An improved hop-by-hop interest shaper for congestion control in named data networking

Yaogong Wang, Natalya Rozhnova, Ashok Narayanan, David Oran, Injong Rhee

An Improved Hop-by-hop Interest Shaper for Congestion Control in Named Data Networking

Yaogong Wang
N.C. State University, USA
ywang15@ncsu.edu

Natalya Rozhnova
Université Pierre et Marie Curie (UPMC), France
natalya.rozhnova@lip6.fr

Ashok Narayanan
Cisco Systems, USA
ashokn@cisco.com

David Oran
Cisco Systems, USA
oran@cisco.com

Injong Rhee
N.C. State University, USA
rhee@ncsu.edu

ABSTRACT

Hop-by-hop interest shaping has been proposed as a viable congestion control mechanism in Named Data Networking (NDN). Interest shaping exploits the strict receiver-driven traffic pattern and the symmetric bidirectional forwarding in NDN to control the returning data rate. In this paper, we point out that both interests and contents contribute to congestion and their interdependence must be considered in any interest shaping algorithm. We first analyze this issue mathematically by formulating it as an optimization problem to obtain the optimal shaping rate. Then a practical interest shaping algorithm is proposed to achieve high link utilization without congestive data loss. We further note that flow differentiation in NDN is complicated and design our scheme independently of traffic flows. We demonstrate our hop-by-hop interest shaper in conjunction with simple Additive-Increase-Multiplicative-Decrease (AIMD) clients using the ns3-based NDN simulator (ndnSIM). Our results show that the proposed shaping algorithm can effectively control congestion and achieve near-optimal throughput.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

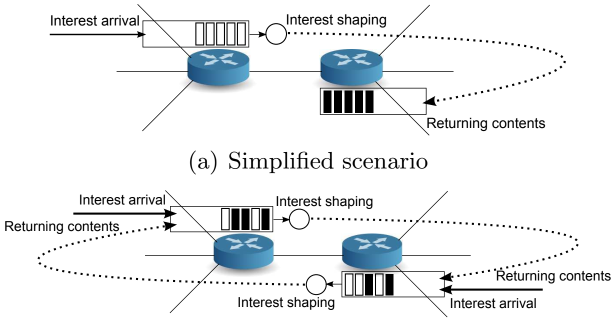
Performance

Keywords

Congestion Control, Information-Centric Networking

1. INTRODUCTION

The communication paradigm of NDN [8] has two prominent features: 1) all traffic is receiver-driven; 2) content retrieved in response to an interest traverses exactly the same links in reverse order. These two unique characteristics make hop-by-hop interest shaping a better option for NDN congestion control than traditional TCP-like mechanisms. TCP congestion control *reacts* to congestion after data packets are lost. By contrast, interest shaping *proactively* prevents data packet loss by regulating the interest rate in the first place. Dropping interest packets early wastes fewer resources than dropping data packets late. More significantly, end-to-end congestion control is severely handicapped in NDN. Extensive content multihoming and caching make it very difficult to identify interests belonging to a single "flow" and sharing the same congestion path. Performing interest shaping in a hop-by-hop manner can significantly alleviate this problem, especially if the shaping scheme does not rely on flow identification. It also enables sophisticated forwarding strategies such as congestion-driven rerouting to higher-cost (but uncongested) paths. Hence, incorporating hop-by-hop interest shaping with a backpressure mechanism appears to be a more viable option for NDN congestion control.



