A TCP Friendly, Rate-Based Mechanism for Nack-Oriented Reliable Multicast Congestion Control

Joseph P. Macker and R. Brian Adamson
Information Technology Division, Naval Research Laboratory

Abstract-**In this paper, we describe ongoing work in adding congestion control extensions to an existing negative acknowledgement (NACK) oriented reliable multicast protocol. Our previous work adopted and used the concept of a dynamic worst path representative (WPR) receiver for equation-based rate adaptation at the multicast source. We have further refined this approach. We present a design overview and simulation results of these extensions within a working reliable multicast protocol (mdp-cc). Our analysis of interflow fairness with steady state TCP unicast sessions demonstrates friendly behavior across a set of scenarios. Testing with more dynamic TCP flows and more complex topologies demonstrates the WPR approach adapts well to changing congestion conditions.**

I. Introduction

The successful day-to-day operation and proliferation of Internet Protocol (IP) technology worldwide has been in a large part due to the existence and wide scale use of a standardized, reliable unicast transport protocol (i.e., Transport Control Protocol (TCP)). In addition to reliable data transport mechanisms, TCP also provides effective, end-to-end congestion control mechanisms [1,2]. At present, reliable multicast transport mechanisms lack such "best practice" approaches to end-to-end congestion control. Effectively addressing congestion control issues remains a key requirement for widespread Internet deployment of reliable multicast (RM) solutions and applications. Another specific concern for the Internet community, at large, is the impact RM traffic has on other coexistent Internet traffic (particularly TCP flows) during times of congestion [3]. There are several important classes of RM protocols and applications and there is no “one size fits all” solution to the set of problems across all of these design spaces. Our particular work described in this document targets congestion control mechanisms for NACK-oriented reliable multicast (NORM) protocols, but some of the techniques can viewed as independent of specific reliability mechanisms and may be more generally applicable.

II. Approach and Previous Work

In recent years, research on equation-based TCP throughput models [4] has indicated that low complexity, steady state model(s) existed for predicting TCP behavior. Also, previous work exploring TCP fairness definitions and methods for applying models to multicast situations [5] outlined issues in applying fairness models to multicast transport. The TCP worst path fairness model is based on having equation-based TCP throughput estimates for all source-receiver paths in a multicast session. By adapting the source rate to the worst TCP predicted path rate amongst the receiver group, a fairness bound on other paths is assumed. Also other work has explored the concept of using a subset of the receiver group, termed *representatives* to provide more rapid feedback for congestion control and scalability purposes [6,7]. This concept provides merit by trading off the need for rapid feedback for congestion control purposes against the need to preserve protocol efficiency and scalability. We take a different approach to electing and applying the concept of representative feedback than in this previous work. We react only to a single representative at any one time for rate-based control purposes. This is similar to recent work on pgmcc that adopts a dynamically elected single ACKer concept for end-to-end TCP friendly window-based control [8]. In addition, our approach elicits timely feedback from multiple candidate representatives simultaneously and has provisions for improved candidate election from the group at large.

In past years, work and discussions within the Internet Research Task Force (IRTF) Reliable Multicast Research Group (RMRG) group significantly contributed to establishing research goals and ideas for applying fairness models and equation-based approaches to multicast flows [9]. Other results demonstrating a fairness and equation-based congestion control model for unicast have been recently published in [10]. Our preliminary work in applying rate-based TCP friendly congestion control and congestion control representatives to NORM protocols was previously presented at the June 1999 RMRG meeting in Pisa [11] and was also documented briefly in [12,13]. That previous work described a novel approach to use path loss and round trip time (RTT) estimates collected at the source to dynamically elect one worst path representative (WPR) amongst the receiver set. This elected receiver provides a rapid feedback control loop for rate-based congestion control and avoided the well-known drop-to-zero problem caused by aggregating multiple uncorrelated feedback sources. Dynamic WPR election is still required to track dynamic congestion conditions. In addition to the WPR, we maintain rapid control loop state on a small number of additional candidate receiver paths, but only the WPR feedback is used for rate adaptation. In this paper, we expand on that previous work and describe more recent design refinements, experiments, and TCP friendliness results.

