Preprint version of the paper Published in IEEE Global Communications Conference 2020,
7-11 Dec. 2020, DOI: 10.1109/GLOBECOM42002.2020.9322585

Hop-by-Hop Congestion Measurement and Practical Active Queue Management in NDN

Yuhang Ye†, Brian Lee‡, Yuansong Qiao†
Software Research Institute
Athlone Institute of Technology
Athlone, Co. Westmeath, Ireland
†{yye, ysqiao}@research.ait.ie, ‡{blee}@ait.ie

***Abstract*—Content replication and name-based routing in Named Data Network (NDN) naturally lead to connectionless multi-source and multipath transmissions. Traditional congestion control designed for end-to-end connections cannot well fit this architecture. Explicit congestion notification (ECN) can better support NDN because congestion is detected where it occurs and ECN can timely notify the traffic initiator of congestion. NDN can be deployed as an overlay protocol (sharing the underlying devices with other protocols), which means the congestion may also occur at an underlying device (e.g. a switch). In this case, the NDN nodes cannot access the queue or other link status at a remote underlying device for congestion detection. A promising ECN scheme must be able to detect congestion happening anywhere (at an NDN node or an underlying device) without using underlying link information. This paper proposes Hop-by-Hop Congestion Measurement (HbHCM) and Practical Active Queue Management (PAQM) to enable detecting congestion and generating ECN at NDN nodes via monitoring the change of transmission delays. HbHCM measures the transmission delay at the hop level and PAQM converts the delay to ECN signals to notify consumers. We compared HbHCM + PAQM with two milestone solutions (router-label and ECN-based). The simulation results show that HbHCM + PAQM can accurately detect congestion, improve bandwidth utilisation and better support multipath transmission, no need to rely on route or link information.**

I. INTRODUCTION

Content-centric applications have dominated Internet usage. To improve the performance of content distribution, Named Data Networking (NDN) [1] is proposed to address the inefficiency in TCP/IP by routing request (Interest) and content (Data) packets using universal resource identifiers (names) instead of host addresses. Name-based routing and possible content replication (e.g. in-network cache) naturally lead to a multipath and multi-source transmission paradigm.

The connectionless and multipath features make traffic control in NDN difficult. A consumer may download the different fractions in a content file from multiple providers concurrently whereas it does not know which fraction is from which provider. In consequence, the congestion control methods proposed for traditional TCP/MPTCP are no longer effective. The majority of SOTA solutions (e.g. [2], [3]) require NDN routers to detect congestion and generate explicit notifications (e.g. NACK and ECN) thus informing consumers. Compared to the early-stage approaches which detect congestion via implicit guesses (e.g. delay or packet loss), the explicit notification is more accurate and timelier and achieves better bandwidth utilisation. However, explicit congestion notification requires NDN routers to monitor the link status (e.g. queue occupancy or link usages) which is not always feasible for NDN nodes especially when NDN is an overlay protocol. For example, if congestion happens at a lower-layer device (e.g. a switch) connecting two NDN nodes, it is not straightforward for the two nearby NDN nodes to detect congestion because they cannot access the link status of the remote switch. This is an issue that cannot be supported by existing ECN approaches.

In this paper, we will fill the gap that *if congestion is not happening at an NDN node but at a lower layer device, how to detect and manage congestion at a nearby NDN node*. The key contributions of this paper are the following:

- Hop-by-Hop Congestion Measurement (HbHCM): it enables estimating congestion for each NDN link according to transmission delays. HbHCM only needs NDN nodes to exchange round-trip time (RTT) thus estimating hop-by-hop queue delay as the congestion level of an NDN link.
- Practical AQM (PAQM): It demonstrates how an AQM algorithm can be implemented at the NDN layer. PAQM is a mapping function from the transmission delay to the chance of generating explicit signals.
- A CoDeL implementation: The CoDeL algorithm is implemented using the PAQM framework and tested with a naïve AIMD algorithm.

HbHCM and PAQM can be viewed as a replacement pillar of the existing ECN-based congestion control approaches when NDN is served as an overlay. The performance is evaluated via ndnSIM. Via comparative experiments, the results show that without using link or route information, HbHCM and PAQM can still achieve fair bandwidth sharing and high link utilization compared to the existing milestone solutions relying on link and/or routing information.

