# Performance of Group Communication over Ad-Hoc Networks

Marc Mosko and J.J. Garcia-Luna-Aceves
Computer Engineering Department,
University of California at Santa Cruz,
Santa Cruz, CA 95064,
{mmosko, jj}@soe.ucsc.edu *

## Abstract

*We study the performance of reliable and unreliable all-node broadcast over ad-hoc networks that use contention-based channel access. To obtain analytical results while preserving hidden-terminal and node clustering characteristics of ah-hoc networks, we introduce a novel differential-equation fluid model for information flow through a network of cluster trees, where a spanning tree joins groups of fully connected nodes. Through numerical analysis and simulations in GloMoSim, we show throughput, goodput, and loss rates for reliable and unreliable networks. For reliable broadcast, we also find NAK rates, NAK loss rates, and retransmission rates. We show that using end-to-end sequence numbers, which are common in reliable multicast, for NAK generation in ad-hoc networks creates substantial unnecessary traffic.*

## 1. Introduction

Wireless communication enjoys an expanding popularity in todays networks. Ad-hoc networks, such as mobile sensor networks or personal area networks, are entering the market. Ad-hoc networks promote flexibility and mobility by not requiring fixed infrastructure such as cell sites or wireless access points. There are several recommendations for multicast and broadcast in ad-hoc networks. The Internet MANET work group has proposed "Simple Multicast and Broadcast" (SMB) [8] and "Adaptive Demand-Driven Multicast" (ADMR) [9]. SMB operates over DSR Route Discovery packets to flood an ad-hoc network. ADMR uses a combination of network flooding and multicast trees. The unreliable broadcast section of our work models such flooding through a multihop ad-hoc network.

Group multicast or all-node broadcast puts strains on a weak aspect of unscheduled ad-hoc protocols. Collision avoidance protocols, such as IEEE 802.11 DCF [2], use unicast request-to-send (RTS) and clear-to-send (CTS) messages to reserve radio time. Busy-tone protocols use side-band carriers. Such mechanisms do not work for group communication, because multiple nodes would respond or busy tones would collide. These protocols become pure CSMA for broadcasts, which offers no hidden terminal collision protection.

When an application attempts to transmit a stream of group or all-node broadcasts, it creates a self-interfering flow through the network. In a multihop wireless ad-hoc network, the collision rate from hidden terminals can dramatically increase. Receivers cannot reserve the channel through a CTS. A two-hop neighbor, when repeating broadcast traffic, can collide with other sources.

Along the same lines, the packet service rate at repeater nodes may cause multi-hop flow problems. Fig. 1 shows a hidden terminal predicament. Node 1 sends broadcast traffic to nodes 2 – 4. They repeat the traffic for nodes 5 – 7. Without a scheduled MAC layer or a multicast relaying scheme [12, 14, 13], there would be a high likelihood of collision at node 6.

In the case of a multi-point relaying scheme, which generally build minimally connected dominating sets (MCDS), we still have a pacing problem. In Fig. 1, the nodes shown with a box are part of the MCDS and nodes shown with a circle are passive. The MCDS does not alleviate the problems at node 6. We must have coordination or randomization between nodes 2 and 4 to avoid collisions at node 6.

In this paper, we examine these phenomenon with a fluid model [1]. We develop differential equations that model the broadcast/rebroadcast idiom. By studying the instantaneous behavior of coupled equations, we are able to explore several important aspects of correlated transmission streams. Our model is a useful complement to simulations, in which correct assumptions are not always known and causal relationships are sometimes unclear.

Our analysis for unreliable broadcast shows throughput, goodput, and loss metrics. Our analysis for reliable broad-

*This work was supported in part by the Office of Naval Research (ONR) under grant N00014-99-1-0167.

cast also shows NAK generation rate, NAK loss rate, and retransmission rates.

We found that simple NAK generation schemes do not work well in an ad-hoc environment. Although we used local recovery and NAK snooping, the fluid model predicted a lower NAK generation rate than was seen in simulation. The excessive NAK rate in simulation was caused by cascading NAK storms. Each out-of-order packet caused all nodes along its path to generate NAKs for missing packets. We found that combining end-to-end sequence numbers with one-hop radius sequence numbers brought the NAK generation rate down to levels estimated by our fluid model. Our one-hop protocol, however, is not yet a practical scheme.

