Enhancing Reliable Multicast Transport to Mitigate the Impact of Blockage

Stephen F. Bush and Orhan Imer
GE Global Research Center,
Niskayuna, NY, 12309, USA
bushsf@research.ge.com

Praveen K. Gopala
IPS Lab, Ohio State University,
DL505 Dreese Labs, 2015 Neil Avenue,
Columbus, OH, 43210, USA
gopala.1@osu.edu

Abstract— **Mobile wireless communication is susceptible to signal blockage, which is loss of signal, typically due to physical obstruction, over a longer duration relative to fading. Measurements indicate that blockage has a significant impact on reliability in both open and rural areas. Reliable multicast, a transport layer mechanism, attempts to gain network performance by eliminating duplicate packets transmitted from a sender to multiple receivers along common paths while providing guaranteed packet delivery to all receivers. The gain achieved by multicasting places limits on the ability to optimize transmission to heterogeneous receiver channel characteristics in an individualized manner at the multicast transport layer. A simple, low overhead protocol extension to mitigate the impact of blockage upon reliable multicast is proposed by piggybacking on reliable multicast congestion control feedback.**

Keywords-component; mobile ad hoc network, wireless communications, blockage, QoS, reliable multicast transport, and satellite communication.

I. Introduction

Reliable multicast, a transport layer protocol, attempts to gain network performance by eliminating duplicate packets along common paths from a sender to multiple receivers while guarantying packet delivery. The gain achieved by reliable multicast places limits on the ability to optimize transmission to receiver channel characteristics in an individualized manner at the multicast transport layer. This is because there is a single logical channel within a multicast session; transmission characteristics within a session impact all receivers joined to a session. Wireless reliable multicast will be subject to signal blockage where blockage is the loss of a signal, typically due to physical obstruction of the signal path over longer durations relative to fading. It is assumed that blockage is most likely to occur between a satellite and a terrestrial receiver, most likely due to movement of receivers into locations without wireless contact to the satellite or the movement of an obstruction such as water vapor through the path of the signal. Measurements indicate that blockage has a significant impact on signal reception in both open and rural areas [7]. A simple, low-overhead protocol extension is presented that mitigates the impact of blockage in reliable multicast via piggybacking key

aggregate blockage information within the existing congestion control feedback mechanism. No new messages or fields are required.

The particular form of reliable multicast assumed in this paper is the NORM protocol [1][2]. The Internet Engineering Task Force (IETF) Reliable Multicast Transport (RMT) working group has made progress in defining a reliable multicast standard that reduces NACK overhead, incorporates congestion control, and forward error correction (FEC) [5]. Because reliable multicast will be competing for network resources with other transport, namely TCP/IP, it is important that reliable multicast share the bandwidth in a fair manner. An equation-based technique has been proposed for use in NORM in which the transmission rate is adjusted such that congestion control reacts in a manner similar to TCP/IP. In order to enhance reliability, forward error correction (FEC) is used such that NORM data are logically segmented into coding blocks and symbols. A NORM_DATA message payload corresponds to one encoding symbol. NACK-Oriented Reliable Multicast (NORM) packets are encapsulated in UDP/IP packets; thus each NORM packet receives best-effort delivery and blocked receivers result in dropped packets. The coding rate is defined as the ratio of the number of original data segments (K) to the total number of segments transmitted (N), which includes data and coding segments. Ideal rateless codes allow an indefinite number of coding symbols to be created and a receiver that successfully receives any K of N symbols will be able to successfully decode the original data. This allows the sender to stretch the transmission (stretch factor N/K) long enough so that lost packets can be recovered. Unfortunately, if the stretch factor is long enough for receivers with high-expected blockage durations to successfully receive the data, receivers in the same NORM multicast session with lower expected blockage durations will receive many additional, but unneeded, coding symbols. There are relationships between congestion control and FEC that this paper explores. First, a brief review of the extant single-layer multicast protocol will be useful.

Multicast, in order to be scalable, tends to be a receiver-driven protocol; receivers decide when to join and when to leave a multicast session. Given that changes to the existing network protocols and infrastructure cannot be taken lightly, a

22

blockage tolerant solution that only modifies edge nodes would be preferred. Ultimately a multicast tree is formed via group multicast management protocols such as IGMP[3], MLDv2 [4][6], and NORM with the sender as root. The hypothesis of this paper is that it will be advantageous to feed an aggregate blockage model from the receivers through the tree towards the root (sender) such that receivers with similar blockage patterns are grouped together in separate sessions. This requires balancing the tradeoff between duplication of packets in multiple sessions and reducing the impact of blockage upon all receivers. Before discussing the specific technique of blockage aggregation, the characteristics of blockage across receivers are discussed.