II. RELATED WORK

Over the last couple of years, there have been several proposals for congestion control in NDN. Early-stage works [4], [5] primarily focus on transplanting traditional TCP congestion control to NDN. Interest Control Protocol (ICP) [4] detects congestion according to whether the RTO timer is timeout. However, the multi-source and multipath of NDN makes traditional congestion detection by RTT measurement or packet losses inadequate. Therefore, researchers started to explore using in-network congestion detection and explicit congestion notification (ECN). One of the popular solutions is Practical NDN congestion control [2] (PCON) which detects link congestion and generates ECNs by monitoring the

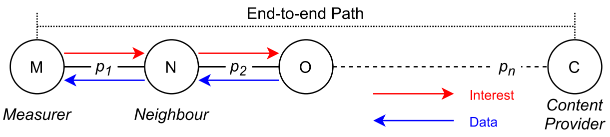

Figure 1 Example Linear Topology

behaviour of link-layer queues. In Backpressure Interest Control Protocol (B-ICP) [6] and Improved Hop-by-hop Interest Shaping (HIS) [7], routers estimate the optimal Interest sending rate for each interface based on the link capacity of the local and remote links. Hop-by-hop interest shaping (HbHIS) [8] and its extension [9] calculate the Interest shaping rate for each interface based on both the number of queued packets and the link capacity. These traffic shaping approaches notify consumers according to the virtual queue. The limitation of the existing ECN-based solutions is that they need underlying link status to infer congestion, which becomes infeasible once the information is unavailable.

In addition to the ECN-based method, there is another type of congestion control, usually called route-label. Route-label approaches allow each consumer to detect congestion or even control the traffic on each end-to-end path separately, using so-called route or path identifiers. Remote Adaptive Active Queue Management (RAAQM) [10] is a pioneering approach which separates the congestion measurement for each path and detects congestion based on the variation of RTTs. The rate-based, multipath-aware ICN congestion control (MIRCC) [11] approach directs Interest forwarding to the specified path. Inspired by Rate Control Protocol [12], MIRCC lets routers calculate the desired Interest requesting rate for each flow and notify consumers. Path-specified Transport Protocol (PTP) [13] similar to MIRCC controls the traffic on each path independently. It is more scalable than MIRCC since it avoids rate estimations at routers. Although the route-label approaches achieves better performance, the practicality and scalability of route-labels have been questioned because of the scalability and security concern [2].

This motivates the development of HbHCM + PAQM which allows routers to accurately detect congestion without 1) path/route information and 2) link status.

III. HOP-BY-HOP RTT MEASUREMENT

HbHCM measures the transmission delay of each NDN link (between two adjacent NDN nodes) to estimate the congestion of the devices sitting between the two NDN nodes. A possible method is to measure the single trip time between the two nodes. However, this needs clock synchronisation which is impractical in large scale networks. Moreover, the deviations in crystal performance may cause cumulative errors. This paper uses the RTT between two adjacent NDN nodes (HbHRTT) for congestion detection.

A. Model

End-to-end RTT can be directly measured at any NDN node, denoting the interval between sending out an Interest and receiving the Data packet. The relationship between queuing delay and end-to-end RTT is given in equation (1).

$$r_P^M = \sum_{n\in P} d_n^I + d_n^D \qquad (1)$$

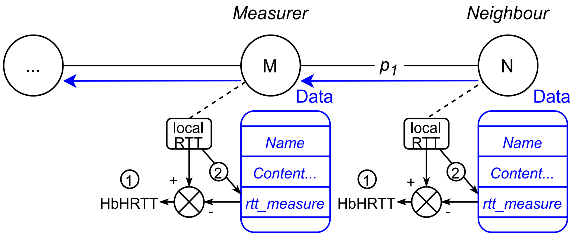

Figure 2 Calculate HbHRTT① and Update the "*rtt_measure*" Field②

Here, r_P^M indicates the end-to-end RTT measured at node *M*, $P=[p_1,\ldots,p_n]$ denotes the Interest-Data forwarding path (*M*–*N*–*O*—*C*) as shown in Figure 1, in which p_1 is the first hop between measurer *M* and its neighbour *N* on the path. d_n^I and d_n^D denote the Interest and Data transmission delay at each hop. For the neighbour *N*, its RTT measurement is given equation (2).

$$r_{P/p_1}^N = \sum_{n\in P/p_1} d_n^I + d_n^D \qquad (2)$$

Here, P/p_1 denotes the shared path (*N*–*O*—*C*) from *N* to *C* after hop p_1. Using equation (2) and (1), we will get the hop-by-hop measure of RTT (HbHRTT) $r_{p_1}^M$ at hop p_1 measured by *M*.

