Channelization for Network Coding in Wireless Networks

Raju Kumar*, Heesook Choi†, JaeSheung Shin‡ and Thomas La Porta*

*Computer Science and Engineering,
Pennsylvania State University,
University Park, PA 16802
{rajukuma,tlp}@cse.psu.edu

†Sprint ATL,
1 Adrian Court,
Burlingame, CA 94010
heesook.choi@sprint.com

‡Converged Access Network Research Team,
Electronics and Telecommunications Research Institute,
DaeJeon, Korea (South)
sjs@etri.re.kr

Abstract—**Network coding is increasingly being investigated as an alternative to routing to increase throughput in packet networks. Like most data transfer schemes, the effectiveness of network coding may be limited by extreme congestion. When using network coding, these congested conditions are mitigated somewhat, but may still occur. We propose a selective channelization scheme in which links that experience congestion at a level that cannot be overcome by network coding are given reserved communication resources. This method has the following benefits. First, the algorithm proposed allows network coding full opportunity to overcome congestion before performing channelization, thus reducing the number of reserved resources used. Second, when triggered, the channelization of severely congested links greatly improves the end-to-end performance of flows that traverse the channelized link. To determine the point at which channelization should be triggered, we perform a thorough analysis of potential coding gains in a network facing errors due to collisions, and determine the point at which network coding loses its effectiveness.**

I. INTRODUCTION

Network coding, proposed by Ahlswede et al. [1], introduced the concept of combining packets before forwarding. By combining packets, less transmission capacity in the network is required to deliver data than with the traditional store-and-forward approach used by routing schemes. Due to the inherent broadcast nature of the medium, network coding has been effectively applied to the problem of data delivery in wireless networks [11].

Network coding has been widely used to deliver multicast data. On a multicast link, called a hyperarc, network coding gains can arise for one of two reasons - multi-path and independent errors at destinations. If a node is a destination of multiple hyperarcs (e.g., in the canonical butterfly network example for network coding [6]), gains due to multi-path may arise. Otherwise, coding opportunities will arise out of independent errors in packet delivery to the destinations of the hyperarc. For example, consider a source that multicasts two packets P_1 and P_2 to nodes N_1 and N_2. Suppose that node N_1 receives only packet P_1 while node N_2 receives only packet P_2. The source may then multicast P_1 combined (using e.g, bitwise-XOR) with P_2 so that both nodes can recover their missing packet in a single transmission. This simple example shows the basic operation of network coding to overcome errors on a multicast flow.

Several factors impact the extent to which network coding gains are achieved when dealing with error recovery, such as link error rates (LER), frequency of collisions, maximum number of packets that can be coded, distribution of errors among destinations of a hyperarc, etc. While network coding greatly reduces the number of transmissions required to deliver data from a source to multiple destinations, its effectiveness is limited under conditions of extremely high errors, as with most data transfer methods. Though it still improves delivery efficiency when compared to routing under such conditions, it is unable to prevent the number of transmissions from increasing drastically.

In this work, we introduce a technique to assist network coding under conditions of extreme contention. Specifically, we address the delivery of multicast traffic using network coding in the presence of intense background traffic. When most errors occur due to collisions, channelization of a link will drastically reduce the error rate by removing errors due to collisions. We use this as motivation to introduce a selective channelization scheme used in concert with network coding. Network coding is relied upon to overcome errors caused by impaired links and collisions in the majority of the network operation; under periods of high collision rates, the affected links are channelized, i.e., assigned TDMA slots, to reduce the probability of collision to zero, leaving network coding to contend only with remaining errors.

With selective channelization only severely congested hyperarcs in a flow are selected for channelization. As a result very few resources are required for reserved channels. Because network coding naturally reduces congestion, channelization is required at a higher threshold of offered load than when traditional routing protocols are used.

The contributions of this paper are :

- Provide an in-depth analysis of potential network coding gains under congested conditions
- Design of selective channelization policies based on our analysis
- Simulation results to show gains brought by selective channelization with network coding

The rest of this work is organized as follows: Section II presents motivations for this work; Section III presents an analysis of expected network coding gains and sheds insight into the potential impact of channelization; Section IV introduces our selective channelization policies; Section V lays out the details of our network coding scheme; results are discussed in Section VI; related work is presented in Section VII; Section VIII concludes the paper.

II. MOTIVATION

Multicast traffic forms both the background and main traffic in this work. 802.11 does not protect multicast traffic by performing an RTS/CTS exchange before transmission. As a result, many collisions occur when the intensity of the background traffic is high. Channelization is one solution to address the issue of large-scale collisions. In this section we provide a brief motivation for selective channelization applied to network coding. We develop a formal model in the next section.

A. Channelization

In this work, we define channels as TDMA slots that are organized into frames which repeat cyclically. If a link is channelized, it has at least one slot in a frame reserved for its use among its two hop neighbors. The remaining slots that are not reserved may be used for transmissions by other links. The other links access these slots using a Slotted ALOHA-based protocol.

We explain this model with a simple example that is for illustrative purposes only. Assume a frame includes 10 slots. If a node is backlogged, i.e., it has data to send at the start of the frame, and it does not have a channelized link, it will send in each slot with a probability of 0.1 until it empties its queue. Now consider a case in which a node has a channelized link, i.e., it has a time-slot in each frame reserved for its exclusive use. The remaining nodes in the neighborhood will now contend for the other 9 slots in each frame.