III. Design Challenges

A key design tradeoff in applying congestion control mechanisms to NORM style protocols is balancing the inherent reduced group feedback mechanisms against the increased need for timely and accurate receiver feedback for dynamic congestion control. Another challenge is in the application of the equation-based TCP model. While this model gives us a steady state target rate to achieve TCP fairness, it does not tell us how to effectively collect metrics or how to integrate and achieve such a design within a dynamic multicast protocol framework. The typical NORM protocol feature set makes this more challenging due to the typical infrequent NACKing and feedback suppression mechanisms. NORM protocols also typically adapt forms of forward error correction (FEC) based packet repairing technique to replace or enhance explicit packet retransmission schemes [15] and this tends to further reduce receiver feedback. These inherent NORM design mechanisms compete against congestion control needs. We review our approach in addressing these challenges below.

IV. DESIGN APPROACH

In targeting TCP friendliness behavior end-to-end, we choose to apply a worst path fairness model [5]. As mentioned, the worst path TCP fairness model requires that only the minimum of the equation-based TCP throughputs across the set of receiver paths be used as a target rate goal. To effectively adapt to dynamics in congestion, timely feedback should be provided on the source-receiver path(s) of interest. Maintaining rapid feedback of congestion control metrics for all paths within a scaled multicast session can be prohibitive to efficient operation of the protocol. In our approach, we hypothesize that maintaining more timely feedback state for a dynamically elected subset of source-receiver paths and only reacting to the worst path receiver amongst this group provides a reasonable compromise to the set of competing feedback and dynamic election requirements.

To investigate our protocol ideas in practice, we implemented extensions to the existing Multicast Dissemination Protocol (MDP) software [14]. We refer to the extended congestion control version as *mdp-cc*. Many of the mdp-cc extensions can be generalized to other NORM protocols and the techniques can be adopted outside of the MDP protocol framework. Using MDP as a research framework provided some advantages in development and experimentation. First, the framework is a well-tested open source implementation of an end-to-end, rate-controlled NACK-based protocol with all the typical esoteric features used for improved scalability (e.g., NACK suppression, FEC repairing, etc). Second, existing work provided us with a detailed protocol simulation model already embedded within the ns2 framework to evaluate various protocol components and TCP fairness issues.

A. Core Design Components

The mdp-cc design extension can be broken down into four principal areas.

1) Receiver loss fraction measurement and collection
2) Source-receiver path RTT measurement and collection
3) Congestion control representative selection and timely feedback mechanisms
4) Source transmission rate adjustment algorithm

To predict the expected source-receiver path TCP throughput from the equation model we adopted [4], we require a loss estimation input for the receiver path in question. Each MDP receiver maintains a running estimate of the current loss event fraction from the multicast source. The loss event fraction corresponds to the inverse of the average interval (in terms of a packet count) between loss events. A loss event is distinct from a raw packet loss in that multiple, individual packet losses occurring within in one RTT "window" of packets are counted as only a single loss event. This loss event definition is consistent with the definition used in the equation-based TFRC work [5]. Whenever a receiver provides any form of feedback (e.g., NACK) to a source, the receiver provides its current loss estimate for that source as part of the feedback message. The source uses this estimate, along with other information, to feed into the equation-based TCP throughput predictors. To facilitate maintenance of a loss fraction estimate, all source packets include a monotonically increasing sequence number that receivers use to measure packet loss. The mdp-cc implementation also keeps track of packets arriving out-of-order and delays counting losses until the possibility of an out-of-order arrival is reasonably reduced. The delay depth for out-of-order packet tracking is dynamically updated when out-of-order packets arrive. The effectiveness of this technique in networks (e.g. mobile wireless) where out-of-order arrivals may be more common and its subsequent effect on congestion control operation (including impact on TCP-fairness) is a subject of future investigation.

The mdp-cc protocol implementation currently includes two different algorithms for loss event estimation. The first of these is a technique similar to the Average Loss Interval (ALI) with discounted, weighted history similar to that described previously in [10]. The other technique is an adaptively smoothed exponentially weighted moving average (EWMA) of the loss event interval developed from previous work on mdp-cc [11,13]. In preliminary evaluation, both techniques produce similar estimates. The performance and complexity trade-offs of these two techniques are still being examined, but both are included within the present implementation allowing cross comparison and tradeoff analysis.

