U. S. Naval Research Laboratory
Washington, DC 20375-5320

NRL/5590/FR--2024/2

Effective Event-Driven Group Communications in Wireless Tactical Edge Networks

Jeffery W. Weston, Joseph P. Macker, R. Brian Adamson

Center for Computational Science
Information Technology Division

May 2, 2024

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION

1. REPORT DATE May 2, 2024	**2. REPORT TYPE** NRL Formal Report	**3. DATES COVERED** **START DATE** September 2022	**END DATE** September 2023

4. TITLE AND SUBTITLE
Effective Event-Driven Group Communications in Wireless Tactical Edge Networks

5a. CONTRACT NUMBER	**5b. GRANT NUMBER**	**5c. PROGRAM ELEMENT NUMBER** 61153N
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER** 6C13

6. AUTHOR(S)
Jeffery W. Weston, Joseph P. Macker, R. Brian Adamson

7. PERFORMING ORGANIZATION / AFFILIATION NAME(S) AND ADDRESS(ES)
U. S. Naval Research Laboratory
4555 Overlook Avenue, SW, Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER
NRL/5590/FR--2024/2

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
One Liberty Center
875 N. Randolph Street, Suite 1425
Arlington, VA 22203-1995

10. SPONSOR / MONITOR'S ACRONYM(S)
ONR

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A: Approved for public release, distribution is unlimited

13. SUPPLEMENTAL NOTES

14. ABSTRACT
This paper provides an overview of design and research results relating to the use of NACK-Oriented Reliable Multicast (NORM) transport within the Zero Message Queue (ZMQ) message bus framework. We outline numerous new NORM modes and options implemented as ZMQ socket options to provide for use of NORM-based ZMQ messaging across a wide variety of scenarios, use cases, and applications. These improvements have been accepted back into the ZMQ source code for use and experimentation. The presented network emulation experiments show that reliable multicast transport improvements and overhead reduction can result from proper configuration of NORM as a reliable ZMQ message-delivery service in wireless networks. NORM ZMQ's "fixed-rate" operation was shown to offer significant benefits over default congestion control when used in combination with more modern network traffic-management approaches. NORM ZMQ interoperability with modern queueing disciplines is also demonstrated, and areas for future improvement are discussed.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT U/U	**b. ABSTRACT** U/U	**c. THIS PAGE** U/U	**17. LIMITATION OF ABSTRACT** SAR	**18. NUMBER OF PAGES** 23

19a. NAME OF RESPONSIBLE PERSON
Jeffery W. Weston

19b. PHONE NUMBER *(Include area code)*
(202) 767-0172

PREVOUS EDITION IS OBSOLETE

STANDARD FORM 298 (REV. 5/2020)

Prescribed by ANSI Std. Z39.18

This page intentionally left blank

CONTENTS

FIGURES

EXECUTIVE SUMMARY

This work focuses on research and development to improve the resiliency and reliability of event-driven architectures (EDAs), focusing on group-based communications at the tactical edge using reliable multicast messaging with support for asymmetric edge networks. The internet standard NACK-Oriented Reliable Multicast (NORM) protocol, with its inherent performance benefits in disruptive wireless network conditions, serves as the reliable multicast protocol, and Zero Message Queue (ZMQ) serves as a popular messaging framework with flexible support for a variety of EDA messaging patterns and underlying transport options. The authors completed work to develop and integrate a variety of NORM modes and options as ZMQ socket options to enable the use of NORM-based ZMQ messaging across a wide variety of scenarios, use cases, and applications.

Multiple NORM congestion-control methods are now supported in addition to the default Internet wired congestion-control mode, including a "fixed-rate" mode that sets a maximum rate for message transmission, as well as more loss-tolerant congestion-control methods, including Explicit Congestion Notification (ECN)-based methods requiring intermediate router traffic notification support. NORM settings controlling the trade-offs between reliability and overhead are now exposed as ZMQ socket options, along with options for controlling multicast forwarding and quality-of-service markings. These improvements extend the present NORM ZMQ source code.

Wireless network emulation experiments using NORM ZMQ transport were conducted across a wide variety of network densities using both random static graphs and clustered mobility. Results demonstrated improvements to reliable ZMQ message delivery and overhead through the use of NORM multicast transport. NORM ZMQ's "fixed-rate" operation was also shown to offer significant benefits over the default TCP-friendly congestion-control mode, delivering more messages more reliably to groups of receivers while using less overhead. For NORM congestion control, utilizing an Essential Connected Dominating Set (ECDS) relay set offers significant advantages over classical flooding approaches in terms of both message delivery and overhead, particularly for denser networks. However, in sparser networks, ECDS routing can result in slightly lower message-delivery rates due to less redundancy. The message-delivery rates of fixed-rate NORM ZMQ are significantly higher than when using congestion control, so while the overhead differences between ECDS and classical flooding apply, the differences are not as significant in terms of message delivery. Many of the NORM ZMQ transport enhancements support enhanced utilization features for envisioned software-defined networks (SDNs) and future tactical edge network architectures. As an example, initial testing with hierarchical fair-service curve (HFSC) queueing support was demonstrated in wireless network emulation with competing network flows. HFSC can simultaneously support (a) hierarchical link-sharing service, (b) guaranteed real-time service with provable tight delay bounds, and (c) decoupled delay and bandwidth allocation. This demonstrates that NORM ZMQ message bus transport design can also easily integrate with networks using advanced queueing and traffic-management disciplines. Finally, we discuss further work that could be done to further improve the NORM ZMQ transport.

This page intentionally left blank

EFFECTIVE EVENT-DRIVEN GROUP COMMUNICATIONS IN WIRELESS TACTICAL EDGE NETWORKS

1. INTRODUCTION

Future Navy wireless communication networks introduce numerous technical challenges relating to distributed collaboration, reliability, efficiency, and dynamic adaptation. The envisioned future operational networks involve an ever-increasing reliance on coordinated and distributed tactical operations amongst heterogeneous platforms and sensors. Effective distributed network operations must support timely and group-oriented updates and exchanges of time-critical information, often over constrained and disrupted wireless communication networks.

In these future distributed tactical information networks, large amounts of critical and actionable data will be generated, and increasing amounts of data will need to be reliably exchanged and actions will need to be coordinated. Within state-of-the-art network technology, publish/subscribe (pub-sub) systems provide a useful means to manage dynamic interest topics and timely information exchanges amongst participants. However, there remains a significant technical challenge in supporting distributed, heterogeneous, constrained network operations needed for tactical effectiveness. Conventional multiparty Internet network message bus services often use network transport and architecture mechanisms that poorly match to the unique reliability and efficiency needs of tactical environment operations. Efficient-and-reliable group network messaging transport, such as multicast, is often poorly supported and information exchange architectures are often reliant on failure-prone centralized client/server architectures. Improving state-of-the-art network message bus options with robust, efficient group data transport is a means to help realize these capabilities.

