

# On the Analysis of Adaptive-Rate Applications in Data-Centric Wireless Ad-Hoc Networks

Md Ashiqur Rahman  
The University of Arizona  
marahman@cs.arizona.edu

Beichuan Zhang  
The University of Arizona  
bzhang@cs.arizona.edu

**Abstract**—Adapting applications’ data rates in multi-hop wireless ad-hoc networks is inherently challenging. Packet collision, channel contention, and queue buildup contribute to packet loss but are difficult to manage in conventional TCP/IP architecture. This work explores a data-centric approach based on Name Data Networking (NDN) architecture, which is considered more suitable for wireless ad-hoc networks. We show that the default NDN transport offers better performance in linear topologies but struggles in more extensive networks due to high collision and contention caused by excessive Interests from out-of-order data retrieval and redundant data transmission from improper Interest lifetime setting as well as in-network caching. To fix these, we use round-trip hop count to limit Interest rate and *Dynamic Interest Lifetime* to minimize the negative effect of improper Interest lifetime. Finally, we analyze the effect of in-network caching on transport performance and which scenarios may benefit or suffer from it.

**Index Terms**—wireless ad-hoc networks, data-centric transport analysis, adaptive-rate applications.

## I. INTRODUCTION

Communication in a wireless environment is complex by nature because of channel access, contention, packet collision, signal interference, to name a few [1]. A multi-hop network makes it more challenging. A mobile ad-hoc network (MANET) exacerbates the packet loss even further as the uncertainty of link breakage between two neighbor nodes increases manifold. Thus, the application-level performance is much lower than in a stable wired network.

A transport-layer protocol ensures end-to-end data delivery and adapts data rate to the available bandwidth in the network, such that applications can fully utilize available bandwidth without causing network congestion. Moreover, sending rate or congestion window also affects wireless contention and throughput [2]. Today’s IP-based transport protocols for adaptive-rate applications based on TCP and its variants assume point-to-point data transportation over a relatively stable path. Mobility, however, often breaks this assumption as end-to-end paths break more often. It results in significant RTT variance, out-of-order data arrival, and higher packet loss. Thus, rate management becomes tricky for transport.

A data-centric communication moves away from the point-to-point abstraction of IP. Here, the network layer can deliver or retrieve data from anywhere in the network, even from multiple data points at the same time. As a result, a data-centric transport can support out-of-order data arrival. On the other hand, the in-order delivery of TCP and its variants have

many limitations in MANETs [3], e.g., head-of-line blocking, which yields low throughput. In theory, a data-centric transport should perform better than traditional TCP/IP in an ad-hoc wireless environment.

One such data-centric architecture is the emerging Named Data Networking or NDN [4]. In NDN, requests (called Interest packets) and replies (called Data packets) carry data *name* instead of IP addresses. The function of the network is to retrieve named data instead of delivering packets to a particular node. Thus data can be supplied from any node, arrive in any order, and via any path. As a result, NDN natively supports out-of-order data retrieval, multicast, multihoming, and in-network caching. It also has a *stateful forwarding plane*, enabling fast detection of network anomalies and recovering from them accordingly. However, due to the inherent challenges of the wireless environment, it is unclear how much NDN’s architectural advantage would translate into actual performance gain and how much optimization or additional engineering would be needed to maximize it even further.

This work analyzes the transport and forwarder behavior in data-centric adaptive-rate applications using NDN in wireless ad-hoc networks. We start with a proof-of-concept using a simple AIMD-based window adaptation. It shows that NDN can outperform TCP in both wired and static wireless linear topology under lossy conditions with a single sender-receiver (IP) or consumer-producer (NDN) pair. However, in a more extensive scenario with multiple consumer-producer pairs, NDN’s out-of-order delivery causes abrupt congestion-window adjustment, degrading its throughput. We identify channel contention as a root cause with almost no congestion at the device queue. We verify and overcome such behavior by applying a congestion window limit (CWL) [5]. NDN offers better throughput with this change than TCP/IP under mobility with caching and multicast utilization.

Next, an NDN consumer application adds a randomly generated NONCE with each Interest irrespective of original or retransmission and helps to detect in-network loops. The consumer application also adds a lifetime to the Interests (usually fixed, e.g., 2s). It helps achieve better in-network multicast by aggregating Interests with the same name and different NONCES in the Pending Interest Table (PIT). However, using a fixed lifetime in an adaptive-rate application can cause PIT aggregation on consumer retransmissions (different NONCES). Consequently, NDN multicast leads to redundant

data over multiple paths to a consumer, causing high collision and contention. We propose a novel Dynamic Interest Lifetime (DIL) mechanism based on the application’s round-trip timeout (RTO) and use a multiplier to delay the timeout event check to negate this effect. It allows the network to clear out stale PIT entries on application retransmission while keeping the aggregation opportunity if multiple consumers ask for the same data within a close interval.

Finally, NDN caching offers better resiliency to packet loss. However, it can also increase redundant data transmissions in the network from disjoint paths. Such transmission overhead inherently leads to increased packet collision and channel contention. Thus we show that caching in some ad-hoc communication scenarios might not be as beneficial.

Through this work, we confirm that a data-centric transport like in NDN improves adaptive-rate application’s performance, especially in small lossy networks. In an extensive wireless network, however, the data-centric mechanisms such as out-of-order delivery, multicast, in-network caching can have undesirable side effects that would reduce throughput. We then present meaningful insights into these critical challenges of developing data-centric adaptive-rate applications in a wireless ad-hoc network and propose solutions to mitigate them for achieving better throughput. Our results call for more research into new mechanisms and designs that can take advantage of the data-centric architecture in a wireless environment to achieve a breakthrough in overall performance.