(a) Simplified scenario

(b) Practical scenario

Figure 1: Interdependence between interests and contents in reverse directions and its impact on hop-by-hop interest shaping

A number of schemes have been proposed along this path (e.g., [6, 5, 2]) but all these interest shapers consider single or multiple unidirectional flows. As Figure 1(a) shows, if we assume that interests are sent in just one direction or the bandwidth consumption of the interests in the reverse direction is negligible, then the design of the interest shaping algorithm is quite straightforward: we simply pace the interests so that the contents they bring back will saturate the reverse link. However, in practice, a link will see interests and contents flowing in both directions simultaneously. Also content names in interests can be long since it includes transactional information in many applications (e.g., [3]). As each interest packet fetches exactly one data packet, interests may consume a non-negligible fraction of the link bandwidth. In light of these factors, the interest shaping algorithm in one direction can no longer assume that the entire reverse link bandwidth is available for returning contents. It needs to shape the interests properly so that enough room is left on the reverse link for interests in the reverse direction. The same logic applies to the interest shaper on the other side of a link and they have a recursive interdependence as shown in Figure 1(b). Since it is non-trivial to determine how much room should be left by the shaper, an imprudent algorithm may cause starvation or link under-utilization.

In this paper, we examine the interdependence between interests and contents in bidirectional flows, and study its impact on the design of hop-by-hop interest shapers in NDN. Section 2 formulates this issue as an optimization problem and obtains optimal solutions under various scenarios. Section 3 presents a practical shaping algorithm based on these solutions, to be executed at the output interface of NDN routers. Section 4 evaluates the performance of our algorithm in conjunction with simple AIMD clients via simulation. Finally, Section 5 discusses related work and Section 6 concludes the paper.

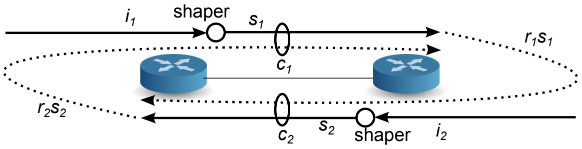


Figure 2: Modeling hop-by-hop interest shaping

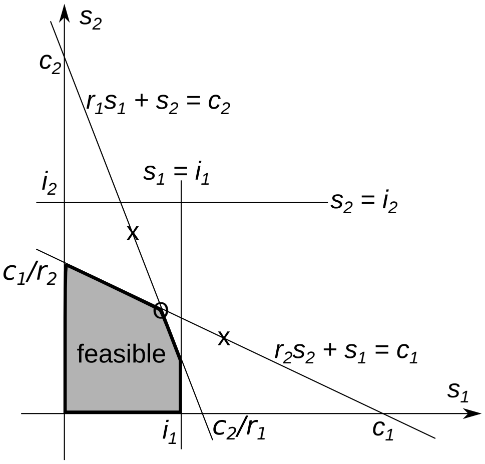


Figure 3: The feasible region of the optimization problem is convex.

2. PROBLEM FORMULATION

As depicted in Figure 2, let i_1 and i_2 denote the interest arrival rate in each direction. They are shaped down to s_1 and s_2 respectively by the interest shapers at the routers. Let r_1 and r_2 denote the average size ratio between contents and interests in each direction and let c_1 and c_2 denote the link capacity in each direction. Note that both r_1 and r_2 must be greater than 1. We can then formulate the interest shaping problem under steady state as follow:

Objective:

$$\max\ u(s_1) + u(s_2) \tag{1}$$

Subject to:

$$0 \leq s_1 \leq i_1 \tag{2}$$

$$0 \leq s_2 \leq i_2 \tag{3}$$

$$s_1 + r_2 s_2 \leq c_1 \tag{4}$$

$$r_1 s_1 + s_2 \leq c_2 \tag{5}$$

where $u(\cdot)$ is the utility function discussed later. The objective of this optimization is to maximize network utility (Eq. 1) subject to demand (Eq. 2 and 3) and bandwidth (Eq. 4 and 5) constraints. As Figure 3 shows, the feasible region of this optimization problem is convex. Hence, if the utility function constitutes a concave objective function, the formulation is a convex optimization problem which is mathematically tractable.

Note that the utility function we pick has profound impacts on the solution we would obtain. The utility function must maximize data throughput while maintaining certain degree of fairness between the traffic in both directions. As mentioned above, reverse interests compete with forward data packets for bandwidth and the shaper needs to appropriately divide resources between them. It has been shown in [4] that logarithmic utility functions can achieve proportional fairness between competing flows. Hence, we present below a closed-form solution to the described optimization problem under logarithmic utility functions (equivalent to maximizing the product of $s1$ and $s2$).