$$r_{p_1}^M = r_P^M - r_{P/p_1}^N = d_{p_1}^I + d_{p_1}^D \qquad (3)$$

Here, $r_{p_1}^M$ reflects the change of the queuing delay at p_1. The relationship between HbHRTT and queuing delay is given in equation (4). The superscript (*M*) is removed for clarity.

$$r_{p_1} = b_{p_1} + d_{p_1} \qquad (4)$$

Here, b_{p_1} denotes the base HbHRTT (the minimum delay to transmit an Interest and the Data via p_1), $d_{p_1} = d_{p_1}^I + d_{p_1}^D$ is the queuing delay (for outgoing Interest and incoming Data).

Both Interest and Data packets may cause congestion whereas a hop is likely to be congested in only one direction rather than both. This section will discuss the queuing delay caused by Data congestion for clarity. The idea is also applicable when the queuing delay is caused by Interest congestion. The relationship between the queuing delay and the number of queued packets is given in equation (5).

$$d_{p_1}^D = q^D / w^D \qquad (5)$$

Here, q^D is the number of queued Data packets at the bottleneck and w^D is the bandwidth of the bottleneck. Obviously, the increase of the $d_{p_1}^D$ indicates that more Data packets are congested at p_1. Because w^D is unavailable to NDN nodes, the proposed solution generates congestion signals based on queuing delays rather than queued packets.

B. Implementation

The implementation of HbHCM is straightforward. According to equation (3), if *M* needs its local measurement r_P^M and the RTT measurement r_{P/p_1}^N from *N* to estimate the

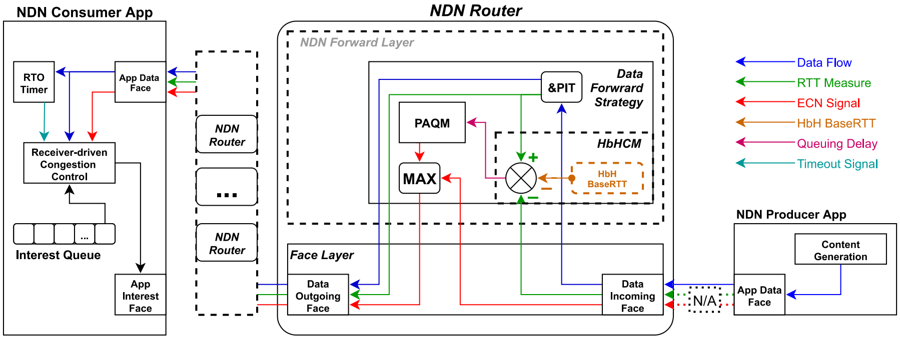

Figure 3 HbHCM + PAQM Architecture

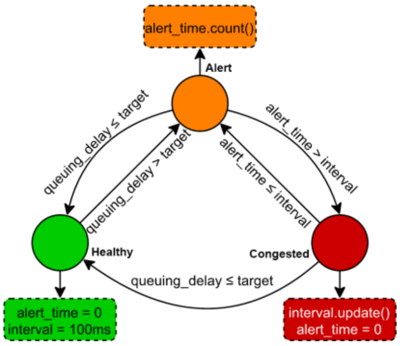

Figure 4 PAQM-CoDeL State Machine

HbHRTT $r_{p_1}^{M}$. HbHCM uses piggyback to exchange RTT measurements. Each Data packet is appended with a new metadata field "*rtt_measure*" as shown in Figure 2. When N receives a Data packet, it first calculates HbHRTT $r_{p_2}^{N}$ based on the local RTT measurement and the value stored in *the rtt_measure* field (step ①). Then, N updates this field with the local measurement (step ②) and forwards this Data packet to M to calculate $r_{p_1}^{M}$. This allows NDN nodes to calculate HbHRTT for each hop. For the content provider, this field is initialised to 0 because there is no previous transmission.

To estimate the hop-by-hop queuing delay d_{p_1} according to equation (4), the node also needs to know the base HbHRTT b_{p_1}. In the current implementation, b_{p_1} is the minimal HbHRTT that has been measured so far. The accuracy of base HbHRTT has less effect on the effectiveness because if b_{p_1} is higher than the true value, the consequence is to have a higher equilibrium point (i.e. more queued packets) than the original one. Because the minimum function is monotonic, the equilibrium point will always converge towards the correct equilibrium point.

