Adapting NORM Unicast Transport for Loss Tolerant and ECN Environments

Jeffery W. Weston, Joseph P. Macker, R. Brian Adamson
Information Technology Division, Naval Research Laboratory, Washington DC
jeffery.weston@nrl.navy.mil, joseph.macker@nrl.navy.mil, brian.adamson@nrl.navy.mil

***Abstract*—This work presents new experimental results examining the loss tolerant unicast congestion control behavior of the NACK-Oriented Reliable Multicast (NORM) transport protocol documented initially as Internet Proposed Standard, Request for Comments (RFC) 5740. While RFC 5740 specifies a reliable multicast UDP-based transport protocol, NORM also has the ability to provide loss tolerant UDP unicast reliable transport that is well-suited for lossy wireless environments where protocols like TCP may not perform well. In addition, we introduce the design and use of a unicast SOCKS proxy for the NORM protocol called the NACK-Oriented Proxy (NORP) and validate the results of NORP performance against other NORM unicast experimental results. The availability of NORP will reduce the transition risk and improve the use of NORM unicast capabilities by not requiring end system application modifications.**

Our results examine a number of unicast congestion control variants of the NORM protocol, including an explicit congestion notification (ECN) mode, within an emulated lossy link modeled network environment. We include COTS hardware routing and radio-to-router interface (R2RI) within the test environment to examine and validate the use and performance of ECN-based packet marking and improved cross-layer flow control when available. The initial experiments demonstrate the significant advantage of using ECN-based assistance within error-prone wireless environments. We also show increased performance of an additional loss tolerant mode that does not require ECN infrastructure support. Coexistent transport fairness results show that NORM unicast modes are friendly to both inter-protocol flows (e.g., TCP flows coexisting) and to intra-protocol flows (e.g., other NORM flows coexisting). We conclude by summarizing findings and discussing future work planned.

I. Introduction

The NACK-Oriented Reliable Multicast (NORM) transport protocol documented as Internet Proposed Standard, Request for Comments (RFC) 5740, in [1] has a set of reliability, congestion control, and flow control features that are separable and somewhat independent components as documented also in [2]. Designed initially as a reliable multicast UDP-based transport protocol, NORM also has the ability to provide reliable UDP unicast transport and its robust performance and behaviors are well-suited for wireless environments where protocols like TCP may not perform well. The NRL NORM software reference prototype [3] also provides extended real-time transport features (e.g. quasi-reliable, realtime streaming) that may be needed for various types of tactical information exchange. Recently, NORM is being considered for use in a variety of challenging unicast applications and this paper addresses initial evaluations and modes of operation for consideration in NORM unicast deployment scenarios. In addition, a NORM Socket Secure (SOCKS) proxy [4] called the NACK-Oriented Proxy (NORP) [5] has been developed, allowing NORM to act as a reliable transport proxy for a single or set of TCP end-to-end application flows. NORP will allow NORM unicast reliable transport capabilities to be considered for deployment and transition without directly modifying a large set of existing end system applications.

Concepts like explicit congestion notification (ECN) [6] based rate control behavior allow NORM to perform dynamic congestion control without having to rely solely on packet queueing loss events. This has additional wireless benefits due to the fact that stochastic wireless channel errors may otherwise be mistaken for router congestion indication events. The initial ideas for NORM TCP-friendly congestion control (NORM-CC) and initial simulation results demonstrating the potential gains in using ECN as a wireless transport enhancement feature were first presented in [7]. More recently, NORM ECN features and congestion control modes, including enhanced unicast capabilities, have been matured and initially discussed in [2]. This paper also presents the use of commercial router hardware and radio-to-router interface (R2RI) software to improve flow control within the testbed.

This paper includes experimental studies of various unicast-based NORM reliability and congestion control modes. This includes ECN-centric congestion control, NORM Congestion Control ECN (NORM-CCE), and a wireless loss tolerant mode, NORM Congestion Control Loss tolerant (NORM-CCL), that was recently developed for use when ECN features may not be available or trustworthy. The paper is organized as follows: the experimentation system is described, an overview of the NORM congestion control modes is provided, the NORP system design is introduced, results from experimentation are presented, and then future issues and summary observations are outlined.