II. BLOCKAGE CHARACTERISTICS

A simplified blockage model based upon a two-state Markov model is described in [7]. Fig. 1 shows the frequency of blocked events based upon the Markov model and a maximum likelihood estimation (MLE) assuming an exponential distribution. The exponential distribution parameter (λ) is determined to be 0.257. Note that the Markov model distribution appears to have a heavier tail than the fitted exponential distribution. This result is for a single receiver and does not yet consider blockage correlation among multiple receivers.
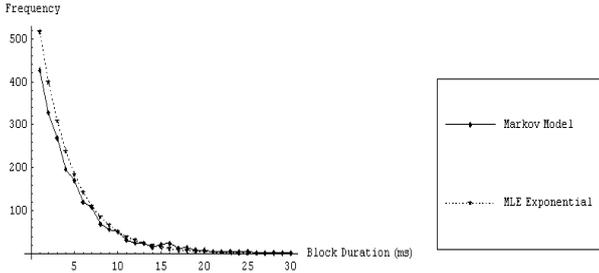

Fig. 1. Blocked and unblocked duration frequency of 100 receivers over a 100-millisecond time period based upon a two-state Markov model (solid line) and an MLE fit to an exponential distribution (dashed line).

Blockage is likely to occur due to movement of both receivers and obstacles within the path of a receiver. Receivers may move as a coordinated group and/or cloud cover may impact multiple receivers. In these cases, sets of receivers are likely to be impacted in a predictable manner. It should be noted that the data used to deduce the existence of the Markov model in [7] is based upon blockage collected from a single mobile receiver traveling through a sample urban and open area. Additional hypothetical receivers, traveling along with the actual receiver would experience a time-shifted duplicate of the measured blockage where the time-shift would depend upon their distance from the measured receiver's blockage. This relationship can be capture by a time-shifted exponential distribution. The probability of overlapping blockage durations for two receivers, where Λ is the random variable for the blockage event duration and τ is the time shift

between the events at the receivers is $P(\Lambda - \tau \geq 0)$. For more than two receivers, the overlapping blockage event probability is $P(\Lambda - \delta \geq 0)$ where δ is the dispersion, that is, the maximum time shift among blockage to all the receivers shown in (1).

$$P_b = 1 - \int_0^\delta \lambda e^{-\lambda \tau} d\tau \qquad (1)$$

The probability of two receivers in the same state is shown in Fig. 2. As the time shift between the receivers increases, the likelihood of being in the same state decreases. This may be due to greater spatial separation in the receivers resulting in the impact of blockage taking longer to impact the other receiver. Note that the slightly heavier tail mentioned earlier helps to increase the likelihood of receivers being in phase, that is, in the same blockage state. In order to make this generally useful, this model should be extended to multiple receivers which is considered next.
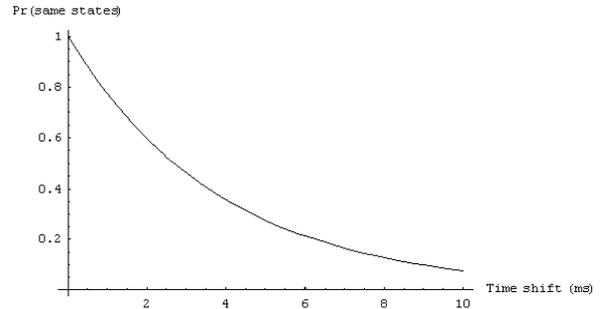

Fig. 2. The analytically derived probability of two blocked nodes in the same state using an exponential model is plotted above. The probability decreases with increasing time shift.

Two mechanisms to extend blockage to the general case of multiple receivers have been considered. The first and conceptually simplest is a double Markov model show in Fig. 3. The blockage Markov model for a single receiver is encapsulated within each state of a double Markov model. The leftmost state is a common blockage model shared by all receivers while the rightmost state is the individual model unique to each receiver. The *dc* parameter impacts the degree to which receivers are correlated.
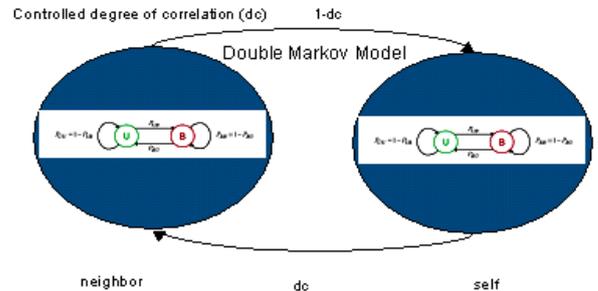

Fig. 3. Multiple receiver blockage models control the degree of blockage correlation among receivers.