First, if we temporarily assume infinite loads in both directions, we can lift the constraints in Eq. 2 and 3. Second, from Figure 3 the optimal solution of the problem lies on the boundary of the feasible region, so we can convert the inequality constraints in Eq. 4 and 5 into equality constraints. Now let us begin by solving the optimization problem under each equality constraint independently. Solving for Eq. 4:

$$\begin{array}{ll} \text{maximize} & s_1 s_2 \\ \text{subject to} & s_1 + r_2 s_2 = c_1 \end{array}$$

The optimal solution for this problem can be determined using Lagrange multipliers:

$$\begin{cases} s_1 = \frac{c_1}{2} \\ s_2 = \frac{c_1}{2r_2} \end{cases} \tag{6}$$

This optimal point has been labeled by an "x" in Figure 3. Similarly, solving for Eq. 5:

$$\begin{array}{ll} \text{maximize} & s_1 s_2 \\ \text{subject to} & r_1 s_1 + s_2 = c_2 \end{array}$$

The solution (also labeled by an "x" in Figure 3) is:

$$\begin{cases} s_1 = \frac{c_2}{2r_1} \\ s_2 = \frac{c_2}{2} \end{cases} \tag{7}$$

Now we consider the solution to the original problem under infinite demand (i.e., the optimization problem consisting Eq. 1, 4 and 5). From Figure 3 we can see that the optimal solution primarily depends on how the two bandwidth constraint lines cross each other. If one of the optimal points is within the feasible region, then the optimal solution is just that point (Eq. 6 or 7). If neither optimal point lies within the feasible region (the case shown in Figure 3), the optimal solution is given by the intersection of the two lines (labeled by a circle):

$$\begin{cases} s_1 = \frac{r_2 c_2 - c_1}{r_1 r_2 - 1} \\ s_2 = \frac{r_1 c_1 - c_2}{r_1 r_2 - 1} \end{cases} \tag{8}$$

It is trivial to show that the only case where both optimal points lie within the feasible region occurs for the degenerate case of $r_1 = r_2 = 1$, in which case both points coincide and represent the optimal solution.

Now we reintroduce the condition of finite load (Eq. 2 and 3). In practice, it is possible that the traffic load in one direction is inherently low (e.g., the first hop link from a client may have lots of outgoing interests but few incoming interests). Under such cases, the reduced load in one direction should result in increased shaping rate in the opposite direction. The limiting case is unidirectional demand in which case:

$$i_2 = 0 \tag{9}$$

$$s_1 = \frac{c_2}{r_1} \tag{10}$$

Hence, the instantaneous shaping rate should be variable. We seek proportional fairness between the two-way traffic only if both directions have excessive demand and are competing for the link capacity in both directions. If the load in one direction is inherently low, our scheme is work-conserving and will let the traffic in the other direction grab as much bandwidth as it can. An adaptive algorithm that adjusts the shaping rate between Table 1 and Eq. 10 based on the fluctuating demand will be presented in the next section.

case	s_1	s_2
$\frac{c_1}{c_2} < \frac{2r_2}{r_1 r_2 + 1}$	$\frac{c_1}{2}$	$\frac{c_1}{2r_2}$
$\frac{2r_2}{r_1 r_2 + 1} \le \frac{c_1}{c_2} \le \frac{r_1 r_2 + 1}{2r_1}$	$\frac{r_2 c_2 - c_1}{r_1 r_2 - 1}$	$\frac{r_1 c_1 - c_2}{r_1 r_2 - 1}$
$\frac{c_1}{c_2} > \frac{r_1 r_2 + 1}{2r_1}$	$\frac{c_2}{2r_1}$	$\frac{c_2}{2}$

