Performance Modeling of Multicast with Network Coding in Multi-Channel Multi-Radio Wireless Mesh Networks

Hoang Lan Nguyen
Department of Computer Science and Engineering
York University, Toronto, Canada
Email: lan@cse.yorku.ca

Uyen Trang Nguyen
Department of Computer Science and Engineering
York University, Toronto, Canada
Email: utn@cse.yorku.ca

***Abstract*—Systems with multiple channels and multiple radios per node have been shown to enhance throughput of wireless mesh networks (WMNs). Recently, network coding has also been proved to be a promising technique for improving network throughput of WMNs. However, the performance of network coding in the context of multicast, a form of one-to-many communication, in multi-channel multi-radio (MCMR) WMNs is still unknown. In this paper, we present analytical models for estimating the average end-to-end delay, throughput and packet delivery ratio of a network-coded multicast session in 802.11-based MCMR WMNs. The proposed models are then validated using numerical analysis and simulations. To the best of our knowledge, our work is the first that studies the performance of network-coded multicast in MCMR WMNs.**

I. Introduction

In a wireless mesh network (WMN), wireless routers provide multi-hop wireless connectivity from a host to either other hosts in the same network or in the Internet. The wireless routers are often stationary and form a wireless mesh backbone. Our work in this article focuses on this mesh backbone, and we will use the terms “routers” and “nodes” interchangeably.

Multicast is a form of communication that delivers information from a source to a group of destinations simultaneously in an efficient manner. The throughput of each multicast source in a random wireless ad hoc network is upper-bounded by $O(1/\sqrt{n^{\epsilon}\log n})$, where $0 \leq \epsilon \leq 1$ and n is the number of nodes in the network [1]. This upper bound indicates that the throughput capacity of multicast in a single-channel WMN becomes unacceptably low as the network size increases. One of the most effective approaches to enhance network throughput is to use systems with multiple channels and multiple radios (MCMR) per node [2], [3].

Also recently, network coding [4], [5] has received much attention as a promising technique for improving network throughput. Previous studies on the benefits of network coding focus mostly on single-channel networks. In an MCMR system, the performance of network coding in the context of multicast communication is yet unknown.

In this paper, we provide analytical models to estimate the average end-to-end delay, throughput and packet delivery ratio of multicast *with* and *without* network coding in 802.11-based MCMR WMNs. The proposed models are validated using numerical analysis as well as simulations under realistic network settings. To the best of our knowledge, our work is the first that studies the performance of multicast with network coding in MCMR WMNs.

The remainder of the paper is organized as follows. A summary of related work is provided in Section II. We describe the system model and our assumptions in Section III. In Section IV, we present our proposed analytical models. Section V show our numerical and simulation results. We conclude the paper and outline our future work in Section VI.

II. Related Work

In this section, we review prior work on performance analysis and modeling of 802.11-based wireless networks, network coding, and MCMR networks.

There exists a number of studies on the performance of both single-hop and multi-hop 802.11 networks [6], [7]. Bianchi [6] models the behaviour of the exponential backoff time at one unicast node as a discrete Markov chain. It determines the transmission probability and analyzes the saturation throughput under the assumption that in each transmission attempt, regardless of the number of retransmissions, a collision occurs with a constant and independent probability. Medepalli et al. [7] extend Bianchi’s model to multi-hop wireless networks and study the effects of hidden and exposed nodes.

Performance analyses of network coding in wireless networks [8], [9], [10], [11] have gained much attention recently. Random linear coding of packets in a multicast flow was first introduced in [8], which provides a lower bound on the probability that all one-hop multicast receivers are able to successfully decode the data sent by the source and shows that this scheme can outperform a traditional store-and-forward routing mechanism. In [9], the authors consider a network with finite queue buffers for storing packets and propose a scheme for coding packets of a single unicast flow that arrive through a random process. They provide a framework that allows for the computations of the delay and queue blocking probability. The work in [10] provides analytical bounds on the completion time and stable throughput for random linear coding across multiple multicast flows. Eryilmaz et al. [11] quantify the performance gains of network coding in terms of completion time from a single source to one-hop receivers with varying channel conditions, modeled as stochastic changes in on/off state.

All the above models consider wireless networks with a single channel. There have existed only a few analytical studies on the performance of *unicast* flows in MCMR wireless networks [12], [13]. The authors of [12] study the capacity region of a multi-channel multi-radio wireless networks using linear programming. Su et al. [13] extend the work in [12] to model the throughput gain of network coding in two-way, star, and general network topologies.

None of the work above, however, considers the performance of multicast flows in MCMR wireless environments, and in combination with network coding. In this paper, we present analytical models to estimate the average end-to-end delay, throughput, and packet delivery ratio of a multicast session with and without network coding in an MCMR WMN.

III. System Model

We assume that each multicast source or destination is associated with a different wireless mesh router. That is, a multicast group with j destinations consists of j distinct destination routers and one source router, since we are interested in the multicast performance of routers in the mesh backbone. Furthermore, the following system model and assumptions are used in both the analytical modeling and simulations, unless otherwise stated.

A. Medium Access Control

The medium access control (MAC) for multicast uses the basic access procedure of the IEEE 802.11 distributed coordination function (DCF) with carrier sense multiple access and collision avoidance (CSMA/CA) without RTS (request-to-send), CTS (clear-to-send) or ACK (acknowledgment) [14]. At the MAC layer, multicast packets are neither acknowledged nor retransmitted if being lost. Although there exist some works in the literature that propose RTS/CTS mechanisms for multicast, they either incur very long delay (e.g., by polling multicast receivers one by one) [15], [16] or require extensive modifications to the 802.11 MAC protocol [17], [18], [19]. We thus adopt the standard 802.11 CSMA/CA protocol without RTS/CTS/ACK in this paper.