II. Relationship to Past Work

Initial *ns2* simulation studies of NORM congestion control modes were provided in [2] and a basic bandwidth utilization vs. loss curve showing goodput vs. bit error rate for TCP, UDP, NORM-CC, and NORM-CCE (included in Fig. 3) was presented. Goodput is defined as the application-level throughput or the useful data in packets delivered by the network to a destination per unit of time and excludes retransmissions,

978-1-4799-6770-4 2014

DOI 10.1109/MILCOM.2014.165

packet overhead, and redundant data packets received. In terms of transport system goodput, the key simulation results from [2] are summarized as follows: NORM-CC closely tracks TCP behavior as loss statistics increase. NORM-CCE performs significantly better under lossy conditions, performing almost as well in terms of bandwidth utilization as unreliable UDP without congestion control. This past simulation work demonstrated the potential for utilizing the available bandwidth more effectively while maintaining reliability under lossy conditions with enhancements such as ECN-based congestion control assistance.

III. Overview of Experimentation System

One of the goals of this work is to validate previous NORM congestion control simulation results and performance trends using actual commercial off-the-shelf (COTS) routing equipment and end system software. To accomplish this, we require router software with relevant features such as ECN packet marking and related router queue management. We also require a means to model and control wireless link and loss characteristics. In our experimentation setup, the wireless link model component uses the Extendable Mobile Ad-hoc Network Emulator (EMANE) [8] to provide programmatic control of link characteristics such as packet loss, bandwidth limitation, and delay. We also require an R2RI as mentioned in Section I to support flow control between the EMANE radio layer and the router.

The experimental testbed (Fig. 1) consists of two pairs of Linux end systems, two Cisco 3825 routers, and another Linux system running EMANE as a link emulator with a basic wireless link module, RF Pipe, that allows us to control characteristics such as bandwidth, loss, and delay between the routers. This testbed configuration was chosen to match the simulation conditions in [2] (1 Mbps bandwidth, 242 ms delay, uniform packet loss). We point out that the purpose of these tests was not to evaluate dynamic routing scenarios, but rather to test initial concepts of wireless-friendly congestion control and reliability using NORM alongside commercial routing hardware and basic emulated radio link models.

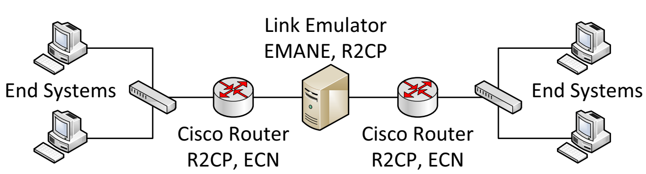

Fig. 1. Testbed diagram

The Cisco routers were configured with static routing, random early drop (RED) ECN queueing, and internal switch modules (shown separately in Fig. 1). For EMANE, we configured an RF Pipe MAC layer with an integrated R2RI that connects to Radio-Router Control Protocol (R2CP) running on the Cisco routers. The R2RI is used to set the interface speed dynamically, preserving flow control between the routers and enabling the routers to effectively mark ECN conditions to assist in distinguishing dynamic wireless loss from congestion-based router loss conditions. Data Link Exchange Protocol (DLEP), a newer protocol being considered for R2RI, was still under development at the time of these experiments and was not yet supported by the Cisco 3825 routers we used, but a similar flow control mechanism to that used is expected to be supported in this newer evolving standard. For RED ECN queueing on the Cisco routers, our goal was to respond to wireless network changes as quickly as possible. To do this, the exponential weight constant was set to 1, the minimum queue threshold to 1, the maximum queue threshold to 2, and the marking probability denominator to 1. There may be more optimum values for these settings, but no effort was made to investigate these issues in this initial study.

IV. End-to-end Protocol Tests

A. TCP and TCP-ECN

TCP-based congestion avoidance is the primary congestion control algorithm used within the Internet [9–11]. For TCP, we used the default TCP implementation in the Linux 3.6.11 kernel. This implementation uses the CUBIC congestion avoidance algorithm [12]. We did not evaluate alternative TCP congestion avoidance algorithms or implementations, although there is extensive literature examining variations of TCP congestion avoidance and fairness issues [13, 14]. For ECN-enhanced TCP, we utilized the same Linux kernel CUBIC implementation and used system settings to enable TCP ECN support.