The benefit for a link that is channelized is that it will not experience contention or collisions during its slot. Thus it is highly advantageous during periods of high congestion when Slotted ALOHA will collapse. A naive approach to solve high collision rates is to channelize all transmissions in a flow or network. This solution suffers from high inefficiencies because in a sizable network, the large number of channels required to allow all flows to be scheduled (inspite of spatial reuse) would be impractical from a standpoint of resources required. Even if ample resources are available, large frame sizes, i.e., frames comprised of many slots, would be required to ensure all links may have a channel. Such large frame sizes will lead to long delays even in a lightly loaded network.

To avoid the problem of large frame sizes we propose selective channelization in which only a portion of the links are channelized.

B. Channelization for Network Coding

Opportunity for network coding to assist in overcoming errors arises out of independent losses in the delivery of data to the destinations of a hyperarc. We consider losses due to link errors and collisions only in this work. Other reasons for packet loss can be dealt with by using the solutions proposed in prior literature.

While network coding improves the throughput of the network by reducing the number of transmissions required [1], [14], under high error rates the number of transmissions required to deliver data will increase drastically. If most of these errors are due to collisions, channelization may be used to mitigate the loss and reduce error rates to the point where network coding regains its effectiveness; if the errors are due to link impairments, channelization will not help the situation.

Channelization may be applied to networks that use routing as well as those that use network coding. The benefit of combining network coding with channelization is that fewer links will require channelization due to the reduced transmissions required when using network coding. *We aim to activate channelization only in circumstances when network performance is highly degraded due to collisions and cannot be overcome by network coding. Our main challenge is to determine when to trigger channelization to fully leverage the benefits of network coding.*

III. GAIN ANALYSIS

In this section we analyze the potential performance of a simple network using routing and network coding. This analysis, while based on some simplifying assumptions, sheds light on the potential gains of network coding, shows regions of operation where its effectiveness may be limited, and provides insight as to when channelization will be beneficial.

At high collision error rates, even with network coding, the number of transmissions for data delivery may reach unacceptable levels. Let the metric for the efficiency of the data delivery scheme (i.e., routing or network coding) be the number of transmissions required to deliver 100% of the data to all the destinations for all flows. The gain g_h achieved over a single hyperarc h is

$$g_h = r_h/n_h \ \forall h \in H \tag{1}$$

where r_h is the number of total transmissions without network coding, and n_h is the total number of transmissions required with network coding. We next analyze the features of network coding gain and its interaction with channelization for one-hop and multi-hop networks.

A. One-Hop Network

Consider a one hop network with N destinations. Since we are interested in channelization, we focus on losses due to collisions only and not due to link errors. Let the number of packets that need to be delivered to all the destinations be B. Note that network coding does not allow combining more packets than the number of destination nodes. Hence the following analysis assumes $B \leq N$. Let the probability of a destination receiving a packet correctly be p_i, i.e., the probability of a collision is $p_b = 1 - p_i$. We assume that errors due to collisions at each destination are independent. While this is not realistic depending on network topology, the assumption is sufficient for allowing us to determine thresholds for triggering channelization.

After transmitting a group of packets, all patterns of data delivery to the destinations can be characterized by l - the number of packets delivered to all destinations, j - the number of packets that are delivered to none of the destinations, and $k = (B - l - j)$ - number of packets that are delivered to some but not all destinations. The performances of routing and network coding differ when $k > 1$.

The expected number of transmissions (denoted by $T(N, B)_{routing}$ for routing and $T(N, B)_{nc}$ for network coding) required to deliver B packets to all N destinations is given by Equation 2.

$$T(N,B)_{scheme} = B(p_i^N)^B + \sum_{l=0,m=B-l}^{B-1} \binom{B}{l} X_1 + \sum_{l=0}^{B-1} \binom{B}{l} X_2 \tag{2}$$

$$X_1 = (B + T(N,m))(p_i^N)^l (p_b^N)^m \tag{3}$$

$$X_2(l,j,k) = p_i^{Nl} \sum_{j=0}^{B-l-1} p_b^{Nj}[1 - p_i^N - p_b^N]^k \binom{B-l}{j} X_3(l,j,k) \tag{4}$$

The difference in the value of $T(N, B)_{routing}$ and $T(N, B)_{nc}$ stems from the term X_3 as derived in the Appendix. This term accounts for the reductions in transmissions when coding is possible.

When $l = B$ all packets reach all destinations and when $j = B$, none of the packets reach any destination. In these two cases network coding will not assist in recovering from errors because there is no scope for coding. When $k = 0$, a subset of packets reaches all destinations, while the remaining packets do not reach any destination. When $k = 1$, one packet reaches some, but not all destinations. But this single packet cannot be combined with any other packet. Hence network coding will not provide a benefit for error recovery when $k \leq 1$. When $k > 1$ different destinations receive different sets of packets. In these cases it may be possible to combine packets and thus achieve network coding gains. Even when $k > 1$ coding is not always possible, and if it is, different gains arise from different combinations of received packets. The term X_3 quantifies these gains.

If the network is operating in a region where the term for $k > 1$ dominates, this means some destinations are receiving packets while others are not, and hence network coding may provide a benefit. More specifically, if this term is large, destinations have received different packets and combining packets for retransmissions may be possible. If the network is operating in a region where terms of Equation 2 for $l = B$ or $j = B$ dominate, the scope for network coding is small. Specifically, if $j = B$, there is little opportunity for network coding due to collisions and channelization gains are vital for the performance of the network.

