



LQCC: a Link Quality-based Congestion Control scheme in named data networks

Hakima Khelifi, Senlin Luo, Boubakr Nour, Hassine Moun gla

► To cite this version:

Hakima Khelifi, Senlin Luo, Boubakr Nour, Hassine Moun gla. LQCC: a Link Quality-based Congestion Control scheme in named data networks. WCNC 2019: IEEE Wireless Communications and Networking Conference, Apr 2019, Marrakech, Morocco. pp.1-6, 10.1109/WCNC.2019.8885909 . hal-02100494

HAL Id: hal-02100494

<https://hal.science/hal-02100494>

Submitted on 16 Apr 2019

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.

LQCC: A Link Quality-based Congestion Control Scheme in Named Data Networks

Hakima Khelifi*, Senlin Luo*, Boubakr Nour[‡], and Hassine Moun gla^{§¶}

*School of Information and Electronics, Beijing Institute of Technology, Beijing, China

[‡]School of Computer Science, Beijing Institute of Technology, Beijing, China

[§]LIPADE, Paris Descartes University and Sorbonne Paris Cite University, France

[¶]CNRS, UMR 5157, Mines Telecom Institute, Telecom SudParis, CEA Nano-Innov Saclay, France

Emails: {hakima, luosenlin, n.boubakr}@bit.edu.cn, hassine.moun gla@parisdescartes.fr

Abstract—Information-Centric Networking (ICN) is a new communication paradigm that replaces the host addresses by the name of content; Named Data Networking (NDN) is a promising ICN architecture that has attracted research attention in recent years. NDN is a receiver-driven architecture implements pull-based communication in the form of one-interest-one-data. This model poses different challenges, especially from the transport layer perspective. In contrast to IP-based networks where the congestion is handled in an end-to-end manner, NDN cannot apply the same concept, while most of the existing solutions are based on hop-by-hop connection. In this paper, we present a new congestion control mechanism for NDN based on link quality estimation. We focus our efforts to provide fast data transmission, decrease packet dropping rate, and maximize the link utilization. The simulation results show that our solution outperforms the NDN schemes in terms of throughput and drop packets.

I. INTRODUCTION

Recently, the research community is exploring new architectures for the future Internet. Information-Centric Networking (ICN) [1], which is a content-oriented paradigm, has been proposed as a new model to replace the current host-centric paradigm. Several projects have been implemented under the umbrella of ICN, all of them focus on the use of the content name instead of the host address. Named Data Networking (NDN) [2] is recent and active ICN project that rapidly gained a considerable research interest.

NDN uses hierarchical names to identify, request, and deliver the content in the network [3]. It also uses three data structures namely Cache Store (CS), Pending Interest Table (PIT), and Forwarding Interest Table (FIB), to provide in-network caching feature [4], ensure a simple receiver-driven data exchange, and provide a name-based routing, respectively. In NDN-based networks, a consumer initializes the communications by sending an interest packet, to request content by specifying its name. NDN nodes forward the interest packets hop-by-hop until reaching the original content producer, or any replicate node that has the content. The forwarding process is based on name-based routing rules and consulting both PIT and FIB tables; the former is used to keep tracing the path in order to use it in the data delivery phase, while the latter is used to select the next hop and forward the interest upstream. When content is available, the said node replies with a data packet that carries the content and the same requested name, and

delivers back to the consumer(s) using the reverse path created by the interest. Moreover, NDN architecture uses different mechanisms and trust models to ensure data-integrity and authentication [5]. However, and as illustrated in Figure 1, the NDN stack does not have a dedicated Transport layer. Hence, all transport mechanisms such as a reliable data delivery, link utilization, and congestion control must be treated by the Strategies and the Application layer.

NDN has been applied in different domains including Internet of Things [6], Vehicular Ad hoc Networks [7], and Wireless Sensor Network [8]. On the other hand, Link Quality Estimation (LQE) [9] is used to estimate and select the stable and reliable link over a set of candidate links that can be used to forward interest. LQE integrates different information and metrics to select the stable link. Indeed, choosing the stable link may increase end-to-end data delivery, avoid the congestion, and improve the overall network performance.