In Fig. 4, three density plots are shown indicating blockage (white bars) for ten receivers (along y-axis) over 100

milliseconds (x-axis). The expected correlation is defined in (2) for each pair of blockage histories b_i and b_j .

$$E[c_{ij}] = E_{ij}[\frac{\text{cov}(b_i, b_j)}{\sigma(b_i)\sigma(b_j)}] \qquad (2)$$

In the left-most density plot, blockages are nearly independent, in the middle plot blockages are more aligned, and in the final plot, blockages are completely aligned.
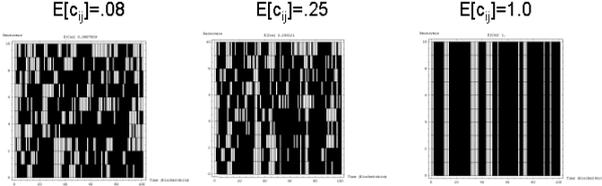

Fig. 4. Blockage is displayed for ten receivers (along y axis) as a function of time (x-axis). White bars indicate blockage intervals and the density plots illustrate blockage correlation for *dc* values of 0.2, 0.5, and 1.0.

A simulation for 10 receivers using the double Markov model was run for 100 milliseconds. The expected correlation for every pair of receivers (2) and the probability of all nodes being in the same state are shown as a function of the *dc* parameter in Fig. 5.
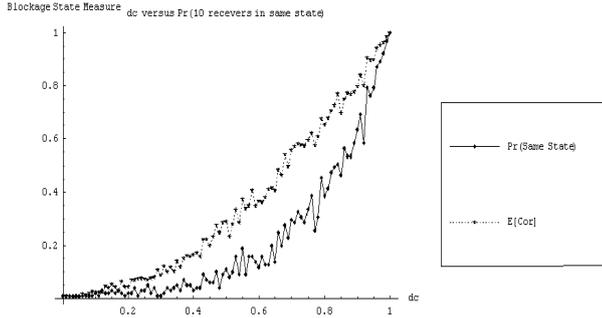

Fig. 5. The expected correlation of all receivers' blockage states (top curve) and the probability of receivers being in the same state (bottom curve) increase with *dc*.

The second approach is an analytical one in which dispersion, as mentioned earlier is used to control the correlation. The intuition is that the farther apart receivers are in space, the greater the time lag a common source of blockage may have in causing an impact across a set of receivers, regardless of whether the blockage or receivers are moving. The dispersion is the maximum blockage time shift among all receivers.

III. NORM PROTOCOL OPERATION

The portions of the reliable multicast protocol most impacted by blockage are data transmission from the sender to the receivers, the corresponding ACK/NACKs from the receivers, forward error correction overhead, and congestion control. FEC overhead is defined as any FEC packet transmitted but not needed by a receiver. A blocked node will induce a NACK to be sent and begin the multicast retransmission process. Blockage will also cause the transmission rate to be reduced as though congestion had occurred. The explicit congestion control (ECN) indicator [9] can be used so that only packets marked as experiencing congestion impact the receiver's model of congestion. The receiver can detect the presence of the ECN indicator and modify its feedback to the NORM sender appropriately.

This document describes, in progressively more detail, a protocol for partitioning receivers in a multicast network into distinct groups (sessions) based on the correlation of their blockage realizations. This enables the sender to tailor its responses to the group with the worse blockage without penalizing the group with fewer blockages. In summary, each receiver determines the session that it should join by a simple extension of the existing NORM congestion control mechanism. Receivers do not communicate with each other directly; they only communicate with the sender. The result of the protocol, namely, partitioning receivers, helps in splitting the multicast session into multiple sessions, one for each of the receiver groups. The use of multiple sessions to improve multicast is not new and has been proposed in layered coding. However, layered coding does not specifically address blockage. This proposed extension to NORM is unique because it facilitates leveraging correlated blockage among receivers in order to best partition receivers into multiple sessions. This yields significant gain in overall throughput since the performance of the entire multicast network is no longer dominated by the globally worst receiver, but by the locally worst receivers within each receiver group. However, this increase in throughput can come at the expense of increased network load, since the some data needs to be repeated in each of the multicast sessions. This algorithm attempts to increase the gain derived from partitioning into correlated blockage groups while limiting the number of multicast sessions to avoid duplicate packet flow. Next, requirements of FEC as a sole guarantor (without ARQ) of reliability are examined.

Blockage has an impact on forward error correction (FEC) overhead. Consider the extreme case in which there is no receiver acknowledgement; the sole guarantee of reliability rests with the ability of FEC to mitigate blockage. This has application when the desire is to eliminate receiver feedback to ensure a stealth factor for the multicast group. Given a multicast receiver set of size R where each receiver is undergoing independent satellite blockage, the problem is to determine the right compromise between (k, n) values of the FEC code, where k is the number of data objects, n is the total number of objects transmitted, and there are n-k objects added as FEC objects, to achieve a desired level of reliability, overhead, and delay. Note that any k objects received from the n that were transmitted guarantees reception. More precisely, one would like to:
1. Maximize the probability of successful reception, i.e. reception of k out of n packets across all receivers
2. Minimize the average delay in reception across all receivers