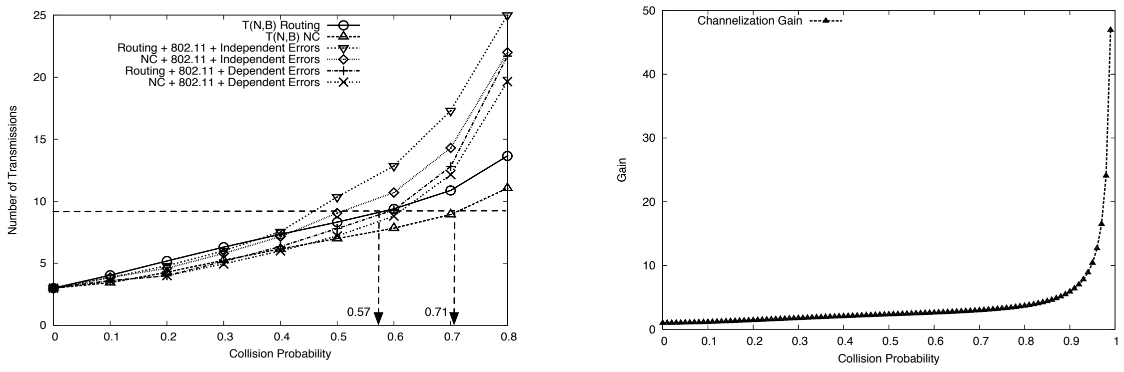
To validate these equations we run an ns-2 [9] simulation with $N = 3$ and $B = 3$ to determine the number of packets required to deliver data for routing and network coding under different collision probabilities. We compare these results with those obtained by Equation 2 in Figure 1(a). In the first set of simulations, we set the destinations so that their collisions are independent. In the second set, two of the destinations are assumed to share 50% of their neighbors with the third destination and hence their packet reception is correlated 50% of the time with the third destination. Our results in Figure 1(a) indicate our model provides a good estimate of the expected number of packets to be transmitted. The values resulting from the model match the empirical results closely for low collision rates. At higher rates, the model diverges but is still within 30% for network coding with independent errors with a probability of collisions of 50% and within 15% for dependent errors when the probability of collisions is below 60%.

The formula for the gain (by using Equation 1 and $r_h = T(N, B)_{routing}$ and $n_h = T(N, B)_{nc}$) brought about by network coding is theoretical in nature. In practice, these gains are lower due to the limited number of packets that can be combined and the limited number of destination nodes of a multicast hyperarc. In addition, implicit to the concept of achieving such a theoretical gain is the presence of a perfect feedback channel to convey the failure of data delivery to any destination of the hyperarc. Without this feedback, the source of the hyperarc has to employ schemes like ACKs and heuristic based guessing schemes [11] to multicast the correct combination of packets. These imperfect feedback schemes further erode the gains of network coding.

Using Equations 1 and 2 for $N = 3$ and $B = 3$ we find that the network coding gain over routing increases as the error rate increases until an error rate of 0.21, and then remains at a value of around 1.21 thereafter. Since our goal in this work is to present the benefits of combining channelization and network coding, we use topologies that can isolate the gains related to collisions during data transfer. The topologies we use to do this are not necessarily the best for network coding in general - e.g., network coding brings large gains when a node is at the junction of numerous flows. Therefore, the absolute gains of network coding over routing may be smaller than those reported in other papers. If optimal topologies are used, these gains will be restored and the benefits we show in this paper will still be achieved.

As discussed above, initially as the error rate increases, network coding gains increase. If a source of errors is removed, e.g., if collisions are eliminated via channelization, network coding gains may also decrease because the scope for network coding has been reduced. However, this loss in coding gain may be offset by a gain due to channelization, i.e., fewer packets are required to be transmitted because channelization has removed errors. We should channelize only if doing so has an overall gain.

Let the error rate due to collisions be e_1 and that due to link errors be e_2 for a link. Hence the combined error rate is $1 - (1 - e_1) * (1 - e_2) = e_1 + e_2 - e_1 * e_2$. Let the gain function for network coding over routing be $g_{nc}(e)$ and that for channelization gain over network coding be $g_{ch}(e)$ where e is the combined error rate. When the total error rate is $(e_1 + e_2 - e_1 * e_2)$, the network coding gain without channelization



(a) Avg. Number of Transmissions Per Packet with Perfect Feedback: N=3, B=3, Packet Size = 512 Bytes

(b) Channelization Gain over Network Coding

Fig. 1. Network Coding and Channelization

is $g_{nc}(e_1 + e_2 - e_1 * e_2)$. Upon channelizing this link, errors due to collisions are removed. Hence network coding gains arise out of link errors only, i.e., network coding gain is now $g_{nc}(e_2)$. In this case the channelization gain is $g_{ch}(e_1)$. Since network coding and channelization gains are independent , the total gain for the link is their product - $g_{nc}(e_2) * g_{ch}(e_1)$.

For a larger overall gain due to channelization, we require

$$\begin{aligned} g_{nc}(e_1 + e_2 - e_1 * e_2) <& g_{nc}(e_2) * g_{ch}(e_1) \\ g_{nc}(e_1 + e_2 - e_1 * e_2)/g_{nc}(e_2) <& g_{ch}(e_1) \\ g_{ch}(e_1) >& 1.24 \quad \text{[From Equations 1,2]} \\ e_1 >& 0.14 \text{ [From Figure 1(b)]} \end{aligned} \tag{5}$$

Hence in our sample network, if channelization is triggered on a link that has a collision probability of more than 0.14, it will always be beneficial. Our channelization policies (discussed in the next section) trigger channelization at higher estimated collision rates. Thus our approach to channelization will always increase the joint gain.