Table 1: Optimal solution under infinite loads in both directions

3. PRACTICAL ALGORITHM

As we have shown, obtaining the optimal interest shaping rate is mathematically tractable if the shaper has knowledge of the content/interest size ratio (r_1 and r_2), link capacity (c_1 and c_2) and demand (i_1 and i_2) in both directions. Due to the symmetric routing of interests and contents in NDN, the shapers can independently measure r_1 and r_2 by observing the interests and contents arriving at and leaving the interface. If we assume that link bandwidths c_1 and c_2 are static parameters, they can also be made known to the shapers easily. However, the offered demand (i_1 and i_2) is constantly varying and cannot be accurately measured. Further, the shaper at one end of a link cannot know the interest load on the other side without additional message exchange. Hence, we have designed an adaptive algorithm that does not require accurate knowledge of the offered demand on both sides, which we present below.

From Eq. 10, we can compute the *maximum* interest shaping rate max_s_1 (which occurs when $i_2 = 0$). We can also determine the *minimum* interest shaping rate (min_s_1) from Table 1. To determine the actual shaping rate, we measure the incoming interest rate and use it as an estimation of the load on the other side. Assuming a similar shaper is running on the other side of the link, if there is sufficient demand in the reverse direction, then the observed incoming interest rate (obs_s_2) should be no less than the rate given by the s_2 column in Table 1 (we call it the expected minimum incoming interest $expmin\_s_2$). Hence, if $obs\_s_2 \ge expmin\_s_2$, the shaping rate is set to min_s_1. Otherwise, the shaping rate is calculated as follows:

$$min\_s_1 + (max\_s_1 - min\_s_1)(1 - \frac{obs\_s_2}{expmin\_s_2})^2 \tag{11}$$

This equation adjusts the outgoing interest shaping rate between min_s_1 and max_s_1. If $obs\_s_2 = 0$, the shaping rate become max_s_1. As we measure higher incoming interest rate, we reduce the shaped outgoing interest rate until it hits min_s_1. We observe that the quadratic control used here is more conservative and robust than a linear control.

Figure 4 shows how the proposed interest shaper is implemented on each interface of an NDN router. The outgoing packets are first classified into interests and contents. Content packets are passed directly to the link output queue without shaping. Interest packets

Scenario	Data Throughput (Mbps)		Packet Loss Rate (%)		Interest Rejection Rate (%)	
	Client/Server1	Client/Server2	Router1	Router2	Client/Server1	Client/Server2
Baseline	9.558421 ± 0.001261	9.559624 ± 0.001550	0	0	0.0150 ± 0.0006	0.0153 ± 0.0011
Randomized Packet Size	9.432117 ± 0.005931	9.434337 ± 0.007859	0	0	0.0180 ± 0.0014	0.0167 ± 0.0015
Asymmetric Size Ratio	9.373692 ± 0.014214	9.326215 ± 0.000921	0	0	0.0074 ± 0.0006	0.0155 ± 0.0006
Asymmetric Link Bandwidth	9.774441 ± 0.001723	0.719525 ± 0.000139	0	0	0.0119 ± 0.0005	0.0576 ± 0.0000

Table 2: Simulation results over baseline topology

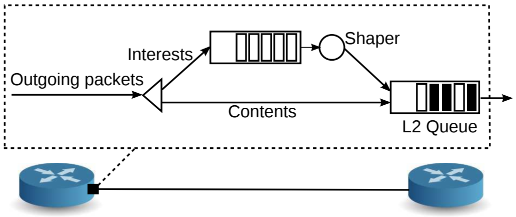


Figure 4: Shaper implementation

join a separate shaper queue, the output of which is fed into the link output queue. The shaping rate of this queue is dynamically computed as per Eq. 11.