3. Minimize the overhead

Some of these are conflicting goals, since, for example, one cannot increase the probability of successful reception while at the same time decreasing the overhead. To analytically study the effect of FEC parameters on our performance objectives, first consider a single sender that sends n encoding packets to a single receiver, i.e. R=1. Suppose the packets are sent at a uniform rate every T seconds, and are lost independently with probability p. If the receiver receives any k out of the n encoding packets, declare that transmission is successful. If the receiver starts transmitting at time t=0, the transmission of the encoding block of n packets will terminate at time t=n T. Let S be the random variable (R.V.) denoting the number of packets received by the receiver at the end of the transmission period, t=n T. Clearly, S is a binomial R.V., and a successful transmission occurs if and only if $S \geq k$. In other words,

$$P = P[success] = P[S \geq k] = \sum_{l=k}^{n} \binom{n}{l}(1-p)^l p^{n-l}$$

for which it would be desirable to maximize over (k, n). Note that for a given p and k, successful reception probability increases with n. In fact, $P \to 1.0$ as $n \to \infty$, for any fixed k. However, the sender overhead, O_s=n-k, also increases with n. Therefore, n cannot be increased indefinitely if overhead is a design concern.

Next consider the average reception time, i.e. the average time it takes the receiver to successfully receive k packets given that it receives at least k packets by time t=n T. Let D be the R.V. that denotes the number of time units needed to receive k packets successfully. Note that D is a negative binomial (or Pascal) R.V. Now, the average reception time can be calculated as

$$T_{avg} = T \sum_{l=k}^{n} lP[D = l \mid S \geq k]$$

$$= T \sum_{l=k}^{n} l \frac{P[D = l, S \geq k]}{P[S \geq k]} = T \frac{\sum_{l=k}^{n} lP[D = l]}{P[S \geq k]}$$

$$= T \frac{\sum_{l=k}^{n} l \binom{l-1}{k-1}(1-p)^k p^{l-k}}{\sum_{l=k}^{n} \binom{n}{l}(1-p)^l p^{n-l}}$$

It can be shown that, as the stretch factor, s=n/k, becomes large, T_{avg} converges to kT/(1-p). In fact, the approximation $T_{avg} \approx kT/(1-p)$ works even for moderately large values of s. Note that the sender transmits n packets in nT seconds, and it takes on average T_{avg} seconds for the receiver to receive k of them.

Finally, the average receiver overhead, O_r, can be calculated as the number of additional packets received by the receiver given that the receiver received k or more packets by time t=nT. This yields,

$$O_r = \sum_{l=k+1}^{n} (l-k)P[S = l \mid S \geq k]$$

$$= \sum_{l=k+1}^{n} (l-k) \frac{P[S = l, S \geq k]}{P[S \geq k]} = \frac{\sum_{l=k+1}^{n} (l-k)P[S = l]}{P[S \geq k]}$$

$$= \frac{\sum_{l=k+1}^{n} (l-k)\binom{n}{l}(1-p)^l p^{n-l}}{\sum_{l=k}^{n} \binom{n}{l}(1-p)^l p^{n-l}}$$

Assuming that packets are lost independently across receivers, one can extend the analysis of this section to multiple receivers. Let S_r denote the number of packets successfully received by receiver r. Then, all R receivers will receive at least k packets if and only if $\{S_1 \geq k, S_2 \geq k, \ldots, S_R \geq k\}$. Therefore, the probability of successful reception by all R receivers is $P_s = P[S_1 \geq k, S_2 \geq k, \ldots, S_R \geq k] = P[S_1 \geq k]P[S_2 \geq k]\ldots P[S_R \geq k]=(P[S \geq k])^R = P^R$, assuming independent packet losses across the receivers.

The average time for a particular receiver to receive k packets, on the other hand, is the same as in one receiver case, and is given by T_{avg}. The average overhead for a particular receiver will also remain the same, and is given by O_r. Finally, the total sender overhead is given by O_s=n-k, as before.