## II. RELATED WORKS

The IP transport layer supports minimal loss recovery as it is decoupled from the network layer. Thus, [6], [7] show how TCP struggles to perform well in mobile wireless networks from well-known head-of-line blocking and packet loss from pushing data in the network. The idea is to maintain a *full pipe* with in-flight packets even *before* receiving acknowledgments for sent data to achieve a throughput close to the theoretical maximum. However, in a wireless network, with or without mobility, a push-based model causes a high amount of packet loss and retransmissions.

The TCP header limits loss recovery even with SACK [8] and adjusts window size on both ACK/SACK, timeouts and explicit congestion notification. Thus adapting TCP in ad hoc networks requires substantial modifications such as [9] or enabling out-of-order data retrieval [10]. More recent protocols like QUIC [11] achieve per-packet loss detection and out-of-order retrieval in web applications showing the networking trend moving towards data-centric communication.

[12] shows that NDN offers a better network forwarder than IP in MANETs but only considers constant bit-rate (CBR) traffic, excluding transport analysis. Furthermore, NDN’s pull-based model controls the consumer’s Interest sending rate as a Data node only replies on Interest packets. As an Interest packet is much smaller than a Data packet, one can assume that Interest loss from a high sending rate is far less precarious. However, in a multi-hop wireless scenario, it can still contribute to channel contention. [13] proposes rate-based

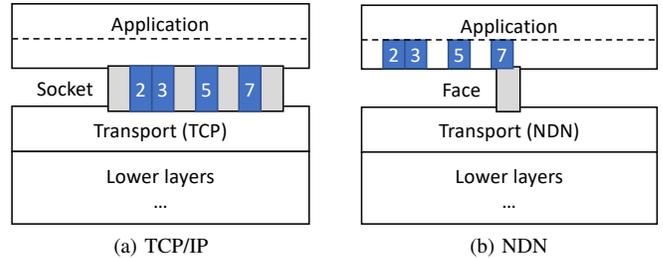


Fig. 1. Packet flow from lower layers to the application in TCP/IP and NDN.

transport for NDN using the inter-data gap (IDG). However, mobility may lead to frequent data node switching, such that a consumer may not get enough data packets for the sampling process. Their data node id-based design also deviates from a complete decentralized data-centric forwarding. [14] proposes a dynamic PIT entry lifetime (DPEL) based on Interest satisfaction rate and hop count. However, the hop count between a consumer and data node will likely fluctuate frequently in a dynamic network. As a result, PIT entries can timeout while data comes back when a data node is further away.

## III. MOTIVATION

### A. Out-of-order Retrieval Offers Better Transport

A fundamental difference between data-centric networking and traditional TCP/IP is how data packets are forwarded from the lower layers to the uppermost application layer. Data-centric communication supports out-of-retrieval by default, while IP requires specialized protocols such as [10] for the same purpose as TCP assumes in-order delivery. Thus to understand the transport benefits of data-centric networking, we show a side-by-side comparison between the NDN transport (or consumer node’s forwarder) and traditional TCP/IP byte-stream management in Fig. 1.

In Fig. 1a, the socket provides an interface between the transport and application layer. It also offers a packet buffer to manage the in-order delivery. The example also shows the head-of-line blocking for packets 1 to 7 because packet 1 was either lost in the network or traveling through a possible longer path (multipath or path change). As a result, even though packets 2 and 3 arrived earlier, they will be buffered until the first arrives. Selective acknowledgment or SACK [8] can help TCP with better loss recovery but is limited to maximum three-hole detection in the byte stream. Such limitation comes from only using the fixed-sized TCP header and not a variable-length design for performance reasons.

NDN’s *face* in Fig. 1b is similar to the socket interface and sits between the transport and the application. However, it does not buffer any data packet from the transport; instead, it relays to the application right away, responsible for packet ordering. Such behavior avoids possible buffer overflow like in TCP socket and provides infinite SACK with per-Interest-Data communication.

Moreover, out-of-order delivery in TCP would mean packet loss in the network. In wired networks, it is highly likely from a congestion drop. In wireless networks, however, it

can also be from packet collision, channel contention or path breakage from mobility. In data-centric networking, the out-of-data retrieval indicates that the network communication is available, irrespective of multipath, congestion, or channel contention. Later, a retransmitted Interest can also retrieve cached data. Therefore out-of-order retrieval enables NDN to advance the congestion window on each data packet, while TCP/IP does not do so by default. Thus data-centric transport promises better performance.

### B. Proof-of-concept for Data-centric Transport

We now show a proof-of-concept on the potentials of using data-centric communication at the transport level without any complex engineering as in IP's host-centric model. It also shows the advantages of NDN's out-of-order data retrieval under lossy conditions. To do so, we consider the well-known TCP-NewReno [15] for TCP/IP. An AIMD algorithm for NDN applications follows the same slow-start and congestion avoidance for congestion window ( $cwnd$ ) maintenance. It starts with an initial congestion window or  $cwnd = 1$  and an initial threshold,  $ssthresh = \infty$ . On each data packet, the slow start or congestion avoidance for  $cwnd$  works as,

$$cwnd+ = \begin{cases} 1.0 & \text{if } cwnd < ssthresh \text{ (slow start)} \\ \frac{1.0}{cwnd}, & \text{otherwise (congestion avoidance).} \end{cases} \quad (1)$$

The NDN AIMD also uses the conservative window adaptation (CWA) [16] on consumer application's timeout followed by the well-known TCP retransmission timeout (RTO) calculation [17]. The difference being for CWA, the Interest packet replaces TCP byte-stream, and the Data packet replaces TCP ACK. Thus, on an application timeout, if the CWA conditions meet, the multiplicative decrease in NDN is as follows,

$$ssthresh = cwnd * \beta \quad (2)$$

$$cwnd = \max(ssthresh, 1). \quad (3)$$

Here, similar to the TCP-NewReno,  $\beta = 0.5$  in Eq.2, while Eq.3 ensures  $cwnd$  never falls below the initial window size. For explicit congestion control, IP uses explicit congestion notification (ECN) while NDN uses congestion marking (CM) directly with returning data packets for outgoing queue build-up as described in [18].