B. Multi-Hop Network

The gains of network coding in a multi-hop network are limited by the fraction of multicast hyperarcs (like COPE [11], unicast hyperarcs do not bring network coding gain in our scheme) and scope for network coding. Suppose that a flow involves m hops (i.e., hyperarcs) and that network coding brings gains g_i where $i \in [1, m]$ over these hops. Let maximum gain over any hyperarc be g_{max}. Let hop i transmit n_i packets when routing is used. The gain G for the end-to-end flow is

$$\begin{aligned} G =& (\sum_{i=1}^{m} n_i)/(\sum_{i=1}^{m} (n_i/g_i)) \\ \leq& (\sum_{i=1}^{m} n_i)/(\sum_{i=1}^{m} (n_i/g_{max})) \\ \leq& g_{max} \end{aligned} \tag{6}$$

As a result, the total gain is not multiplicative (or even additive) of individual gains. In fact, the total gain is less than the maximum gain achieved over any hyperarc.

Now let the fraction of hyperarcs that bring gain be f and let $n_i = n$ and $g_i = g \ \forall i \in [1, m]$. Hence

$$\begin{aligned} G =& (\sum_{i=1}^{m} n_i)/(\sum_{i=1}^{m} (n_i/g_i)) \\ =& (mn)/(\sum_{i=1}^{m} (n/g_i)) \\ =& m/((1-f)m + \sum_{i=1, g_i \neq 1}^{m} (1/g_i)) \\ =& m/((1-f)m + fm/g) = g/((1-f)g + f) \end{aligned} \tag{7}$$

As f increases, G increases. In fact, if a large fraction of the hyperarcs on a flow have little network coding gain, the end-to-end flow will also have low coding gains. As a result, the fraction of hyperarcs that bring gain is also an important contributor to total gain. Hence our approach to channelization helps the network in two respects. First, it increases the joint gain of hyperarcs on which network coding gains are already present (note that we never reduce joint gain over a hyperarc). Second, since network coding does not bring any gains for unicast links, channelization of such a link will bring pure channelization gains. As a result, channelization is a compelling scheme to improve the performance of networks employing network coding.

IV. CHANNELIZATION

We assume an underlying TDMA MAC layer and hence channelization refers to the allocation of slots (i.e., channels) in a frame to hyperarcs. We use the canonical two-hop reuse constraint of channelization [15].

Our channelization is per flow, i.e., the same physical hyperarc may be assigned multiple channels - one for each flow that utilizes it, thereby isolating the performance of one flow from another. Suppose there are s slots in a TDMA frame. Out of these slots, s_{chan} slots are marked for channelization (i.e., reservation) while the rest of the slots are used by unchannelized hyperarcs to contend for transmission using a Slotted ALOHA-based mechanism. These contention channels are also used to carry control packets.

In the following subsections, we describe our algorithms for deciding when to channelize and how to channelize.

A. When To Channelize

As discussed, one benefit of applying network coding is to reduce the number of transmissions required to overcome errors. However, even with network coding the number of transmissions required to deliver data may rise dramatically due to high error rates. To address this, we propose to integrate network coding with channelization. We propose two channelization schemes of varying complexity and adaptation to network coding. These methods adhere to the following philosophy. A node determines if the number of transmissions required to successfully deliver packets exceeds a threshold. If it does, the node channelizes the link, i.e., we trigger channelization if the threshold is exceeded by errors due to collisions.

- *Count Based Channelization:* In this first scheme a node simply maintains an average of the number of transmissions required to successfully deliver a packet. When this average becomes larger than a threshold, this hyperarc is channelized. This approach is accurate in determining the cost of transmitting a packet while accounting for network coding, but does not separate errors due to collisions from those due to link errors. Therefore, it may channelize links to no avail in some circumstances.
- *Model-Based Channelization using Network Coding Estimate:* In this method we attempt to isolate the impact of collisions on the number of transmissions. We do this by setting a target threshold based on $T(N,B)_{nc}$ from Equation 2. Based on this equation we determine a target p_b at which the threshold is exceeded. Nodes gather information, as described below, from their neighbors to estimate p_b, and if this value is above the target p_b, the link is channelized. This method isolates collisions from link errors and accounts for the benefits of network coding. Thus, if the collision rate is such that network coding is still operating effectively, this method will not trigger channelization.

  How to estimate p_b of neighbors: Each node appends a flag in each transmitted packet indicating whether it has more packets waiting to be transmitted in the contention slots. Based on such overheard information from all neighbors, a node can calculate its own collision probability, p_b. Let n_{cont} be the number of neighbors that have data to send in one of the s_{cont} channels. Each node maintains a weighted moving average of n_{cont} by counting the number of neighbors that transmitted packet with the aforementioned flag set. p_b can be estimated as $1-[1-p_{send}]^{n_{cont}}$ where p_{send} is the probability of a backlogged node transmitting in a contention slot. The local p_b in each node is maintained as a weighted moving average as well. This p_b is also appended to all transmitted packets. Using overhearing, nodes gather the p_b of all neighbors. This value of p_b will tend to be aggressive because a node does not contend with itself but the information received from its neighbors includes its own transmissions.