C. *Impact of Packet Size*

The packet size will impact the base HbHRTT. Using the same bandwidth at a hop, transmitting a larger packet costs longer time delay, therefore the estimations of queuing delay can be inaccurate. A compensation method is introduced here to compensate for the estimation. The relationship between base RTT b and the packet size p is given below:

$$b = \sum_{u} \frac{p}{w_u} + c = p \sum_{u} \frac{1}{w_u} + c = p\mathbf{W} + c$$

Here, u denotes the hop counts at the underlying layers, w_u denotes the bandwidth of each hop and c denotes the constant processing time that is irrelevant to packet size. Obviously, for a stationary underlying connection, $1 / w_u$ for each u is a constant value, i.e. $\mathbf{W} = \sum_u (1/w_u)$ is a constant. The round-trip time b is an affine function to the packet size p with two unknowns $\mathbf{W}$ and c so they can be easily estimated by linear regression using samples $\{p, b\}$.

IV. PRACTICAL AQM

Based on the hop-by-hop queuing delay, the intermediate NDN nodes can generate explicit signals to notify upstream[1] nodes (e.g. consumers). This section will introduce an NDN layer AQM based on the queuing delay, namely PAQM.

A. *PAQM Architecture*

HbHCM and PAQM are implemented as an NDN forwarding strategy for Data packets. The main function of HbHCM is to estimate queuing delay while that of PAQM is to attach and update the ECN flag on each Data packet. The system architecture is shown in Figure 3. Each Data packet will carry an ECN flag and the "*rtt_measure*" field

To realised congestion detection and feedback, the strategy operates based on three flows (Data, RTT and ECN). For a received Data packets, the node (e.g. M) first locates its request history (an existing PIT entry) to calculate the local RTT between M and the content provider. Because the Data packet also carries the nested RTT which denotes the RTT between its neighbour node N and the provider, M can estimate the HbHRTT between M and N (the two green input wires to $\otimes$). By eliminating base HbHRTT (the brown input wire to $\otimes$) from the HbHRTT, the router can get the queuing delay (the purple output wire from $\otimes$) between M and N. The queuing delay is fed into the PAQM module which maps the queuing delay to an ECN output. Then, PAQM updates the ECN flag carried by the Data packet if the ECN output is 1. The MAX operator indicates any detected congestion will trigger the ECN flag to 1 thus notifying upstream nodes of congestion. For example, if ECN is solely used for the receiver-driven congestion control, once a Data packet (with ECN=1) is received by a consumer, the consumer will reduce its congestion window according to the control law.

B. *PAQM-CoDeL*

The architecure facilitates an NDN node to realise versatile AQM schemes at the NDN-layer thus interacting with the traffic control at upstream nodes. In this section, we will present an effective PAQM approach (PAQM-CoDeL) based on the queuing delay generated by HbHCM.

PAQM-CoDeL follows the design principle of CoDeL, detecting congestion according to packet sojourn time without accessing the real link-layer queues. The principle of CoDeL is to distinguish good and bad queues via monitoring the change of packet sojourn time. For a good queue (caused by

---

[1] Interest forwarding: downstream; Data forwarding: upstream.

busty traffic), CoDeL does not generate congestion signals (e.g. drop packets). For a bad queue, CoDeL starts to generate signals if it recognises that the queue length does not converge to a low level due to the overmuch traffic.

To emulate the behaviour of CoDeL, PAQM-CoDeL lets the router monitor the queuing delay (from HbHCM) as the packet sojourn time. PAQM-CoDeL operates based on a state machine (Figure 4) with three states (Healthy, Alert and Congested). By default, the interface is in *Healthy*. If the queuing delay of a received Data packet is smaller than the target value (5ms), the interface will return to *Healthy* and will not set ECN to the packet. If a healthy interface receives a Data packet with a queuing delay larger than the target value, the interface will switch to *Alert*. Once an interface enters *Alert*, it will set up a timer to count how long the interface has been in *Alert*. If the state is *Alert*, the interface does not need to set ECN to Data packets. Once the value of the timer hits a predefined time interval (100ms by default), the interface will switch to *Congested* and start set ECN to the Data packet. Then, the interface resets the timer, updates the time interval Δ according to CoDeL's control law [14] according to equation (6), and returns to *Alert*.

$$\Delta := \Delta/\sqrt{N} \tag{6}$$

Here N denotes the total number of Data packets that have been marked with ECN since the interface enters *Alert*. Regardless of whether the interface is in *Alert* or *Congested* if the queuing delay of a newly received Data packet is less than the target value, the interface will return to *Healthy*.

