalities are still at an early stage, or have been considered in very simple contexts.

CCN decouples the sender from the receiver in a mode that is similar to the Publish and Subscribe service model. Content names are used instead of network addresses to convey the information to the interested parties. The content might be located anywhere in the network thanks to extensive caching capabilities. Therefore, the data is not necessarily associated with the content publisher as the content can be delivered by any cache in the network. In CCN, the request from the content publisher as the content can be delivered by any cache in the network thanks to extensive caching policies based on Markov chains. CCN Congestion control has not been studied yet, arguing that its hop-by-hop control will enforce a reasonable level of performance. Some preliminary work on a transport protocol for CCN is presented by S. Arianfar and al. in [13]. This protocol is based on the TCP congestion window principles but uses Data packets as acknowledgments to enforce decisions to increase or decrease the congestion window. Congestion can appear in CCN and therefore, we promote the utilization of an algorithm to control the level of congestion of CCN routers filled by Data packets. We follow similar principles as in [14], where the authors present ERAQLES, a rate control mechanism for Available Bit Rate (ABR) ATM communications, and define an analytical
method to compute the advertised explicit rate to which the sources have to adapt. A similar work is presented in [15]. In [16], the authors describe a rate-based hop-by-hop congestion control mechanism in which a desired service rate is computed at each switch as a function of the target queue occupancy. Feedback information is then exchanged between neighbor switches so that they can dynamically adjust the service rate for each connection.

III. CCN NODE MODEL

In this section we describe the CCN node model and discuss the most meaningful properties of CCN for our work.

The general CCN node model was introduced in [1]. A simple router model is illustrated in Figure 1. In this figure, we omit the forwarding information base (FIB) and the pending interest table (PIT). The former is used to forward Interests toward the data source while the later keeps track of the forwarded Interests so that the returned Chunks can be sent to its requestor. We first introduce our model of a CCN router in order to illustrate where congestion can appear.

In [1] the authors advocate that “CCN does not have FIFO queues between links but rather an LRU memory (the cache)”. However, in a CCN router, it is important to differentiate the memory allocated for transmitting packets from the one used for caching. If a single Content-Store was used to store both packets waiting for transmission and cached Data chunks, the whole cache will become congested by Data chunks waiting for transmission through the congested interface and so it will penalize another conversation going through non-congested interfaces. Thus, it would be hard to conceptualize the CCN node behaviour when there is a congestion of a single output interface. Another undesirable side is the huge delay induced by the filling of a very large cache used as a transmission buffer, before congestion detection. Finally, it is crucial that at any given time, each output interface schedules which packet is the next one to be transmitted on the physical channel. This can be achieved with the use of per-interface queue rather than a single shared cache.

IV. TRAFFIC CONTROL IN CCN

One key CCN property is that one Interest retrieves at most one Data packet. This behaviour enforces a flow balance that is maintained in the network and allows the communication between the different machines and at different speeds. The flow balance enables multiple Interests to be issued at once. The CCN node behaviour is described thereafter; when a Data-chunk is received on one face or is retrieved from the cache by an incoming matching Interest, it must be transferred to the output interface(s). Depending on the results of the PIT lookup, it is then queued on one or more transmission buffers. At the same time the Data chunk can also be stored into the shared cache to enable further retrieving. Because some Data chunks will present very few incentives for being cached (short-lived or non-popular Data chunk), the copy of data-chunks to the content store is not mandatory. Caching decisions are out-of-scope of this paper.

Congestion in a CCN router is defined as the overflow of the buffer associated to an output interface and manifests itself by the loss of data chunks.

V. HOP-BY-HOP INTEREST SHAPING MECHANISM (HoBHIS)

In this section, we present the hop-by-hop Interest Shaping mechanism (HoBHIS). Every CCN router will control the rate of individual Data chunks conversations by appropriately shaping the rate of the associated sending Interests.

The congestion control scheme based on hop-by-hop interest shaping was preferred to an end-to-end mechanism such as TCP. Our mechanism is proactive and the share of the network capacity allocated to different conversations is controlled by the algorithm implemented in each CCN node. Interest shaping allows us to anticipate the drop of data packets due to buffer overflow. This is another advantage over TCP congestion control scheme that starts to react only after the drop of one segment, unless a mechanism such as RED is used. In addition, using a hop-by-hop control provides a feedback information more quickly thanks to the smaller distances between hops.