Section 2 presents our fluid models of ad-hoc broadcast networks. Section 3 illustrates our findings with several numeric examples and a comparison with simulation. Section 4 concludes the paper.

## 2. Fluid Models

The majority of work on broadcasting over wireless networks concerns scheduled networks. These networks require synchronized clocks between network nodes and provide two-hop collision free transmissions by constructing schedules. Only over the last few years have we seen studies on randomized channel access broadcast protocols. These protocols work over ad-hoc wireless networks without centralized base stations or synchronized clocks. These studies do not, however, present significant analysis of broadcast flows or solve the hidden-terminal problem described earlier. In works devoted to broadcast distribution, most use simulation to validate proposals (e.g. [12, 5, 18, 11, 17, 16, 13]). Other works do not consider contention-based channel access and the resulting hidden-terminal losses (e.g. [4]). We have not seen an attempt at broadcast flow analysis over multi-hop contention-based networks.

We present a fluid model for broadcast stream distribution in a cluster tree topology ad-hoc wireless network over a contention-based MAC layer. A cluster tree contains clusters of fully connected nodes. The clusters form a source-based spanning tree, with a single in-bound receiver. Several previous works use a cluster approach for ad-hoc networks [5, 7, 10, 17], though our specific design for analysis is novel. We use an ALOHA-type MAC layer without capture for broadcasts. The MAC does not transmit if it is currently receiving a packet, but otherwise there is no carrier sense and no channel reservation (RTS) or ACK.

The models below address both unreliable and reliable broadcast flows. In the unreliable model, a source node transmits a flow with Poisson rate $\mu_0$ and other nodes rebroadcast the flow at the same rate they receive it. As we use a fluid model, information travels in infinitesimal pieces, with each piece taking $\Delta t$ seconds. This is a *rate* based model, not an *information* based model. Nodes adjust behavior based on the rate of information, not the quantity of information, and we ignore the accumulation (queuing) of information at a node. From simulation results, there is no significant queuing for unreliable broadcast without cross traffic as long as the packet scheduler works faster than the traffic source. The reliable model extends the unreliable model by adding NAKs and retransmissions. The term *rebroadcast* means the first transmissions of a given piece of information by a repeater node and *retransmission* means the repetition of information in response to a NAK. Simulation shows that reliable broadcast may have significant queuing depending on service rates.

We make the following assumptions in our model. A node may not transmit and receive in the same $\Delta t$. If a node hears two or more transmissions in the same $\Delta t$, it understands none of them. A node may instantly detect inbound information rates and adjust it's out-bound rate accordingly in the next $\Delta t$. A node may instantly detect the exact amount of lost information flow and adjust it's NAK rate accordingly in the next $\Delta t$. Likewise, a node may instantly detect the incoming NAK rate and adjust its retransmission rate in the next $\Delta t$.

We assume that all rates are Poisson. We may sum arrivals rates at a sink: $\lambda = \lambda_1 + \cdots + \lambda_k$. The probability of zero arrivals in $\Delta t$ seconds is $(1-\lambda)\cdot\Delta t$ and the probability of exactly one arrival is $\lambda \cdot \Delta t$, with more than one arrival $o(\Delta t)$.

The model uses the following notation for flow rates. For the time being, let node $z$ be the receiver and node $i$ be any 1-hop neighbor of node $z$. $\mu_i$ is the total out-bound flow rate from node $i$. The probability that node $z$ hears node $i$ in a collision free $\Delta t$ is $\pi_{i,z}$. The total collision-free in-bound flow rate from node $i$ to node $z$ is $\mu_i\pi_{i,z}$. $\varepsilon_{i,z}$ is the useful data density. The *useful* flow rate is the amount of non-redundant information for node $z$. Below, we call redundant information *noise*. The useful in-bound flow rate at node $z$ from node $i$ is $\lambda_{i,z} \equiv \mu_i\pi_{i,z}\varepsilon_{i,z}$ and the total useful in-bound flow rate at node $z$ is $\lambda_z = \sum_i \lambda_{i,z}$. For reliable broadcast, node $z$ generates NAKs at the rate $\eta_z$. Non-leaf nodes generate retransmissions at rate $\zeta_j$.

Let the set $N(z)$ be all nodes adjacent to $z$. Let $C(z)$ be the children of node $z$ in a given source-based cluster spanning tree. Let $P(z)$ be the parent of node $z$. $S(z)$ are the siblings of node $z$ (children of $P(z)$ excluding $z$).