It is important to note that hop-by-hop interest shaping alone is inadequate to solve the entire congestion control problem. If a client issues more interests than the network can handle, the excessive interests are discarded by the shaper. This interest loss must be signaled back to the client so that it can slow down its request rate. In this paper, we use the simplest drop-tail policy for the interest queue in the shaper and reject interests with negative acknowledgments (NACK). A similar NACK mechanism has been proposed in [7]. Compared with explicit congestion notification in the current Internet, we believe that using NACK to signal congestion in NDN networks with hop-by-hop interest shaping has some unique advantages. When a neighboring router forwards an interest across a link, enough bandwidth has been accounted for in the reverse direction of this link to accommodate the expected returning content. If this interest is then rejected with a NACK towards the client, this NACK takes the place of the accounted-for content. As long as NACKs are smaller on average than contents, there should be enough bandwidth in all the downstream links for NACKs and they should never get lost due to congestion. Hence, using our shaping scheme, NACKs are a much more reliable and timely method of congestion notification than existing proposals (e.g., the timeout mechanism in [1]).

Clients should react to congestion-triggered NACKs and throttle their outstanding interest rate. Intermediate routers can also react to this congestion signal and implement some sophisticated forwarding strategies such as dynamic rerouting to alternative paths, or mid-stream throttling of flows. In this paper, we demonstrate our shaping algorithm with a simple window-based control on clients only. We defer the design of more sophisticated router reaction to our future work.

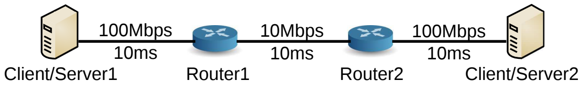


Figure 5: Baseline topology

4. PERFORMANCE EVALUATION

We have implemented our proposal in ndnSIM and evaluated its performance across a number of different scenarios. In our current implementation, each shaper estimates r_1 and r_2 by monitoring the sizes of the interests and contents passing in both directions. The average size of interests (or contents) is calculated by smoothing out the observed samples similar to TCP round-trip time (RTT) estimation. Also we allow for 2% headroom in the shaping to accommodate traffic burstiness. The sizes of the shaper queues as well as the Layer 2 queues are all set to 60 packets. In-network caching is not enabled at this point. All simulations last for 70s and the session start times are randomly picked between 0s and 5s. Metric measurement ignores the first 10s to eliminate any transient effect during the warm-up phase. All test cases have been repeated 12 times and the confidence intervals are calculated.

In the baseline scenario shown in Figure 5, clients at the two ends of the network issue interests for the contents served at the other end. Content payload size is fixed at 1000B and the interest size is 24B (which slowly increases to 28B due to the increasing number of digits in the content name: $/prefix/1$, $/prefix/2$, ...). Table 2 shows the simulation results under this scenario as well as a few other scenario that will be discussed later. As can be seen from the table, the hop-by-hop interest shaper effectively controls congestion so that the packet loss rate at the bottleneck link (due to queue overflow) is zero. About 0.015% of interests issued by the clients are rejected by the shapers with NACK and they serve as congestion signals propagating back to the clients. The throughputs achieved by the clients on both sides of the bottleneck are close to 9.56Mbps. This illustrates the bandwidth consumed by interests and the motivation for our optimization: to achieve 9.56Mbps of data throughput in each direction, we need an additional $9.56 \times 26 \div 1000 \approx 0.24856$Mbps of interest rate in each direction. Therefore, the total throughput of interest and content traffic is 9.80856Mbps. Taking into account the 2% headroom reserved by the shaper, our shaping algorithm achieves the optimal possible throughput.

Scenario	Data Throughput (Mbps)		Packet Loss Rate (%)		Interest Rejection Rate (%)	
	Client/Server1	Client/Server2	Router1	Router2	Client/Server1	Client/Server2
Homogeneous RTT	5.142089 ± 0.505369	4.692407 ± 0.505271	0	0	0.0515 ± 0.0112	0.0620 ± 0.0129
Heterogeneous RTT	5.209043 ± 0.384781	4.624094 ± 0.380328	0	0	0.0513 ± 0.0092	0.0428 ± 0.0067
	Client/Server1	Client/Server4	Router1	Router2	Client/Server1	Client/Server4
Heterogeneous RTT	9.565575 ± 0.000762	9.419777 ± 0.007525	0	0	0.0148 ± 0.0004	0.0116 ± 0.0005