In LQE, there are two types of estimators: (a) hardware-based estimators that are based on using signal strength of a received packet that is obtained directly from radio transceiver, and (b) software-based estimators that are collected via the persistent tracking of messages losses. The main advantage of this type of estimators is that they do not depend on specific hardware and correlate directly with the application layer [10], [11]. The link quality estimation process consists of three steps:

- 1) *Link Monitoring*: three kinds of link monitoring can be used: active link monitoring, passive link monitoring, and hybrid link monitoring.
- 2) *Link Measurements*: are performed by retrieving useful information from received packets and acknowledgments, or from the sent packets such as sequence numbers, timestamp, Received Signal Strength Indication (RSSI), Link Quality Indicator (LQI), and packet retransmission count.
- 3) *Metric Evaluation*: metrics are evaluated based on the link measurements to produce an estimation of the link quality.

In this paper, we focus on the NDN transport layer issues, more specifically the congestion control and reliable communication. Hence, we apply the Link Quality Estimator to provide

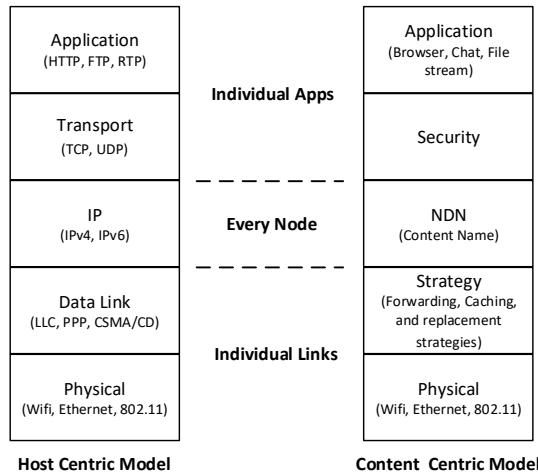


Fig. 1: Host Centric Model VS Content Centric Model

a reliable data delivery along the best path, and maximize the link utilization. We define new dynamic metrics that have a direct impact on the best alternative link selection, rather than using only one or two static metrics such as bandwidth or Round-Trip Time (RTT). Thus, we propose *Link Quality-based Congestion Control (LQCC)* scheme that can be coupled with the smart NDN forwarding plane to persistently monitor the connected links, detect the congestion in the earlier phase, avoid the congestion based on the link metrics, and provide hop-by-hop fragmentation. The simulation results show that LQCC scheme outperforms the existing NDN forwarding schemes in terms of throughput, and less packet drops.

The remainder of the paper is organized as follows. Section II reviews the existing NDN congestion control solutions. Section III describes the used network model and the defined LQE metrics. Section IV details the proposed scheme and its working process. In Section V, we evaluate the proposed scheme and discuss the obtained results. Finally, Section VI concludes the paper.

II. RELATED WORK

NDN uses Interest/Data exchange model to retrieve content from multiple sources through multiple paths. Hence, network congestion can appear during the data transmission (on both content discovery or delivery phases). The traditional congestion control mechanisms that have been designed for end-to-end communication model, cannot be adapted in NDN [12]. Various research efforts have been shown in the literature addressing this issue, categorized as follows:

TCP-like Congestion Control Schemes: Carofiglio *et al.* [13] propose a delay-based congestion control mechanism. Authors define an interest window that is decreased based on the past RTT measurements. In addition, route-labeling and remote adaptive active queue management have been proposed to overcome the problem of mixing RTT from different packet sources. However, Nguyen *et al.* [14] argue that route-labels have scalability and practicality issues. The number of potential routes increases exponentially when the

number of nodes increases. Thus, authors propose a congestion control scheme by estimating the available bandwidth, without using route-labeling or any cooperation mechanism.

Implicit-Feedback Congestion Control Solutions: Schneider *et al.* [15] propose a practical congestion control scheme. This scheme measures packet queuing time to detect congestion, then marks certain packets to signal them to the consumer. Hence, the downstream routers can forward packets to alternative paths, while consumers can reduce their interest sending rates. In a similar manner, Nour *et al.* [16] propose a control protocol for NDN, they also introduce a Bottleneck Notification message to reduce the interest sending rate and avoid the congestion.