B. NORM-CC: TCP-friendly congestion control

By design, NORM-CC mimics the behavior of TCP throughput vs. congestion events via a rate-based congestion control fundamentally developed from concepts presented in the seminal paper [15]. NORM-CC is not designed to be optimal over wireless links, but was important to develop for RFC 5740 Internet standardization, where the goal was for NORM reliable multicast transport to be able to coexist with TCP flows and do little harm in terms of congestion control fairness. NORM-CC flow should impact other TCP flows only as much as an additional TCP flow would, and it is in many ways less aggressive than TCP because it is additionally designed for careful scaling and use within multicast flow scenarios. NORM-CC reduces end-to-end deployment risk given a wide range of wired deployment scenarios and considerations, but this is not a desired behavior for many error-prone tactical wireless deployments where default TCP congestion control behavior has been shown to be ineffective [2].

C. NORM-CCE: ECN-enhanced congestion control

NORM-CCE provides an ECN congestion control component for NORM end-to-end operations. The basic design is largely motivated by the potential gains in wireless channel use, where distinguishing wireless channel loss from router queue congestion becomes a critical performance concern [7]. In lossy wireless network channel environments, available bandwidth often goes underutilized when end-to-end congestion control algorithms mistake channel packet loss for network congestion events. NORM-CCE relies heavily on ECN markings to determine congestion events and packet loss is ignored as a first-order indication of congestion. This improves the determination of whether congestion is actually imminent or whether there is normal statistical channel loss events occurring in the wireless network. An interesting side effect of ECN use is that end-to-end congestion control can be performed largely without router queue packet drops actually occurring, since ECN marking are triggered early via the router queueing threshold. Extreme packet loss (e.g. link disconnection) degrades the sender/receiver feedback loop and the NORM protocol will decrease its rate to avoid pathological behavior.

D. NORM-CCL: Loss tolerant congestion control

NORM-CCL provides an end-to-end congestion control capability designed to work better than NORM-CC on wireless links, without requiring the ECN infrastructure capability needed by NORM-CCE. The distinction of NORM-CCL from the TCP-friendly NORM-CC is in how the receiver(s) interpret packet loss as a congestion indication. With TCP-friendly NORM-CC, any sender packet loss within a window of one round-trip time (RTT) is considered a congestion event and the NORM receiver updates its loss/congestion event estimate that is fed back to the sender accordingly. With NORM-CCL, a single, isolated packet loss within one RTT is assumed to be a non-congestion related loss and is ignored for purposes of congestion event estimation. If multiple sender packets are lost within an RTT, then it is identified as a congestion event. NORM-CCL allows for some congestion control performance improvement for networks with modest (and non-bursty) bit error rate links.

E. MGEN

To provide test traffic for our experiments, we use the well-known Multi-Generator (MGEN) network traffic test tool [16]. MGEN supports both UDP and TCP test traffic scenarios and is used to conduct end-to-end TCP and NORM traffic testing with logging of received data packets for use in goodput measurements.

For all emulation tests, 1400 byte MGEN IPv4 packets (effective user/application data size) were used to prevent fragmentation. With various protocol headers (not including a 14 byte MAC header for each packet type), this resulted in 1428 byte UDP packets, 1468 byte NORM/NORP packets (using 1400 byte segment size), and 1492 byte TCP packets. It should be noted that while MGEN created TCP messages of 1400 bytes, the underlying kernel TCP implementation chose to create packets with a 1440 byte payload, thereby including an additional 40 bytes of the next MGEN message in each of its packets. This means that for every 36 MGEN TCP messages, only 35 TCP data packets were required.

MGEN logs were used to calculate resultant system goodput values and our current congestion and rate control techniques were evaluated in terms of steady state performance characteristics by eliminating slow start data periods at the beginning of each recorded test.

V. NORP Design and Issues

Deploying protocols such as NORM for unicast application purposes is often met with a variety of deployment challenges, including the cost and difficulty of updating application software. The NRL NORM reference prototype [3] includes a flexible application programming interface (API) allowing programmers to adapt NORM behaviors and operations to numerous challenging conditions including bulk delivery, messaging, and real-time streaming applications. The API use and integration is sometimes the best engineering choice, but requires additional design and application integration investment. The ubiquitous deployment of TCP and UDP application designs often makes application redesign or network extensions expensive or undesirable. This challenge can be partially addressed via the standardized transport proxy design presented here.