With this NDN AIMD design, we divide the proof-of-concept for transport behavior analysis into two scenarios, 1) linear wired topology and 2) static, linear wireless topology.

1) *Linear wired topology*: We simulate single consumer-producer (NDN) or server-client (TCP/IP) communication over different wired chain lengths using ndnSIM [19]. We also consider per-byte error (or loss) probability at the end nodes' interfaces to emulate loss from wireless collision and contention. Furthermore, we test with and without bottleneck links in the chain to observe congestion signals. By default, all links are 5 Mbps with a 1 ms propagation delay. We set only one link in the middle as 1 Mbps with a 10ms propagation delay when the bottleneck is enabled. Each data packet payload is 1460 bytes. We collect an average of five 300 second runs and show the results in Fig. 2.

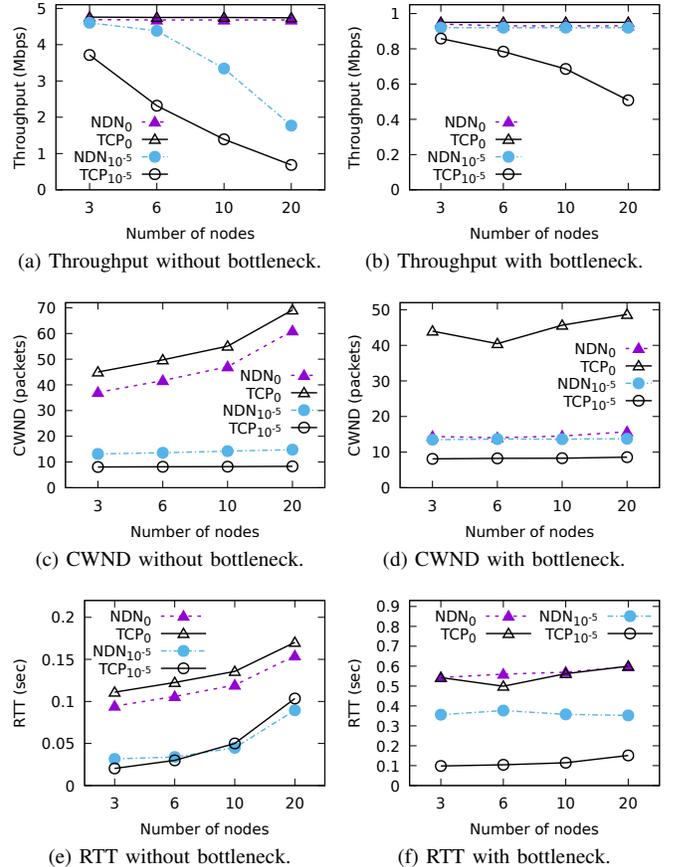


Fig. 2. Effect of packet error and number of nodes on throughput, congestion window (CWND), and round-trip time (RTT) with and without bottleneck link in a linear wired topology. Subscript shows per-byte corruption probability = 0 or  $10^{-5}$  ( $\approx 1.5\%$  per-packet) at the sender and receiver interfaces.

**Without bottleneck:** TCP and NDN have similar throughput without packet loss in Fig. 2a. However, under lossy conditions, their throughput go down with increasing chain length, with TCP performing the worse. TCP tries to keep the socket buffer full and yields high CWND (Fig. 2c) and RTT (2e) when there is no loss. With packet loss, the in-order byte-stream cannot increase CWND due to SACK limitations (max. three-hole filling) in the header and socket buffer capacity. The resulting low RTT verifies our claims. On the other hand, NDN mainly lowers CWND on application timeout as there is minimal congestion marking without a bottleneck. Under packet loss, its infinite-SACK keeps the CWND higher than TCP, helping to achieve higher throughput.

**With bottleneck:** TCP and NDN show similar throughput in Fig. 2b without error. However, with error, only TCP's throughput degrades over chain length. Despite ECN, the SACK limitation keeps the CWND very low (Fig. 2d), which we verify with the low RTT in Fig. 2f. NDN maintains a stable CWND even with packet error using CM and out-of-order data. Thus throughput change is minimal over chain length.

Fig. 3 shows the application throughput,  $cwnd$ , and RTT over the first 100 seconds in a single simulation and further verifies the advantages of NDN and shortcomings of TCP

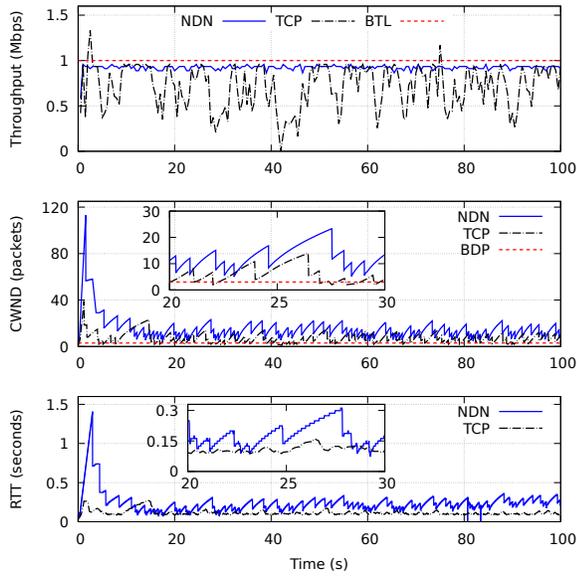


Fig. 3. Transport behavior over time in a single simulation with per-byte loss probability =  $10^{-5}$  ( $\approx 1.5\%$  per-packet) at the sender and receiver interfaces with bottleneck-link = 1 Mbps in 10 nodes linear topology. Red lines in Throughput and CWND show the ideal bottleneck (BTL) and bandwidth-delay product (BDP), respectively.