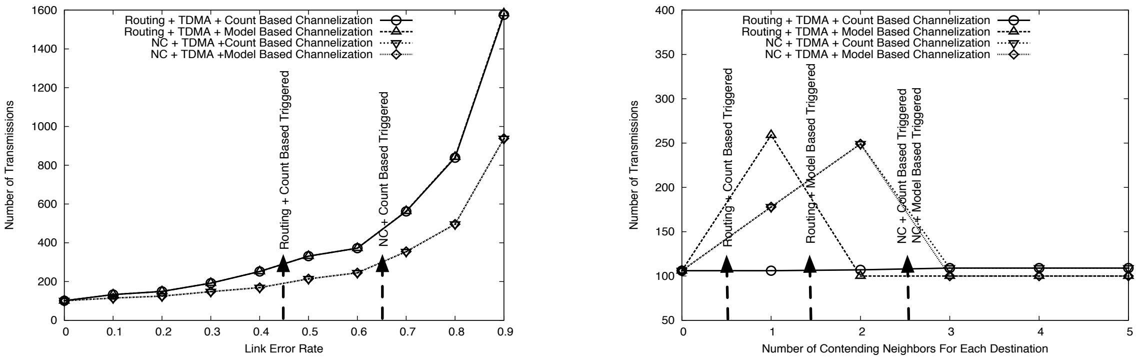
Comparison of Channelization Schemes: To understand the behavior of these two channelization schemes we perform two simulations in ns-2 [9]. In both cases we have a single hop multicast flow from a source to three destinations. For comparison, we apply methods similar to that described above onto a network that uses traditional routing. The count based scheme is easily extended for routing. For the model based scheme, we determine a target p_b using $T(N,B)_{routing}$ from Equation 2 at which the threshold is reached.

We use this scheme with routing to show the benefits of combining network coding and channelization.

For a fair comparison of the channelization policies, we set the threshold as M transmissions per packet. For the model-based policies if the value of $T(N,B)$ is larger than $M*B$, channelization is invoked. We use $M=3.0$ in these simulations. For this threshold, Figure 1(a) shows that the p_b at which model based channelization schemes trigger are 0.57 and 0.71 for routing and network coding, respectively. This shows that the network can tolerate more load (higher collisions) using network coding than with routing alone.

In the first simulation, there are link errors but no collisions. Figure 2(a) shows that as the link error rate increases, the number of transmissions required to deliver the same amount of data also increases. The vertical arrows show the points at which the count algorithms trigger the channelization of the hyperarc due to the crossing of the threshold. The results show that this channelization does not reduce the number of transmissions because the errors are not due to collisions (curves for count and model based schemes for both routing and network coding overlap in Figure 2(a)). Note that neither of the model-based schemes trigger. This shows that the count based channelization algorithm has false positives in the presence of sufficient link errors and wastes resources by channelizing when there is no need.

In the second simulation, there are no link errors and losses occur only due to collisions. As the probability of collision increases, both count and model based algorithms eventually trigger channelization and improve performance as shown in Figure 2(b). Both routing schemes channelize before any scheme employing network coding because, without the reduction in transmissions achieved through network coding, the target collision probability is reached at a lower contention level. The results for both the count and model-based algorithm using $T(N,B)_{nc}$ indicate they trigger at higher background loads because of the reduction in transmissions due to network coding. Note that in this example, they trigger at virtually the same load. In this case, the count algorithm does not result in false positives as all errors are due to collisions.



(a) Varying LER: Number of Background Flows = 0

(b) Varying Contention: Link Error Rate = 0

Fig. 2. Number of Transmissions: Number of Flows = 1, Number of Packets = 100, Packet Size = 512 Bytes

These results indicate that under conditions in which errors occur due to both link impairments and collisions, the model-based algorithm using $T(N, B)_{nc}$ as its estimate will perform best. The count algorithm is likely to be too aggressive in cases of high link errors.

B. How To Channelize

There are several existing mechanisms to channelize links. We define an efficient method below that fits well with our selective channelization algorithm. This method is based on similar assumptions as other previously proposed algorithms use for similar purposes (like Bao et al. [3] and Tang et al. [16]).

Since our channelization scheme is per-flow, we append channel information to each entry in the multicast tables of all the nodes. When routes are destroyed, channelization information is lost, and channels are implicitly unchannelized.

If the hyperarc on which a packet is to be forwarded is not channelized and the selective channelization policy indicates it should be channelized, the node will attempt to channelize the hyperarc. Hence it needs to be aware of the channels used by its two hop neighbors. Since explicit messages to gather two-hop channel utilization information will further burden a network, an implicit mechanism is used. To each data packet that a node broadcasts, it appends bitmaps of the channels that it uses and that its neighbors use (based on best available information). These bitmaps keep the overhead of such piggybacked transmission to a minimum. Each node keeps track of such 1-hop and 2-hop information from all of its neighbors. When a node needs to channelize an outgoing hyperarc, it uses this locally available information to rule out the channels that are being used in the neighborhood. It then picks a random channel from among the available reservation channels.

A drawback of using implicit information is that, at low loads, the information may be stale. Even with fresh neighborhood information, reservation conflicts may arise. If a conflict is detected, a node retains the channel for the hyperarc with a probability of 0.5. If the channel is not retained, the hyperarc is assigned another available reservation channel, based again on the locally stored information. If no reservation channel is available, contention slots are used to transmit packets meant for this hyperarc. This resolution mechanism does not impose any overhead of explicit messages.

V. NETWORK CODING SCHEME

There are several proposed ways in which network coding can be implemented. The network coding scheme that we employ uses most of the salient features of COPE [11]. COPE uses a feedback mechanism built using asynchronous ACKs such that a source, in ideal circumstances, knows which destinations have received which packets and can perform packet combination intelligently. This feedback mechanism is hop-by-hop. The quality of this feedback channel is vital for the performance of network coding.

The focus of our work is on channelization for network coding and not network coding itself. Hence we assume the presence of a perfect feedback channel, i.e., a node is always aware of whether a downstream node received its transmission or not. In fact, channelization aids the design of the feedback scheme. Nodes upstream of the source of a channelized hyperarc will be able to overhear all transmissions that may have been otherwise lost due to contention errors. While this will not make the feedback channel perfect, it will make it more reliable.