Table 3: Simulation results over dumbbell topology

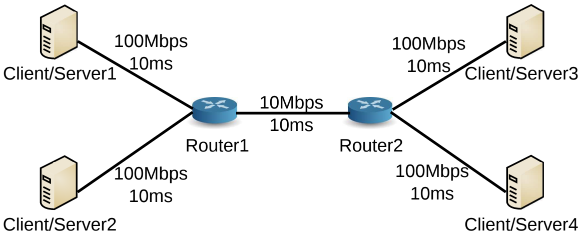


Figure 6: Dumbbell topology

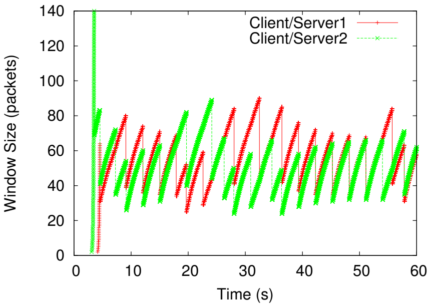


Figure 7: Client window size evolution under homogeneous RTT

We now investigate a few variations of the baseline scenario. First, we evaluate the shaper under randomized packet size. Recall that the shaper estimates r_1 and r_2 by smoothing packet size samples. When the sizes vary over time, the shaper may have inaccurate estimation of r_1 and r_2, leading to sub-optimal shaping behavior. However, as shown in Table 2, when the interest packet size is uniformly distributed between 27B and 62B, and the content payload size between 600B and 1400B, the shaper still achieves zero packet loss and we only experience minor throughput loss. Second, we evaluate the scenario where the content/interest size ratios in two directions are asymmetric (i.e., $r_1 \neq r_2$). In this scenario, the content payload size on Client/Server2 is set to 500B while the other side still has 1000B payloads. Due to the reduced payload size, Client/Server1 needs to send interests at twice the original rate to achieve the same data throughput. Hence, we experience lower data throughput at Client/Server2 because more link bandwidth is allocated to the interests from Client/Server1. Finally, we simulate the asymmetric link bandwidth scenario (i.e., $c_1 \neq c_2$), as is commonly found in today's residential access networks. Here the bottleneck link bandwidth from Router1 to Router2 is reduced to 1Mbps while the reverse link remains at 10Mbps. As simulation results show, our hop-by-hop interest shaper still manages to control congestion under this highly asymmetric scenario. Client/Server2 achieves approximately 0.72Mbps data throughput out of the 1Mbps link, leaving the rest to the interests from Client/Server1 that bring back 9.77Mbps of contents. This also outlines the importance of the logarithmic utility function which avoids starvation of either side.

Next, we evaluate our scheme under the dumbbell topology shown in Figure 6. Three different scenarios have been simulated. First, we launch two clients on the left side of the bottleneck link. They retrieve contents from the two servers on the right side respectively. Here the two flows have homogeneous RTTs of 60ms and their performance is shown in Table 3 along with other scenarios under the same topology. As Figure 7 shows, the two flows roughly converge to fair bandwidth sharing despite the randomized session start time. Note that, in this scenario, there are no interests coming from the right side of the bottleneck link, and therefore all this bandwidth can be used for the returning contents. Our shaping algorithm has correctly adapted to this situation, as evidenced by the sum of the throughputs of these two flows being around 9.8Mbps.

We reran this test but set the link latency between Router2 and Client/Server4 to 20ms so that the two flows now have heterogeneous RTTs. The results are also shown in Table 3. Figure 8 compares the evolution of the bottleneck queue length over time between the homogeneous RTT case and the heterogeneous RTT case. As the figures show, we have minimum queues (oscillating between zero and one packet) under homogeneous RTTs since the interests going out from Router1 to Router2 are perfectly paced by the shaper. With heterogeneous RTTs, we see more fluctuations in the queue length due to the burstiness of the traffic incurred by RTT heterogeneity. However, even in this case the queue is well controlled (maximum of 17 packets). Note that our shaper is an open-loop controller that aims at long-term fairness and stability of the system. An alternative is a closed-loop controller that adjusts the shaping rate based on the instantaneous queue length on the other side. However, such a solution may be much more complicated and costly since the queue states need to be communicated using some messaging or piggybacking technique. We believe that with proper parameter tuning, our open-loop controller is able to make the system converge to the optimal steady state.