V. LIMITATION

There are two major limitations of HbHCM. Firstly HbHCM cannot estimate the exact number of queued packets in a hop because NDN nodes may not be able to access the bandwidth of the bottleneck. As a result, the congestion signal needs to be generated according to the queuing delay rather than the number of queued packets. In consequence, PAQM cannot support the queue length-based AQM scheme. A potential solution is to utilise packet pair to probe the bottleneck bandwidth of each link. The second limitation is that PAQM cannot directly control the queue as the real AQMs. For example, when the original CoDeL detects continuously high sojourn time of queued packets, it can fast drain the queue by dropping the long-wait packets. Unfortunately, PAQM does support this feature because the real congested queue is not accessible to PAQM. Instead, PAQM can only notify upstream nodes to reduce requesting rate thus to reduce queue occupancy. This longer control loop causes a delayed reaction to congestion. Thus, the queue occupancy will be higher than the target value. Section VI will demonstrate this issue. From the experiment results, we found that the increased queuing delay is trivial to most applications.

VI. EVALUATION

HbHCM and PAQM are implemented through the NDN Forwarding Daemon and evaluated via ndnSIM [15]. The performance will be evaluated from 4 perspectives: 1) bandwidth utilisation, 2) user fairness, 3) end-to-end latency and 4) packet loss rate. Its implementation will be compared with the popular and widely acknowledged traffic control approaches 1) Optimal Multipath Congestion Control and Request Forwarding (OMCC-RF) [10] and 2) Practical Congestion Control (PCON) [2]. OMCC-RF is a route-label method. It includes a forwarding strategy called Request Forwarding Algorithm (RFA) and a congestion control scheme called Remote Adaptive Active Queue Management (RAAQM). PCON is an ECN-based method. It includes an ECN-based forwarding strategy and a BIC-like congestion control scheme (also based on ECN). Because the scope of this paper does not cover adaptive forwarding or congestion control, HbHCM +PAQM only employs the concise methods (RFA and AIMD) for a demonstration purpose.

A. Simulation Setup

The payload of Data packets is set to a constant (1024) for each application. If a node can cache content, its buffer will hold the data packets for at least 20s. For certain bandwidth figures, we present the results of the first 20 (50 or 100) seconds for clearer illustrations.

B. Scenario 1: Multi-flow + Cache

This experiment targets to validate the proposed approach in a multi-flow case when the in-network cache is only available for one flow. The topology is shown in Figure 5. The single-direction latency is 10ms for each link. Router B is enabled with cache. Started from 0s, Consumer 1 downloads content_1 from Router B and Producer 1. During the first 20s, Consumer 1 can retrieve content from Router B then Consumer 1 will downloading content from Producer 1. Started from 5s, Consumer 2 downloads content_2 from Producer 2. Figure 6 –10 report the performance of the three approaches. We can see that all of the three approaches can well support in-network caches by near-perfect utilisation of the bottleneck bandwidth (15Mbps). For the downloading rate, we could see that HbHCM achieves a near-perfect bandwidth utilisation (96.4%) whereas the performances of OMCC-RF and PCON are worse (94.6% and 92.5%). For OMCC-RF, although it implemented a remote AQM at the consumer-side for congestion detection, the detection accuracy is somehow affected by the bursty change of queuing length, e.g. multiple packets arriving within a short period. PCON employs the TCP BIC window adaptation algorithm however, we observed that the consumer sometimes overly react to the congestion signals emitted by the queue thus causing the under-utilisation of the bottleneck link. For exchange, HbHCM requires a higher queue occupancy than the other two. In Figure 7, we can see that HbHCM introduced 7.8ms queuing delay in average which is slightly higher than the target sojourn value (5ms). Because the bottleneck is sometimes under-utilised, the averaged queuing delays of OMCC-RF and PCON are slightly lower (4.5ms and 5.9ms). We measure the multi-flow fairness using the Jain's index as shown in Figure 6. Between 0s to 5s, because only Consumer 1 is downloading, the fairness is 0.5. Between 5s to 20s, Consumer 1 downloads content from Router B while Consumer 2 downloads content from Producer 2. None of the proposed approach can achieve a fair share of the bandwidth because of the RTT fairness issue. The necessity of RTT fairness is controversial and is beyond the scope of this paper, therefore we only calculate the fairness for the period from 20s to 50s. From the legend in Figure 6, we can see that HbHCM achieves the highest score.