Hop-by-Hop Congestion Control Mechanisms: Other researches show their efforts in Hop-by-Hop Interest Sharing mechanism. This mechanism uses the interest sending rate in the opposite direction to control the number of data returning on its links. Carofiglio *et al.* [17] propose a joint Hop-by-Hop and Receiver-driven Interest Control Protocol. This protocol aims to control user requests using per-flow in each output interfaces at every router. This will improve the network scalability and overall performance. Similarly, Wang *et al.* [18] propose a hop-by-hop interest shaping algorithm for NDN. This solution is based on controlling the interest packet rate by taking into account the interdependence between interests and contents in opposite directions. Hence, it tends to maximize the use of available bandwidth, and minimize the loss of content packets.

Shaping-based Congestion Control Solutions: Lei *et al.* [19] propose a shaping-based algorithm that combines Rate Control Protocol (RCP), rate-based, hop-by-hop, and explicit congestion control algorithm. The objective of the proposed algorithm is to achieve a high link utilization, and increase overall network throughput. Zhou *et al.* [20] propose an Explicit Congestion Notification based on proactive interest sending rate control mechanism. All receivers send their RTT measurements to the routers to calculate the available bandwidth, and adopt a smart forwarding using Software Defined Network-style controller to select the best forwarding path. Wang *et al.* [21] design a passive and proactive mechanism that aims to distribute the computation among different routers, and hence avoid the congestion in the network. The scheme takes advantage of the available resources in a neighborhood to balance the service load, and reduce service latency and request drop rate. Miyoshi *et al.* [22] propose a Multi-Source Congestion Control scheme that is based on the combination of consumer window control and router-assisted branch probability control.

Although the aforementioned solutions, most of them take only one (static) metric into consideration, or try to bring IP-based congestion mechanisms to NDN network. In fact, none of them consider the link quality with multiple dynamic metrics. To authors' best knowledge, there are no existing efforts on NDN and LQE. This paper proposes a congestion control scheme based on LQE to reduce the packet dropping rate and maximize the link utilization.

TABLE I: Notations used throughout this paper.

Notation	Definition
$l_{i,j}$	Link between nodes i and j
\mathcal{B}	Bandwidth
\mathcal{U}	Link utilization
\mathcal{LQ}	Link quality
C_i	Cache capacity for node i
C_o	Occupied cache size
C_f	Free cache size
$\mathcal{I}(t)$	Interest sending rate
$\mathcal{D}(t)$	Data delivery rate
\mathcal{V}	Mobility variable
DRR	Data Reception Rate
ISR	Data Success Rate
IRR	Interest Reception Rate
\mathcal{ACV}	Avoid Congestion Value
\mathcal{CV}	Congestion Value

III. SYSTEM DESCRIPTION

The network congestion control is one of the most important aspects of networking [23]. In this section, we describe in detail our proposed congestion control mechanism. In fact, we apply Link Quality Estimator on top of NDN network, and define four methods that can deal with different cases. Combined all methods, we achieve a high link utilization, fast data transmission, as well as a low packet dropping rate.

A. Network Model

Consider a connected multi-hop network modeled by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ comprising a set nodes \mathcal{N} and links \mathcal{L} . A connection link $l \in \mathcal{L}$ between two nodes $(i, j) \in \mathcal{N}$ is denoted by $l_{i,j}$, we denote \mathcal{B} and \mathcal{U} , the bandwidth and link utilization of link $l_{i,j}$, respectively. Also we denote \mathcal{LQ} as the link quality value of link $l_{i,j}$. In addition, we define a new data structure namely Link Quality Table (\mathcal{LQT}) where nodes store all \mathcal{LQ} values. Table I provides a summary about the notations used in this paper.

B. LQE Metrics

For a give node $n \in \mathcal{N}$, $\mathcal{I}(t)$ denotes the interest sending rate at time t over the link l , where $\mathcal{D}(t)$ denotes the data delivery rate at time t . Let C_o be the size of data occupied on the local cache, and C_f the free cache size. $\mathcal{V} = \{0, 1\}$ is a boolean variable indicating if the node will move to another network or not.