The goal of the work presented here is to improve the resiliency and reliability of event-driven architectures (EDAs), with a particular focus on distributed, group-based communications at the tactical edge using decentralized multicast dissemination. The Negative-Acknowledgement (NACK) Oriented Reliable Multicast (NORM) protocol [1, 2] is a key building block for supporting this goal, offering inherent benefits for scaled group communications, asymmetric loss and delay, and temporally disrupted environments like those often present in tactical edge networks. NORM is an existing Internet standard protocol for reliable multicast communications that NRL helped develop and transition and NRL's public implementation [3] includes a variety of additional modalities and settings that support more effective group and disrupted network operations. The NRL NORM implementation has a well-defined, mature application programming interface (API) supporting unicast transport as well as multicast, along with a variety of congestion-control modalities and protocol behavior settings that can help manage the inherent complex trade-offs between reliability, overhead, and latency needs for particular applications or network scenarios.

ZeroMQ (ZMQ) [4] was chosen as an initial decentralized EDA framework to target for integration because it is a popular open-source framework with support for a variety of underlying transport methods and multiple EDA messaging patterns. Compared to other EDA frameworks, ZMQ is flexible, allowing for a wide variety of designs and approaches, including flexible brokering designs. We specifically targeted the

---

Manuscript approved March 29, 2024.

pub-sub messaging pattern for initial work, as more advanced architectures are commonly derived from this basic functional network messaging pattern.

This paper is organized as follows. First, we discuss past work, then provide an overview of the NORM ZMQ integration improvements. NORM ZMQ performance results are presented and compared against both reliable and best-effort protocols, for both static and mobile scenarios. NORM ZMQ operation with standard queueing mechanisms is demonstrated. Finally, we discuss future work and conclusions.

2. PAST WORK

Previous work on tactical EDA frameworks largely focuses on improving higher-layer EDA middleware, utilizing EDA concepts such as store-and-forward caching to improve system response to network disruptions and delays. Our previous work in [5] provides a more detailed overview of the state of the art of tactical EDA, but most studies are limited by the use of reliable unicast transport. Our research differs from these related works by focusing on improving the efficiency and effectiveness of the underlying event transport that higher-layer EDA middleware depends on. Our goal is to demonstrate that group-centric multicast transport and efficient forwarding mechanisms at the transport layer enable an EDA to more efficiently deliver events across a constrained tactical network.

Previous efforts implemented NORM as a ZMQ transport option, alongside Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and a number of other preexisting ZMQ transport engines. This initial integration supported both unicast and multicast transport, but was otherwise hard-coded to use a particular NORM configuration, including a modality called congestion-control (CC) mode. Congestion-control mode is a sensible choice for an open-source Internet software product, as the mode is meant to operate in a manner similar to TCP congestion-control methods so that NORM flows will coexist fairly with TCP flows on common Internet networks. It is well known that in many tactical edge networks, a TCP-based congestion-control method leads to significant overhead and reduced performance in response to packet loss [6].

Results from our related work in [5] show improvements to reliable ZMQ data delivery and overhead under a variety of topology densities by using multicast NORM transport, and in particular, the ”fixed-rate” mode of NORM instead of the congestion-control mode. The term ”fixed-rate” here is a reference to the more general NORM mode, but it may be a bit misleading in the context of ZMQ transport — it is effectively setting a fixed maximum datarate, including any error-control-based parity packets needed to repair missing transport data. Initial testing in [5] was limited in a few ways. First, the scenarios are random static topologies with no mobility effects. Secondly, the specific NORM modalities used were hard-coded into a series of custom ZMQ implementations as a proof of concept, rather than being flexible options integrated into the code that could be utilized by others. We will discuss the more adaptive designs that have recently been achieved and we will also present mobile emulation test results.

3. NORM ZMQ TRANSPORT IMPROVEMENTS

Previous work used a custom ZMQ implementation with hard-coded settings for NORM ”fixed-rate” operation. To improve on that design and to allow for wider usage of this functionality, improvements have been made to the NORM ZMQ transport integration to allow for more flexibility by implementing a variety of modes and settings that may be altered at runtime to control NORM using ZMQ socket options. These changes are detailed below, followed by a discussion of a future design that has been partially implemented.

3.1 NORM ZMQ Socket Options

This section provides an overview of improvements the authors made to the NORM ZMQ transport. All options are implemented as ZMQ ”socket options” and currently need to be set before calling *connect* or *bind* on the ZMQ socket to initialize communications.

The main, new socket option controlling the publisher's NORM congestion-control mode is *ZMQ_NORM_MODE*, which uses the previous default TCP-friendly congestion-control method (*ZMQ_NORM_CC*), but also supports ”fixed-rate” mode (*ZMQ_NORM_FIXED*) using a maximum datarate set by the preexisting *ZMQ_RATE* socket option that performs similar functions for other ZMQ transports. Additionally, *ZMQ_NORM_MODE* supports the use of a moderately wireless loss tolerant version of congestion control (*ZMQ_NORM_CCL*). Finally, there are two versions of congestion control that use Explicit Congestion Notification (ECN), which requires support by intermediate routers to mark packets: *ZMQ_NORM_CCE* uses ECN markings in addition to normal congestion-control methods, whereas *ZMQ_NORM_CCE_ECNONLY* relies solely on ECN markings for congestion control. ECN modes have the ability to drastically increase protocol performance by distinguishing random wireless channel loss from congestion-induced loss in networks where ECN can be utilized, as shown in [6].

The standard NORM settings controlling efficiency and repair trade-offs are implemented via several new ZMQ socket options. The *ZMQ_NORM_SEGMENT_SIZE* socket option controls the maximum message payload size for individual NORM transport layer messages. ZMQ messages may be split over multiple NORM messages, so this does not affect maximum ZMQ message size. For maximum efficiency and lower overhead, *ZMQ_NORM_SEGMENT_SIZE* should be set to the system or network maximum transmission unit (MTU), less any additional NORM message headers of up to 48 bytes. NORM uses forward-error-correction (FEC) coding methods to repair missing data and can send proactive parity packets to automatically repair some amount of missing data without needing additional communications (i.e., a NACK from the receiver and an explicit repair packet from the sender). The publisher's *ZMQ_NORM_BLOCK_SIZE* controls the number of segments in each FEC coding block, which comprises data segments along with a number of optional proactive repair segments controlled by *ZMQ_NORM_NUM_AUTOPARITY*. The *ZMQ_NORM_BLOCK_SIZE* is currently limited to 255 segments, but the number of calculated repair packets (*ZMQ_NORM_NUM_PARITY*, which must be greater than or equal to *ZMQ_NORM_NUM_AUTOPARITY*) is limited to 255−*ZMQ_NORM_BLOCK_SIZE*. There is a direct trade-off between the number of proactive repair segments and overhead — each proactive repair segment increases overhead, but that may be a useful trade-off for scenarios where lower latency is desired, as regular repairs will not happen until the block boundary has been reached. This may also be valuable in scenarios where network communications have asymmetric characteristics.