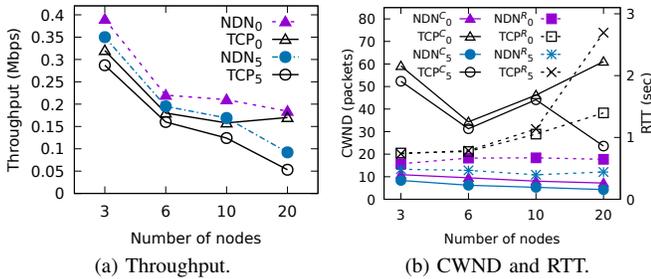


Fig. 4. Effect of packet error and number of nodes on throughput, congestion window (CWND), and round-trip time (RTT) in a static linear wireless topology. Superscript shows  $CWND=C$  or  $RTT=R$ . Subscript shows per-packet loss probability percentage at each node’s wireless interface.

under lossy conditions and bottleneck. We see that TCP shows high fluctuations in throughput because of in-order delivery. As a result, application throughput goes above the link capacity at time = 2s and 75s as it processes more bytes from the receiver socket than the bottleneck capacity. On the other hand, NDN maintains a stable throughput over time with out-of-order retrieval and maintains a higher CWND and RTT, as discussed earlier. NDN’s CWND and RTT spikes at the beginning verify the out-of-order effect with an initial RTO of 2 seconds (TCP reacts comparatively early with *ACK/SACK*).

2) *Static, linear wireless topology*: We use a static linear topology similar to Sec. III-B1 for different chain lengths in multi-hop static wireless networks. However, each node has a single wireless interface covering a maximum of two neighbors. They communicate using IEEE 802.11b over a single channel at 1 Mbps with RTS/CTS disabled and 512 bytes payload. We collect five runs of 300 seconds per chain length. Each node’s interface has a queue of 25 packets with a packet reception error rate of 0% or 5% to emulate added

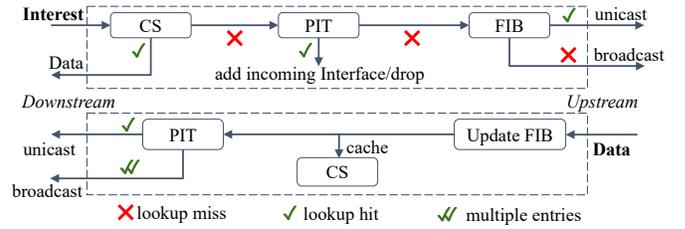


Fig. 5. Forwarding pipeline of the DAF strategy for MANETs.

loss (e.g., signal jamming in military networks). Fig. 4 shows the NDN and IP transport behavior.

We can see that both TCP and NDN’s maximum throughput are lower than their wired scenario and sharply drops with increasing chain length. It is because channel contention starts taking over NIC queue congestion, and the hidden terminal problem is also possible from bi-directional traffic. Our simulation logs also show only a few congestion signals. As a result, NDN controls *cwnd* primarily based on application timeout, while TCP does so with *ACK/SACK* and timeout.

TCP still tries to keep the receiver socket buffer full, which leads to high *cwnd* and RTT at the sender in Fig. 4b with and without additional packet error. However, TCP’s in-order delivery leads to lower throughput than NDN’s out-of-order retrieval in Fig. 4a. A single simulation test also showed CWND and RTT spikes at the beginning (separate plot not shown), similar to Fig. 3, verifying NDN’s out-of-order retrieval. However, they both go below TCP’s as there are no separate buffers in NDN transport. As a result, NDN reacts to timeouts much faster compared to TCP. NDN shows an average of 21.31% and 38.42% more throughput than TCP/IP with 0% and 5% packet error, respectively.

### C. Data-centric Forwarding Improves Network Performance

The data-centric ad-hoc forwarding (DAF) [12] shows that NDN can be a better fit than IP in a MANET environment while keeping the similar characteristics of broadcast-based learning, like the AODV [20] routing. DAF significantly reduces the retrieval latency, network load and thus improves the retrieval success rate. It does so by allowing any node to fall back to discovery for quick mobility and loss reaction. Each node maintains a weighted moving average of the RTT for each producer prefix and next-hop pair. Thus any node can expire its Forwarding Information Base (FIB) entry using the same TCP-RTO principle [17].

The forwarding pipeline of DAF is in Fig. 5. A node receiving an Interest first checks the Content Store (CS) and only returns data on a cache hit; otherwise, it forwards to the Pending Interest Table (PIT). The PIT aggregates and suppresses an Interest if it finds the same name with a different *NONCE* or drops if the same *NONCE* (loop); otherwise, forwards the unique Interest to the FIB. A node then unicasts the Interest to a valid FIB entry and broadcasts otherwise.

A node receiving a Data packet on the downstream first updates its FIB with the sender information and then opportunistically caches the data to the CS. It benefits the network under mobility with uncertain link breakage. A node then

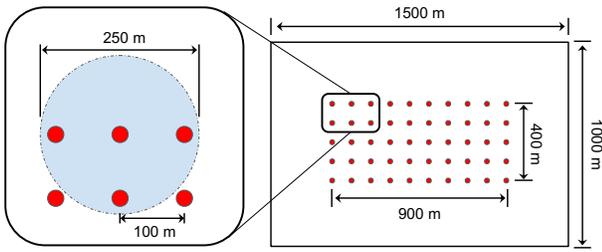


Fig. 6. Initial system setup with stationary nodes. Right side shows the grid topology when nodes are stationary, and left side shows inter-node distance and transmission range (dotted large circle).

performs a PIT lookup and unicasts data if there is one downstream, broadcasts on two or more entries to reduce transmission overhead, and drops if no entry is found.