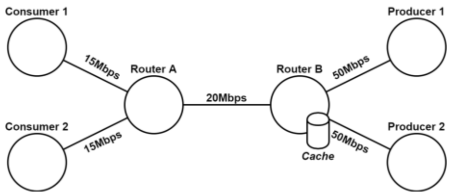

Figure 5 Two-flow and Cache Scenario

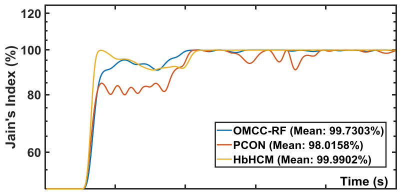

Figure 6 Jain's Index

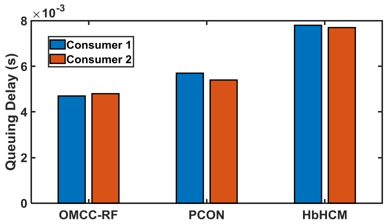

Figure 7 Averaged Queuing Delay

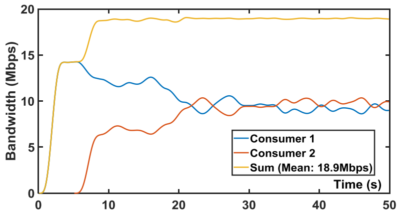

Figure 8 Downloading Rate of OMCC-RF

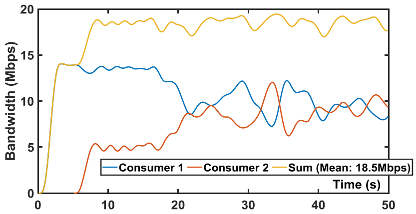

Figure 9 Downloading Rate of PCON

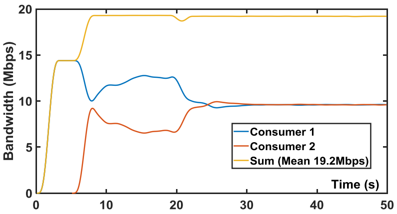

Figure 10 Downloading Rate of HbHCM

C. *Scenario 2: Multi-source Transmission*

The second experiment validates the effectiveness of HbHCM to support multi-source transmission. The topology is shown in Figure 11. Figure 12 compares the downloading rate of the three protocols. We can see that HbHCM receives the highest averaged downloading speed comparing to the others. HbHCM uses the same adaptive forwarding strategy – RFA as OMCC-RF. HbHCM and RFA determine traffic allocations for each interface based on the number of pending Interests. HbHCM achieves a certain level of bandwidth improvement compared to OMCC-RF because the congestion signal generated by the PAQM-CoDeL is more reliable whereas the congestion control in OMCC-RF is more sensitive to RTT variations, i.e. the consumer reduces the congestion window earlier than it should do. PCON uses on ECNs to adjust the multipath traffic distribution which can better balance the loads on bottlenecks. This is observed as PCON achieves the highest instantaneous rate. Nevertheless, the curve shows that the allocation algorithm of PCON is unstable, which results in the periodic reductions of bandwidth usage. Figure 13 reports the averaged queuing delay of the three approaches. We can see that HbHCM introduces a slightly higher latency due to its delayed response to the queued packets (as we mentioned in Section V) to the congestion. The delay of 4.9ms usually will not have significant impacts on normal applications.