Additional, new ZMQ socket options to control NORM behavior include *ZMQ_NORM_BUFFER_SIZE*, which controls the size of various publisher and subscriber buffers and limits how far back data can be repaired. *ZMQ_NORM_UNICAST_NACK* is a subscriber option controlling whether NACK packets will be sent back to the subscriber via unicast instead of multicast. NORM transport endpoints that specify a unicast address will use unicast NACKs by default, but the default for multicast transport endpoints is to send the NACK via multicast to suppress NACKs at other receivers and to avoid duplicate repair requests. The publisher setting *ZMQ_NORM_PUSH* enables NORM's stream push mode, which changes the behavior of enqueueing new data. By default, NORM will stop accepting new messages while waiting for old data to be transmitted or repaired, but push mode discards the oldest data, which may be pending repair or may have never been transmitted, in favor of accepting newer data. This may be beneficial for some message types such as periodic messages, where it may be more desireable to deliver new data quickly instead of reliably delivering all older data.

Lastly, recent changes have been made to allow NORM ZMQ transport to make use of preexisting ZMQ socket options to set fields in the underlying Internet Protocol (IP) header of NORM packets. A packet's time to live (TTL) value controls how many times a packet can be forwarded through a network before being dropped. This was previously fixed to the maximum TTL value of 255 in the NORM ZMQ transport, but is now controllable using the same *ZMQ_MULTICAST_HOPS* socket option that is used for this purpose by other ZMQ transports (e.g., UDP), and the new default is 1 hop. ZMQ also had an existing socket option to control Type of Service (ToS) packet markings, *ZMQ_TOS*, but it was not supported by NORM ZMQ transport. TOS, in its modern form, consists of a 6-bit Differentiated Services Code Point (DSCP) field and a 2-bit ECN field. The DSCP portion can be used by queueing mechanisms to give priority to certain packets, while the ECN portion is used to implement ECN by marking a packet as ECN-capable and allowing an intermediate router to change the marking to a state representing that congestion was experienced. Given the added support for ECN-aware congestion-control methods in the NORM ZMQ transport, TOS was very important to control. As implemented within NORM ZMQ, if an ECN-aware congestion-control method is used, the NORM transport will mark the flow as ECN-capable, whether the specified TOS value does so or not.

These NORM transport socket option improvements are now part of ZMQ's *libzmq* project source code, so they are available to use by future efforts for further experimentation and development. The improvements are included in stable releases of *libzmq* starting with version 4.3.5, though they require building with the ”DRAFT API” currently. The socket options are usable under programming language-specific bindings such as the *pyzmq* Python library, though they may need to be used by the numeric value assigned to each socket option until the language-specific bindings update their socket option lists. Future improvements for these socket options include allowing some of them (e.g., *ZMQ_RATE*) to be changed after *connect* or *bind* has been called to allow the transport to adjust more easily to changing network capacity.

3.2 NORM ZMQ Listener and Connector

The improved NORM ZMQ transport has a few important limitations due to the way the engine was originally implemented. Currently, publishers and subscribers using NORM unicast transport need to use the same port on each end, and need to specify the other endpoint's IP address directly, which means that using unicast NORM transport among a group of publishers and subscribers requires a large number of port assignments and significant a priori knowledge. Other ZMQ transports (e.g., TCP, UDP) use a ”listener”-and-”connector” paradigm, which allows a publisher or a subscriber to connect or bind to a port only so that an indefinite number of unicast endpoints can connect to that same port using ephemeral ports for the other endpoint. The ”listener”-and-”connector” design would also allow the full array of ZMQ message patterns to take advantage of NORM multicast transport. The connection-oriented nature would also enable ZMQ multicast publish/subscribe sessions to learn client subscription topics as they connect to the multicast publisher, and this would allow for automated strategies to filter topics of interest. Under this effort, progress was made toward implementation of such a NORM ZMQ ”listener”-and-”connector” transport using a new NormSocket API extension, but that work is incomplete and is planned to be finalized and matured in future work.

4. NORM ZMQ PERFORMANCE

Initial testing for the new NORM ZMQ transport consisted of functional testing to ensure that each socket option works correctly. As part of this testing, a late-joining subscriber's error in the original transport code was discovered and fixed. After initial functional testing, the next step was to reproduce the static testing results from [5] in order to validate the performance of the new code, followed by an exploration of the performance under mobility. To accomplish this, a series of tests were carried out in a wireless network emulation environment using the Extendable Mobile Ad Hoc Network Emulator (EMANE) [7] and the Common Open Research Emulator (CORE) [8]. The 802.11abg EMANE model was used with unicast and multicast rates set to 1 Mbps. For this testing, *AdaptNet* k-X Topology Control (kXTC) [9] was used to generate topologies of varying density and resilience from a single set of node positions by adjusting the transmit power of each node.

In addition to NORM ZMQ transport, TCP is also used as an example of a de facto pub-sub reliable transport protocol. TCP use is limited to unicast, it suffers under high loss and error, and it scales poorly when used in multiple parallel broadcast network scenarios. For best-effort results, we use unicast and multicast UDP, implemented in ZMQ's DRAFT API using the RADIO-DISH pattern rather than reliable pub-sub. Shortest-path unicast routing is calculated ahead of time based on calculated topologies and is set statically, rather than using a dynamic unicast routing protocol such as Optimized Link State Routing (OLSR) [10]. Multicast results here use Simplified Multicast Forwarding (SMF) [11] to deliver the traffic to all nodes in the network using two different forwarding mechanisms. Classical flooding (CF) is a simple mechanism wherein each node forwards each packet once, keeping recent state to prevent a node from forwarding the same packet multiple times. The Essential Connected Dominating Set (ECDS) [11] method calculates a minimal backbone of nodes that forward traffic, using the same duplicate detection mechanisms used by CF, while nodes that are not selected as relays do not forward any multicast traffic. This calculation is designed to be done in real time on an individual-node basis using 2-hop neighborhood information, but in this case, we precalculate the ECDS relay set based on expected topologies. SMF-ECDS provides lower overhead and reduced contention, as fewer nodes need to forward each packet, but can result in more loss due to decreased redundancy when SMF-CF is able to operate without causing significant contention and congestion.

To study the performance of the ZMQ transport protocols, we will use the message-delivery ratio (MDR) metric from [5], a ratio of unique ZMQ messages delivered vs. transmitted that is calculated as

$$MDR_{i,j} = \frac{Rx_j}{Tx_i}, \tag{1}$$

where i is the ith traffic flow at a receiver j;

$$MDR_n = \frac{\sum_{i,j} MDR_{i,j}}{\sum_i numRx_i} \tag{2}$$

$$numRx_i\text{: total receivers per } flow_i.$$

MDR is presented for each individual flow in each individual test via a bar-and-whiskers plot with an additional, white circle denoting the mean value. Each plot includes 6000 data points: 600 individual flow

MDR values per test for all-to-all communications among 25 nodes (discounting any messages to self) across 10 tests. It should be noted that ZMQ is a message bus — therefore, each receiver is expected to receive all messages. This means that even reliable protocols can have low MDR values because some nodes receive fewer messages than other nodes.