Our general network topology uses a tree of fully connected wireless node clusters. Fig. 3 shows an example of a binary tree of height five. There may be any number of nodes per cluster and the tree may be of any degree. The figure shows clusters of size 3, which form a binary tree. The trees do not need to be regular. The specific topology we use ensures that a node either receives data directly from

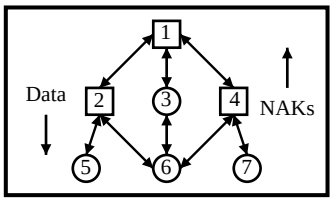


**Figure 1. Multiple Hidden Terminal Collisions**

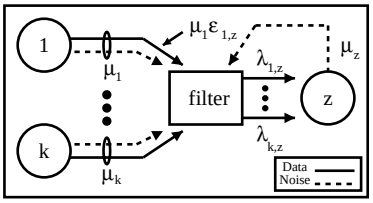


**Figure 2. Cluster flow through filter**

it's cluster root node, or from one of it's siblings. Useful data does not flow up the tree.

Fig. 2 shows how we filter the in-coming data rates. The input to the filter is all out-flow rates at and around node $z$. Rates $\mu_i$ are for all $k$ neighbors of node $z$. As above, the filter output is $\lambda_{i,z} \equiv \mu_i \pi_{i,z} \varepsilon_{i,z}$. The figure shows $\mu_i \varepsilon_{i,z}$ as the useful part of $\mu_i$.

The filter is defined per node per neighbor per $\Delta t$. For node $z$ and neighbor $i$, the filter is based on the sum of all transmit rates in the vicinity of $z$, including $z$ but excluding $i$. This is the combined Poisson arrival rate of traffic that may interfere with the reception of traffic from node $i$.

$$\nu_{i,z} = \sum_{j \in N(z) \cup \{z\} \setminus \{i\}} \mu_j \tag{1}$$

Given the interference rate per $\Delta t$, $\nu_{i,z}$, we find the steady-state probability that node $z$ can hear node $i$ in a given $\Delta t$. The arrival rate at node $z$ from node $i$ may be either 0 or $\mu_i$, modulated by $\nu_{i,z}$. The "on" state represents that a Poisson source is on and received at rate $\mu$. The "off" state means that other transmissions interfere with the source and the received transmission rate is 0. In a $\Delta t$, the state transition matrix $\boldsymbol{P}(t, t+\Delta t)$ for the filter is

$$\boldsymbol{P}_{i,z} = \begin{bmatrix} \nu_{i,z} & 1-\nu_{i,z} \\ \nu_{i,z} & 1-\nu_{i,z} \end{bmatrix} \Delta t$$

where the probability to transition to an "off" state is $\nu \cdot \Delta t$ and to an "on" state is $(1-\nu) \cdot \Delta t$.

The Markov transition rates are

$$q_{ij} = \lim_{\Delta t \to 0} \frac{p_{ij}(t, t+\Delta t)}{\Delta t}$$

for $i \neq j$ [15]. The diagonal elements are $q_{ii} = -\sum_{j \neq i} q_{ij}$.

$$\boldsymbol{Q}_{i,z} = \begin{bmatrix} \nu_{i,z} - 1 & 1-\nu_{i,z} \\ \nu_{i,z} & -\nu_{i,z} \end{bmatrix}$$

From $\boldsymbol{\pi}_i \boldsymbol{Q}_{i,z} = 0$ and $\pi_i(\text{off}) + \pi_i(\text{on}) = 1$, we find $\pi_i(\text{off}) = \nu_i$ and $\pi_i(\text{on}) = 1 - \nu_i$, so $\pi_{i,z} = 1 - \nu_{i,z}$.