According to the basic access procedure of the 802.11 DCF CSMA/CA protocol, a multicast node with a new packet to transmit monitors the channel activity until it is measured idle for an interval of distributed interframe space (DIFS). After the DIFS, a backoff time is randomly selected in the range $[0, W-1]$, where W is the *minimum contention window*. The backoff time counter is decremented as long as the channel is sensed idle, paused when a transmission is detected on the channel, and resumed when the channel is sensed idle again for a DIFS interval. The node transmits when the backoff counter reaches zero. Note that the contention window in the backoff scheme for multicast does not increase in size due to the absence of ACK packets. Note that this is different from *unicast communication*, wherein, after each unsuccessful transmission (indicated by whether or not an ACK is received after the transmission), the contention window is doubled, up to a maximum value [14].

DCF employs a discrete-time backoff scale, meaning the backoff time following a DIFS interval is slotted. The contention window denotes the number of slots a node should wait before transmitting. The values of a *slot_time*, DIFS and W depend on the physical layer settings. For example, for the Direct Sequence Spread Spectrum (DSSS) modulation scheme used in the 802.11b standard, the *slot_time*, DIFS and W are set to 20 μs, 50 μs, and 32 (*slots*), respectively. We also assume that the packet size does not exceed the maximum size allowed by the physical layer so that the packet can be transmitted in one transmission with no fragmentation.

B. Network Coding

With respect to network coding, we use the *intra-flow* random linear coding to combine packets within a single multicast flow. When a multicast source with network coding function has a file to deliver to its multicast group, it divides the file into batches, each having K packets. (In theory, the whole file could be treated as a batch, but the encoding/decoding time at the multicast forwarders would be high if the file size is very large.) To simplify the analysis, we assume $\omega = 1$ in our analytical models, i.e., the file consists of one batch; in other words, the file is simply divided into K packets. These K uncoded, original packets are called *native packets* and K is the *batch size*. When the source is ready to send a batch, it creates a coded packet, which a random linear combination of the K native packets, and broadcasts the coded packet. When a multicast forwarding node receives a coded packet, it checks whether the packet is an *innovative packet* and keeps only innovative packets. A packet is innovative if it is linearly independent from the packets the node has previously received. The node then creates a new coded packet by generating a random linear combination of the innovative coded packets it has received, and broadcasts it to its neighbors. Upon receiving a packet, a multicast destination checks whether the packet is innovative and discards the packet if it is not. Once the destination receives K innovative packets belonging to a batch, it can decode the batch and obtain the native packets using a simple matrix inversion [20].

C. Multi-Channel Multi-Radio Systems

We consider multi-channel wireless mesh networks with multiple radios per node. Two nodes u and v are directly connected and form a communication link (u, v) if they are within the transmission range of each other and share a common channel. Each node is equipped the same number of radios r and the network has C orthogonal (non-overlapping) channels. We assume that the network uses a channel assignment (CA) scheme that ensures that, at any point in time, the number of distinct channels assigned to any node is less than or equal to the number of radios the node possesses. As a result, each radio is bound to a specific distinct channel and no channel switching is needed. There exist many such CA algorithms [21], [22], [23] that produce results satisfying the above condition.

D. Queueing Model

We model a router with r radios and queue capacity of Q using the $M/M/r/Q$ queue [24] with the inter-arrival time and the service time being exponentially distributed. Each router is assumed to have r independent servers, as each of the r radios can process and broadcast packets in parallel over different channels. All multicast packets waiting for service and those being served are kept in one multicast queue of capacity Q, meaning that the queue can hold up to a maximum of Q packets (we assume that each router maintains a separate queue for multicast flows). When the queue is full, all newly arriving packets are dropped. We assume that Q is greater than or equal to r; otherwise some radios would fail to operate due to lack of queueing buffers [24]. The service policy is first in first out (FIFO) and each server has a mean service rate of μ, which will be derived later for regular and network-coded systems in Sections IV-B1 and IV-C1, respectively. We assume that if the multicast source sends data packets at a constant bit rate of λ, then the average packet arrival rate at a multicast router is approximately λ.

Let p_m be the probability that there are m packets in a router, including packets being processed and those waiting to be processed. According to the $M/M/r/Q$ model, we have

$$p_0 = \left[1 + \frac{(1-\rho^{Q-r+1})(r\rho)^r}{r!(1-\rho)} + \sum_{i=1}^{r-1} \frac{(r\rho)^i}{i!}\right]^{-1}, \quad (1)$$

and

$$p_m = \begin{cases} \rho^m r^m p_0/m! & m \in [1, r-1] \\ \rho^m r^r p_0/r! & m \in [r, Q], \end{cases} \quad (2)$$

where $\rho = \lambda/(r\mu)$ is the traffic utilization, p_0 represents the probability that the router is empty, and p_Q represents the probability that the router is full. Using the expressions for p_m in (1) and (2), we can determine the expected number of packets $E[m]$ in a router as follows:

$$E[m] = \sum_{m=1}^{Q} m p_m = p_0 \left[\sum_{m=1}^{r-1} \frac{\rho^m r^m}{m!} + \sum_{m=r}^{Q} \frac{\rho^m r^r}{r!}\right] \quad (3)$$

IV. The Proposed Performance Models

In this section, we present our analytical models and provide closed-form expressions for estimating the performance of a multicast session with and without network coding in multi-channel multi-radio wireless mesh networks. Our analysis is divided into two parts. First, we analyze the performance of a multicast session without network coding, which we refer to as **Re**gular **M**ulticast (ReM). Previous work on performance analysis for ReM has been mostly experiment-based, protocol-specific, and focused on systems with a single channel. Here, we provide a theoretical analysis, which is independent of any multicast routing protocol, and in MCMR networks. Second, we present performance models of **Net**work-**Co**ded **M**ulticast (NetCoM) in MCMR networks, a problem that has not been addressed prior to this work.

In the analyses, we consider the following performance metrics:

- **Average end-to-end delay**. The end-to-end delay of a packet received at a multicast destination is defined as the latency between the time the packet is transmitted from the multicast source and the time the packet is received at the destination. The average end-to-end delay is the average of the end-to-end delays of all the packets received at all multicast destinations.
- **Average packet delivery ratio**. The packet delivery ratio (PDR) of a multicast destination is the ratio of the number of packets received by the destination and the number of packets sent by the multicast source. The average end-to-end PDR of a multicast group is the average of the PDRs of all multicast destinations in the group.
- **Average throughput**. The throughput of a multicast destination is defined as the total number of packets the destination receives divided by the interval starting from the time the multicast source begins transmitting the first packet to the time the destination receives its last packet. The average taken over the throughputs of all multicast destinations is the average throughput of the group.