Additionally, we present a similar plot of network load measurements for each trial, calculated as the mean rate, in Kbps, of ZMQ packets emitted by all nodes in the network. This includes all packets sent and forwarded, as well as any protocol overhead such as retransmissions, acknowledgements (ACKs), NACKs, etc. These graphs will include fewer outliers than the MDR graphs, as they are calculated from only 10 data points — a single, mean network-wide value per test for 10 overall randomized tests. A white circle will again denote the overall mean value.

4.1 Static Test Results

In order to reproduce and validate test results from previous work in [5] using the new NORM ZMQ transport code, we used the same scenario-generation setup as that paper: 25 nodes randomly distributed in an area with radius 5000 m, with variable-density topologies generated with kXTC using $k \in 2, 3, 5, 8$ (higher k creates a denser network) and test traffic of two 64-byte Poisson-distributed messages per second in an all-to-all message-traffic pattern for 5 minutes. We did not use the same positions as in the original paper, but instead generated new positions using the BonnMotion [12] RandomWaypoint motion generator by taking the initial positions. Table 1 provides an overview of the graph metrics across all 10 static random topologies using each k connectivity value. Despite using the same position- and topology-generation methods, we note that the average metrics for degree and number of edges differ significantly from those values in the original paper. The reason for this is not clear, but is likely to be a miscalculation in the original paper, partially due to neglecting to take into account extra links added when a node chooses the largest of its minimum transmit powers necessary to communicate with each of its intended neighbors as is required to form a topology-controlled local neighbor multicast network model.

Table 1—Graph Metrics for 10 Random Static Topologies

	Degree	Number of Edges	Diameter
k2	6.6000 (SD 1.6767)	82.5000 (SD 4.8132)	5.8000 (SD 0.6325)
k3	8.6720 (SD 2.2326)	108.4000 (SD 4.7656)	4.6000 (SD 0.5164)
k5	11.7520 (SD 2.9252)	146.9000 (SD 4.7481)	3.3000 (SD 0.4830)
k8	15.8880 (SD 3.4808)	198.6000 (SD 5.0155)	3.0000 (SD 0.0000)

The MDR and network load results of these tests for best-effort UDP ZMQ transport are shown in Figs. 1 and 2. The graphs here show the same trends present in the original paper for nonreliable, best-effort UDP transport. All tests performed were done using ZMQ messaging, but we distinguish the various test cases by labeling the transport protocol used within ZMQ and the packet-routing mechanism. Multicast UDP significantly outperforms unicast while creating a lower network load because the large number of individual unicast flows necessary for an all-to-all communications pattern creates significant network congestion, leading to additional loss. Multicast with SMF-ECDS has significantly lower network load vs. SMF-CF, but MDR values for SMF-ECDS are slightly lower at lower k values due to decreased redundancy and the absence of Request-to-Send/Clear-to-Send (RTS/CTS) signaling in 802.11 multicast used here.

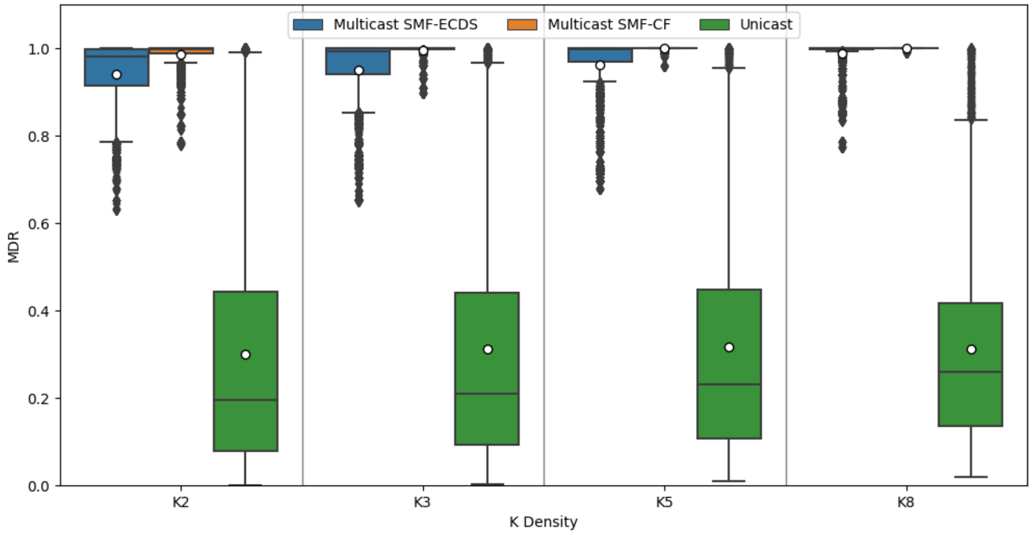

Fig. 1—Best-effort MDR (static topologies)

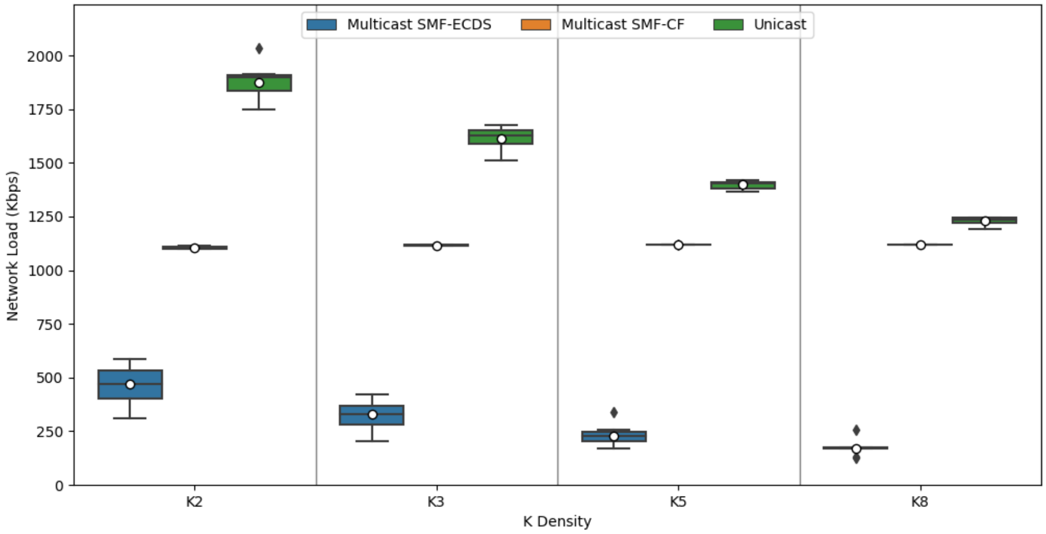

Fig. 2—Best-effort network load (static topologies)

Figures 3 and 4 show the MDR and network-load measurements for ZMQ reliable transport, including TCP as well as NORM with congestion control and fixed-rate modes of operation. Again, for the reliable transports studied here, lower MDR does not generally mean that messages were lost, but that some subset of the receivers were not able to receive as many messages as other receivers in the given time. The graphs here validate the trends from the original paper for reliable transport. TCP ZMQ performs poorly due to its use of a large number of unicast flows, which causes network congestion and loss leading to significant retransmissions and decreased transmit rate. Overall, TCP ZMQ experiences both low MDR and low network load because many individual flows fail to get any messages through. Multicast with NORM congestion-control mode (NORM-CC) performs well, with SMF-ECDS mode resulting in higher MDR in denser networks at higher k values when compared with SMF-CF. Fixed-rate NORM outperforms the congestion-control mode, resulting in significantly higher MDR and more overall data transmitted while simultaneously creating less network load, particularly at lower k values.