To optimize over the FEC parameters (k, n), fix p and R to given values and plot the performance metrics (P_s, T_{avg}, O_r, O_s) as k and n are varied. In surface plots of Fig. 6 below, T is normalized to 1 second, p is set to 0.3, and R=100. The value of k increases from 1 to 8, as $n \geq k$ is increased from k to 4k for each fixed value of k.
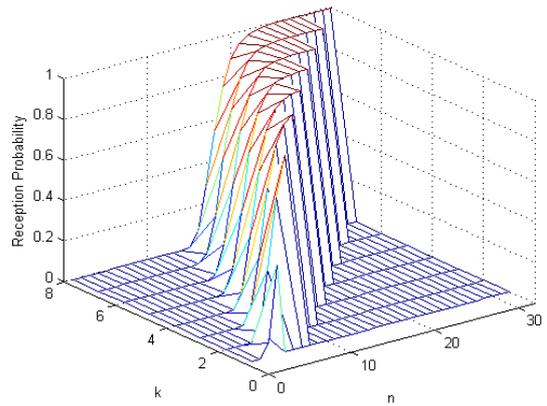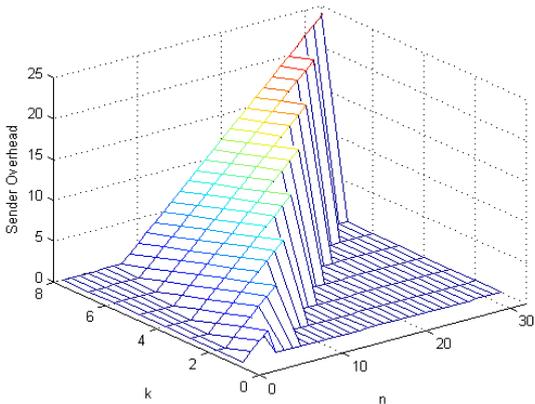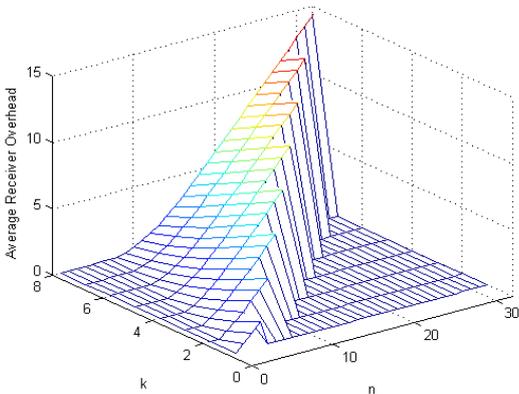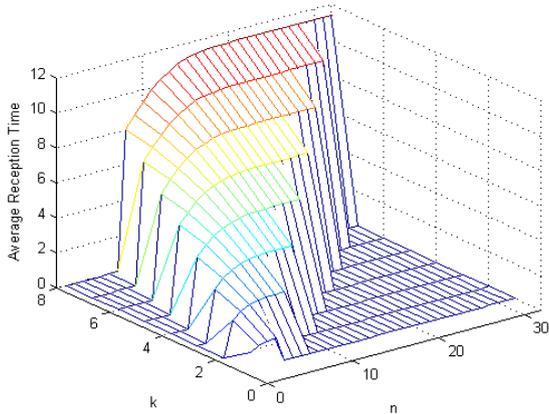
Fig. 6. Plot of the performance metrics (P_s, T_{avg}, O_r, O_s) for p=0.3, R=100, and T=1.

As can be seen from Fig. 6, for a given p and R, both the reception probability, P_s, and the average reception time, T_{avg}, increase with the ratio n/k. However, T_{avg} converges much faster than P_s. Hence in determining how large n should be as

compared to k, the reception probability is the determining factor. Both the receiver and sender overheads increase with the difference n-k, with the rate of increase being linear. Note that the curves in Fig. 6 can be repeated for different values of p and R. Trade-offs between these performance metrics must be taken into account when the FEC code parameters need to be determined for a particular application scenario.

Now, assume FEC automatically adjusts to meet worst-case blockage. The overhead is defined as the FEC stretch factor that required guaranteeing reception of packet at all receivers. A larger stretch factor is required for channels with larger blockage or higher bit error rates. When blockage is uncorrelated using NORM, a single layer multicast protocol, the same FEC overhead must be sent to all receivers. As shown in Fig. 7, FEC overhead goes down as channel blockage becomes correlated across receivers. The overhead is for 10 receivers using the double Markov model of blockage in an open environment and the x-axis is the value of *dc* in the Markov model.
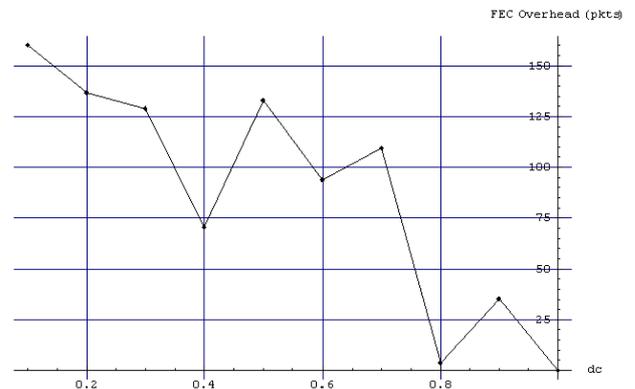

Fig. 7. FEC overhead decreases as blockage becomes more correlated.