In MDP, without congestion control, the source is responsible for collecting RTT measurements from receivers to determine both NACK suppression and repair cycle timing based upon the greatest observed RTT. In mdp-cc, RTT information is needed by the source as part of the congestion control algorithm to calculate the TCP throughput estimate for different receivers. For general protocol operation, receivers provide the opportunity for the source to collect path RTT measurements when they transmit NACK messages. For reliability purposes alone, this technique seems sufficient, but we feel a complete rate-based congestion control solution requires consideration of additional issues. While it may be sufficient to receive feedback only from NACKing receivers in many scenarios, an additional mechanism is deemed useful for protocol startup conditions and for operation among more heterogeneous source-receiver paths.

Within the experimental mdp-cc implementation, the source uses a long-term feedback mechanism to excite explicit feedback responses in addition to general NACK and the rapid congestion control representative collection processes. The long-term feedback excited from the group provides an opportunity for receivers not NACKing (or whose NACKs are suppressed) to provide feedback and to actively participate in the congestion control process. The mechanism for collecting this long-term feedback from the group at large is described later.

From the loss event estimates and RTT measurements gathered from received feedback, the source calculates the estimated steady state throughput rate predicted by the analytical model of TCP for individual receivers. The source keeps a list with state for a small set of receivers with the lowest predicted TCP throughput rates. This list is dynamically updated as feedback is progressively received from the group. Members of this list are termed congestion control representatives (CCR). CCRs are considered the expected candidates for worst path TCP fairness and are rapidly probed by the source for continued loss estimate and RTT measurement updates. The source uses the feedback from CCRs and the previous methods described to find the receiver with the minimum transmission rate predicted by the TCP throughput model. This receiver is selected as the WPR and its feedback is subsequently used for rate-based control until a different WPR is selected. The source maintains smoothed RTT estimates for the CCRs and tracks RTT variation for

calculation of the retransmission timeout value (T0) used in the analytical TCP throughput model.

The source uses a control message to excite responses from the current CCR set. This message contains a list of the current CCRs along with their respective RTT measurements. This allows a CCR to properly filter detected packet losses as loss events (i.e. counting multiple packet losses within one RTT as a single loss event). The source also advertises its current transmission rate and the current representative set metrics (e.g., CCR RTT estimates) for more accurate CCR loss event estimation. The message also contains a flag to mark when feedback is expected from non-CCR receivers. A field dictating the random backoff time window for receivers to respond to this *wildcard* probe is also included in the message. The non-CCR receivers backoff their response with a uniform random distribution and the period of the *wildcard* probing also corresponds to this interval. The rate of this non-CCR feedback is a function of the group size and this interval setting. This interval is presently set conservatively, but further work is underway to examine alternative methods for this group-wide feedback.

The current algorithm for selecting the CCR set is very simple. The source keeps state for a small number (currently 5) of receivers with the lowest transmission rate predicted by the TCP analytical model. The potential that receivers lying behind a common bottleneck link within the network may monopolize the current representative set somewhat defeats the role of rapidly exciting congestion control feedback from multiple receivers simultaneously. However, NACK suppression may help reduce this potential. Additionally, algorithms to dynamically populate the CCR list with receivers with uncorrelated metric sets (e.g. RTT, loss event fraction, loss patterns) are being considered. If such approaches can be refined, they would help the source select the most significant, heterogeneous paths to monitor with rapid feedback.

In addition to congestion control feedback collection, the mdp-cc source needs to have a method for adjusting the multicast transmission rate to that which provides TCP friendliness. The source uses the rate predicted from the present WPR (single representative) to establish a goal rate for transmission. The source begins adjusting its rate towards this goal rate. A rule-based approach is used to adjust the rate and this approach includes techniques for dealing missing expected feedback from the representative set and receivers leaving the group. As noted previously, the source presently transmits congestion control probes to the group at a rate of once per RTT of the current WPR, denoted as WPR_RTT. At startup, before any receivers have responded, the source transmits "wildcard" probes that are acknowledged by the group at large within a distributed random backoff time. Unlike unicast transport, multicast startup issues tend to involve a consideration for longer delay in order to capture heterogeneous group effects and to preserve efficient protocol overhead characteristics.