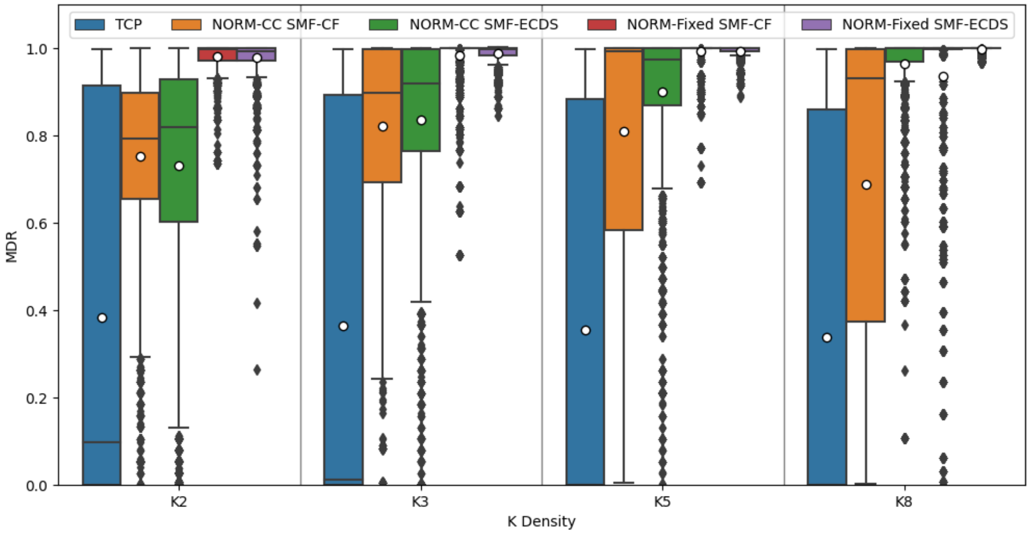

Fig. 3—Reliable MDR (static topologies)

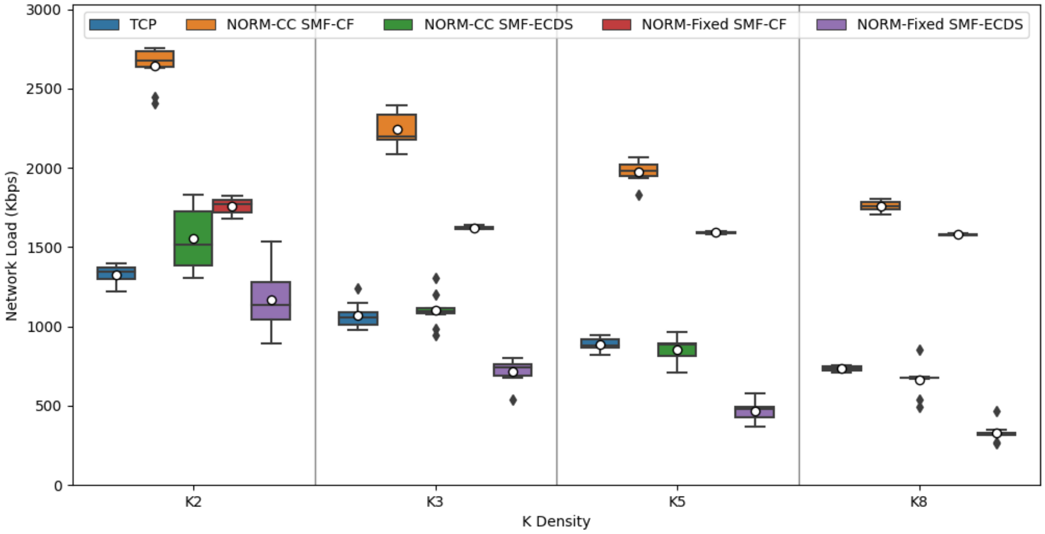

Fig. 4—Reliable network load (static topologies)

4.2 Clustered Mobility Test Results

The static test results confirmed that the new NORM ZMQ transport implementation performs similarly to the hard-coded versions used in initial research project testing. Next, we wanted to look at how well this transport operates under more challenging conditions such as mobility. For this testing, we used the BonnMotion Reference Point Group Mobility (RPGM) model to create a clustered-mobility scenario with five clusters of five nodes within a 5-km-by-5-km area. Maximum distance to cluster center was 250 m and nodes remain in the same cluster for the duration of the 10-minute scenario. We again used *AdaptNet* kXTC with $k \in 2, 3, 5, 8$ to create networks of varying density from the same physical topologies by controlling node transmit power. Node positions and graph topologies (based on link cost and transmit power settings) are changed on 10-second intervals. As in the static testing, test traffic consisted of two 64-byte Poisson-distributed ZMQ messages per second in an all-to-all message-traffic pattern. As in the static testing, shortest-path unicast routes are calculated for each temporal network graph and are updated every 10 seconds when the topology changes. This approach makes the results somewhat independent of any particular routing protocol implementation specifics in order to focus on the transport characteristics. Similarly, for multicast routing, ECDS relay sets are calculated in advance for each network graph and are updated as the topology changes. Multicast with classical flooding is the same regardless of topology, so it requires no changes.

Table 2 provides an overview of the graph metrics summarized across each 10-second graph period and across all 10 randomly generated mobile-cluster scenarios. When compared to the random static topologies, the degree and number of edges are slightly lower for clustered mobility, while the network diameter is significantly larger for lower k values. The larger diameter may be an effect of the clustered mobility model — the kXTC algorithm used is not cluster-aware, but it may naturally form more links among cluster members and limit links between clusters.

Table 2—Graph Metrics for 10 Clustered Mobile Topologies

	Degree	Number of Edges	Diameter
k2	6.4353 (SD 1.7348)	80.4413 (SD 3.6538)	7.2587 (SD 1.0289)
k3	8.2519 (SD 2.2631)	103.1492 (SD 5.6029)	5.5365 (SD 0.6603)
k5	11.6526 (SD 3.2577)	145.6571 (SD 7.9648)	3.6810 (SD 0.4695)
k8	15.7385 (SD 3.9010)	196.7317 (SD 6.5859)	2.9937 (SD 0.0794)

Trends in the ZMQ best-effort test results for clustered-mobility topologies largely mirror those present in the static test results. Figure 5 shows drops in mean MDR across all scenarios when compared against static results in Fig. 1, but the difference is minimal. Network load measurements in Fig. 6 are nearly identical to the static topology results.