We developed the NACK-Oriented Proxy (NORP) to support simplified proxying of existing TCP-based (and potentially UDP-based) applications. NORP uses the SOCKS5 protocol [4] to intercept TCP connections and mediate them to remote destinations using the NORM protocol. The SOCKS5 capability is available on a large set of end host systems and requires no modification to the actual end system software applications. The NORP software daemon allows for multiple deployment models including, in its simplest form, a small software installation on the client and server platforms for which NORM transport enhancement is desired. An example of a traditional SOCKS-enhanced deployment across a network to support NORM-enabled end system connection proxying is shown in Fig. 2. The NORP daemon proxies system application sockets initiated through a standard SOCKS client path. The NORP daemon also proxies and redirects NORP SOCKS sessions initiated from remote machines. As shown via the dotted connection in Fig. 2, if an end system is detected as not supporting the capability, then TCP and UDP connections proceed as normal. Once deployed and configured, this is a fairly turnkey solution and supports interoperability with end systems that are not NORP-enabled. Other NORP deployment options are also possible, including a “bump-in-the-wire” gateway or performance-enhancing proxy middleware approaches. No special configuration is required for the baseline NORP software-only deployment model and full backwards compatibility with existing TCP applications and services is maintained. This opens up a set of transition opportunities lowering program risk on engineering development, planning, procurement, and coordination in future systems. NORP also provides cutting-edge features such as NORM-CCE and NORM-CCL, allowing a degree of network awareness in a standard low complexity way to achieve increased wireless network resilience.

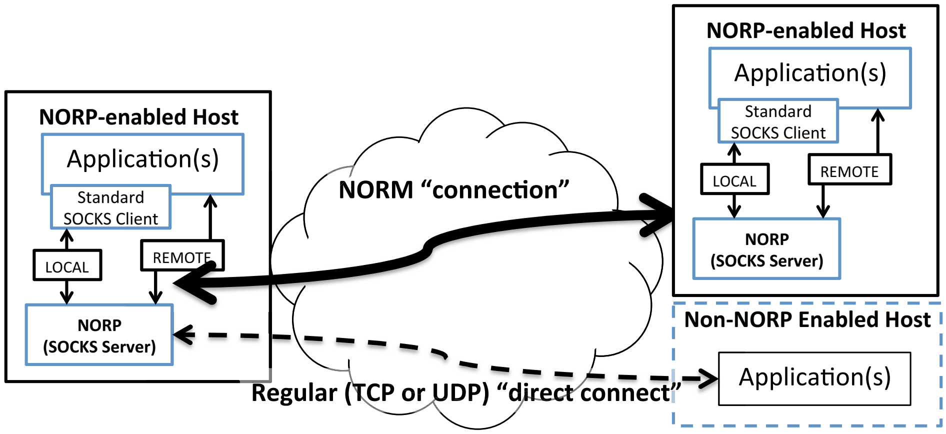

Fig. 2. NORP network connection proxying

VI. Experimental Results

A. Simulation of NORM-CCL

The original *ns2* simulation performance curves in [2] that we wish to roughly validate with actual software and hardware components are reproduced in Fig. 3, along with an additional simulation curve for NORM-CCL that was not presented in that previous publication. The simulation environment and settings were the same as in [2]: 1 Mbps link, 242 ms one-way delay, 1250 byte packets. The results show that NORM-CCL provides more steady state goodput than TCP and NORM-CC across all bit error rates tested. As expected, NORM-CCL does not perform as well as NORM-CCE, but it provides a moderate degree of loss tolerant improvement without requiring intermediate router ECN support.