Some of the present rules used for rate adjustment in mdp-cc are as follows. If the source receives a response from the WPR in a timely fashion (within 2*WPR_RTT of the time the corresponding probe was sent) and its current transmission rate is less than the predicted WPR rate, the source increases its rate quickly (exponentially) towards the goal WPR rate. If a response is received later than 2*WPR_RTT after the corresponding probe was sent, no rate adjustment occurs. If the predicted WPR throughput rate is less than the current source transmission rate, the source rapidly (exponentially) reduces its rate towards the goal bottleneck rate. Rate decreases occur at the timeout interval at which the congestion control representative probing is done. In addition, the source decreases its transmission rate if no response is received from the current WPR within 4*WPR_RTT of the probe transmission time. The rate is decreased once per WPR_RTT when the response is late. This measure serves as a congestion collapse avoidance measure. However, if a representative fails to respond at all after a large number of probes, the representative is bumped from the list and the next ranking candidate assumes the bottleneck role.

When the CCR list is completely emptied due to lack of response, the protocol quickly reverts to a minimal transmission rate with long term, wildcard probing of the group in preparation for resuming steady-state operation. It is anticipated that the rule-based algorithm for rate adjustment will help maintain stability when network dynamics or measurement uncertainties cause the WPR to flip-flop among more than one CCR candidate. Although the rate adjustment of this approach for multicast is slower in response than TCP, early simulation results show that this approach maintains good long-term steady state interflow fairness with co-existing TCP flows, even when new flows are dynamically initiated and terminated.

Another design factor that should be mentioned is the provision for dynamic packet sizes within the equation-based approach. In the present mdp-cc design, the possibility of varying source packet sizes is accommodated in the TCP throughput equation by keeping an EWMA measurement that used as the nominal packet size for transmitted source packets. This number is provided as input to the TCP throughput equation calculation.

V. Simulation Performance Results

A. Basic TCP Friendliness Trials and Results

To test the TCP friendliness of the approaches outlined here for mdp-cc, we have studied many basic simulation topologies and congestion scenarios to gain insight on steady state behavior of the protocol and to examine interflow fairness. All of our simulations have performed within the ns2 simulation environment using a highly detailed model of mdp-cc. In one set of tests, we have adopted graphing methods used in [10] to help examine rate-based TCP friendliness. The results in Figure 1 represent and analysis of long-term steady state interflow fairness between the TCP-Sack model and mdp-cc. The central value of 1.0 on the y-axis represents the fair normalized average throughput given that there are *n* flows competing on the congested bottleneck. For example, if there are 64 intercompeting flows, as shown in the right part of the graph, the expected steady state throughput per flow is 512kbps/64 = 8kbps. The graph plots the ratio between the observed value for a flow and the expected value, a ratio of 1.0 being ideal. The solid lines are averages for all sample points of a particular flow type. Even under high degrees of statistical multiplexing the interflow fairness trials we have examined seem encouraging and within bounds demonstrating friendliness as the number of flows increases. In additional tests, we have seen that the results using mdp-cc are very comparable to those presented in [10] for unicast friendliness. Even though these initial tests are simple scenarios, it is important to note that this is a fully functioning reliable protocol with NACKing, feedback suppression, FEC repairing, CCR feedback mechanisms and reelection [12].

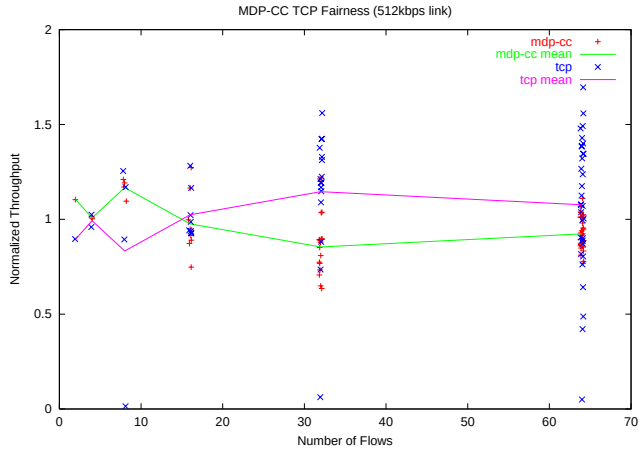

Figure 1: mdp-cc and TCP interflow fairness

B. WPR Switching Tests and Representative Plotting

Additional simulations were constructed to evaluate the operation of mdp-cc in environments with dynamic changes in the worst path rate and location within a multi-bottleneck topology. Figure 2 illustrates an example topology generated by our simulation toolset. In this particular example, a source cluster (Nodes 0-7) populated with a mix of mdp-cc and TCP generator agents sends traffic to five other receiver clusters. Persistent steady-state TCP flows and a single mdp-cc flow are transmitted across the five links feeding the receiver cluster. These five links dynamically play bottleneck roles in the simulation through the starting and stopping of additional TCP flows sharing those links. The simulation toolset is capable of random and/or deterministic generation of additional dynamic TCP flows as needed. We have also performed tests with a variety of TCP models within ns2 and include TCP-Sack results here as an example.