Moreover, DAF avoids negative acknowledgments or NACKs to reduce network traffic and allows a node to broadcast data for aggregated Interests in the PIT to avoid multiple data transmissions. Later in this paper, we use the DAF and AODV for NDN and IP, respectively, in wireless ad-hoc simulations. It ensures that the network level forwarding behavior is similar to provide a fair comparison at the transport level.

#### IV. SYSTEM MODEL

We use the random geometric graph [21] to analyze NDN's adaptive-rate applications in wireless networks. We randomly place 50 nodes in a 10x5 grid setup with 100 meters of separation in X and Y coordinates and 250 meters (diameters) transmission range for each node. Therefore, one node can communicate to a maximum of four nodes at any given time. It allows us to analyze the transport and network behavior when nodes are stationary, with a valid path between a consumer and producer. The simulation area is 1500m x 1000m, and the system model topology is in Fig. 6.

With mobility (speed  $> 0$  m/s), nodes start to move around at time  $t > 0$  using a random-walk 2D mobility model with 5 seconds duration for each randomly selected direction and no pause before selecting the next one. Speed ranges between 0 and 8 m/s, and in each simulation, all nodes maintain a constant speed. Each node has similar NIC properties as in Sec. III-B2. The network includes ten consumers or clients, 10 or 2 producers or servers, while the rest act as packet forwarders. The purpose of such mobility is to solely focus on the transport behavior in a generalized MANET scenario, and unlike in more specialized cases such as vehicular networks where mobility direction may play a vital role. We expect future research to tackle such design considerations.

#### V. DATA-CENTRIC ADAPTIVE-RATE APPLICATIONS

With NDN's advantages at the application, transport, and network layer, we now analyze the behavior of data-centric adaptive-rate applications in a more extensive wireless network with multiple consumers, producers, and mobility.

##### A. Channel Contention from Out-of-order Data Retrieval

In Sec. III-B2, we saw that wireless communication induces channel contention, overshadowing queue congestion. Thus

NDN adjusts  $cwnd$  mostly on Interest timeouts. Consequently, the  $cwnd$  increases on every data received, ignoring potential loss in the network, while channel contention keeps getting higher, causing further packet loss. This effect exaggerates in a multi-consumer-producer scenario, leading to no queue congestion altogether. We consider this phenomenon as  **$cwnd$  overestimation**, which significantly degrades NDN throughput, as we will see later in Sec VI-B.

To mitigate this overestimation, we use a Congestion Window Limit (CWL) approach from [5]. It uses the round-trip hop-count (RTHC) to impose a tighter upper bound on the  $cwnd$  and minimizes channel contention. It also considers the possibility of an asymmetric path for data and ACK. In NDN, however, the Interest-Data path is symmetric, i.e., data strictly follows PIT entries downstream. Even with potential disjoint paths, the symmetric behavior between one Interest, one Data holds. Thus after a consumer receives a data packet and increases  $cwnd$  using Eq. 1, it can use the packet's  $hop\_count (= \frac{RTHC}{2})$  from the data node to calculate the CWL ( $cwl$ ) as follows,

```

if (hop_count <= 2) cwl = 2;
else if (hop_count <= 4) cwl = 1;
else if (hop_count <= 6) cwl = 2;
else if (hop_count <= 10) cwl = 3;
else if (hop_count <= 13) cwl = 4;
else if (hop_count <= 15) cwl = 5;
...

```

Finally, we update the  $cwnd$  with,

$$cwnd = \min(cwnd, cwl) \quad (4)$$

Eq. 4 preserves congestion avoidance's linear growth and employs the tight upper bound to minimize channel contention. Simulation results in the next section also show significant throughput improvement using CWL in data-centric transport.

##### B. Effect of Interest Lifetime

Interest lifetime dictates how early or late an Interest will be evicted from the PIT. A small value means less aggregation, lower multicast opportunity, and vice-versa. Moreover, if a node receives a duplicate Interest with different NONCE before the existing PIT entry expires, it updates the PIT entry lifetime with the new Interest, increments the PIT count, and makes either one of two choices, (a) suppress, if it arrives within a suppression interval, or (b) forward. With RTO-based retransmission at the application level, case (b) will likely occur with a large and fixed Interest lifetime.

An example in Fig. 7 shows  $c$  as a consumer,  $d$  as a data node, and the rest as forwarders. The dotted lines represent wireless reachability. Assume that  $c$  broadcasts an Interest  $/a/img.png$  with 2s lifetime in Fig. 7a. Both  $x$  and  $y$  receive it, but  $x$  broadcasts first from the layer-1 random timer during transmission, and  $y$  detects a duplicate. Even though  $x$  and  $y$  both forward the Interest, the smaller packet size has a lower collision chance at  $z$ . Thus  $z$  receives the Interest from  $x$  and creates a PIT entry with  $x$  as downstream (loop detected

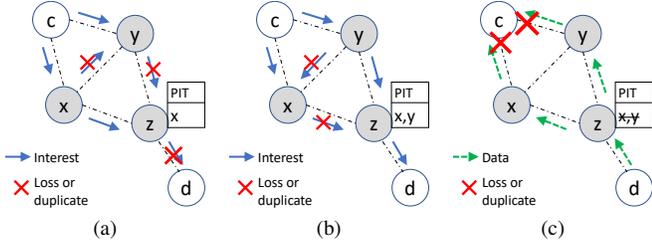


Fig. 7. Example of data redundancy from multicast/broadcast on Interest aggregation upon application retransmission.

for  $y$ , dropped). However, the Interest fails to reach  $d$  from collision/contention from another node nearby. At time=1s, the Interest times-out at  $c$ 's application and issues a retransmission (Fig. 7b). This time  $z$  receives the Interest from  $y$ . Thus  $z$  now has two downstreams,  $x$  and  $y$  for  $/a/img.png$ , forwards Interest to  $d$  as the PIT considers case (b). Now,  $d$  receives the Interest and sends the data to  $z$  (Fig. 7c). Node  $z$  will now either make two unicast transmissions as multicast or a single broadcast (according to the DAF strategy). Thus both  $x$  and  $y$  receive the data. As data packets are usually much larger than Interests, even with wireless layer-1 randomness, the chance of collision/contention at  $c$  is highly likely by data transmission from both  $x$  and  $y$ .