Briefly, our network coding scheme adopts the following approach. Each node stores all data packets for all flows that it forwards. Nodes also decode all packets that they receive. Hence we perform encoding as well as decoding on a per-hop basis. A source of a hyperarc estimates which destinations of the hyperarc have received which packets based on the feedback. Any packet that has been received by all destinations of the hyperarc need not be retransmitted in any combination. Among the remaining packets, multiple packets can be combined only if the destinations that do not have these packets have no common nodes (this policy is from COPE [11]). As a result, combining more packets than the number of destinations of a hyperarc is not possible. Packets are combined using XORs as is widely prevalent in network coding literature.

VI. RESULTS

The results presented here were obtained using ns-2.30 [9]. An 802.11 air interface operating at 11Mbps was used for the non-TDMA results. ns-2 provides a single-hop TDMA-MAC that uses the 802.11 physical layer module. We extended this MAC layer for multi-hop networks for the TDMA results. The COPE-like network coding scheme presented in Section V was also implemented in ns-2.

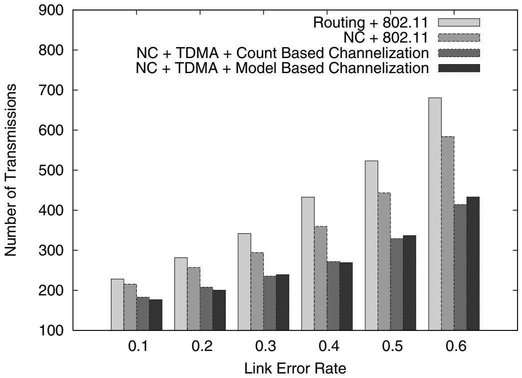
We simulate multicast flows from a source to multiple destinations across multiple hops. Our measured multicast flow has seven destinations. The source transmits to three destinations three hops away and two destinations each one hop and two hops away. Hence, there are three hyperarcs in this flow. Because the source and intermediate nodes have at most three downstream nodes, we set the limit on the number of packets that can be combined to three as network coding does not combine more packets than the number of downstream nodes. Also, the number of contention and reservation channels was each set to 4 in these simulations.

In the first simulation, the hyperarc with the flow source as its root is subjected to severe contention, the second hyperarc is under mild contention, and the third hyperarc had little contention. Throughout the simulation, the offered load is kept constant. In these simulations, the threshold for triggering channelization is set to $M = 2.0$ transmissions per packet. These results are an average of 5 runs.

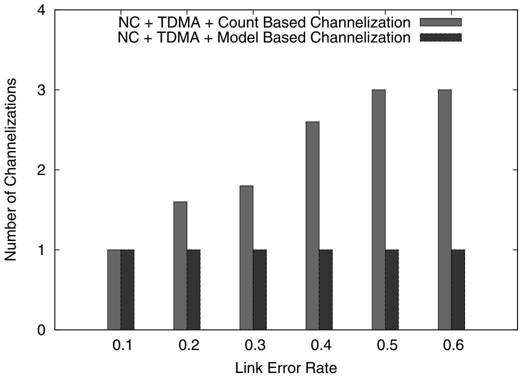
Figure 3(a) shows that as the link error rate increases, the number of transmissions increases for all schemes. The network coding based schemes using TDMA outperform routing and network coding over 802.11. Even at low error rates, the high contention hyperarc is channelized by both the model based and count based schemes. At sufficiently high link error rates, the other two hyperarcs are also channelized by the count based scheme (Figure 3(b)). This illustrates the drawback of false positives in the count based scheme. On the other hand, the model based scheme successfully realizes that only one of the links has high contention error and channelizes only this link for all link error rates. From Figure 3(a), we observe that the performance of both the count based and model based schemes is almost the same even though the count based scheme uses more channels.

In the next set of simulations, the number of links under high contention were varied. The link error rate was set to 0.4 throughout these simulations. The channelization trigger threshold was set to $M = 3.0$ transmissions per packet.

Results in Figures 4(a) and 4(b) were averaged over 5 runs. As the number of hyperarcs under high contention error is increased, both routing and network coding over unchannelized TDMA face performance degradation (Figure 4(a)). Both the count based and model based schemes still maintain their performance level inspite of the increasing number of contention links.



(a) Number of Transmissions



(b) Number of Channelizations

Fig. 3. Varying Link Error Rate: Number of Flows = 1, Number of Packets = 50, Packet Size = 512 Bytes

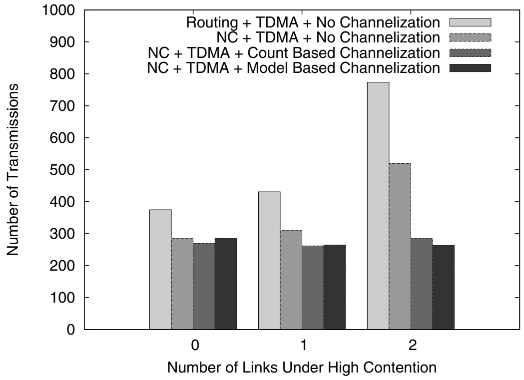
When there are no links under high contention, and just one link is under mild contention, the count based channelization scheme channelizes links (Figure 4(b)), indicating false positives. As the number of links under contention increases, the number of channelized links increases accordingly for the model based scheme. The count based scheme is always more aggressive in terms of channelization even though both schemes result in a similar level of performance.

In summary, the model based scheme results in the same performance in terms of number of transmissions as the count based scheme. In the presence of link errors, the count based scheme is prone to false positives. The model based channelization scheme effectively detects the need for channelization, thereby preventing performance degradation while using fewer channels at the same time.