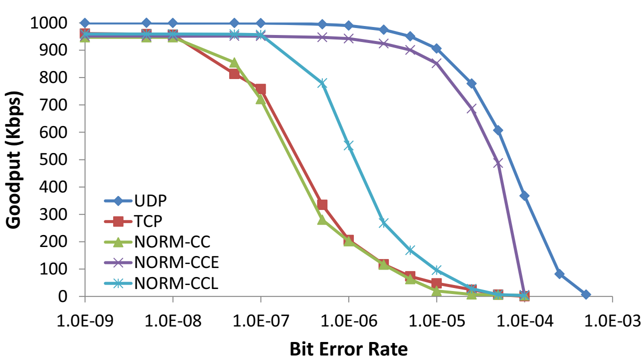

Fig. 3. *ns2* simulation results with NORM-CCL

B. Validation of Previous Simulation Results

The same simulation experiments represented by Fig. 3 were repeated in emulation using settings as close to simulation as possible. The only differences are packet sizes as noted in Section IV-E, and the use of packet error rate rather than bit error rate due to the particular model used for link loss emulation. As in simulation, Fig. 4 shows that NORM-CC performs slightly better than TCP at low loss rates (<0.1%), and slightly worse under more lossy conditions, but roughly tracks TCP's basic loss vs. goodput performance trend. NORM-CCE again closely tracks the unreliable UDP throughput, providing a significant amount of bandwidth utilization improvement as packet loss rate increased. As in simulation, NORM-CCL provides better goodput vs. loss than both TCP and NORM-CC, but cannot match the high utilization performance of NORM-CCE using explicit congestion information. In addition to the above protocols, ECN-enhanced TCP (TCP-ECN) is also plotted in this graph, but it has significantly less goodput than regular TCP in a low-loss environment, and is almost identical to TCP at higher loss rates. TCP-ECN was not included in further tests due to this issue. We predict the TCP-ECN behavior in lossy environments is due to the fact that TCP-ECN only used ECN as an early warning mechanism, rather than more aggressively as a means to help distinguish wireless channel loss from congestion loss as NORM-CCE does.

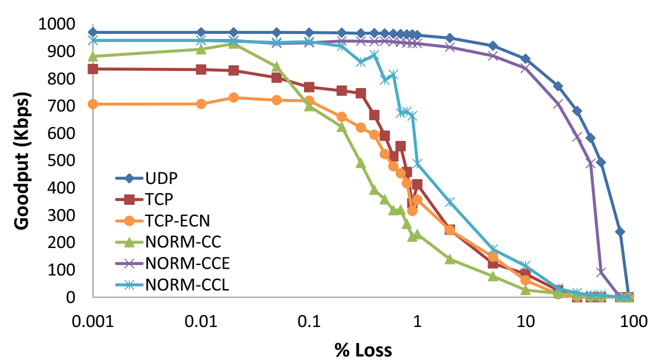

Fig. 4. NORM validation tests

In order to also test the performance of the NORP SOCKS proxy application previously introduced, the results are presented both using *norm* and the *norp* proxy software. As shown in Fig. 5, NORM and NORP values closely track each other for all congestion control methods, so our results further validate the use and performance prediction of NORP-supported connections in these scenarios. All further tests were conducted using the native NORM application.

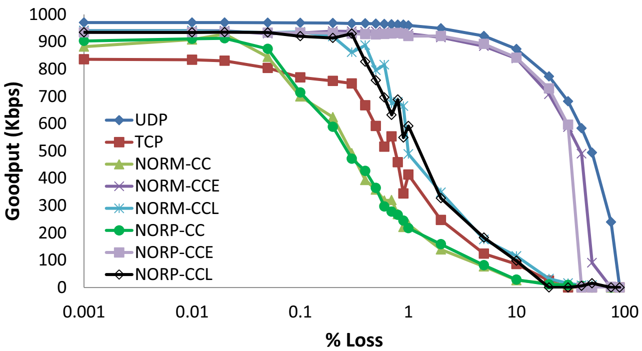

Fig. 5. NORP validation tests

C. Multiple Flow Utilization Results

In addition to single flow steady state congestion control performance under lossy conditions, we also conducted a series of basic interflow utilization and fairness experiments to examine the variety of congestion control modes presented. In this series of experiments, we used the identical experimental setup described and increased the number of end-to-end transport flows from one to five for each congestion control mode, all in the same direction across a single bottleneck network link. We also used a second pair of end systems with an additional five flows to send up to 10 flows through the network in the same direction.