Thus, an extensive network can lead to a very high collision and contention. We use a Dynamic Interest Lifetime (**DIL**) technique to minimize the *retransmitted Interest aggregation* effect. We use the most recent RTO (same as [17]) as the subsequent Interest's lifetime and use a multiplier  $\gamma > 1.0$  when checking for timeout events at the application as,

$$timeout = RTO * \gamma. \quad (5)$$

Using the calculated *timeout* from Eq. 5 gives the existing PIT entry (or entries) for an Interest in the network a *chance* to timeout before the application issues retransmission (RTx). It is because RTO over time varies, and at any given time, the calculated *timeout* can be less than the assigned calculated lifetime of the most recent timed-out Interest. In our simulation,  $\gamma = 2.0$  offers the best results. Using an RTO-based lifetime and timeout checker lowers the RTx aggregation effect, proportional to  $\gamma$ 's value. Some redundant data from the RTx aggregation is still helpful when the network is mobile and sparse.

### C. Effect of In-network Caching

The data-centric design enables in-network caching in NDN with content-store (CS), which can be highly effective under mobility and lossy networks in various ways. For example, if a data packet gets lost on its way to the consumer, a retransmission Interest may retrieve it from a potential closer cache node than reaching the actual producer. A cache hit can also occur if another consumer asked for the same-named data recently. However, caching can also lead to redundant data packets. In the example of Fig. 8a, we see two paths to data nodes  $d_1$  and  $d_2$  from consumer  $c$  during the discovery phase. In case one of the paths breaks or suffers collision/contention,

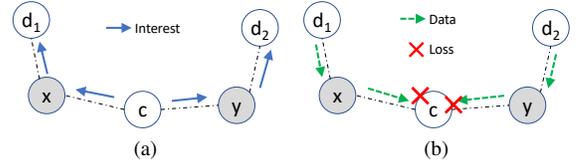


Fig. 8. Example of data redundancy through caching.

TABLE I  
NOTATIONS AND VALUES

Notation	Meaning	Value
cwl	CWL enabled	-
nocwl	CWL disabled (default AIMD)	-
cwl/dil	Both CWL and DIL enabled	-
$\gamma$	Interest timeout checker multiplier	2.0
CS	NDN content store capacity (packets)	0 or 200
1-1	1-to-1 communication	-
m-1	many-to-1 communication	-
m-m	many-to-many communication	-

the other has the chance to retrieve data successfully. On the other hand, there is also a high probability of redundant data coming back to the consumer from both paths and causing a collision at  $c$  (Fig. 8b). Thus caching can sometimes act as a double-edged sword depending on the network structure.

## VI. PERFORMANCE ANALYSIS

We now analyze the NDN AIMD performance in different setups of multiple consumer-producer communication with the discussed challenges and potential solutions in mind.

### A. Simulation Setup

We use ndnSIM [19] to simulate the default NDN AIMD and the proposed CWL and DIL for reducing contention and redundant transmissions. We also use TCP-NewReno [15] for TCP/IP, with and without CWL. However, we do not consider added packet loss like in Sec. III-B, as multi-consumer-producer communication is good enough to induce packet loss from collision/contention. Each consumer or sender application runs for the entirety of the simulation duration, which is 100 seconds. RTS/CTS is disabled as a proper solution is not available for broadcast Interest and Data. Under mobility, all nodes move at the same speed in a single simulation. Each data packet payload is 512 bytes, and when DIL is disabled, the default Interest lifetime is 2 seconds. We also collect an average of 10 runs per simulation. Notations and values used in the plots and simulations are in Table I.

### B. Analyzing Channel Contention and Mobility

We begin our analysis of the data-centric adaptive-rate application in wireless ad-hoc networks for multiple 1-to-1 consumer-producer communication, enabling and disabling mobility, in Fig. 9. Here, each consumer-producer pair exchanges data under a unique prefix, i.e., each consumer retrieves data from only one specific producer and vice-versa. This setup shows the simplistic data-centric communication with out-of-order data retrieval. Moreover, multicast occurs only on the aggregation of retransmitted interest as different

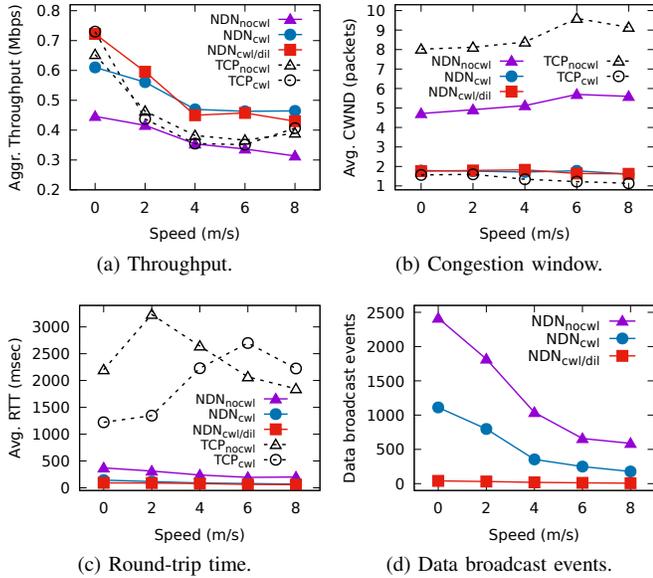


Fig. 9. Effect of channel contention, CWL and DIL on adaptive-rate application in 1-to-1 communication setup. CS=200 packets in NDN.

consumers ask for data under different name prefixes. TCP throughput inclusion is only for referential purposes.