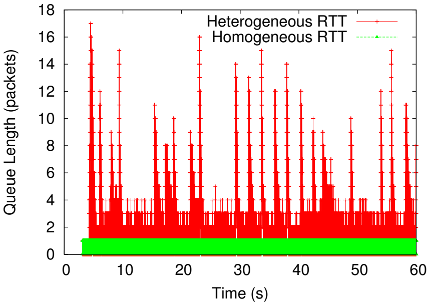


Figure 8: Comparison of bottleneck queues

Finally, we simulate the following heterogeneous RTT scenario: one client resides on Client/Server1 and retrieves contents from Client/Server3. Another client resides on Client/Server4 and retrieves contents from Client/Server2. Since the link latency between Router2 and Client/Server4 is 20ms, these two flows sharing the bottleneck link in reverse directions have different RTTs. The results in Table 3 show that our hop-by-hop interest shaper works perfectly in this scenario, too.

5. RELATED WORK

A number of hop-by-hop interest shaping algorithms have been proposed in the literature. [7] proposed the simplest algorithm where the interest rate is shaped so that the returning data rate matches the reverse link capacity multiplied by a pre-determined coefficient between 0 and 1. A more sophisticated shaping algorithm was proposed in [6] where the shaping rate is dynamically adjusted based on the instantaneous queue length and available capacity to send data packets at time t. A window-based mechanism described in [1] controls the interest sending rate of a client in TCP-like fashion. This mechanism was extended in [2] by adding a hop-by-hop interest shaping scheme where each flow is allocated a fair share of the link bandwidth and shaped accordingly. [5] also presented a per-flow bandwidth fair sharing mechanism. Interests exceeding the fair share are discarded with NACKs and simple AIMD schemes are used to adjust the client's interest rate.

Our scheme differs from these existing efforts in a few ways. First, our scheme works well in bidirectional traffic scenarios, where we believe that fair allocation of bandwidth between interests and contents requires the consideration of their interdependence. This is missing in all prior proposals. Second, we note that flow identification at intermediate routers in NDN is difficult due to the absence of any client identifying information in interest packets. Our scheme does not require any awareness of flows by the shaper. Third, the proposed shaping algorithm is well-suited to handle highly asymmetric link bandwidth and highly varying packet sizes, maximizing utilization in both directions while preventing starvation in either. Fourth, NACKs generated by our shaper offer a timely congestion signal to clients. Our shaping scheme ensures that NACKs have a very low probability of getting dropped, thus avoiding the traditional problem with congestion-generated NACKs exacerbating the congestion problem. These NACKs can also be used by intermediate routers to enforce client shaping. Finally, our scheme detects congestion *downstream* of the bottleneck link. This enables congestion-aware rerouting of interests at the correct point without any additional messaging.

6. CONCLUSION

In conclusion, we have presented a hop-by-hop interest shaping algorithm for NDN to avoid network congestion and achieve optimal network resource utilization. Our proposed solution accounts for the interdependence between interests and contents in opposite directions and is capable of optimally sharing link bandwidth without extra message exchange. We evaluated the performance of our mechanism when combined with simple AIMD clients using ndnSIM, and studied its behavior under varying conditions of bandwidth, load, and packet size ratios. Our future work includes the design of more sophisticated backpressure mechanisms as well as congestion-aware multipath routing.

7. REFERENCES

[1] 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.

[2] 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.

[3] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard. VoCCN: Voice over Content-Centric Networks. In *ACM ReArch'09*, Dec. 2009.

[4] F. Kelly. Charging and Rate Control for Elastic Traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

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

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

[7] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang. A Case for Stateful Forwarding Plane. Technical Report NDN-0002, July 2012.

[8] L. Zhang and et al. Named Data Networking Project. Technical Report NDN-0001, Oct. 2010.