To model the performance of a multicast session using the above metrics, we need to consider the following time components incurred by a packet, from the time it arrives at a node to the time it is forwarded to a neighbouring node:

- *processing time*: time to react to an incoming event. Since the processing time is typically negligible compared to the other time components, we ignore this latency in our analyses.
- *queueing time*: time the packet waits in the queue (buffer) before being considered and forwarded.
- *backoff time*: time incurred by 802.11 CSMA/CA algorithm.
- *transmission time*: time for the transmitter to send out all bits of the frame.
- *propagation time*: time for a bit to travel from the transmitter to the receiver.
- *coding opportunity delay*: time the packet has to wait before it can be combined with other packets of the same flow for coding [25]. The waiting is caused by packets from different flows interleaving in the buffer. We do not consider the coding opportunity delay in this paper because we assume an *intra-flow* network model where all packets buffered at a node belong to the same flow, as in [20], [26]. We will extend the models to *inter-flow* network coding in our future work.

We begin the analysis by estimating the average backoff time of multicast nodes at the MAC layer. We then present the performance models of ReM and NetCoM in Sections IV-B and IV-C, respectively.

A. Estimating the Average Multicast Backoff Time

In this section, we model the average backoff time a node waits before transmitting a multicast packet. The model follows the standard 802.11 CSMA/CA protocol described in Section III-A.

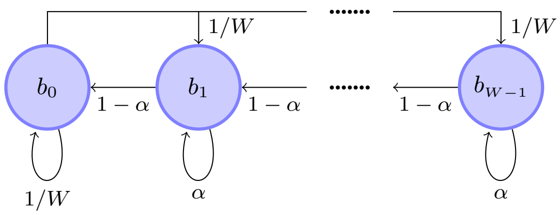


Fig. 1: Markov chain model for the backoff time at multicast node.

We assume that a channel is busy, i.e., some node is transmitting on the channel, with a constant and independent probability α. Then, it is possible to model the 802.11 backoff scheme with a discrete-time Markov chain depicted in Fig. 1. For a given multicast node, let b_k be the steady-state stationary distribution that the backoff time counter is equal to k, where $k \in [0, W-1]$, and $P\{i \rightarrow j\}$ be the one-step transition probability from state i to state j. In this Markov chain, the non-null one-step transition probabilities are as follows:

$$\begin{cases} P\{k \rightarrow k\} & = \alpha & k \in [1, W-1] \\ P\{k \rightarrow k-1\} & = 1-\alpha & k \in [1, W-1] \\ P\{0 \rightarrow k\} & = 1/W & k \in [0, W-1] \end{cases} \tag{4}$$

The first equation in (4) models the fact that the backoff counter pauses when the node senses that the channel is busy. The second equation accounts for the fact that the backoff counter decrements when the node senses that the channel is idle. The third equation considers the fact that once the backoff counter reaches zero, the node can transmit its packet and starts a new backoff period for a new transmission. The new backoff time interval is again chosen randomly in the range $[0, W-1]$, and thus having the probability $1/W$ of being in one of the W states, from 0 to $W-1$.

According to the chain regularities, we have

$$\begin{cases} b_1 & = b_0/W + \alpha b_1 + (1-\alpha)b_2 \\ b_2 & = b_0/W + \alpha b_2 + (1-\alpha)b_3 \\ \vdots & \vdots \\ b_{W-2} & = b_0/W + \alpha b_{W-2} + (1-\alpha)b_{W-1} \\ b_{W-1} & = b_0/W + \alpha b_{W-1} \end{cases} \tag{5}$$

If we rewrite (5) and express b_k values as functions of b_0, we get

$$b_k = \frac{W-k}{W} \times \frac{1}{1-\alpha} b_0, \qquad k \in [1, W-1] \tag{6}$$

The value b_0 is determined by imposing the normalization condition that the sum of all the state probabilities must be one and by substituting b_k values with $k \in [1, W-1]$ using (6):

$$1 = \sum_{k=0}^{W-1} b_k = b_0 + \sum_{k=1}^{W-1} \frac{W-k}{W} \times \frac{1}{1-\alpha} b_0 \tag{7}$$

From (7), we obtain:

$$b_0 = \frac{2(1-\alpha)}{W - 2\alpha + 1} \tag{8}$$

Since the equation for b_0 in (8) still has the variable α whose value is yet unknown, we now show how to obtain α and b_0. As a transmission occurs when the backoff counter equals zero, b_0 is also the probability that a node transmits at a randomly chosen time. Hence, the probability that a node does not transmit is $(1-b_0)$. Let $n \geq 1$ denote the number of nodes in the network. To simplify the analysis, we assume that nodes are within the interference range of each other. In case of a single-channel network, n nodes contend for the sole channel; the probability that all n nodes do not transmit (i.e., the channel is idle) is thus $(1-b_0)^n$. We are, however, interested in networks with multiple channels. Given C as the number of non-overlapping channels available in a multi-channel network, n nodes can be grouped into C autonomous regions, each of which does not interfere with the others and consists of n/C nodes. Therefore, the probability that a channel is idle is $(1-b_0)^{n/C}$. Assuming that all C channels are fully utilized, there should always be at least one node contending for a channel, even when $n < C$. (In practice, $n > C$.) We thus rewrite the probability that a channel is idle as $(1-b_0)^{\max\{n/C,1\}}$. (The single-channel equation is a special case of the multi-channel equation in which $C = 1$.) The probability α that a channel is busy in a multi-channel network is then:

$$\alpha = 1 - (1-b_0)^{\max\{n/C,1\}} \tag{9}$$

Using (8) and (9), we obtain the following polynomial equation of degree $(\max\{n/C,1\}+1)$ with $(1-b_0)$ being the unknown:

$$2(1-b_0)^{\max\{n/C,1\}+1} + (W-1)(1-b_0) - (W-1) = 0 \tag{10}$$

Given known values for constants n, C and W, equation (10) can be solved for b_0 using numerical techniques or mathematical tools such as MATLAB, after n/C is rounded to the nearest integer.