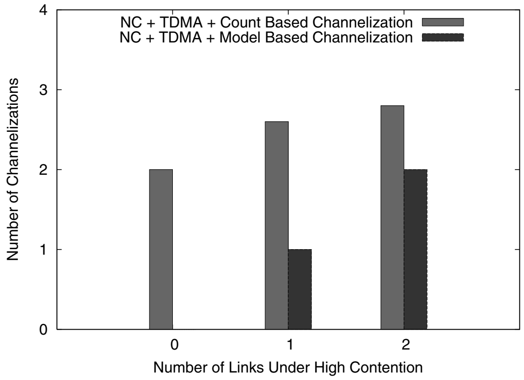
VII. RELATED WORK

Various approaches have been adopted to increase throughput of wireless networks. Ahlswede et al. [1] proposed the concept of network coding and have shown that the multicast capacity of a network cannot be achieved by traditional store-and-forward approach of routing. This capacity can be realized by using network coding. Linear network coding has been shown to be an effective way to approach this capacity [14], [13]. Random linear network coding proposed by Ho et al. [8] approaches this multicast capacity as the code length increases.

Network coding is increasingly being adopted to address problems of bandwidth constraints in different areas. Gkantsidis et al. [7] proposed a scheme which uses network coding



(a) Number of Transmissions



(b) Number of Channelizations

Fig. 4. Varying contention level: Number of Flows = 1, Number of Packets = 50, Packet Size = 512 Bytes, Link Error Rate = 0.4

to enhance delivery of blocks of a file in a peer-to-peer network. Research has also been conducted to bring network coding in the realm of practical use by Chou et al. [4] among others. Katti et al. [11] proposed COPE to leverage scope for combining packets in wireless networks by using opportunistic policies. Katti et al. [10] and Zhang et al. [18] have also proposed to use network coding at the physical layer to avoid dropping packets due to collisions in the medium at a receiver.

Another way to improve network capacity is through scheduling and spatial reuse of channels. Specially at high contention levels for medium access, scheduling has been an important mechanism to avoid large-scale contention. Since the number of channels at disposal is limited, spatial reuse of multiple available channels has been intensively researched. But optimal link or broadcast scheduling in wireless networks has been known to be an NP-complete problem [2], [5]. Works such as [12], [16] and [17] have investigated this spectrum of research. Distributed schemes for link and node scheduling were also proposed in Bao et al. [3]. Note that in this work we have considered link channelization as compared to broadcast or node channelization.

To the best of our knowledge, our work is the first to foray into channelization schemes specific to network coding.

VIII. CONCLUSIONS

Even when network coding is employed, under highly congested circumstances the number of transmissions required to successfully deliver data may become prohibitive. We provide an analysis of the gains to expect out of network coding in a multicast setup. Based on this analysis, we propose a selective channelization scheme to recognize the opportunity to use channelization to prevent performance degradation of network coding in the face of high contention errors. Our results show that our model based channelization trigger criteria is an effective scheme to use fewer channels while maintaining the performance of network coding.

ACKNOWLEDGMENT

This work was supported by DARPA grant BAA 05-42. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of DARPA.

REFERENCES

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, “Network information flow,” *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] E. Arikan, “Some complexity results about packet radio networks (Corresp.),” *IEEE Transactions on Information Theory*, vol. 30, no. 4, pp. 681–685, 1984.

[3] L. Bao and J. J. Garcia-Luna-Aceves, “A new approach to channel access scheduling for ad hoc networks,” in *MobiCom ’01*. New York, NY, USA: ACM Press, 2001, pp. 210–221.

[4] P. Chou, Y. Wu, and K. Jain, “Practical network coding,” *Allerton Conference on Communication, Control, and Computing*, 2003.

[5] A. Ephremides and T. Truong, “Scheduling broadcasts in multihop radio networks,” *Communications, IEEE Transactions on*, vol. 38, no. 4, pp. 456–460, 1990.

[6] C. Fragouli and J. Widmer, “Network coding: an instant primer,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, 2006.

[7] C. Gkantsidis and P. Rodriguez, “Network coding for large scale content distribution.” in *INFOCOM*, 2005, pp. 2235–2245.

[8] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inform. Theory*, 2004.

[9] http://www.isi.edu, “The Network Simulator - ns-2,” 2000. [Online]. Available: http://www.isi.edu/nsnam/ns/

[10] S. Katti, S. Gollakota, and D. Katabi, “Embracing Wireless Interference: Analog Network Coding,” *SIGCOMM*, 2007.

[11] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “Xors in the air: practical wireless network coding,” in *SIGCOMM ’06*. New York, NY, USA: ACM Press, 2006, pp. 243–254.

[12] T.-S. Kim, J. C. Hou, and H. Lim, “Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks,” in *MobiCom ’06*. New York, NY, USA: ACM Press, 2006, pp. 366–377.

[13] S. Li and R. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[14] M. Médard and R. Koetter, “Beyond routing: An algebraic approach to network coding.” in *INFOCOM*, 2002.

[15] R. Nelson and L. Kleinrock, “Spatial TDMA: A Collision-Free Multihop Channel Access Protocol,” *IEEE Transactions on Communications*, vol. 33, no. 9, pp. 934–944, 1985.

[16] Z. Tang and J. Garcia-Luna-Aceves, “Collision-Avoidance Transmission Scheduling for Ad-Hoc Networks,” *Proc. IEEE ICC*, 2000.

[17] R. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar, “Component based channel assignment in single radio, multi-channel ad hoc networks,” in *MobiCom ’06*. New York, NY, USA: ACM Press, 2006, pp. 378–389.

[18] S. Zhang, S. C. Liew, and P. P. Lam, “Hot topic: physical-layer network coding,” in *MobiCom ’06*. New York, NY, USA: ACM Press, 2006, pp. 358–365.