The aggregated throughput in Fig. 9a shows the summation of application throughput in the ten consumers' applications. We can see that NDN has an average 17.25% lower throughput than TCP when CWL is disabled, with the lowest 32.3% when nodes are stationary. This result validates our claim that a default NDN AIMD suffers from *cwnd* overestimation without CWL and DIL. With CWL alone, NDN's average throughput over different speeds exceeds 12.6% compared to TCP with CWL. Note that TCP with CWL also has just about < 1% throughput improvement over *nocwl*, on average, which is noticeable when nodes are stationary. It shows that CWL can help both TCP and NDN to lower channel contention in such scenarios.

However, a default Interest lifetime of 2 seconds still leads to unexpected Interest aggregation in the network on application retransmissions, leading to high data broadcast. Fig. 9d verifies our claim showing *nocwl* leads to the highest Interest aggregation, thus the highest data broadcast. The *cwl* helps to reduce it to some extent, but not enough. Using both *cwl* dynamic interest lifetime (*dil*) offers the best overall throughput in NDN (16.44% more than TCP with *cwl*) as Data broadcast is the lowest, by allowing most in-network PIT entries to expire before retransmission. We further verify our claim that *cwl* and *dil* together reduce contention by looking into Fig. 9b, where high *cwnd* represents high channel contention possibility and vice-versa.

Fig. 9c further verifies that in NDN, lower RTT is a direct result of lower network contention and *cwl/dil* offers the lowest values. It is also due to DAF's Interest-Data only communication avoiding any explicit routing exchange, unlike IP-AODV. The very high RTT over different speeds using TCP is mostly from AODV's route setup and maintenance.

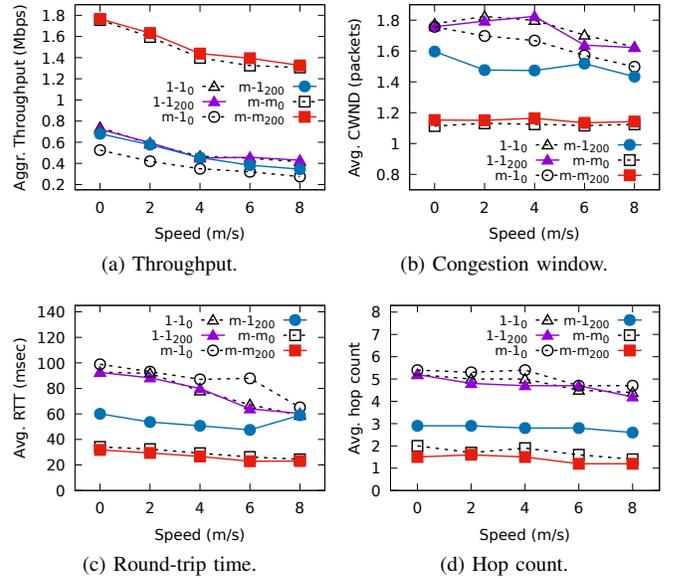


Fig. 10. Effect of caching on adaptive-rate applications under different communication setups. Subscript shows per-node CS capacity in packets.

On average, NDN with *cwl* and *dil* offers an aggregated throughput of 0.53 Mbps while only *cwl* and *nocwl* yield 0.51 and 0.37 Mbps, respectively. Furthermore, *NDN<sub>cwl/dil</sub>* shows a maximum of 19.67% more throughput than *NDN<sub>cwl</sub>*, indicating that *dil* works better in static networks.

### C. Analyzing Effects of Caching

Next, we analyze the effect of caching in NDN-based wireless ad-hoc networks with adaptive-rate applications and different communication setups in Fig. 10, such as 1-to-1 (1-1), many-to-1 (m-1), and many-to-many (m-m). We do not show TCP results in the plots as IP does not support caching by default.

While 1-to-1 communication is the same as in Sec. VI-B, in many-to-1 communication, there are two producers, serving data with two different prefixes, e.g., A, and B, respectively. Five consumers ask for the same set of data packets from one producer, and the other five consumers ask for the same set of data from the second one. In IP, however, one server sends five copies of the same data to five clients as multicast is not built-in to IP. This scenario expects the most NDN multicast, cache (when enabled) hits, and traffic reduction at (or near) an actual producer node. In many-to-many communication, five consumer-producer pairs exchange the same data set under the same prefix, e.g., A, while the other five pairs exchange data under another prefix, e.g., B. In IP, both 1-to-1 and many-to-many communications have the same behavior because of host-centric communication. However, in NDN, any eligible Data node (producer or cache) can serve a consumer following its location-agnostic design. Thus it shows the effect of retrieving data from the closest producer out of many by utilizing NDN multihoming and learning the shortest path towards data with DAF. All the communication schemes have *cwl* and *dil* enabled by default as they offer the best aggregated application throughput in our simulations.

Fig. 10a verifies our claim in Sec. V-C that the advantage of caching depends on the consumer-producer communication model. We see that only many-to-1 communication gains the most throughput with caching (28.83% more than without). On the other hand, gains in the case of 1-to-1 and many-to-many are minimal. In fact, in a 1-to-1 scenario, caching results in about 0.4% less throughput than without caching when nodes are stationary.