The average backoff time depends on the value of the backoff counter and the duration for which the counter pauses when the node detects transmissions from other nodes [27]. Let us first consider the average backoff time interval $E[k]$ in terms of number of slots, *without* taking into account the duration for which the counter is paused. This interval is given by: $E[k] = \sum_{k=1}^{W-1} k b_k$, i.e., when the backoff counter is at state b_k, a time interval of k slots is needed for the counter to reach zero. Substituting (6), (8), (9) into $E[k]$, we get

$$E[k] = \frac{(W+1)}{3}(1-b_0) \tag{11}$$

Next, let us denote φ as the average duration for which the backoff counter remains paused. Since the mean number of consecutive idle slots the backoff counter decrements before it pauses due to the channel being busy is $(1-\alpha)/\alpha$, the average number of times the counter pauses during the $E[k]$ backoff period before it reaches zero is $(\frac{E[k]}{\max\{(1-\alpha)/\alpha,1\}}-1)$. The max function ensures that there should be at least one idle slot in every busy slot; otherwise the backoff counter would never reach zero. Given that once the counter pauses, it remains paused for the duration of a packet transmission, which is equal to S/B, where S is the packet size and B is the channel bandwidth at the physical layer, plus a period of DIFS, the total pause time is

$$\varphi = \left(\frac{E[k]}{\max\{(1-\alpha)/\alpha,1\}}-1\right) \times (\frac{S}{B} + \text{DIFS}) \tag{12}$$

The average backoff time β is the sum of the initial DIFS period, the backoff time without pausing $E[k]$, and the pause duration φ.

$$\beta = \text{DIFS} + E[k] \times \textit{slot\_time} + \varphi \tag{13}$$

Since $E[k]$ in (11) is expressed in terms of number of slots, to convert it to a proper time unit (e.g., microsecond), we multiply it by the *slot_time*.

B. Regular Multicast (ReM) Performance Modeling

1) Average End-to-End Delay: The average service time a node takes to serve a packet is the sum of the average backoff time β given by (13) and the transmissions time S/B. The average service rate μ at a regular multicast router is thus

$$\mu = \frac{1}{(\beta + S/B)} \tag{14}$$

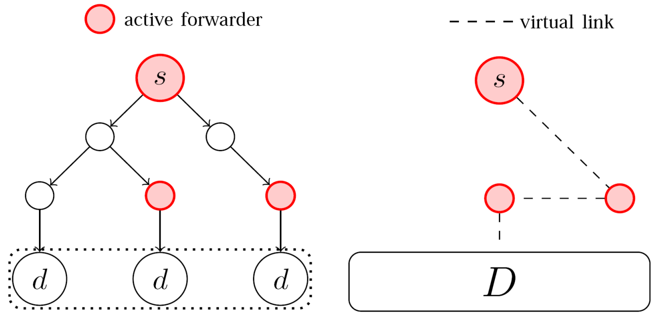
Let L denote the average node latency experienced by a packet at a node u, which is the period from the time the packet enters u to the time u completely transmits the packet back into the network. More specifically, L is the sum of the service time $(\beta + S/B)$ and the time waiting in the queue at u. L can be determined by Little's law [24] as follows:

$$L = \frac{E[m]}{\lambda(1-p_Q)}, \tag{15}$$

where λ is the packet arrival rate, p_Q is the probability that there are Q packets in u, and $E[m]$ is the mean number of packets in u, defined in (2) and (3), respectively. Since packets are dropped when the node is full, i.e., when there are Q packets, p_Q can be considered as the dropping probability and $(1-p_Q)$ as the absorbing probability. The term $\lambda(1-p_Q)$ is hence also known as the effective arrival rate of node u. From the value of μ given by (14), we compute the traffic utilization $\rho = \lambda/(r\mu)$, which then allows us to compute p_Q and $E[m]$ using (1), (2) and (3).

Consider a link (u,v) with u being the transmitter and v, the receiver. The sum of the average node latency L and the propagation time, which can be estimated as the transmission range R divided by the speed of light c, is the average point-to-point delay δ it takes for node u to deliver a packet to node v:

$$\delta = L + \frac{R}{c} \tag{16}$$



(a) Multicast structure at time t

(b) A virtual path at time t

Fig. 2: Virtual path from s to D.

For a multicast session, let $|F|$ denote the number of forwarding nodes needed in the multicast structure (e.g., a tree or mesh) to forward data packets to the multicast destinations. We note that the shortest possible path length from the multicast source s to a multicast destination d, in terms of number of hops, is one. The shortest end-to-end delay would thus be the point-to-point delay δ. On the other hand, in the worst-case scenario where the multicast structure is a straight line, the longest possible source-to-destination path is $|F|$ hops long. The longest end-to-end delay would thus be $(|F| \times \delta)$. To derive the average end-to-end delay of a group of multicast destinations with different distances from the source, we approximate the average path length of the multicast group as follows. We model the multicast group at an instance in time as one *super-destination* D, a concept introduced in [28] and commonly seen in performance analysis for representing a group of entities [29], [30], [31]. Let ξ, $0 \le \xi \le |F|$, denote the number of active forwarders whose queues are not empty. At any given time t, only non-empty forwarders are active to forward packets to the multicast group or, more specifically, to the super-destination D. Therefore, at time t, the multicast structure can be viewed as a virtual path connecting ξ forwarders to D, and thus D can be said to be ξ hops away from the source, as illustrated in Fig. 2. The average number of hops from the source to D can thus be approximated by the expected number of non-empty forwarders $E[\xi]$ at any moment in time. As a result, an average of $E[\xi]$ transmissions are needed to deliver a packet. In a single-channel network, this would result in an average end-to-end delay of $(E[\xi] \times \delta)$, as each transmission requires a point-to-point delay of δ. However, in a multi-channel network with C channels, $E[\xi]$ transmissions can be parallelized over the C channels, potentially reducing the average end-to-end delay Δ to:

$$\Delta = \max\{E[\xi]/C, 1\} \times \delta \tag{17}$$

The max function ensures that the average end-to-end delay Δ should never be less than the average point-to-point delay δ, as any packet arrived at a destination must have traversed over at least one link.