According to the status of NDN network, we define three dynamic LQE metrics:

1. *Data Reception Rate (DRR)*: The Data Reception Rate (DRR) metric is calculated at the consumer level, and measures the average of successfully received data. DRR is defined as follows:

$$DRR = \frac{\text{Number of received data packets}}{\text{Number of transmitted interest packets}}$$

2. *Data Success Rate (DSR)*: The Data Success Rate (DSR) metric is calculated at the intermediate node level, and mea-

Algorithm 1: LQCC Algorithm

```

1 while (True) do
2   Populate  $\mathcal{LQT}$  Table;
3    $\triangleright$  Persistent Link Monitoring
4   if ( $\mathcal{LQ} \geq \mathcal{ACV}$ ) then
5     if ( $\mathcal{LQ} \geq \mathcal{CV}$ ) then
6       if ( $Path \geq 1$ ) then
7          $\triangleright$  Hop-by-Hop Fragmentation
8         Enable data caching;
9         Enable hop-by-hop fragmentation;
10      else
11         $\triangleright$  Congestion Detection
12        Select alternative path;
13      end
14    else
15       $\triangleright$  Congestion Avoidance
16      Send notification;
17    end
18  end
19 end

```

sures the average of successfully received data. DSR is defined as follows:

$$DSR = \frac{\text{Number of received data packets}}{\text{Number of forwarded interest packets}}$$

3. *Interest Reception Rate (IRR)*: The Interest Reception Rate (IRR) metric is calculated at the original producer level, and measures the average of successfully received interest. IRR is defined as follows:

$$IRR = \frac{\text{Number of received interest packets}}{\text{Number of sent data packets}}$$

IV. LQCC: LINK QUALITY-BASED CONGESTION CONTROL SCHEME

In order to control the congestion in NDN network, we define two thresholds: *Avoid Congestion Value* (\mathcal{ACV}), and *Congestion Value* (\mathcal{CV}). When \mathcal{LQ} reaches the \mathcal{ACV} value, the node sends a notification packet to all connected consumers in order to reduce their interest sending rate. This mechanism is used to avoid the congestion before it happens. Otherwise, if the \mathcal{LQ} reaches \mathcal{CV} value, here we distinguish two cases based on the number of available paths (faces): (i) one-path scenario: we enable data caching and hop-by-hop fragmentation, (ii) multi-path scenario: an alternative path from \mathcal{LQT} table is selected based on the highest value. A pseudo-code of LQCC scheme is presented in Algorithm 1, and detailed as follows:

Step 1 - Persistent Link Monitoring: By using the aforementioned link quality metrics, we monitor each link $l_{i,x} \in \mathcal{L}$, and store their estimated \mathcal{LQ} in \mathcal{LQT} . \mathcal{LQT} table contains the name of the interface and link quality value. We sort the table based on the link quality values to achieve a fast selection of an alternative path in case of congestion. To

calculate \mathcal{LQ} value, we define our function as the sum of bandwidth, link utilization, mobility, and cache capacity, as well as the metrics (\mathcal{DRR} , \mathcal{DSR} , and \mathcal{IRR}). The result is an estimation of the link quality. If \mathcal{LQ} reaches $\mathcal{mathcal{ACV}}$ the *Step 2* is applied. Otherwise, if \mathcal{LQ} reaches the \mathcal{CV} , then the *Step 3* or *Step 4* is applied according to the context. The node updates this value for each incoming interest or data packet. The calculation of \mathcal{LQ} is based on the node type, and is presented in the following cases:

(1) *Consumer node*: from the consumer side, the link quality \mathcal{LQ} is calculated by using the proposed metric \mathcal{DRR} , and \mathcal{U} , \mathcal{C} , \mathcal{V} and $\mathcal{D}(t)$, and is defined in the following equation:

$$\mathcal{LQ} = \alpha \mathcal{DRR} + \beta \mathcal{D}(t) + \gamma \mathcal{U} + \delta \mathcal{C} + \theta \mathcal{V}$$

(2) *Intermediate node*: from the intermediate node side, the \mathcal{LQ} is calculated by using the defined metric \mathcal{DSR} , and \mathcal{U} , \mathcal{C} , \mathcal{V} , and $\mathcal{D}(t)$, and is defined in the following equation:

$$\mathcal{LQ} = \alpha \mathcal{DSR} + \beta \mathcal{D}(t) + \gamma \mathcal{U} + \delta \mathcal{C} + \theta \mathcal{V}$$