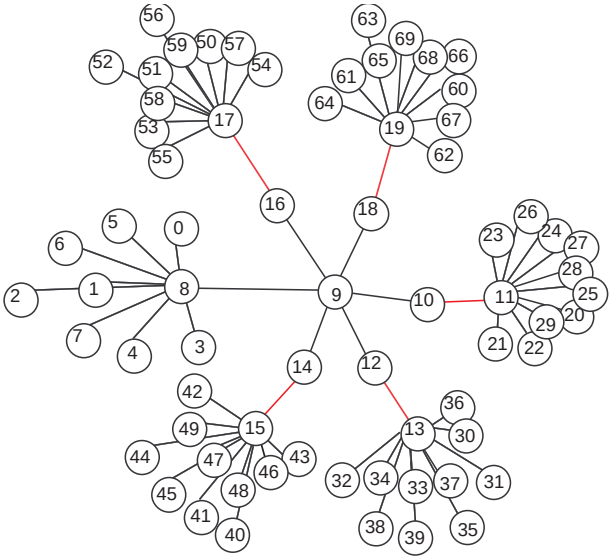

Figure 2: Dynamic Multi-bottleneck Simulation Topology

Figure 3 is a plot of the observed transmission rates of mdp-cc and TCP flows during a simulation using the above topology with 500 kbps receiver cluster feed links. In this simulation, a single TCP flow (in addition to the persistent, background flow per bottleneck link) was added to one of the feed links from time 0 to 400 sec, creating a congestion bottleneck. Then, as that added flow terminated, two TCP flows were added to another feed link from time 400 to 800 sec, creating a different, slightly more severe bottleneck as the mdp-cc flow is forced to share the link with three other TCP flows (one steady-state and the two added flows). Finally, from time 800 to 1200 sec, a single additional TCP flow was placed on yet another feed link, once again changing the worst path within the topology and the congestion control rate.

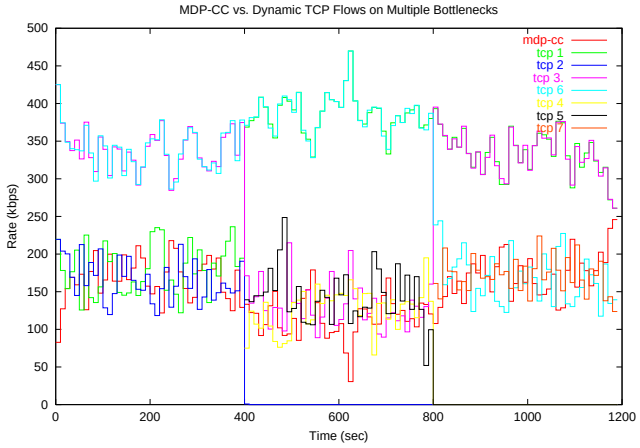

Figure 3: Dynamic TCP, multiple paths, and WPR reaction

As the bottom sets of flows in Figure 3 show, the transmission rate of the mdp-cc flow is friendly with the transmission rates of the competing TCP flows on the dynamically changing worst congestion path. The flow rate plots in the top portion of Figure 3 correspond to TCP flows on the remaining feed links, during the period in which these links are not worst path links. Note that the transmission rate of mdp-cc appropriately adapts in response to changes in worst path location and the dynamic congestion rate. These results are representative of what has been observed in other mdp-cc and TCP friendliness simulations run to date.