The system under study consists of a set of CCN routers forwarding Interest packets issued by a consumer. As a response, Data chunks are forwarded back to the consumer (namely, the source of the Interests). Once an Interest has been issued by a given CCN router, the corresponding Data chunk will be piggybacked to that router after a variable delay \( A(t) \) named “Response Delay”, assuming no loss. This parameter differs from the usual RTT from TCP as the Data might be stored in any cache on the path or at the publisher site (the source of the Data). Thus, in such architecture as CCN it is possible sometimes to observe large variations of \( A(t) \) that leads to the queue size oscillations. But thanks to the congestion control mechanism, the queue size will be fast stabilized. We also infer that \( A(t) \) should not change drastically on short time scales as it is likely that when the data is stored in a given cache, it should stay there for some time, at least until the network and demand conditions have largely evolved. The objective of HoBHIS is to avoid congestion in CCN transmission queues by enforcing the queue size to converge to a given objective, defined as a percentage of the capacity of that buffer. We achieve this objective by shaping the Interest rate. In order to avoid the losses of Interests when shaping is enforced we
need some back-pressure mechanism which is out-of-scope of this paper and will be provided in future work.

Upon arrival of a Data chunk in the transmission queue, the router computes the Interest rate based on the queue occupancy and the available resources for each conversation. It adjusts the Interest rate according to this information. If the actual number of queued packets is less than some threshold value $r$, then the router can temporarily increase the shaping rate. On the other hand, if the number of queued packets is higher than $r$, then the router will decrease its shaping rate.

The system under study is presented in figure 2. The Shaping time component is responsible for the computation of the shaping delay that the Interests packets will have to satisfy. Every Interest packet will be shaped according to the occupancy of the transmission queue, as well as the parameter $A(t)$.

### VI. Single router model

For the purpose of the analysis, we consider the simplified model presented in figure 3. It is characterized by a single conversation. In addition, all packets have the same size. The control is applied to each chunk entering the transmission queue. We do not consider any caching policy or routing mechanism. The parameters and notations are given in Table I.

#### A. Computation of the shaping rate

Let $e(t)$ define the occupancy of a CCN router transmission queue expressed in number of chunks.

We consider the following function that represents the maximum shaping rate at time $t$, while still being able to control the transmission queue under a feedback delay equal to $A(t)$:

\[
\gamma(t) = C(t) + \frac{B - e(t)}{A(t)}, \quad \text{or}
\]

\[
\gamma(t) = C(t) + h \frac{r - e(t)}{A(t)},
\]

where $h$ is a design parameter that aims to control the dynamicity of our scheme towards the objective $r$. In the general case when there are $F$ conversations flowing through a CCN node, we need to divide the available buffer capacity between all active conversations at time $t$. We allow each conversation to share a part equal to $\frac{e}{F}$ of the total buffer capacity. If a conversation is using less than $\frac{e}{F}$, the remainder resources may be used for queue size oscillations caused by $A(t)$ (cf. Section V). Let $e_i(t)$ be the number of packets from conversation $i$ waiting in the transmission queue. The shaping rate for conversation $i$ is expressed as follows:

\[
\gamma^i(t) = C(t) + \frac{r - e_i(t)}{A_i(t)},
\]

Note that the shaping rate is bounded by $C_{int}(t)$ defined as the maximum capacity to send the Interests at time $t$. The expression for $\gamma^i(t)$ becomes:

\[
\gamma^i(t) = \min[\max[C(t) + \frac{r - e_i(t)}{A_i(t)}, 0], C_{int}(t)],
\]

It is possible from equation (2) that the shaping rate becomes negative. In this situation, the Interests are blocked until either the transmission queue size $e(t)$ becomes less or equal to $r$, or a Chunk arrives at the queue and the Interest shaping rate is re-evaluated. We use an exponential weighted moving average mechanism to estimate the value of $A(t)$.

#### B. Queue convergence

In this section we demonstrate that the transmission queue length converges to the objective $r$. We assume a single conversation that can use the entire link capacity to send its Data. Let $A_j$ (respectively $A_{j+1}$) be the Response Delay for packet number $j$ (respectively packet $j + 1$). The queue evolution for a single conversation is written as follows:

\[
e(t + 1) = e(t) + \gamma(t).A_j^* - C.A_{j+1}^*
\]

and

\[
\gamma(t) = C + h \frac{r - e(t - 1)}{A_j^*},
\]
Based on above we can write:

\[ e(t+1) = e(t) - h \cdot \frac{A_{j+1}}{A_j} \cdot e(t-1) + [h \cdot r \cdot \frac{A_{j+1}}{A_j}] \]  \hspace{1cm} (7)