We now show how to derive $E[\xi]$. Given in (1) the probability p_0 that a multicast queue is empty, the probability that the queue of a multicast forwarder is non-empty is $(1-p_0)$. Hence, among all $|F|$ forwarders, the expected number of forwarders with non-empty queues is

$$E[\xi] = (1 - p_0)|F| \tag{18}$$

In our model, $E[\xi]$ is considered as the *effective number of forwarders* of a multicast group and the set of effective forwarders is called the *effective forwarding set*. Using (3), (15), (16), (18), we can rewrite the average end-to-end delay Δ in (17) as follows:

$$\Delta = \max\left\{\frac{(1-p_0)|F|}{C}, 1\right\} \times \left[\frac{\sum_{m=1}^{Q} m p_m}{\lambda(1-p_Q)} + \frac{R}{c}\right], \tag{19}$$

where the p_m values are computed using (1), (2) and variable μ derived in (14). Other variables such as $|F|$, C, R, λ are known parameters whose values depend on the network settings. Note that the number of forwarding nodes $|F|$ can be obtained from the underlying routing multicast structure.

2) Average Packet Delivery Ratio: We assume a packet transmission over a link (u, v) from node u to node v fails if either the transmission collides with other transmissions from other nodes, or the packet is dropped by v because the queue at v is full. Suppose that there are n nodes in a single-channel network. The probability that a transmitted packet from u encounters a collision would be approximately the probability that at least one of the $(n-1)$ remaining nodes transmits. Given a network with C channels, if we group n nodes into C autonomous, non-overlapping regions, each of which does not interfere with the others and consists of n/C nodes, then the collision probability in the network can be approximated as the probability that at least one of $(n/C - 1)$ nodes transmits. As the probability that a node transmits is b_0, the probability that collision occurs is

$$P[\text{collision}] = (1 - (1 - b_0)^{\max\{n/C,1\}-1}) \tag{20}$$

The $\max$ function ensures that there should be at least one node per region regardless of how large C is.

Since p_Q is the probability that v is full, it is the packet dropping probability at v. The probability that v is not full and no collision occurs is

$$P[\text{not full, no collision}] = (1 - q_Q)(1 - P[\text{collision}]) \tag{21}$$

Then, the link error probability ε that a packet transmission over a link (u, v) fails due to either a collision or a full queue is:

$$\varepsilon = 1 - P[\text{not full, no collision}] \tag{22}$$

We again use the "super-destination" concept introduced in Section IV-B1 to approximate the average packet delivery ratio Ω of a multicast group. In particular, since the super-destination D is $E[\xi]$ hops away from the source, and each hop experiences the link error probability ε, the probability that a packet is successfully delivered to the multicast group is

$$\Omega = (1-\varepsilon)^{E[\xi]} = (1-\varepsilon)^{(1-p_0)|F|} \tag{23}$$

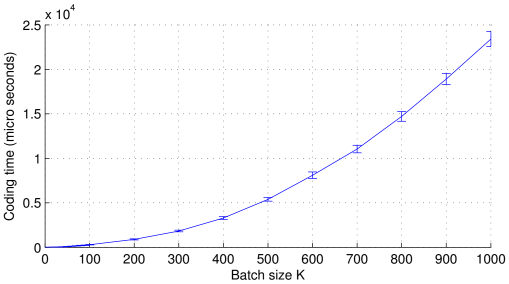


Fig. 3: Coding time versus batch size K.

3) Average Throughput: We begin by estimating the throughput Γ_d of an arbitrary multicast destination d. It is defined as the total amount of multicast packets d received divided by the time it takes d to receive all this data.

The total number of packets d received can be estimated as the number of packets ℓ the source sent multiplied by the average packet delivery ratio Ω given in (23). The time it takes d to receive all the packets is the time it takes the source to transmit all the packets, which ℓ/λ, plus the time it takes the last packet to travel from the source to d, which can be approximated by the average end-to-end delay Δ determined in (19). The throughput of d is thus

$$\Gamma_d = \frac{\ell \times \Omega}{\ell/\lambda + \Delta} \tag{24}$$

Since Γ_d is computed using the average packet delivery ratio and average end-to-end delay of the multicast group, it can be considered as an approximation of the average throughput Γ of the multicast group as defined at the beginning of Section IV. We also substitute the average packet delivery ratio Ω using (23) and obtain the average throughput Γ as follows:

$$\Gamma = \frac{\ell(1-\varepsilon)^{(1-p_0)|F|}}{\ell/\lambda + \Delta} \tag{25}$$

C. Network-Coded Multicast Performance Modeling

In this section, we present our closed-form expressions for estimating the average end-to-end delay, packet delivery ratio and throughput of a network-coded multicast (NetCoM) session in MCMR WMNs. In the following analysis, unless otherwise stated, we use the prime symbol ($'$) to indicate a variable in NetCoM. For example, if μ is the average service rate of a regular node, μ' is the average service rate of a network-coded node.

1) Average End-to-End Delay: Compared to ReM, besides the overall backoff time β and the transmission time S/B, the average service time at a multicast forwarder in NetCoM has an additional element, the *coding time*. The coding time ϕ depends on the batch size K, which is the number of native packets coded in one batch. By measuring the coding time as a function of K, our empirical results (Fig. 3) show that ϕ can be estimated as a quadratic polynomial function of K: $\phi = \sigma_2 K^2 + \sigma_1 K$, where σ_1 and σ_2 are polynomial coefficients and depend on the processing power of routers. Note that $\phi = 0$

when $K = 0$. From (14), the average service rate of a NetCoM forwarder is then:

$$\mu' = \frac{1}{\beta + S/B + \phi} \tag{26}$$

Using the same analysis as in ReM, we derive the average point-to-point delay for a NetCoM forwarder as $\delta' = (L' + R/c)$, where the average node latency L' of a NetCoM forwarder is computed using the NetCoM service rate μ' in (26).