To further examine CCR election and WPR selection we have developed tools to plot their behavior. The graph in Figure 4 plots the receiver nodes comprising the source representative set at different points in time during the simulation. The mdp-cc software was configured for maximum set of five representative receivers in its WPR candidate list at one time. Thus, there is a maximum of five parallel points on the plot shown at any one point in time. The simulation topology assigns consecutive node identifiers to receivers within the same topology cluster allowing for easier interpretation of the graph. As this plot shows, the CCR list membership generally includes the location of the congestion bottleneck throughout the course of the simulation run. However, note that occasionally, the list includes receivers that are not behind the current bottleneck link. Furthermore, it was observed, that even the short term selected WPR, would sometimes be a receiver from a non-bottleneck cluster. This is likely due to dynamics of interacting with the steady-state TCP flow on the other corresponding feed link. It is anticipated that frequent probing of multiple CCRs in mdp-cc helps maintain

improved fairness and stability in light of such phenomena by allowing the source to quickly correct its choice of WPR. Yet, we have observed reasonable simulation results with a minimal CCR set of one, in this case the WPR and CCR are equivalent. From those results, it appears that dynamic CCR election process reasonably tracks the present worst path condition even without multiple candidates in the set.

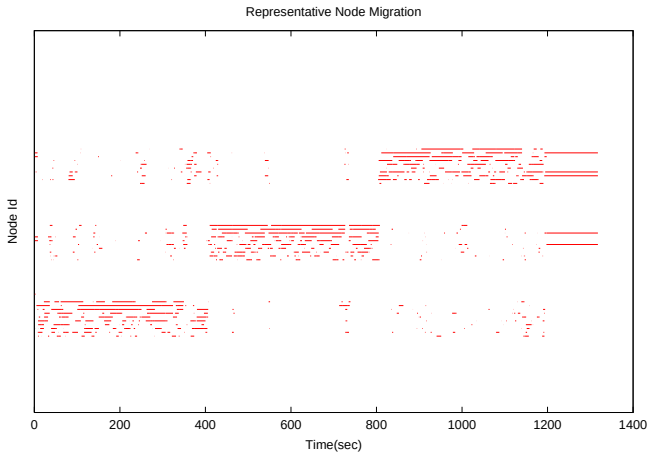

Figure 4: Migration of CC Representative Set

Further simulations will be conducted and more data collected to specifically evaluate the value of the multiple CCR approach. And, as the above graph suggests, further work in selecting appropriately uncorrelated CCRs to better span independent candidate worst paths is of interest. Some WPR candidate list clustering to a single bottleneck is highlighted in Figure 4, although there may be some value in the implicit hysteresis provided by having multiple representatives from the same congestion path. These tradeoffs will be examined in further work.

C. *Feedback Loading with CCRs*

As previously mentioned, maintaining the scalability of a NACK oriented protocol is an important design goal for an applicable congestion control scheme. The quantity of feedback traffic to the source in a reliable multicast session is a principal factor in determining scalability. Figure 5 is graph of the volume of feedback traffic from the receivers to the sources of the mdp-cc and TCP flows from the simulation described above.

The flat line near the bottom of the graph plots the rate of all mdp-cc feedback traffic, including NACK messages for reliability as well as RTT-oriented ACK messages from the CCR set of 5 nodes. In this simulation, there were 50 receivers in the mdp-cc group. The other plots in the graph represent the feedback traffic of TCP flows during the simulation. It is interesting to note that the total feedback traffic volume of mdp-cc to a group of 50 receivers is far less than the feedback generated by any of the competing TCP flows. Note that the quantity of TCP feedback traffic is relative to the transmission rate of the TCP flow. The principal source of mdp-cc feedback is the explicit response by representative receivers to source congestion control probes. The volume of this traffic is mainly a function of the topology RTT and the size of the source's CCR list. This portion of the feedback should remain relatively constant irrespective of group size.