D. *Scenario 3: Simulated Overlay*

The current ndnSIM does not support NDN as an overlay. To this end, we simulate an overlay scenario by adding a pair of "simplified" NDN nodes (to mimic lower layer devices) between two normal NDN nodes. The simplified nodes can be configured to disable the adaptive forwarding and congestion detection functions, i.e. they work like switches. The topology is given in Figure 14. In this scenario, we consider two cases: 1) each simplified node works like a normal NDN router and can detect congestion (Active) and 2) all simplified nodes cannot detect congestion (Inactive). In Figure 15, it presents the downloading bandwidths of HbHCM and PCON in the two cases. HbHCM outperforms PCON in both cases. This is because no matter if the simplified nodes can detect congestion or not, the closest NDN node (Consumer 1) is capable to measure congestion based on the change of transmission delay between Consumer 1 and Router 1. By contrast, PCON works fine if the simplified nodes can access the queue information and generate ECN. Once the congestion detection is not available for the simplified nodes, the nearby NDN nodes also cannot detect congestion because neither Consumer 1 nor Router A can access the queue of the bottleneck. As a result, the PCON operates as a pure loss-based approach. In this case, the PCON consumer tries to fill the bottleneck queue and reduce requesting until packet loss is detected. Because of the aggressive behaviour of filling the queue and overreacting to packet loss, the consumer's downloading bandwidth is highly decreased. The queuing delays of the two approaches and two cases are given in Figure 16. For the same reason, we can see that the averaged queuing delay of PCON in the Inactive case is significantly higher than the rest. In HbHCM, the queuing delays are not much different in the two cases because any NDN nodes nearby the bottleneck can detect congestion and timely reply with signals. In both cases, the averaged queuing delays of HbHCM are slightly higher than PCON in the active case because of the larger queue occupancy.

E. *Scenario 4: Complex Topology*

This experiment compares the performance of HbHCM with others in a multi-user and multi-path scenario. Three consumers download content from three servers (Amazon, Warner and Google) via an Abilene topology as shown in Figure 17. The multipath routing is enabled which allows a consumer to retrieve packets from the server via multipath paths. Figure 18 demonstrates the downloading bandwidths of the 3 consumers over time. HbHCM outperforms the others in terms of stability and convergence. Figure 19 reports the averaged bandwidth, Jain's index and averaged queuing delay for the three approaches. We can see that HbHCM achieves the highest bandwidth utilisation and fairly sharing of bandwidth amongst users. For the same queuing occupancy reason, the delay is slightly higher than the other two.

VII. CONCLUSION

NDN brings unique challenges to detect congestion due to the multi-source (multipath) and connectionless features. This paper described HbHCM and PAQM, a novel method to detect congestion at any device in an NDN network. In contrast to conventional solutions, the proposed method