We group the multicast destinations into a super-destination set D, as discussed in Sections IV-B1 and IV-B2 and, additionally, replace the set of forwarding nodes with a *super-forwarder* F [29], [30], [31] as shown in Fig. 4. Although in practice, F and D may overlap (i.e., a multicast destination may act as a forwarding node) we assume that they are disjoint to simplify the analysis. Given that there are effectively an average of $E[\xi']$ forwarders with non-empty queues in the super-forwarder F, and the average link error probability is ε', which is in the same form as (22) but computed using μ', the average error probability between any pair of forwarders within F is denoted by ε'_F and estimated to be

$$\varepsilon'_F = 1 - (1 - \varepsilon')^{E[\xi']} = 1 - (1 - \varepsilon')^{(1-p'_0)|F|} \tag{27}$$

By assuming that at any given time, the effective number of forwarders in F transmitting to D is also $E[\xi']$, each with an average link error probability of ε', the overall transmission error probability from F to D is estimated to be ε'_F.

A multicast destination d can decode to obtain K native packets when it receives K innovative coded packets. Since there may exist non-innovative packets among those received by d, given a finite field of size q from which coding coefficients are selected, the expected number of coded packets $\bar{K}$ that d should receive before K innovative packets are collected is given by [11]:

$$\bar{K} = \sum_{i=1}^{K} \frac{1}{1 - (1/q)^i} \tag{28}$$

The expression for $\bar{K}$ in (28) is upper-bounded by $Kq/(q-1)$, which is close to K even with reasonably low values of q [11]. For instance, for a Galois field of size $q = 2^8$, on average it is sufficient for a multicast destination d to collect K innovative packets if the total number of coded packets it has received is K. Note that if d receives less than K packets, the decoding at d will fail.

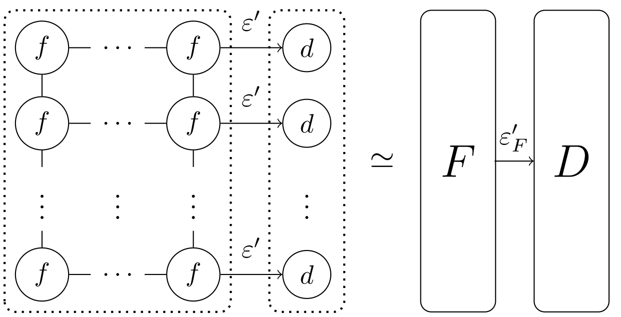


Fig. 4: Super-forwarder F and super-destination D.

Therefore, for the super-destination D to obtain K native packets, F must send to D at least K coded packets. Because in NetCoM, a multicast forwarder uses a new, random coding coefficient set for every transmission, any coded packet transmitted by any forwarder may possibly be an innovative packet. Each forwarder f in F can thus contribute a portion to the required K packets.

We assume that each forwarder in F contributes an equal number of packets. Let π denote the number of packets each forwarder f contributes so that the total number of packets the super-forwarder F sends to D is K. A forwarder f will also overhear and forward (linear combinations of) the coded packets generated by the other forwarders to D. Since the average error probability between any pair of forwarders in set F is ε'_F, f will receive $\pi(1 - \varepsilon'_F)$ packets from any other forwarder. As there are $E[\xi']$ effective forwarders in F, f will receive a total of $\pi(1 - \varepsilon'_F)(E[\xi'] - 1)$ packets from the other effective forwarders. In total, f will have sent κ packets to d, where

$$\kappa = \pi + \pi(1 - \varepsilon'_F)(E[\xi'] - 1) \tag{29}$$

We want κ to be at least K so that D may receive at least K coded packets. Replacing κ in (29) by K, we obtain π as follows:

$$\pi = \frac{K}{1 + (1 - \varepsilon'_F)(E[\xi'] - 1)} \tag{30}$$

In NetCoM, coded packets are transmitted in batches as linear combinations of native packets. We thus consider the average end-to-end delay Δ' of a *batch* instead of individual native packets. On the other hand, Δ' is also the average end-to-end delay of each individual native packet included in the batch (as if the native packets were encapsulated in a virtual data segment).

Using the same model as in ReM, Section IV-B1, each coded packet travels an average distance of $E[\xi']$ hops to reach a multicast destination d. Given the average point-to-point delay δ' and the possibility of parallel transmissions over C channels, a coded packet will experience an average end-to-end delay of $(\delta' \times \max\{E[\xi']/C, 1\})$. In order for the destination d to receive a batch (having at least K coded packets), f must send (contribute) π coded packets. Therefore, the average end-to-end delay Δ' is

$$\Delta' = \pi \times \delta' \times \max\left\{\frac{E[\xi']}{C}, 1\right\} \tag{31}$$

2) Average Packet Delivery Ratio: Using the transformation model in Fig. 4, the total effective number of packets N sent by $E[\xi']$ effective forwarders in F to D is $N = \lceil E[\xi']\pi \rceil$, because each effective forwarder contributes π coded packets to super-destination D. Substituting π using (30), we obtain

$$N = \left\lceil \frac{E[\xi']K}{[1 + (1 - \varepsilon'_F)(E[\xi'] - 1)]} \right\rceil \tag{32}$$

It can be seen that $N \geq K$ because the denominator in (32) is less than or equal to the term $E[\xi']$ in the numerator:

$$[1 + (1 - \varepsilon'_F)(E[\xi'] - 1)] \leq [1 + (E[\xi'] - 1)] = E[\xi']$$

Given the transmission error probability ε'_F from F to D, the average end-to-end packet delivery ratio Ω' of the super-destination D, or the multicast group, is the probability that D receives at least K out of N transmitted packets. This probability can be obtained using a binomial distribution:

$$\Omega' = \sum_{i=K}^{N} \binom{N}{i} (1 - \varepsilon'_F)^i (\varepsilon'_F)^{N-i} \qquad (33)$$

3) Average Throughput: Similar to ReM, the throughput Γ'_d of an arbitrary multicast destination d is defined as the total number of native packets received by d, divided by the interval starting from the time the source begins transmitting to the time d receives the last native packet. In NetCoM, the average of this time interval is actually the average end-to-end delay Δ' of the batch, since all native packets are encapsulated in one batch, as explained in Section IV-C1. Once d decodes the batch successfully, the number of native packets that d receives in the batch is K, and thus the throughput Γ'_d is equal to:

$$\Gamma'_d = \frac{K}{\Delta'} = \frac{K}{\pi \times \delta' \times \max\{E[\xi']/C, 1\}} \qquad (34)$$

Since we use the average end-to-end delay Δ' in (34), the throughput Γ'_d of the arbitrary destination d can be considered as the average throughput Γ' of the multicast group.