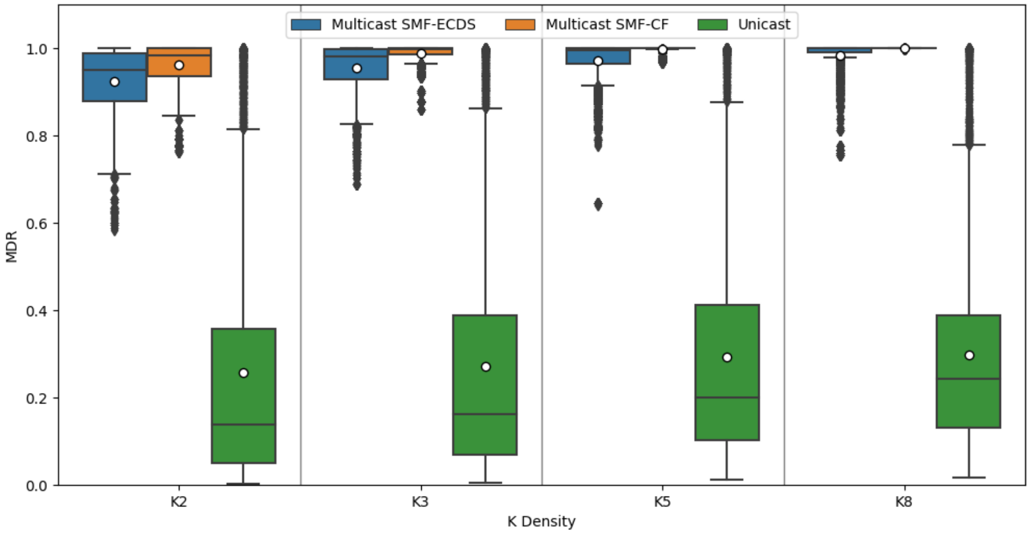

Fig. 5—Best-effort MDR (clustered mobility)

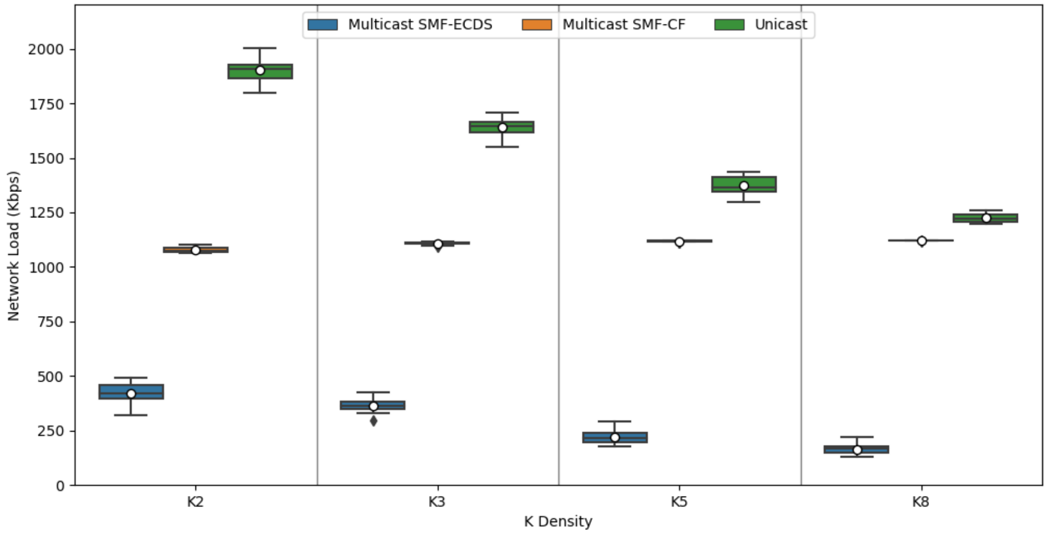

Fig. 6—Best-effort network load (clustered mobility)

When comparing the ZMQ reliable transport results for clustered mobility to static testing results, we again see the same trends present. For NORM-CC with SMF-CF, Fig. 7 shows slightly lowered MDR performance at lower k and slightly higher MDR at higher k. NORM-CC with SMF-ECDS shows slight improvement across all k values, which may be a result of the larger network diameter. NORM fixed-rate transport outperforms the NORM-CC and performance is largely identical to static testing results. Reliable transport network load measurements in Fig. 8 again show the same general trends as in static testing. These results confirm that the NORM ZMQ transport works well under mobile scenarios in addition to static experiments.

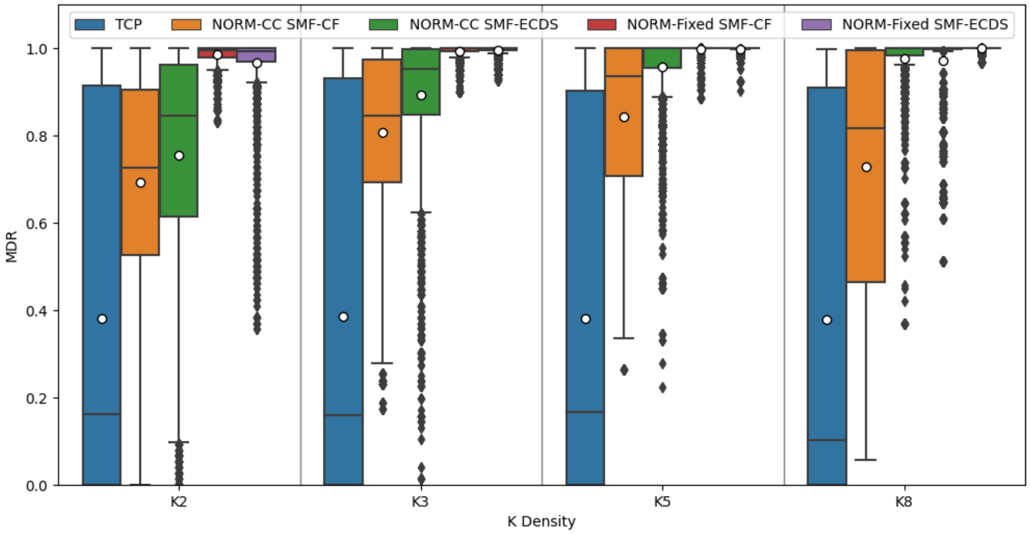

Fig. 7—Reliable MDR (clustered mobility)