(2) *Data producer node*: from the data producer node, the link quality \mathcal{LQ} is calculated by using the defined metric \mathcal{IRR} , and \mathcal{U} , \mathcal{C} , \mathcal{V} , and $\mathcal{I}(t)$, and is defined in the following equation:

$$\mathcal{LQ} = \alpha \mathcal{IRR} + \beta \mathcal{I}(t) + \gamma \mathcal{U} + \delta \mathcal{C} + \theta \mathcal{V}$$

Where $\alpha, \beta, \gamma, \delta$, and θ are weigh parameters that may be used to balance the preferable metrics, with the constraint that $(\alpha + \beta + \gamma + \delta + \theta) = 1$

Step 2 - Congestion Avoidance: During the data retrieval, when the link quality $\mathcal{LQ} \geq \mathcal{ACV}$, the node n sends a notification packet to all directly connected nodes in the said interface informing them to reduce their interest sending rate. This notification packet format is inherited from the regular interest packet, and contains the coefficient of the delay α . If the congestion is still happing, the node re-sends another notification packets, with another bigger coefficient $\alpha' < \alpha$. It is important to highlight that notification packets are not rout-able, they are used only by the neighbor nodes.

Step 3 - Congestion Detection: In case of the congestion is still happening after *Step 2*, and the link quality $\mathcal{LQ} \geq \mathcal{mathcal{ACV}}$. We propose two different mechanisms based on the type of communication:

- 1) *Single-Path Communication*: if a node n has only one path/interface (Figure 2(a)), we apply a hop-by-hop fragmentation mechanism (*Step 4*). We take the benefits of content naming and caching to enable this feature. It is worth noting that the content integrity and security are preserved by using the content-based security concept adopted in NDN design.
- 2) *Multi-path Communication*: if a node n has more than one path/interface p (Figure 2(b)), we select the best alternative link (interface) by consulting the \mathcal{LQT} table.

Step 4 - Hop-by-Hop Fragmentation: The local node fragments the packet as required and according to the MTU

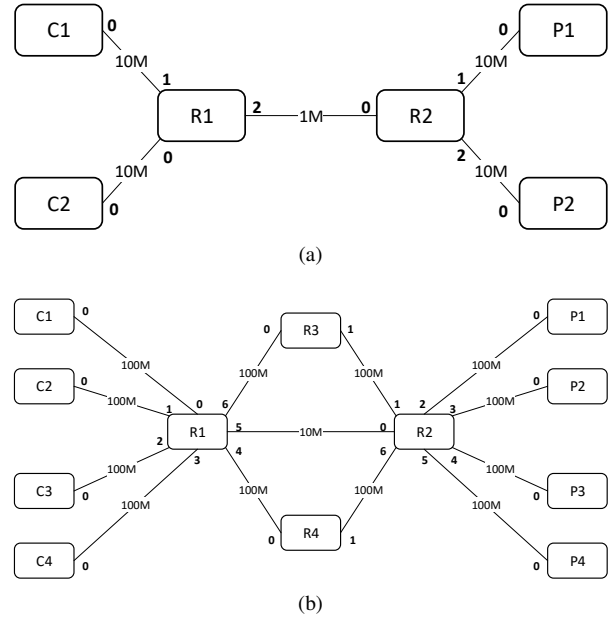


Fig. 2: Bottleneck Network Topology. (a) single-path, (b) multi-path.

value (Maximum Transmission Unit) if needed, while the next hop assembles the packet immediately. Also, each segment is signed separately by the node who fragment the content. In case of losing a fragment, the node sends a simple interest packet that carries the content name as well as the segment number for the requested segment. This interest is satisfied by the node who applied the fragmentation.

Naming Convention: during the fragmentation phase, we use the same NDN hierarchical naming scheme. However, we extend it by adding the number of the segment at the end of naming ($\text{naming} := \text{naming}/\text{SegNumber}$). In case of transmitting the last segment, we add the flat *FIN* to indicate that all segments have been sent ($\text{naming} := \text{naming}/\text{SegNumber}/\text{FIN}$) and the next node can assemble them.

V. PERFORMANCE EVALUATION

We have implemented the LQCC scheme on top of ndnSIM [24], which is an NS3-based simulator for NDN. We have used the bottleneck topology that illustrated in Figure 2, and benchmarked the LQCC against NDN forwarding strategies (i.e., Best-route, NCC, Multicast, and Broadcast).