In the following section, we present experimental results that validate the proposed models.

V. Numerical and Simulation Results

Using Qualnet [32], a software that provides scalable simulations of wireless networks, we simulate a network of $n = 50$ static nodes uniformly distributed in a 1200m × 1200m area. The channel bandwidth at the physical layer is $B = 11$ Mbits/s; the transmission range of the wireless routers is $R = 315$m, according to the specifications of the wireless routers manufactured by Tropos [33]. The IEEE 802.11 DCF CSMA/CA protocol without RTS/CTS/ACK exchange is chosen as the medium access control protocol for multicast transmissions, as explained in Section III-A. The slot time, DIFS interval, and minimum backoff window size W are set at 20 μs, 50 μs, and 32, respectively, as dictated by the DSSS modulation scheme. The data packet size, excluding header size, is $S = 512$ bytes and the queue capacity at each router is $Q = 50$ packets. The queueing policy is FIFO. The path loss model is two-ray and there is no channel fading.

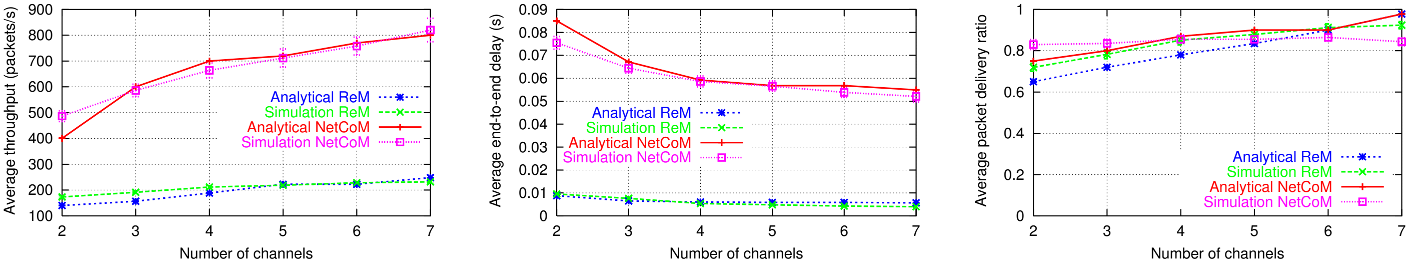
We use UDP at the transport layer in order to evaluate the network performance without any flow, congestion control or reliable mechanisms. The multicast group has one source placed at the center of the network, and 30 destinations randomly selected. The underlying routing algorithm is shortest path trees, built by applying the Dijkstra's algorithm [34] for each source-destination pair. The number of forwarding nodes $|F|$ is computed for each routing tree. The source transmits at a specified constant bit rate $\lambda = 250$ packets/s for 100 seconds of simulated time. The simulator then continues to run for 100 seconds of simulated time to give the last packets time to be routed. Each simulation data point is averaged from 50 runs using different network topologies and random seeds and plotted with a confidence interval of 95%.

We increase the number of channels C from 2 to 7 and measure the average end-to-end delays, packet delivery ratios and throughputs of both ReM and NetCoM as functions of C. The number of radios per node r is set at two for 2 channels and three for 3 to 7 channels. Channels are assigned to wireless links so that the number of distinct channels assigned to a node is not more than the number of radios of the node (as stated in Section III-C). To obtain a fair comparison, the same number of packets $\ell = 25000$ is transmitted by the ReM and NetCoM sources with the ReM source sending uncoded original packets and the NetCoM source sending coded packets. For NetCoM, we select $K = 32$, a common batch size used in network coding experiments [20], [26]. Random coefficients for each linear combination are chosen from a Galois field of size $q = 2^8$, same as in [20], [26]. With this setting, the coding time ϕ is empirically found to be approximately 80 μs.

Using these same input parameters, the numerical results of the proposed models are computed by MATLAB and then plotted against the simulation results obtained from Qualnet, as shown in Fig. 5. The graphs in Fig. 5 show that the numerical results computed from the proposed analytical models are similar to the simulation results. Both the analytical and simulation results show that as the number of channels increases, the performance of ReM and NetCoM improves, as expected. Specifically, increasing the number of channels leads to higher throughput (Fig. 5(a)), shorter end-to-end delay (Fig. 5(b)), and higher packet delivery ratio (Fig. 5(c)).

In addition, we observe that network coding does indeed help improve network throughput significantly (Fig. 5(a)), in agreement with the objective of network coding. In particular, NetCoM average throughput is 2.5-3.5 times higher than ReM throughput. However, this gain comes at the expense of longer end-to-end delay (Fig. 5(b)): NetCoM average end-to-end delay is 8-10 times longer than ReM end-to-end delay. There are two main factors that cause the longer end-to-end delay of NetCoM. First, NetCoM forwarding nodes require additional time for coding the received packets before transmitting. A ReM forwarder would simply forward the (native) packets it just received. Second, upon receiving a coded packet, a NetCoM destination may not be able to decode it right away. It has to wait to receive enough innovative packets (K of them as discussed in the above analysis, where K is the batch size) before decoding them to obtain the native packets enclosed in the batch. The delay to obtain the native packets is thus longer compared with ReM.

On the other hand, the packet delivery ratios of ReM and NetCoM are very close to each other (Fig. 5(c)). This implies that network coding does not help or worsen the PDR of a multicast group, given the above network settings and



(a) Average throughput (b) Average end-to-end delay (c) Average packet delivery ratio

Fig. 5: Analytical results versus simulation results.

simulation parameters.