Table 1 (three surface plots below) shows the FEC overhead as a function of the time window and blockage correlation. As previously determined, overhead decreases as receiver blockages become correlated. However, the frequency with which information must be transmitted as well as the sampling accuracy may have an impact on the overhead. The rough surface for the frequently sampled receivers (every 2ms) also shows the lowest FEC overhead as the system responds quickly to adjust overhead. As the frequency is reduced, the surface becomes smoother and the overhead rises. The expected overhead is 21, 78, and 107 packets respectively. The overhead without partitioning, that is, with no attempt to react to blockage correlation, is 300 packets. Thus, sampling at 2ms yields approximately 15 times improvement in FEC overhead.
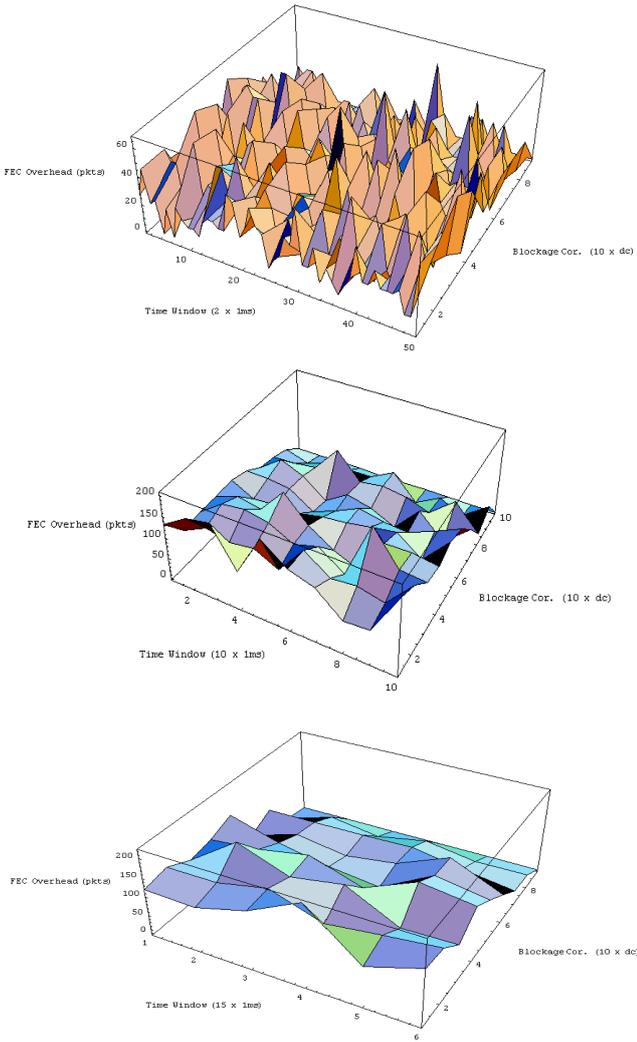
Table 1. FEC overhead decreases as blockage correlation increases. Variation in the smoothness of the surface is due to the blockage sampling rate; the smoother the surface, the less often sampling occurs and the greater the FEC overhead. However, sampling adds its own overhead and may not be feasible beyond a given rate.

IV. BUSH-GOPALA-IMER BLOCKAGE MITIGATION ALGORITHM IN NORM

This section describes the Bush-Gopala-Imer (GPI) algorithm for partitioning receivers into multicast groups based on correlation of blockage patterns to reduce the impact of blockage on both congestion control and FEC overhead. Each receiver generates a compact representation of its past blockage experience. This information is simply the description of a single blocked/unblocked cycle that best fits its most recent sampled blockage history. This computation is relatively simple; because the compact representation is meant to be small and simple rather than extremely accurate, sampling error at the receiver is not expected to be a significant factor. A millisecond-sampling rate that records time and either blocked or unblocked state will be sufficient. Receiver clocks need not be synchronized. Receiver times are computed relative to a common start time that is transmitted

from the sender. It is assumed that receiver clocks do not drift beyond one microsecond from the time the time reference message is received from the sender to the time blockage statistics are reported.

Blockage information piggybacks inside existing NORM congestion control messages. This minimizes changes to the standard as well as reduces overhead. These messages are transmitted back to the sender at random exponentially delayed time intervals to reduce feedback congestion. The expected value from the MLE in Fig. 1 is 3.89 milliseconds and the variance is 15.14. The goal is to reduce feedback overhead, therefore transmitting complete blockage history from each receiver is unfeasible. Instead, an approximation is used. The intuition is to detect the presence of blockage impact by looking for common expected value and variance of blockage durations within receiver sets. While it is possible that receivers may have the same expected value and variance; there is likely to be a time shift. This is detected using the difference between the time of the first blockage and the reference time from the sender. Thus, the goal of the sender is to group receivers with common expected blockage durations, variances, and time shifts together.

In summary, the algorithm operates as follows. Each receiver maintains a local history of expected blockage duration, blockage duration variance, and time shift. Receivers transmit this information to the sender by piggybacking on NACK and congestion control ACK messages. The sender examines each set of parameters arriving from each receiver. The sender places the parameters in one of two bins as follows. If the bins are initially empty, the parameters for the first receiver information received are placed in Bin 1. If Bin 1 already has one set of parameters, then compare the receiver's incoming parameters with the Bin 1 parameters. If the incoming receiver's expected blockage duration exceeds that in Bin 1 by one standard deviation from the exponential MLE value, then place the element in Bin 2. If Bin 1 and Bin 2 both have receiver parameters, then place the incoming parameters in the bin in which the variance of the bin with the newly received parameters would be smaller. Old receiver parameters are removed when new ones arrive for the same receiver. When the union of both bins contains all monitored receiver blockage values, the expected value and variance of Bin 1 (group one) and Bin 2 (group two) are multicast back to each receiver in order to aid in their decision to form new sessions. The process continues in a manner such that a binary tree of sessions may be formed; each session may split into two separate sessions.