TABLE II: Simulation Parameters

Parameter	Settings
Scenario	Bottleneck
No. of Nodes	[1, 12]
No. of Routers	[1, 4]
Packet size	1024 Kbyte
Capacity	[10, 100] Mbps
Sim. Time	30 sec
CS Buffer Size	1000 pkts
Interest Rate	500 Interest/sec

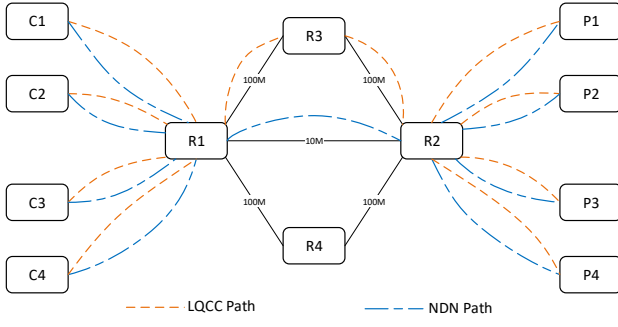


Fig. 3: Forwarding Path

All strategies have been compared using the same parameters as shown in Table II, and performed on Intel Core 5 Duo CPU at 2.4 GHz, and 8 GB DDR3 SDRAM. shows the simulation parameters.

Figure 3 illustrates the path selection for LQCC and best-route scheme. Here, we emphasize on best-route only as the other schemes (i.e., NCC, Multicast, and Broadcast) produce the same results in the bottleneck topology.

A. Simulation Metrics

We have made a comparison in terms of throughput, the number of drop packets, and end-to-end delay:

- *Throughput*: has been measured as the size of well-received data in compared to the requested interests.
- *Number of Drop Packets*: is measured as the total number of dropped packets in the network including data and interest packets.
- *End-to-End Delay*: is used to evaluate the network performance and calculated as the time of sending interest and receiving its corresponding data packet.

B. Simulation Results

In the following, we discuss the obtained numerical results:

Figure 4 shows the throughput measurement results. In contrast to NDN that uses only one static metric (e.g., a small

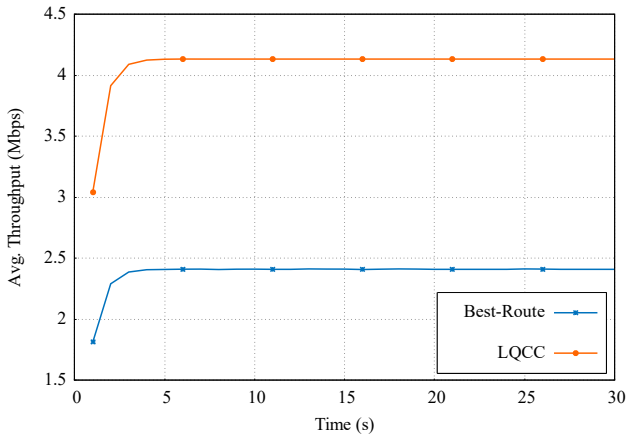


Fig. 4: Throughput Measurement.

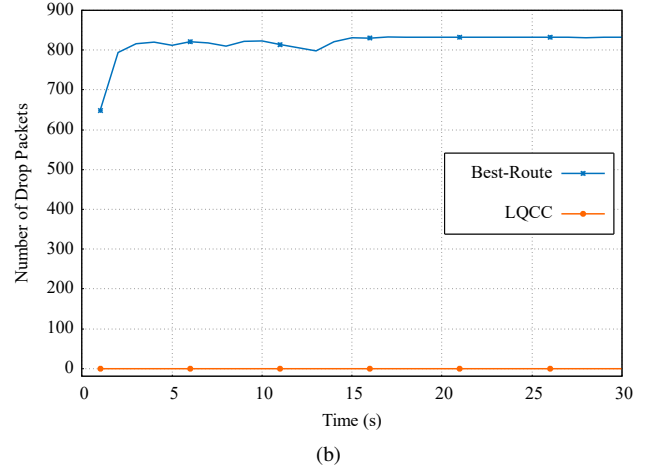
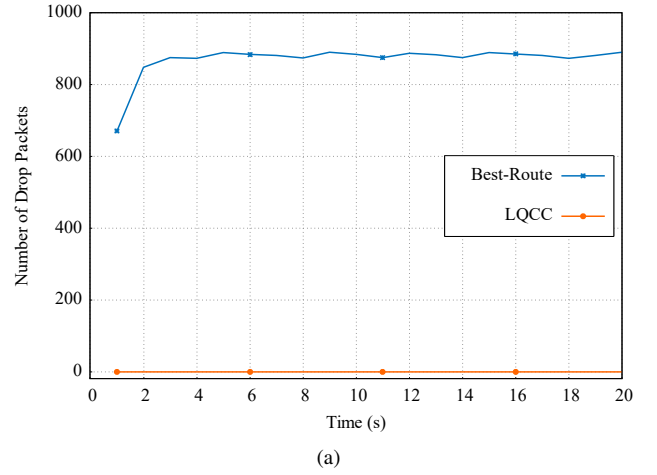