TABLE I
AGGREGATE GOODPUT WITH MULTIPLE FLOWS

		Goodput (Kbps)		
Protocol	Loss %	1 Flow	5 Flows	10 Flows
	0.1	770	827	921
TCP	1	413	774	902
	10	86	327	603
	0.1	700	915	923
NORM-CC	1	231	875	907
	10	27	143	247
	0.1	932	940	937
NORM-CCE	1	929	934	925
	10	839	820	833
	0.1	936	936	937
NORM-CCL	1	489	918	933
	10	115	502	780

Table I summarizes the results from this experiment. The introduction of multiple flows allows TCP, NORM-CC, and NORM-CCL to recapture, in aggregate, some of the underutilized link bandwidth vs. single flow conditions. We note that these utilization results will be dependent upon various scenario variables including loss, packet duplicates, and delay. A key observation is that NORM-CCE was able to achieve high goodput utilization with a single flow and maintained consistent behavior as additional flows were added to the system.

Table II uses the Jain's fairness index [17] as an intra-protocol throughput equality measure. The Jain's fairness value ranges between [0:1], where higher values indicate more equal sharing between flows. We present results for each of the protocols in steady state with 10 concurrent flows at various loss values. The results show that all protocols are relatively fair to other similar flows at steady state.

TABLE II
JAIN'S FAIRNESS INDEX (10 FLOWS)

Protocol	Lossless	0.1% Loss	1% Loss	10% Loss
TCP	0.9862	0.9918	0.9934	0.9785
NORM-CC	0.9850	0.9817	0.9793	0.9647
NORM-CCE	0.9908	0.9912	0.9877	0.9856
NORM-CCL	0.9912	0.9949	0.9870	0.9939

D. Multiple Flow Fairness Results

In order to test the performance of concurrent TCP and NORM flows, we ran up to 5 TCP flows between one pair of machines, and up to 5 NORM flows between a different pair of machines in the same direction across the testbed router infrastructure. For interflow comparison, 10 TCP flows were also run, split evenly between the end system pairs. The results in Fig. 6 show that TCP is able to get higher aggregate goodput when run concurrently with NORM-CC than with itself, so NORM-CC is indeed less aggressive than CUBIC TCP in this case. NORM-CCL and NORM-CCE are both shown to have minimal impact on concurrent TCP flows. The only exception is the high loss case, where NORM-CCE has a somewhat more significant impact on concurrent TCP performance, but this exception only emerged in aggregate with multiple flows of each protocol. With a single flow, TCP achieved more goodput when run with NORM-CCE than with a second TCP flow.

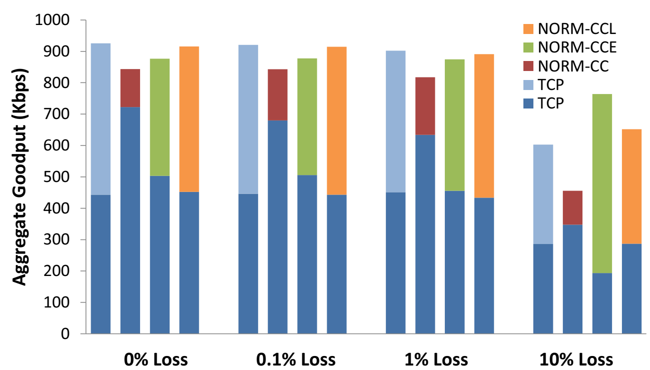

Fig. 6. Fairness of TCP (blue) to itself vs. NORM

VII. FUTURE WORK

The main purpose of our paper was to validate previous simulation performance curves at steady state for a variety of NORM-based congestion control modes and to introduce the investigation of a loss tolerant variant not requiring ECN functionality. Our initial experimentation was limited to a basic bottleneck scenario and future work is planned to examine more heterogeneous topologies including multiple cross links, dynamic bandwidth, bursty loss conditions, and mobility.

Future work planned for NORM-CCL will include investigation of more sophisticated heuristics, including consideration of delay variation, to improve the identification of congestion events. The flexibility of the rate-based NORM congestion control protocol allows for easy exploration of different options. We also plan to investigate how well NORM-CCE works on networks where there may be only partial support for ECN, and whether a hybrid mode of operation that might combine ECN and loss tolerant modes is of benefit. There are a number of components in our testbed that we also plan to further examine including router queue settings (e.g., RED ECN) to further observe the impact on various results in this initial study.