A. Receiver Processing

The first stage of the algorithm is implemented on the receiver side. Each receiver calculates the following three parameters based on the blockage that it experiences. The first parameter, δ_r, is the time the receiver transitions into the first blocked state evaluated between the sender reference time and the current time. The reference time is a common time reference, sent by the sender to all receivers, such that receivers can form a common alignment of their blockage

27

history. Receivers base their local computations from this reference point. The second parameter is the best estimate of the expected blockage duration, μ_{b_r}, (i.e., the period between the unblocked-to-blocked state transition and the next blocked-to-unblocked state transition). The third parameter is the variance (error) of the actual blocked durations from the above blockage duration estimate, $\sigma^2_{b_r}$. Each receiver feeds back these three parameters to the sender as previously mentioned.

B. Sender Operation

The sender examines each set of parameters arriving from each receiver. The binning process takes place as previously mentioned. When the union of both bins contains all monitored receiver blockage values, the expected value and variance of Bin 1 and Bin 2 are fed back to each receiver in order to aid in their decision to form new sessions. The process continues in this manner such that a binary tree of sessions may be formed because each session may split into two separate sessions. Thus, during the second stage of the protocol, the sender generates the following two parameters. The first parameter is the expected value (across receivers) of the expected blockage times that it received from all receives in Bin 1 (μ_1) and Bin 2 (μ_2). The second parameter is the variance of the expected blockage durations from all the receivers in Bin 1 (σ_1^2) and Bin 2 (σ_2^2). The sender multicasts these two values to all the receivers.

C. Receiver Processing

The third stage is the receiver partitioning stage wherein each receiver may join a new session based on the specified partitioning rule. The receiver must decide which of two possible sessions to join based upon the aggregate bin blockage information. Note that the receiver's first order statistical history has been included in the bin information. However, by the time this information has been fed back to the receivers, the receiver's history has likely advanced because new samples have been added to the receiver's history. Thus, the receiver should compare its latest statistics with the bin information from the sender.

If the receiver's blockage time fraction is lower/higher than the corresponding bin expected value calculated at the sender in stage two, i.e. if $e_{b_r} \leq \mu_1$, then remain in the current session, otherwise join a new session that will be created. All the receivers joining the first group (Bin 1) are expected to have longer unblocked periods and hence better blockage performance, thus not being limited by the performance of the remaining half of the receivers partitioned into the second session.

The process of binning may continue within each newly formed session causing a further refinement of the alignment of unblocked periods (and thereby improve throughput performance) by again applying the same algorithm within each group. Hence $\log_2(r/n)$ iterations of this algorithm partition r receivers into groups of size n.

The third stage logically partitions the receivers into two distinct groups. Once this stage is complete, each receiver knows the group to which it belongs. The sender starts a new multicast session and allocates it for transmission to the new partition. The same information stream is transmitted in both multicast sessions. Each receiver joins its corresponding multicast session (this can be handled using the IGMP protocol [8]). Note that the new multicast sessions can be loosely considered to be an analog to an operating system "fork" since transmission continues in the new sessions at the point where they had been in the original session. It is the responsibility of the receivers to properly join the information from the new session to the information received from a previous session.

For synchronizing receivers and starting their local computation windows from the same time reference, the sender needs to transmit a common time reference point (start time) to all the receivers. This information can be piggybacked on the NORM_CMD(CC) messages sent out by the sender for GRTT collection before starting the actual data transmission. The NORM_CMD(CC) message has an 8-bit "reserved" field that is currently not used for NORM operation. The sender chooses a reference point (start time) and sends the difference between this reference point and the current send_time (may require quantization) in this 8-bit reserved field. For more accuracy, the sender could also use the 8-bit reserved field in the NORM_CC Rate Header Extension of the NORM_CMD(CC) message that is also currently not used for basic NORM operation.

V. EXPERIMENTAL VALIDATION: QUALNET SIMULATION

There were 11 nodes, one sender and 10 receivers using three different stretch factors. For each stretch factor, simulations were run with blockage that varies from 0 thru 50% mean blockage time. The NORM parameters used are shown in Table 2. The mean blockage characteristics are the same for all receivers except that receivers 1 thru 5 have low correlation in blockage events while receivers 6 thru 10 have a higher correlation. The network topology was kept simple; it emulated a satellite multicast to all receivers in one hop. Note that congestion control was disabled in order to examine FEC performance separate from congestion control. Also note that similar results were obtained for each stretch factor, however, space limitations do not allow for a discussion of pro-active parity.