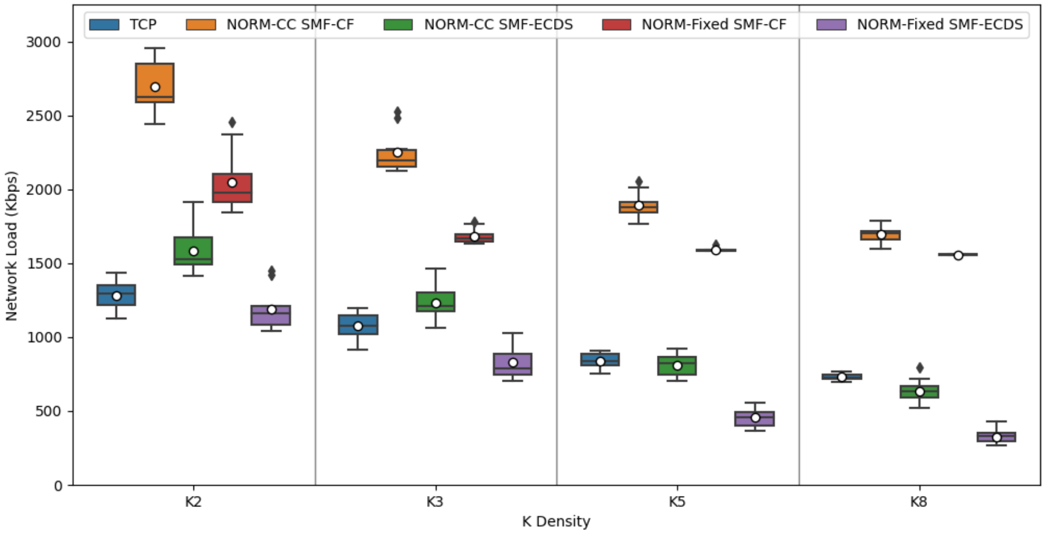

Fig. 8—Reliable network load (clustered mobility)

5. NORM ZMQ QUEUEING DISCIPLINE SUPPORT

Modern network architectures can use a variety of advanced traffic queueing and scheduling disciplines, and ZMQ should be able to provide mechanisms to assist with traffic classification and management. Some of the NORM ZMQ transport improvements added in this work are directly applicable to those goals. In particular, support for the *ZMQ_TOS* socket option allows packets to be marked with a DSCP value, which can be used by the queueing discipline and any associated rules to identify and classify NORM ZMQ traffic. Additionally, the NORM fixed-rate mode, which performed well in both static and dynamic network conditions, ensures that a ZMQ flow stays within allotted capacity and/or delay limits, if so required. In the future, it may be possible to allow NORM's fixed-rate mode to rapidly change its operational rate limit in response to fluctuating network conditions, but that is presently set when establishing sessions.

In order to demonstrate NORM ZMQ transport operation under a widely used advanced network queueing and scheduling mechanism, we used wireless network emulation to create a dynamic network traffic experiment using the Hierarchical Fair Service Curve (HFSC) [13] queueing discipline. HFSC is a complex queueing discipline allowing for a hierarchical arrangement of classes of traffic resulting in the proportional assignment of bandwidth to individual classes as well as allowing for latency constraints. For this test, three main leaf classes were set up: two classes that guarantee a minimum of 100 Kbps each, and a class that guarantees a minimum of 800 Kbps bandwidth, as shown in Fig. 9. All classes are allowed to use up to the full 1000 Kbps bandwidth available to the root class when there is demand and unused bandwidth available. In order to utilize these classes, a series of *tc* traffic control filters were set up. Low-bandwidth control traffic (e.g., routing protocol packets to keep links established) was assigned to one of the 100 Kbps class. A high-bandwidth, but low-priority, flow of NORM ZMQ traffic was assigned to the other 100 Kbps class — this traffic will attempt to fill the available bandwidth when there is unused bandwidth. Finally, a high-throughput, high-priority, intermittent flow of NORM ZMQ traffic was assigned to the 800 Kbps class.

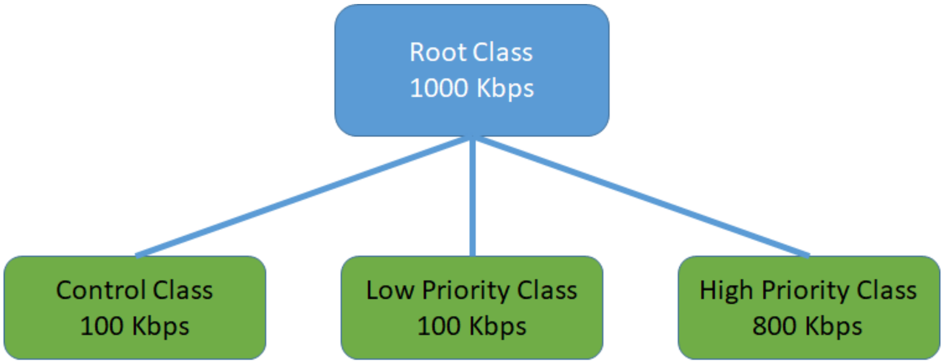

Fig. 9—HFSC queueing class configuration

Figure 10 shows the low-priority NORM ZMQ flow operating above its guaranteed bandwidth by taking advantage of available bandwidth in a flexible manner. Another NORM ZMQ flow, set to high priority using a *ZMQ_TOS* marking matching the appropriate *tc* filter, is activated, using its allotted bandwidth and limiting the additional bandwidth available to the lower-priority flow. Once the intermittent high-priority flow has completed or paused, the lower-priority flow is able to resume utilization of unused capacity. In this example, both NORM ZMQ flows are using NORM CC mode, which explains why the curves do not change more abruptly, as they are using congestion control to sense the effective capacity available to each traffic class.

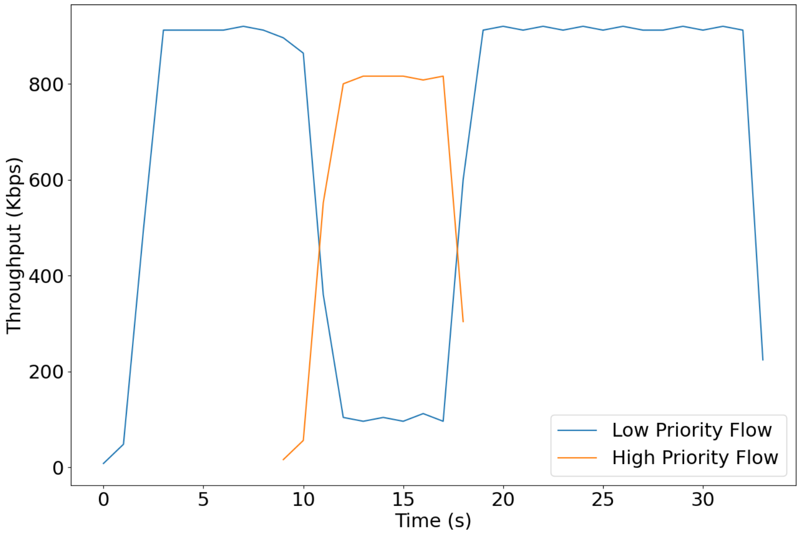

Fig. 10—NORM ZMQ flows in an HFSC queueing network

6. FUTURE WORK

Current NORM ZMQ socket options are required to be set before starting communications using *connect* or *bind* on the ZMQ socket. This means that once ZMQ connections are established, the values cannot be changed. For most of the implemented NORM ZMQ socket options, it would not be useful or possible to change the assigned values once the connection has been established. However, if the value for *ZMQ_RATE* could be changed on the fly when using NORM ”fixed-rate” mode, that would enable NORM ZMQ fixed-rate mode to adjust to changing network capacity. This can be accomplished in the ZMQ code using existing NORM API calls but requires additional programming to track NORM ZMQ sessions.