VIII. Conclusions

In this paper, we presented new experimental results examining various unicast-based congestion control behaviors of the NORM protocol. While NORM is specific in RFC 5740 as a reliable multicast UDP-based transport protocol, we demonstrated that NORM provides effective UDP unicast reliable transport well-suited for lossy wireless environments. We presented the design and use of a unicast SOCKS proxy for the NORM protocol called the NACK-Oriented Proxy (NORP) that is useful in reducing the cost and risk of deploying NORM-based loss tolerant unicast connections without requiring end system application modifications. Our experimental test setup involved off-the-shelf Cisco router systems with ECN-enabled packet marking and also included an existing radio-to-router interface (R2RI) capability that supported dynamic flow control between the radio and router layers.

We measured the performance of a variety of congestion control modes of the NORM protocol alongside TCP behaviors. For lossy wireless environments supporting a single unicast flow, we demonstrate that ECN-enhanced NORM (NORM-CCE) is able to utilize a significant percentage of available wireless bandwidth while more classic end-to-end congestion control modes leave the channel severely underutilized. We also show moderate performance gains with an additional loss tolerant mode (NORM-CCL) that does not require ECN infrastructure support, but is rather less aggressive in declaring loss events. Multiple flow fairness test results show that NORM unicast modes are friendly to both inter-protocol flows (e.g., TCP and NORM flows coexisting) and to intra-protocol flows (e.g., NORM flows coexisting) although the NORM-CCE mode may be slightly more aggressive than TCP under certain scenarios.

In conclusion, we roughly validated earlier simulation results that examined congestion control bandwidth utilization and demonstrated potential benefits of both NORM-CCE and NORM-CCL for unicast operations. These results were obtained using several off-the-shelf components to provide ECN-capable router marking and cross-layer radio-to-router flow control. The ability of the NORP SOCKS proxy to support these congestion control modes for existing TCP (and UDP) application flows provides a potential way forward to deploy these capabilities while minimizing the need for system redesign or development. As an aggregate unicast flow proxy for troublesome wireless scenarios, NORP can potentially regain a significant portion of underutilized steady state bandwidth.

References

[1] B. Adamson, C. Bormann, M. Handley, and J. Macker, "RFC 5740: NACK-oriented reliable multicast (NORM) transport protocol," *IETF*, 2009.

[2] B. Adamson and J. P. Macker, "Reliable messaging for tactical group communication," in *Military Communications Conference Proceedings, IEEE*, 2010, pp. 1899–1904.

[3] NRL. (2014, August) NACK-oriented reliable multicast (NORM). [Online]. Available: http://www.nrl.navy.mil/itd/ncs/products/norm

[4] M. Leech, "RFC 1928: SOCKS protocol version 5," *IETF*, 1996.

[5] NRL. (2014, August) NACK-oriented proxy (NORP). [Online]. Available: http://www.nrl.navy.mil/itd/ncs/products/norp

[6] K. Ramakrishnan, S. Floyd, and D. Black, "RFC 3168: The addition of explicit congestion notification (ECN) to IP," *IETF*, 2001.

[7] J. P. Macker and R. B. Adamson, "The multicast dissemination protocol (MDP) toolkit," in *Military Communications Conference Proceedings, IEEE*, 1999, pp. 626–630.

[8] NRL. (2014, August) Extendable mobile ad-hoc network emulator (EMANE). [Online]. Available: http://www.nrl.navy.mil/itd/ncs/products/emane

[9] V. Jacobson, "Congestion avoidance and control," in *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4. ACM, 1988, pp. 314–329.

[10] W. Stevens, "RFC 2001: TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," *IETF*, 1997.

[11] M. Allman, V. Paxson, and E. Blanton, "RFC 5681: TCP congestion control," *IETF*, 2009.

[12] S. Ha, I. Rhee, and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, 2008.

[13] J. Widmer, R. Denda, and M. Mauve, "A survey on TCP-friendly congestion control," *Network, IEEE*, vol. 15, no. 3, pp. 28–37, 2001.

[14] G. Hasegawa and M. Murata, "Survey on fairness issues in TCP congestion control mechanisms," *IEICE Transactions on Communications*, vol. 84, no. 6, pp. 1461–1472, 2001.

[15] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4. ACM, 1998, pp. 303–314.

[16] NRL. (2014, August) Multi-generator (MGEN). [Online]. Available: http://www.nrl.navy.mil/itd/ncs/products/mgen

[17] R. Jain, D.-M. Chiu, and W. R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation, 1984.