Table 2. NORM QualNet Simulation Parameters.

	Stretch 0	Stretch 2	Stretch 3
NORM-AUTO-PARITY-SIZE	0	64	128
NORM-STATISTICS	NO		
NORM-DEBUG-LEVEL	6		
NORM-CONGESTION-CONTROL	OFF		
NORM-XMIT-RATE	1000000.0		
NORM-BLOCK-SIZE	64		
NORM-PARITY-SIZE	0	64	128
NORM-XMIT-FILE-SIZE	83886080		

As shown in Fig. 8, the total number of erasures occurring in the more correlated partition was significantly less than for

the un-partitioned, single session simulation. Repairs appeared to be more effective within the more correlated partition.
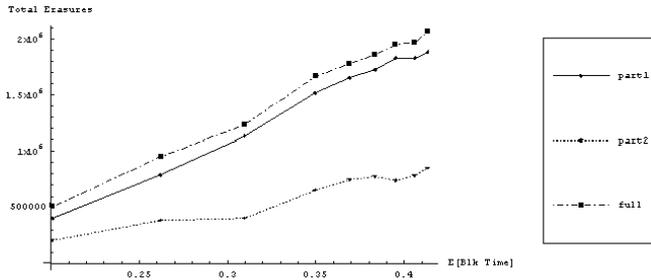

Fig. 8. Total erasures with no session partitioning (top curve) versus BGI partitioned sessions (lower two curves) as a function of expected blockage time. Intelligent partitioning significantly reduces the number of erasures.

As shown in Fig. 9, when using the BGI algorithm, the partitioned multicast sessions had much less overhead than a single multicast session that included all receivers. The total number of UDP packets required to multicast the file was significantly less using the BGI partitioned receiver set.
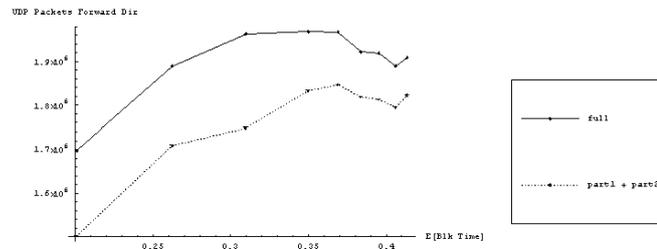

Fig. 9. Total Number of UDP Packets transmitted with no session partitioning (top curve) versus BGI partitioned sessions (lower curve) as a function of expected blockage time. Intelligent partitioning significantly reduces the number of packets (overhead) required.

VI. CONCLUSION

Blockage was characterized in terms of its impact on the NORM protocol and an algorithm to leverage correlated receiver blockage was developed. A simple low-overhead protocol extension to NORM that mitigates the impact of blockage in reliable multicast is feasible with very minor modifications to extant protocol. The blockage mitigation algorithm as presented in this paper is distinct from Asynchronous Layered Coding (ALC) in that ALC creates multiple parallel sessions with receivers in order to increase the FEC throughput. While the technique proposed in this paper could aid receivers in joining such sessions, the proposed Bush-Gopala-Imer (BGI) approach partitions receivers into distinct single sessions such that receivers with similar blockage patterns are grouped together. The BGI extension to NORM also improved the impact of blockage on congestion control because the current limiting receiver (the receiver assumed to be experiencing the most congestion) does not hold back receivers in the less impacted sessions. It was shown that the impact of blockage on FEC overhead alone could be reduced by up to 15 times using the proposed algorithm on a model derived from actual blockage measurements.

REFERENCES

[1] Adamson, B., Bormann, C., Handley, M., Macker, J., Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol, IETF RFC 3940, November 2004.
[2] Adamson, B., Bormann, C., Handley, M., Macker, J., Negative-Acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Building Blocks, IETF RFC 3941, November 2004.
[3] Cain, B., Deering, S., Kouvelas, I., Fenner, B., Thyagarajan, A., RFC3376, Internet Group Management Protocol, Version 3, Oct. 2002.
[4] Haberman, B. Martin, J., RFC4286, Multicast Router Discovery, December 2005.
[5] Luby, M., Vicisano, L., Haken, A., "Reliable Multicast Transport Building Block: Layered Congestion Control", RMT Working Group, draft-ietf-rmt-bb-lcc-00.txt, November 2000.
[6] Vida, R., Ed., Costa, L., Ed., RFC3810, Multicast Listener Discovery Version 2 (MLDv2) for IPv6, June 2004.
[7] Yao, H., EHF Satellite Communications-On-The-Move Blockage Channel Modeling Technical Report 1098, Lincoln Laboratory Massachusetts Institute Of Technology Lexington, Massachusetts, November 2004.
[8] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, Xerox PARC, November 1997.
[9] Ramakrishnan, K., Floyd, S. and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168 September 2001.