VI. Conclusion

We present analytical models to estimate the average end-to-end delay, throughput and PDR of a multicast session without and with network coding in an MCMR WMN. The accuracy of the proposed models is validated via simulations using a commercial simulator software, and realistic network settings. From the obtained results, we also show the performance gains and the throughput-delay tradeoff of network-coded multicast in MCMR networks. Our future work includes a comprehensive evaluation of network-coded multicast under different network sizes, group sizes and traffic loads.

References

[1] S. Shakkottai, X. Liu, and R. Srikant, “The Multicast Capacity of Large Multihop Wireless Networks,” in *ACM MOBIHOC’07*, Sep. 2007, pp. 247–255.

[2] R. Draves, J. Padhye, and B. Zill, “Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks,” in *Proc. ACM MOBICOM’04*, 2004, pp. 114–128.

[3] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, “A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks,” in *Proc. BroadNets’04*, Oct. 2004, pp. 344–354.

[4] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, “Network Information Flow,” *IEEE Transactions on Information Theory*, Jul. 2000.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the air: Practical wireless network coding,” in *Proc. ACM SIGCOMM’06*, 2006.

[6] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[7] K. Medepalli and F. Tobagi, “Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications,” in *Proc. IEEE INFOCOM’06*, 2006.

[8] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, “The benefits of coding over routing in a randomized setting,” in *Proc. IEEE International Symposium on Information Theory*, 2003.

[9] D. S. Lun, P. Pakzad, C. Fragouli, M. Medard, and R. Koetter, “An analysis of finite-memory random linear coding on packet streams,” in *Proc. WiOpt’06*, 2006.

[10] R. Cogill, B. Shrader, and A. Ephremides, “Stability analysis of random linear coding across multicast sessions,” in *Proc. IEEE International Symposium on Information Theory*, 2008.

[11] A. Eryilmaz, A. Ozdaglar, and M. Medard, “On Delay Performance Gains From Network Coding,” in *Proc. IEEE CISS’06*, 2006.

[12] M. Kodialam and T. Nandagopal, “Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks,” in *Proc. ACM MOBICOM’05*, 2005, pp. 73–87.

[13] H. Su and X. Zhang, “Modeling Throughput Gain of Network Coding in Multi-Channel Multi-Radio Wireless Ad Hoc Networks,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 593–605, Jun. 2009.

[14] IEEE Std 802.11-2007, “Specific requirements: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” 2007.

[15] K. Tang and M. Gerla, “Random Access MAC for Efficient Broadcast Support in Ad Hoc Networks,” in *Proc. IEEE WCNC’00*, 2000.

[16] M.-T. Sun, L. Huang, A. Arora, and T.-H. Lai, “Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks,” in *Proc. International Conference on Parallel Processing*, 2002.

[17] A. Chen, D. Lee, G. Chandrasekaran, and P. Sinha, “HIMAC: High Throughput MAC Layer Multicasting in Wireless Networks,” in *Proc. IEEE MASS’06*, 2006.

[18] J. Xie, A. Das, and S. Nandi, “An Improvement to the Reliability of IEEE 802.11 Broadcast Scheme for Multicasting in Mobile Ad Hoc Networks,” in *Proc. IEEE SECON’04*, 2004.

[19] X. Wang, L. Wang, Y. Wang, Y. Zhang, and A. Yamada, “Supporting MAC Layer Multicast in IEEE 802.11n: Issues and Solutions,” in *Proc. IEEE WCNC’09*, 2009.

[20] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, “Trading structure for randomness in wireless opportunistic routing,” in *Proc. ACM SIGCOMM’07*, 2007.

[21] A. K. Das, H. M. K. Alazemi, R. Vijayakumar, and S. Roy, “Optimization Models for Fixed Channel Assignment in Wireless Mesh Networks with Multiple Radios,” in *Proc. IEEE SECON’05*, 2005, pp. 463–474.

[22] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot, “Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks,” in *Proc. IEEE INFOCOM’06*, 2006, pp. 1–12.

[23] A. P. Subramanian, H. Gupta, and S. R. Das, “Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks,” in *Annual IEEE SECON’07*, 2007, pp. 481–490.

[24] R. Jain, *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, 1991.

[25] Y. Yuan, K. Wu, and W. Jia, “On the Queueing Behaviour of Network Coding,” in *Proc. IEEE NetCod’10*, 2010.

[26] D. Koutsonikolas, Y. C. Hu, and C. C. Wang, “Pacifier: High-Throughput, Reliable Multicast without “Crying Babies” in Wireless Mesh Networks,” in *Proc. IEEE INFOCOM’09*, 2009.

[27] E. Ziouva and T. Antonakopoulos, “CSMA/CA performance under high traffic conditions: throughput and delay analysis,” *Elsevier Computer Communications*, vol. 25, pp. 313–321, 2002.

[28] R. Khalili, M. Ghaderi, J. Kurose, and D. Towsley, “On the Performance of Random Linear Network Coding in Relay Networks,” in *Proc. IEEE MILCOM’08*, 2008.

[29] R. Khalili and K. Salamatian, “A tighter Cut-Set bound for the multi-terminal erasure channel without side information,” in *Proc. IEEE International Symposium on Information Theory*, 2006.

[30] M. Chen, S. Serbetli, and A. Yener, “Distributed Power Allocation Strategies for Parallel Relay Networks,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, pp. 552–561, Feb. 2008.

[31] J. Liu, N. B. Shroff, and H. D. Sherali, “Understanding the Optimal Power Allocation Structure in AF-Based MIMO Cooperative Networks with Multiple Relays,” Department of Electrical and Computer Engineering, The Ohio State University, Tech. Rep., Aug. 2010.

[32] Qualnet Simulator. [Online]. Available: http://www.qualnet.com

[33] TROPOS Networks. [Online]. Available: http://www.tropos.com

[34] E. W. Dijkstra, “A Note on Two Problems in Connexion with Graphs,” *Numerical Mathematics*, vol. 1, pp. 269–271, Dec. 1959.