Fig. 5: Drop Packets Measurements: (a) single-path, (b) multi-path.

number of hops regardless of the link status), we observe that LQCC scheme achieves a better throughput as it selects the stable link based on different dynamic metrics to avoid the congestion.

Figure 5 shows the drop packets measurements in a bottleneck link. Results are plotted for two scenarios: single-path (Figure 5(a)), and multi-path (Figure 5(b)). As the path selection in NDN is limited to the number of hops, more packets are dropped in both scenarios. However, LQCC integrates different dynamic metrics to select the best path (according to the current network status) in case of multi-path and hop-by-hop fragmentation in the case of single-path, hence it did not drop any packet during the communication.

The end-to-end delay measurements are represented in Figure 6. We observe that the delay in NDN is smaller than LQCC scheme. We argue this by the fact that NDN uses the short path (in terms of hops) regardless of the path reliability or stability during the data forwarding, while LQCC uses the optimal and stable path that can forward the data with higher throughput, fewer drop packets, and a reasonable delay.

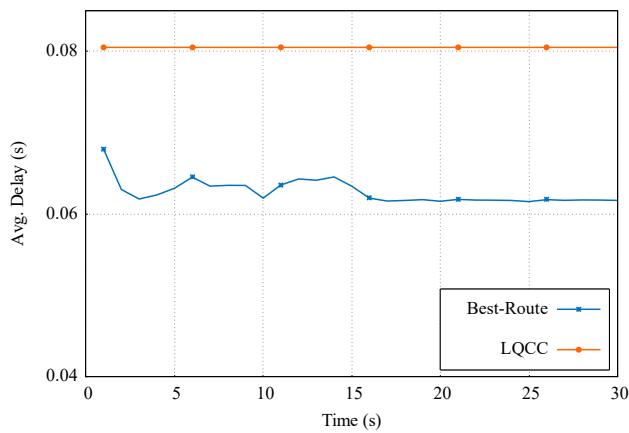


Fig. 6: Network Delay Measurement.

VI. CONCLUSION

In this paper, we propose a Link Quality-based Congestion Control (LQCC) scheme for Named Data Networks in order to avoid congestion in the network and achieve an efficient data delivery. We define different dynamic metrics related to NDN communication. These metrics, along with other metrics such as link utilization, cache store, and the mobility impact, are used to select the optimal and stable path that can be used to forward the interest and deliver that data without facing any congestion issues. Simulation results show that LQCC achieves an optimal throughput and fewer drop packets. Moreover, the proposed scheme can be integrated with different wired and wireless environment including Internet of Things, and vehicular applications. As future work, real-world implementation of LQCC is planned with a large-scale network.

ACKNOWLEDGEMENTS

The work of S. Luo was supported by the National 242 Project under Grant No. 2017A149. Dr. Luo is corresponding author.