APPENDIX A
NETWORK CODING ANALYSIS

Here we analyze network coding gain over a single hyperarc based on the average number of transmitted packets. Let there be N destinations on this hyperarc on which B packets are to be delivered. For simplification, assume that all the destinations of the hyperarc have independent packet loss probability of p_b. Let p_i denote the probability that the channel is idle (i.e., $p_i = 1 - p_b$). All possible outcomes of a set of transmissions can be classified into the following groups:

1) All B packets are received at all N destinations
2) Only $l(< B)$ packets are received at all N destinations.
    a) All $m = (B - l)$ packets not received at any destination
    b) Only $j(< m)$ packets not received at any destination. Hence $k = (m - j)$ packets are received by at least one but not all destinations.

Hence the average number of transmissions $T(N, B)$ is defined as:

$$T(N,B) = B(p_i^N)^B + \sum_{l=0,m=B-l}^{B-1} \binom{B}{l} X_1 + \sum_{l=0}^{B-1} \binom{B}{l} X_2 \tag{8}$$

The second term of Equation 8 represents case 2(a) and X_1 is defined as:

$$X_1 = (B + T(N,m))(p_i^N)^l (p_b^N)^m \tag{9}$$

The third term of Equation 8 represents case 2(b) (when some of $m = (B - l)$ packets are received by some of the destinations) and X_2 is defined as:

$$X_2(l,j,k) = p_i^{Nl} \sum_{j=0}^{B-l-1} p_b^{Nj} [1 - p_i^N - p_b^N]^k \binom{B-l}{j} X_3(l,j,k) \tag{10}$$

Equation 8 is a general definition of the average number of transmissions. The differences in the expected number of transmissions for routing and network coding arise in the expression for X_3. We now analyze expressions for X_3 for routing and network coding.

A. Routing

For routing, the source has to retransmit k packets which need to be delivered to only some of the destinations and hence:

$$\begin{aligned} X_3^{rout}(l,j,k) =& [B + T(N,j) + \sum_{a=1}^{k} T(n_a,1)] \\ \approx& [B + T(N,j) + kT(N,1)] \end{aligned} \tag{11}$$

, where n_a is the number of destinations that do not receive packet a. For simplification, we assume that $T(n_a, 1) = T(N, 1)\ \forall n_a \in [1, N - 1]$.

We can compute $T(N, 1)$ (note that it is the same for routing and network coding) which is the average number of transmitted packets for a source to deliver a packet to N destinations as:

$$\begin{aligned} T(N,1) &= p_i^N + [1 + T(N,1)]p_b^N \\ &+ \sum_{k=1}^{N-1} \binom{N}{k} p_i^{N-k} p_b^k (1 + T(k,1)) \\ &= \frac{p_i^N + p_b^N + \sum_{k=1}^{N-1} \binom{N}{k} p_i^{N-k} p_b^k (1 + T(k,1))}{1 - p_b^N} \end{aligned}$$

B. Network Coding

With N destinations and B packets, the delivery of packets can result in any of 2^{NB} possibilities. Also, the number of cases when $B - l$ packets are not received by any node is $\sum_{l=1}^{B-1} \binom{B}{l} = 2^B - 2$ while the number of cases when some of $B - l$ packets are received by some destinations is $Y_b = 2^{NB} - \left(\sum_{l=1}^{B-1} \binom{B}{l} + 2 \right) = 2^{NB} - 2^B$.

For ease of analysis, we use a network coding policy in which the source tries to find a network coding packet that codes the maximum number of lost packets. Also, if different packets are lost at the same destination, they cannot be combined. We designate the first destination as a basis. For $i \in [1, k-1]$ lost packets at the basis user, one encoded packet and $i - 1$ native packets need to be delivered. The encoded packet is generated by combining a packet lost at the basis user with other $k - i$ packets that the basis user received but other users might not have received. The number of cases in which we can apply network coding with the policy described above, Y_i, is defined as

$$\begin{aligned} Y_i = \binom{k}{i} (2^{N-1} - 2)^{i-1} \sum_{x_1=0}^{N-2} \\ \left[\binom{N-1}{x_1} \prod_{s=2}^{k-i} \left(\sum_{x_s=1}^{N-1-\sum_{t=1}^{s-1} x_t} \binom{N - 1 - \sum_{t=1}^{s-1} x_t}{x_s} \right) \right] \end{aligned} \tag{12}$$

For each case, the number of packets that the source retransmits is $(i - 1)T(N, 1) + T(N, 1) = iT(N, 1)$. Also, the total number of cases that we can apply network coding is defined as $Y_{nc} = \sum_{p=1}^{k-1} Y_p$. Hence, the number of ways in which we cannot apply network coding is $Y_b - Y_{nc}$. In these cases, the source should retransmit k packets to the destinations as routing. The number of transmissions is $kT(N, 1)$ based on earlier-mentioned assumption $T(g_a, 1) \approx T(N, 1)$.

For k packets that at least one, but not all, destination receives, all possibilities can be classified into two groups. The first group, defined by X_{no-nc}, consists of the cases that the source cannot apply network coding.

$$X_{no-nc} = (1 - \frac{Y_{nc}}{Y_b})(B + T(N,j) + kT(N,1)). \tag{13}$$

The other group, denoted by X_{nc}, is for the source to apply network coding.

$$X_{nc} = \sum_{i=1}^{k-1} \frac{Y_i}{Y_b} (B + T(N,j) + iT(N,1)) \tag{14}$$

Based on these two groups, we can evaluate X_3 as:

$$X_3^{nc}(l,j,k) = (X_{no-nc} + X_{nc}) \tag{15}$$