This formula can be simplified as

\[ e_{i+1} = e_i + \alpha \cdot e_{i-1} + \beta \]  \hspace{1cm} (8)

with

\[ \alpha = -h \cdot \frac{A_{j+1}}{A_j}, \quad \beta = h \cdot r \cdot \frac{A_{j+1}}{A_j} \quad \text{and} \quad e_{i-1} = e(t-1) \] \hspace{1cm} (9)

We write \( e_{i+1} \) as

\[ e_{i+1} = f_{i+1} - \frac{\beta}{\alpha} \] \hspace{1cm} (10)

Then we obtain

\[ f_{i+1} = f_i + \alpha \cdot f_{i-1}, \] \hspace{1cm} (11)

which is the series of Fibonacci. So, due to [12] we have

\[ f_{i+1} = P \cdot z^{i+1} \] \hspace{1cm} (12)

where \( z \) is the root of equation

\[ z = z^0 + \alpha = 1 + \alpha \] \hspace{1cm} (13)

Thus, we need that

\[ |z| < 1 \] \hspace{1cm} (14)

and if \( i \) tends to infinity then \( f_{i+1} \) converges to 0. But the condition \( |z| < 1 \) will be true only if:

\[ 0 < h < 2 \cdot \frac{A_j}{A_{j+1}} \] \hspace{1cm} (15)

And finally,

\[ \lim_{i \to \infty} e_i = \lim_{i \to \infty} \left[ -\frac{\beta}{\alpha} \right] = r \] \hspace{1cm} (16)

We found that the transmission queue is converging to \( r \) as expected. We will observe that we can face a burst of Data Chunks during the initialization period. This is because our algorithm starts only after the reception of the first data packet. Until then, the system sends the packets with the maximum available rate. This problem is solved by limiting the initial sending rate according to the shaping rate formula.

We now consider \( F \) conversations. Our shaping rate formula is defined by:

\[ \gamma_i(t) = C(t) + h \cdot \frac{r - e_i(t)}{A_i(t)}, \] \hspace{1cm} (17)

It appears that the queue size for each conversation converges to \( \frac{r}{\gamma} \). So, the total queue size will converge to \( r \) as expected.

We now extend our simple initial scenario to the case of a network of nodes. This model, described in Figure 4, consists of \( N \) nodes using HoBHIS. We have now a hop by hop congestion control mechanisms using the shaping rate computed at each router. The notations are presented in Table II. We start with a single conversation flowing through the system. The Response Delay is defined by parameter \( A \). The Chunks arrive into the nodes with the rate \( \lambda^c \) chunks/second.

\begin{table}[h]
\centering
\caption{Notations}
\begin{tabular}{|c|c|}
\hline
\( \lambda^c_i \) & arrival Chunks rate to node \( i \) \\
\( \lambda^t_i \) & arrival Interests rate \\
\( \mu_i \) & service rate of node \( i \) \\
\( \gamma_i \) & shaping rate of node \( i \) \\
\( A_i \) & delay from Interest to the related content for node \( i \) \\
\( e_i(t) \) & number of queued packets at time \( t \) of node \( i \) \\
\( B \) & buffer size \\
\( r \) & queue threshold \\
\hline
\end{tabular}
\end{table}

A. Response delay evaluation

The Response Delay \( A \) of one node depends on downstream nodes up to the router caching the Data or the source itself. Let’s define \( A_i \) and \( A_{i-1} \) the Response Delays for nodes \( i \) and \( i-1 \) respectively. \( A_i \) can be expressed as follows:

\[ A_i = A_{i-1} + \sum_{j=1}^{X} \frac{1}{\gamma_j^{i-1}} + \frac{e_{i-1}}{\mu_{i-1}} \] \hspace{1cm} (18)

The Response Delay for a given packet \( p \) in node \( i \) depends on the number of packets \( X \) queued for transmission ahead of \( p \) towards the downstream node and also depends on the number of packets \( e_{i-1} \) queued before the response will be sent-back to node \( i \). The quantity \( \sum_{j=1}^{X} \frac{1}{\gamma_j^{i-1}} \) means that we recompute the shaping rate \( \gamma_{i-1} \) for each Chunk entering the queue and hence the Interests could be sent with different rates.

B. Convergence property

We have demonstrated for a single router model that, if the minimal rate among all nodes is \( \gamma_i \), associated to node \( i \), then the queue of node \( i \) converges to \( r \). In addition, we know that

\[ \lim_{t \to \infty} e_i(t+1) = r \quad \text{and} \quad \lim_{t \to \infty} e_i(t+1) = r' \] \hspace{1cm} (19)

So, according to the limit inequality theorem we have: \( r' \leq r \) for all other routers sharing the same conversation’s paths as router \( i \).