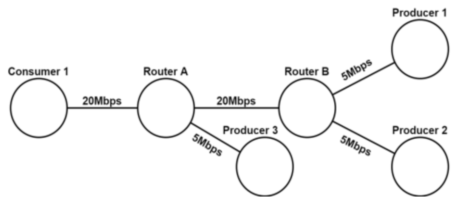

Figure 11 Multi-source Topology

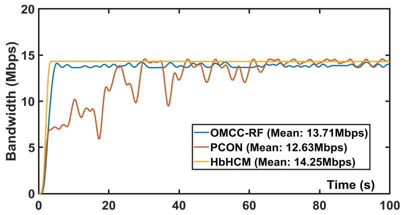

Figure 12 Downloading Bandwidth

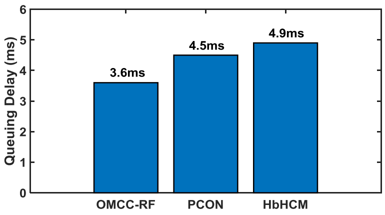

Figure 13 Averaged Queuing Delay

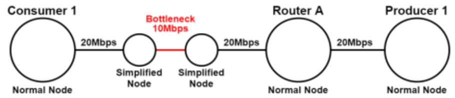

Figure 14 Simulated Overlay Topology

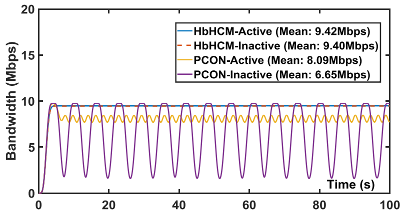

Figure 15 Downloading Bandwidth

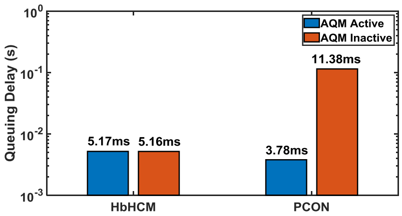

Figure 16 Averaged Queuing Delay

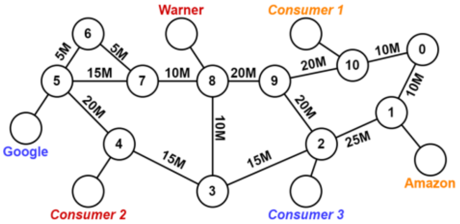

Figure 17 Abilene Topology

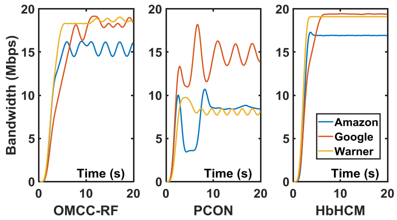

Figure 18 Downloading Bandwidth

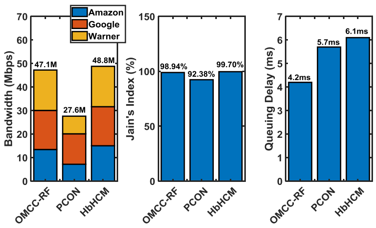

Figure 19 Bandwidth, Fairness and Delay

avoids using route labels, and it does not need any link information. HbHCM only requires the RTT measurements from local and the downstream neighbours for congestion estimation. This endows HbHCM the ability to detect congestion that is not happening at NDN nodes. PAQM supports developing a CoDeL scheme at the NDN layer. The experiment results show that even with a plain forwarding strategy (RFA) and a naive congestion control algorithm (AIMD), the system can achieve decent performance. The main limitation of the current HbHCM and AQM is slightly higher queue occupancy because of the longer control loop. In the future, we consider adding an Interest shaping scheme to shorten the control loop. The shaping scheme will facilitate implementing a QoS control module for the data flows with different QoS requirements. It is worth mentioning that HbHCM and PAQM do not conflict with route-label approaches. Rather, they can help to generate more accurate congestion signals thus further improving their performance.

ACKNOWLEDGEMENT

This publication has emanated from research conducted with the financial support of Irish Research Council under ID GOIPD/2019/874, Science Foundation Ireland (SFI) under Grant Number SFI 16/RC/3918 and 19/FIP/ZE/7383, co-funded by the European Regional Development Fund.

REFERENCE

[1] L. Zhang *et al.*, ‘Named Data Networking’, *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014, doi: 10.1145/2656877.2656887.

[2] K. Schneider, C. Yi, B. Zhang, and L. Zhang, ‘A practical congestion control scheme for named data networking’, in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, 2016, pp. 21–30.

[3] F. Zhang, Y. Zhang, A. Reznik, H. Liu, C. Qian, and C. Xu, ‘A transport protocol for content-centric networking with explicit congestion control’, in *2014 23rd international conference on computer communication and networks (ICCCN)*, 2014, pp. 1–8.

[4] G. Carofiglio, M. Gallo, and L. Muscariello, ‘ICP: Design and evaluation of an Interest control protocol for content-centric networking’, in *2012 Proceedings IEEE INFOCOM Workshops*, Mar. 2012, pp. 304–309, doi: 10.1109/INFCOMW.2012.6193510.

[5] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, and N. Blefari-Melazzi, ‘Receiver-driven interest control protocol for content-centric networks’, in *Proceeding of ACM SIGCOMM Workshop on Information Centric Networking (ICN)*, 2012.

[6] Y. Ye, B. Lee, R. Flynn, N. Murray, and Y. Qiao, ‘B-icp: Backpressure interest control protocol for multipath communication in ndn’, in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[7] 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’, *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 55–60, 2013.

[8] N. Rozhnova and S. Fdida, ‘An effective hop-by-hop interest shaping mechanism for ccn communications’, in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, 2012, pp. 322–327.

[9] N. Rozhnova and S. Fdida, ‘An extended Hop-by-hop Interest shaping mechanism for Content-Centric Networking’, in *IEEE GLOBECOM 2014*, 2014.

[10] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang, ‘Optimal multipath congestion control and request forwarding in information-centric networks’, in *Network Protocols (ICNP), 2013 21st IEEE International Conference on*, 2013, pp. 1–10.

[11] M. Mahdian, S. Arianfar, J. Gibson, and D. Oran, ‘MIRCC: Multipath-aware ICN rate-based congestion control’, in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, 2016, pp. 1–10.

[12] N. Dukkipati, Rate Control Protocol (RCP): Congestion control to make flows complete quickly. Citeseer, 2008.

[13] Y. Ye *et al.*, ‘PTP: Path-specified transport protocol for concurrent multipath transmission in named data networks’, *Comput. Netw.*, vol. 144, pp. 280–296, 2018.

[14] K. Nichols and V. Jacobson, ‘Controlling queue delay’, *Commun. ACM*, vol. 55, no. 7, pp. 42–50, 2012.

[15] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, ‘ndnSIM 2.0: A new version of the NDN simulator for NS-3’, *NDN Tech. Rep. NDN-0028*, 2015.