REFERENCES

- [1] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.
- [2] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.
- [3] B. Nour, K. Sharif, F. Li, H. Moun gla, and Y. Liu, "M2HAV: A Standardized ICN Naming Scheme for Wireless Devices in Internet of Things," in *International Conference Wireless Algorithms, Systems, and Applications (WASA)*. Springer, 2017, pp. 289–301.
- [4] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moun gla, and F. Naït-Abdesselam, "An Optimized Proactive Caching Scheme based on Mobility Prediction for Vehicular Networks," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, Dec 2018, pp. 1–6.
- [5] H. Khelifi, S. Luo, B. Nour, and S. C. Shah, "Security and Privacy Issues in Vehicular Named Data Networks: An Overview," *Mobile Information Systems*, vol. 2018, p. 11, 2018.
- [6] B. Nour, K. Sharif, F. Li, and H. Moun gla, "A Distributed ICN-based IoT Network Architecture: An Ambient Assisted Living Application Case Study," in *IEEE Global Communications Conference (GLOBECOM)*, 2017, pp. 1–6.
- [7] H. Khelifi, S. Luo, B. Nour, H. Moun gla, Y. Faheem, R. Hussain, and A. Ksentini, "Named Data Networking in Vehicular Ad hoc Networks: State-of-the-Art and Challenges," *IEEE Communications Surveys and Tutorials*, 2019.
- [8] M. Amadeo, C. Campolo, A. Molinaro, and G. Ruggeri, "Content-centric wireless networking: A survey," *Computer Networks*, vol. 72, pp. 1–13, 2014.
- [9] H.-J. Audéoud and M. Heusse, "Quick and Efficient Link Quality Estimation in Wireless Sensors Networks," in *IEEE Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2018, pp. 87–90.
- [10] H. Moun gla, K. Haddadi, and S. Boudjit, "Distributed interference management in medical wireless sensor networks," in *IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2016, pp. 151–155.
- [11] M. Ali, H. Moun gla, M. Younis, and A. Mehaoua, "Distributed scheme for interference mitigation of WBANs using predictable channel hopping," in *International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE, 2016, pp. 1–6.
- [12] Q. Chen, R. Xie, F. R. Yu, J. Liu, T. Huang, and Y. Liu, "Transport Control Strategies in Named Data Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2052–2083, 2016.
- [13] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang, "Optimal Multipath Congestion Control and Request Forwarding in Information-Centric Networks," *Computer Networks*, vol. 110, pp. 104–117, 2016.
- [14] D. Nguyen, M. Fukushima, K. Sugiyama, and A. Tagami, "Efficient multipath forwarding and congestion control without route-labeling in CCN," in *IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 1533–1538.
- [15] K. Schneider, C. Yi, B. Zhang, and L. Zhang, "A Practical Congestion Control Scheme for Named Data Networking," in *ACM Conference on Information-Centric Networking (ICN)*, 2016, pp. 21–30.
- [16] B. Nour, K. Sharif, F. Li, H. Khelifi, and H. Moun gla, "NNCP: A Named Data Network Control Protocol for IoT Applications," in *IEEE Conference on Standards for Communications and Networking (CSCN)*, 2018, pp. 1–6.
- [17] G. Carofiglio, M. Gallo, and L. Muscariello, "Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, p. 491, 2012.
- [18] 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 SIGCOMM workshop on Information-Centric Networking*, vol. 1, 2013, p. 55.
- [19] K. Lei, C. Hou, L. Li, and K. Xu, "A RCP-Based Congestion Control Protocol in Named Data Networking," in *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2015, pp. 538–541.
- [20] J. Zhou, Q. Wu, Z. Li, M. A. Kaafar, and G. Xie, "A proactive transport mechanism with explicit congestion notification for NDN," in *IEEE International Conference on Communications (ICC)*, 2015, pp. 5242–5247.
- [21] L. Wang, M. Almeida, J. Blackburn, and J. Crowcroft, "C3PO: Computation Congestion Control (ProActive)," in *ACM Conference on Information-Centric Networking (ICN)*, 2016, pp. 231–236.
- [22] J. Miyoshit, S. Kawachit, M. Bandait, and M. Yamamoto, "Multi-Source Congestion Control for Content Centric Networks," in *ACM Conference on Information-Centric Networking (ICN)*, 2016, pp. 205–206.
- [23] Y. Ren, J. Li, S. Shi, L. Li, G. Wang, and B. Zhang, "Congestion control in named data networking - A survey," *Computer Communications*, vol. 86, pp. 1–11, 2016.
- [24] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," *Tech. Rep. NDN-0005*, pp. 1–7, 2012.