The caching effect is also visible in Figs. 10b, 10c, and 10d, where many-to-1 and many-to-many schemes reduce the *cwnd*, RTT, and average hop count between consumer and data, respectively. Such behavior is expected as NDN's in-network caching helps bring data closer to the consumer under lossy and unstable conditions. Again, many-to-1 shows the highest RTT and hop count reduction as NDN multicast and caching reduce network load at and near the actual producer. Our dynamic interest lifetime does not hamper the performance for different consumers as *dil* only reduces Interest aggregation on retransmissions from the same consumer using RTO and  $\gamma$ . However, in a 1-to-1 setup, *cwnd* and RTT reduction is almost none because redundant data packets from caching end up increasing network transmission, and thus, collision and contention also increase. Fig. 10a also shows that many-to-1 yields the lowest throughput without caching because of higher contention/collision possibility near a producer.

One noticeable result is in many-to-many communication with caching, where the average aggregated throughput is 1.51 Mbps (Fig. 10a), an average of 184.90% more than the 1-to-1 setup with caching. The reason is NDN's location-agnostic design, where a consumer retrieves data from the closest producer or cache node. The lowest average hop count in Fig. 10d and the resulting low *cwnd* and RTT in Fig. 10b and 10c, respectively, verify our claim.

Although we do not show the TCP/IP plots, on average, many-to-1 communication in NDN, with and without caching, shows 38.89% and 7.66% more throughput than TCP, respectively. In many-to-many communication with NDN, with and without caching, the throughput is on average 231.43% and 223.69% more than TCP. Such results reflect data-centric transport and forwarding advantages when it adequately handles *cwnd* adaptation and Interest lifetime.

## VII. CONCLUSION

NDN's data-centric architecture can help achieve better throughput in adaptive-rate applications with and without loss in linear topologies than TCP/IP. However, we also find that in an extensive wireless ad-hoc network, NDN's out-of-order retrieval and improper Interest lifetime setting lead to mismanaged congestion window and redundant data. Consequently, high packet collision and channel contention degrade application throughput. We verify these effects by simulating synthetic data and an AIMD approach for NDN applications. We show that applying a congestion window limit and an RTO-based dynamic interest lifetime can significantly reduce the adverse effects. Together, they improve transport performance and show that one can achieve better throughput

without changing the decentralized norm of NDN. We further show the effects of caching and that it is most beneficial when multiple consumers are asking for the same data. Finally, our analysis paves a path for future research on improvements and applications of data-centric architectures in real-world wireless ad-hoc networks.

## REFERENCES

- [1] A. Goldsmith, *Wireless communications*. Cambridge univ. press, 2005.
- [2] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on tcp throughput and loss," in *IEEE INFOCOM 2003*, vol. 3, 2003, pp. 1744–1753.
- [3] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks," *Wireless Networks*, vol. 8, no. 2-3, pp. 275–288, 2002.
- [4] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM Computer Communication Reviews*, Jun. 2014.
- [5] K. Chen, Y. Xue, and K. Nahrstedt, "On setting tcp's congestion window limit in mobile ad hoc networks," in *IEEE ICC'03*. IEEE, 2003.
- [6] T. D. Dyer and R. V. Boppana, "A comparison of tcp performance over three routing protocols for mobile ad hoc networks," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, 2001, pp. 56–66.
- [7] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad hoc networks*, 2003.
- [8] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "Tcp selective acknowledgment options," RFC 2018, Tech. Rep., 1996.
- [9] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and A. Sivakumar, "Atp: A reliable transport protocol for ad hoc networks," *IEEE transactions on mobile computing*, vol. 4, no. 6, pp. 588–603, 2005.
- [10] F. Wang and Y. Zhang, "Improving tcp performance over mobile ad-hoc networks with out-of-order detection and response," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002, pp. 217–225.
- [11] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 183–196.
- [12] M. A. Rahman and B. Zhang, "On data-centric forwarding in mobile ad-hoc networks: Baseline design and simulation analysis," 2021. [Online]. Available: <https://arxiv.org/abs/2105.07584>
- [13] M. Amadeo, A. Molinaro, C. Campolo, M. Sifalakis, and C. Tschudin, "Transport layer design for named data wireless networking," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2014, pp. 464–469.
- [14] S. H. Bouk, S. H. Ahmed, M. A. Yaqub, D. Kim, and M. Gerla, "Dpel: Dynamic pit entry lifetime in vehicular named data networks," *IEEE Communications Letters*, vol. 20, no. 2, pp. 336–339, 2015.
- [15] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The newreno modification to tcp's fast recovery algorithm." Internet Engineering Task Force (IETF), RFC 6582, RFC 6582, April 2012. [Online]. Available: <https://tools.ietf.org/rfc/rfc6582.txt>
- [16] E. Blanton, M. Allman, K. Fall, and L. Wang, "Rfc3517: A conservative selective acknowledgment (sack)-based loss recovery algorithm for tcp," RFC Editor, USA, Tech. Rep., 2003.
- [17] V. Paxson, M. Allman, H. J. Chu, and M. Sargent, "Computing tcp's retransmission timer," 2011. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6298.txt>
- [18] 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*, ser. ACM-ICN '16. ACM, 2016.
- [19] S. Mastorakis, A. Afanasyev, and L. Zhang, "On the evolution of ndnsim: An open-source simulator for ndn experimentation," *ACM SIGCOMM Computer Communication Review*, pp. 19–33, 2017.
- [20] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, ser. WMCSA '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 90–.
- [21] J. Dall and M. Christensen, "Random geometric graphs," *Physical review E*, vol. 66, no. 1, p. 016121, 2002.