VIII. PERFORMANCE EVALUATION

The aim of this section is to analyze, through simulations, the performance of the CCN Interest shaping mechanism. We have developed the CCN module in Network Simulator 2 (ns2). According to our knowledge, there is no version of CCN in ns2 at the time of writing.
A. Simulation scenario

We start with the single router model. We consider both the single conversation and the multi-conversation scenarios presented in Figure 5 and 6 respectively.

As depicted in figure 5 and 6, our network consists of the client, router and server components. The client sends the Interests to the router that takes the next decision according to the shaping mechanism. The server role is very simple, as for each received Interest it responds with the corresponding Chunk and sends it back to the router. The shaping rate is computed according to the algorithm presented in previous sections. \( e(t) \) is the instantaneous measured value for the number of Chunks at time \( t \). The Interest and Chunk packet sizes are fixed on 500 and 1500 bytes respectively. The buffer parameters are \( B=500 \) Chunks, \( r = 100 \) Chunks, \( h = 0.1, 0.4 \) and 0.7. The clients generate Interests with rate 833.3 Chunks/s. The links rates are given in figure 5 and 6. The bottleneck is the 1Mb link between the client and the router. Parameter \( A(t) \) is a random value uniformly distributed in \([0,1]\) and it is generated in the server for every packet to take into account the variability of the path between the router and the server.

The Network topology used for the evaluation is presented in figure 7. We consider two conversations from clients 1 and 2 respectively. Data for conversation 1 is in the cache of node 3. Conversation 2 flows from Client 2 to the Server. The interest rate is 2500 Chunks/s for each client. The parameters for the buffers are \( B=500 \) Chunks, \( r = 250 \) Chunks and \( h=0.7 \). The capacity of each link is 2500 Chunks. The conversation 2 starts before conversation 1 and after some time it stops. We are interested in the buffer state for node 3, as well as the rate for each conversation over time.

B. Simulation results

The simulation results for the single conversation scenario are presented in Figure 8. We have tested our algorithm for different values of \( h \). In Figure 8 we can see that the queue converges to the objective \( r \) as expected (cf. Section VII). The bursts in these curves are due to the initialization period when the algorithm is not yet in operation. The different values of \( h \) illustrates its influence for the control mode and convergence rate. The higher \( h \) the slower the convergence rate towards the objective and the harder the control. The situation repeats for the multi conversation scenario in Figure 9. The average queue size converges to \( r \) as expected (cf. Section VII).

Figures 10 and 11 illustrate the network scenario. As we use the same capacities for each link, the queue of every node is empty until conversation 1 starts. Then, the queue for node 3 starts to fill up and converges to the objective. Figure 11 illustrates the Chunks arrival rates for each conversation in Chunks/s. At time 20, conversation 1 becomes active. At time 60, conversation 2 becomes inactive. Between time 20 and 60, when the two conversations are active, each of them receives a faire share of the router resources. The Chunks arrival rate, shared between the two conversations are controlled by the minimal shaping rate. In figures 10 and 11 we see that our congestion control mechanism is able to adapt the rate and to maintain the queue length at the expected level as it was designed.

IX. CONCLUSION

We presented (HoBHIS), the first hop-by-hop Interest shaping congestion control mechanism designed to avoid the congestion that can occur in the output interface of a CCN node. HoBHIS monitors the transmission buffers of a CCN
We demonstrated analytically the convergence property of our algorithm. In addition, we have developed a CCN implementation in ns2. We performed various experiments with different settings and progressive complexity. We analyzed the single and multiple conversation scenarios in a single router model. Finally, a network case was studied to demonstrate the behaviour of our algorithm in more complex conditions. We have seen that the shaping mechanism performs as designed.

Future work will extend the analysis as well as the design of HoBHIS. We will consider the case where the sharing of the resources among competing conversations will be different from a fair share in order to favor important content. We will also explore the complexity of our algorithm and its scalability. Monitoring all conversations is far too expensive but we will concentrate on the active conversations only, those for which packets are queued in the buffer, to reduce the number of states stored in the router. In addition, the interaction with TCP will be considered. Finally, we will study more complex scenarios to better assess the behaviour of our solution.

ACKNOWLEDGMENTS

This work is partly funded by the French national research agency (ANR), CONNECT project. The authors would also like to thank Naceur Malouch for many useful discussions and for valuable comments.