While we implemented the most important NORM settings here, there are many additional NORM API options that could be useful to parameterize as ZMQ socket options. For instance, it is possible to use existing NORM API calls to assign upper and lower rate bounds for NORM congestion-control mode, which could allow for bandwidth management in use cases where congestion-control mode is desired. Additionally, NORM maintains a group round-trip time (GRTT) estimate, which can be measured in a variety of ways, including probing. The GRTT estimate affects various internal timers, such as repeat-request timers for reliable operation. It may be desired to adjust the GRTT probing method and to manually set the initial GRTT estimate in order to optimize performance and to minimize protocol overhead in some use cases, so ZMQ socket options to adjust NORM parameters such as those could also be supported.

As noted previously, work was begun on a ”listener”-and-”connector” paradigm for the NORM ZMQ transport that is similar to what is used for TCP and UDP, but that work remains unfinished. The existing NORM ZMQ transport works well, but this effort would allow for topic filtering and would enable use by the full array of ZMQ message patterns. It would also make it easier to connect unicast endpoints together, requiring the use of fewer ports and less a priori knowledge. Work was done in this effort to begin developing this code, but we were not able to complete or test it. This would be a very useful change for future use and could still benefit from the ZMQ socket options implemented here.

7. CONCLUSIONS

We implemented numerous improvements to the NORM ZMQ transport implementation through the use of ZMQ socket options. Various NORM settings were implemented to control the trade-offs between reliability and protocol efficiency and overhead. Support was added to the NORM ZMQ transport for existing ZMQ socket options used to control ToS markings and multicast TTL. Finally, multiple NORM congestion-control methods are now supported in addition to the TCP-friendly congestion-control method previously implemented. These congestion-control methods include a fixed-rate mode that sets the maximum throughput used by the transport, a more wireless-loss-tolerant congestion-control method, and several modes that utilize ECN. These changes were integrated into the existing ZMQ source code, vastly improving the flexibility and utility of NORM ZMQ for a variety of use cases, including wireless tactical edge networks.

Wireless network emulation testing of the improved NORM ZMQ transport confirms the trends seen in previous work and extends that to cover mobility experiment in addition to initial static network topology experiments. Overall, it was shown that multicast ZMQ transport offers significant advantages over unicast transport for both message delivery and network overhead. NORM ZMQ's fixed-rate mode of operation significantly outperforms the default congestion-control mode, delivering more messages more reliably to all receivers while using less overhead. For NORM ZMQ congestion-control mode, utilizing ECDS relay sets for multicast forwarding offers significant advantages over classical flooding in both message delivery and overhead, particularly for denser networks, but in sparser networks, it may result in slightly lower message-delivery rates due to less redundancy. The message-delivery rates of fixed-rate NORM ZMQ are significantly higher than when using congestion control, so while the overhead differences between ECDS and CF apply, the differences between ECDS and CF are not as significant in terms of message delivery for the scenarios examined.

Many of the NORM ZMQ transport improvements support utilization within future tactical edge networks. Small-scale testing with HFSC queueing showed that NORM ZMQ transport can operate well in networks using advanced queueing disciplines, allowing an intermittent NORM ZMQ transport to reserve bandwidth to use when needed but to cede that bandwidth to other flows when it is inactive or when a higher-priority flow reduces its effective capacity.

Future work could implement a number of less important NORM parameters within the NORM ZMQ transport as well as modify some parameters (e.g., fixed-rate value) to be changed for already-established sessions. Work was begun on a more advanced ”listener”-and-”connector” paradigm to allow for better port reuse and less a priori knowledge of connection points, but completing that remains an area of further development and testing.

REFERENCES

1. J. P. Macker, C. Bormann, M. J. Handley, and B. Adamson, “NACK-Oriented Reliable Multicast (NORM) Transport Protocol,” Request for Comments RFC 5740, Internet Engineering Task Force, Nov. 2009. URL `https://datatracker.ietf.org/doc/rfc5740/`.

2. B. Adamson and J. P. Macker, “Reliable messaging for tactical group communication,” Proceedings of the 2010-Milcom 2010 Military Communications Conference (IEEE), 2010, pp. 1899–1904.

3. “NACK-Oriented Reliable Multicast (NORM).” URL `https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/NORM/`.

4. “ZeroMQ.” URL `https://zeromq.org/`.

5. C. Bowers, J. P. Macker, J. W. Weston, and B. Adamson, “Multicast Enhancements to Event-Driven Tactical Networking,” Proceedings of the MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM), San Diego, CA, USA (IEEE), Nov. 2021, pp. 175–180. ISBN 9781665439565, doi:10.1109/MILCOM52596.2021.9653084. URL `https://ieeexplore.ieee.org/document/9653084/`.

6. J. W. Weston, J. P. Macker, and R. B. Adamson, “Adapting NORM Unicast Transport for Loss Tolerant and ECN Environments,” Proceedings of the 2014 IEEE Military Communications Conference, Baltimore, MD, USA (IEEE), Oct. 2014, pp. 962–967. ISBN 9781479967704, doi:10.1109/MILCOM.2014.165. URL `http://ieeexplore.ieee.org/document/6956885/`.

7. “Extendable Mobile Ad-hoc Network Emulator.” URL `https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/EMANE/`.

8. “Common Open Research Emulator (CORE).” URL `https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/CORE/`.

9. J. Macker, C. Bowers, S. Kompella, and C. Kam, “Wireless Network Topology Control: Supporting Link Cost Constraints and Resiliency,” NRL/FR/5522- -20-10,404, Naval Research Laboratory, https://apps.dtic.mil/sti/pdfs/AD1116830.pdf, Dec 3, 2020.

10. T. H. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” Request for Comments RFC 3626, Internet Engineering Task Force, Oct. 2003. URL `https://datatracker.ietf.org/doc/rfc3626/`.

11. J. P. Macker, “Simplified Multicast Forwarding,” Request for Comments RFC 6621, Internet Engineering Task Force, May 2012. URL `https://datatracker.ietf.org/doc/rfc6621/`.

12. N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, “BonnMotion: a mobility scenario generation and analysis tool,” Proceedings of the Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, Malaga, Spain (ICST), 2010. ISBN 9789639799875, doi:10.4108/ICST.SIMUTOOLS2010.8684. URL `http://eudl.eu/doi/10.4108/ICST.SIMUTOOLS2010.8684`.

13. I. Stoica, H. Zhang, and T. S. E. Ng, “A hierarchical fair service curve algorithm for link-sharing, real-time and priority services,” *ACM SIGCOMM Computer Communication Review* **27**(4), 249–262 (Oct. 1997), ISSN 0146-4833, doi:10.1145/263109.263175. URL `https://dl.acm.org/doi/10.1145/263109.263175`.