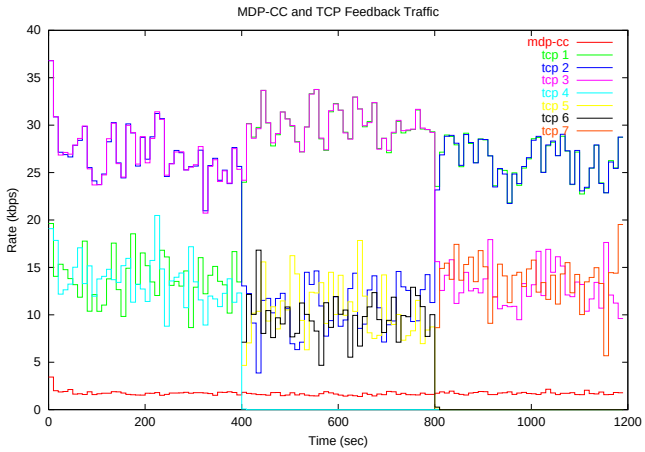

Figure 5: Session Feedback Loading

It is possible that the size of the representative list could be expanded if proved beneficial while maintaining low levels of feedback traffic compared to TCP. The explicit feedback from the group at large (excited by the *wildcard* probes mentioned earlier) is a minority of the traffic level in the figure. Proper prediction or estimation of group size will allow the volume of this type of feedback traffic to remain low.

VI. Conclusions

We have provided an overview of the mdp-cc design that provides rate-based TCP friendly congestion control within a NORM protocol framework. We also described a number of mechanisms that effectively tradeoff protocol scalability and overhead, while providing improved rapid response for congestion control reaction. We also presented example simulation results illustrating TCP friendliness and dynamic behavior.

We mentioned ongoing investigations into such issues as the membership characteristics of the CCR set for improving WPR election. Also, additional work is planned to investigate robustness issues under highly heterogeneous scenarios. We are also performing ongoing work on an algorithm to improve distributed receiver loss event estimation for non-CCRs to improve worst path calculations using purely source-based RTT estimators.

We believe our simulation and operational tests have demonstrated effective TCP fairness and inter-protocol fairness across a number of network scenarios. While ongoing work remains, we feel that the mdp-cc approach is safe to deploy in moderately scaled scenarios and preserves TCP friendliness. A public version of the protocol with the mdp-cc extensions is available at [16] and runs on a variety of environments (e.g., Win32, freeBSD, Linux, Solaris, ns2).

VII. REFERENCES

[1] V. Jacobson, *"Congestion Control and Avoidance"*, Proc. of SIGCOMM 1988, pp. 314-328.

[2] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", Internet RFC 2001, January 1997.

[3] A. Mankin, A. Romanow, S. Bradner, V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols", RFC 2357, June 1998.

[4] J. Padhye, V. Firoiu, D. Towsley, J. Kurose: Modeling TCP Throughput: A Simple Model and Its Empirical Validation. SIGCOMM 1998: 303-314.

[5] B. Whetton, J. Conlan. *"A Rate Based Congestion Control Scheme for Reliable Multicast"*, Technical White Paper, Globalcast Communications, Oct 1998.

[6] D. DeLucia, K. Obrascka, "Multicast Feedback Suppression using Representatives", in Proc. INFOCOM 97.

[7] D. DeLucia, K. Obraczka, *"Congestion Control Performance of a Reliable Multicast Protocol.,"*. Proc. of IEEE ICNP'98, August 1998.

[8] L. Rizzo, "pgmcc: a TCP-friendly single rate multicast congestion control scheme", ACM SIGCOMM 2000, Aug 2000.

[9] M. Handley, and S. Floyd, "*Strawman Specification for TCP Friendly* (Reliable) Multicast Congestion Control (TFMCC)", Reliable Multicast Research Group, December 1998.

[10] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based Congestion Control for Unicast Applications", ACM SIGCOMM 2000, Aug 2000.

[11] B. Adamson, J. Macker, "Recent MDP Congestion Control Research Status", slides from RMRG meeting, Pisa, Italy, June 1999, http://tonnant.itd.nrl.navy.mil/mdpcc/mdpcc-pisa.pdf

[12] J. Macker, R. B. Adamson, "Reliable Multicast Congestion Control", in Proc. IEEE MILCOM 2000, Los Angeles, USA, Oct 2000.

[13] J. Macker, R. B. Adamson. "The MDP Protocol Specification Version 1.6", Draft Technical Specification, Oct 1999, http://manimac.itd.nrl.navy.mil/MDP/DraftMdpSpec-1.6.txt

[14] J. Macker, R. B. Adamson, *"The Multicast Dissemination Protocol Toolkit"*, in Proc. IEEE MILCOM 99, Oct 99.

[15] J. Macker, "Reliable Multicast Transport and Integrated Erasure-based Forward Error Correction", Proc. IEEE MILCOM 97, Oct 1997.

[16] MDP Software Toolkit distribution, downloadable software and documentation, http://manimac.itd.nrl.navy.mil/MDP