## 2.1. Unreliable Broadcast

For each neighbor $i$ of $z$, information flow $\mu_i \pi_{i,z}$ reaches $z$. This quantity contains both noise and data. To determine the amount of useful data, and thus $\lambda_{i,z}$, we define the quantity $\varepsilon_{i,z}$ for each neighbor of $z$. Eq. 2 states that all information from the cluster source, $P(z)$, is useful while all information from the children of $z$ is noise. The children only transmit noise because they only repeat what $z$ has already transmitted. Siblings of $z$ transmit a mixture of noise and data. As an approximation, we shall only consider the first rebroadcast of information by a sibling; we exclude receiving rebroadcasts of rebroadcasts. We further approximate by not considering the delay between loss and rebroadcast.

$$\varepsilon_{i,z} = \begin{cases} 1 & \text{i} = \text{P(z)} \\ 0 & \text{i} \in \text{C(z)} \\ \pi_{i,P(z)} \sum_{j \in C(z)} \mu_j & \text{i} \in \text{S(z)} \end{cases} \tag{2}$$

The probability that a rebroadcast from a Sibling is useful may be defined as the probability that at a given instant, a sibling received data from the cluster source while the node in question did not. For each sibling $i \in S(z)$ with parent $p$, $\varepsilon_{i,z} = \Pr\{\text{node } i \text{ hears } p \text{ and } z \text{ does not}\}$. For any $i$ to hear $p$, it means that $z$ does not transmit. The only way for $z$ to not hear $p$ is for one or more of $z$'s children to transmit over $p$'s transmission.

$$\lambda_{i,z} = \mu_i \pi_{i,z} \varepsilon_{i,z} \tag{3}$$

$$\lambda_z = \sum_{all\ i} \lambda_{i,z} \tag{4}$$

$\lambda_z$ is the combined, assumed Poisson, useful arrival rate at node $z$. The probability of a useful arrival is thus $\lambda_z \cdot \Delta t$, where the $\Delta t$ term applies to each $\mu_i$, as $\pi_{i,z}$ and $\varepsilon_{i,z}$ are already probabilities.

We may now write the differential equation for the flow through a node. At node $i$, the difference equation is $\mu_i(t+\Delta t) - \mu_i(t) = [\lambda_i(t) - \mu_i(t)]\, \Delta t$. This says that in

the next $\Delta t$, node $i$ adjusts its output rate by the change in the useful input rate and the current output rate. The form looks deceptively like a simple birth death process. The $\lambda$ terms, however, are non-linear couplings between nodes. Expanding the right hand side and rewriting as a differential equation, we have

$$\frac{d}{dt}\mu_z = -\mu_z + \sum_{i \in N(z)} \mu_i \pi_{i,z} \varepsilon_{i,z} \tag{5}$$

The summation adds the useful information flow from all of $z$'s neighbors. $\mu_i \pi_{i,z}$ is the filtered transmission rate per neighbor. $\varepsilon_{i,z}$ is the useful density of the flow. By our assumptions, each node adjusts it's instantaneous transmission rate to match the inflow rate of useful data. Thus, at each instance, node $z$ drains at rate $\mu_z$ accounting for the final term.

We may set the initial condition $\mu_i(0)$ to any reasonable value, such as zero or $\mu_0$. For stable networks with suitable initial values, the long-term rates converge on their steady-state value. The initial values affect the short-term transient behavior. We use a value of zero.

### 2.2. Reliable broadcast

In the case of reliable transmission, a node generates NAKs at a rate proportional to the loss from the cluster root. In general, one retransmitted packet could aid several nodes in addition to the node that transmitted the NAK. In what follows, we assume that retransmissions only help the node that sent a NAK. When a cluster root retransmits data, that flow rate only directly helps one cluster child. If that child rebroadcasts the data, the rebroadcast may help any or all siblings. Likewise, when a sibling retransmits data for a child, that transmission cannot aid any node except the child sending the NAK. We treat retransmissions and rebroadcasts as separate types of information, which is not necessarily true of a real network.

Define $\eta$ as the amount of NAK traffic generated at a node. We assume that nodes transmit NAK packets as ALOHA traffic without the benefit of RTS/CTS/ACK handshake. Define $\zeta$ as the amount of retransmission traffic due to $\eta$. The equation for $\lambda$ expands to account for NAK and retransmission traffic. We write these quantities as differential equations. They must be computed at each instant along with the $\boldsymbol{Q}$ matrix.

The function $\gamma(z,p)$ computes the total useful in-flow to node $z$ from it's parent $p$ at a particular instant. $\gamma$ contains two pieces of inflow. The useful information received on first transmission from the parent is $\lambda_p \pi_{p,z}$, where $\lambda_p$ is the useful inflow received at the parent and $\pi_{p,z}$ is the reception probability from the parent to node $z$. This is an approximation, because $\lambda_p(t)\pi_{p,z}(t)$ should be $\lambda(t-\Delta t)_p \pi_{p,z}(t)$. The retransmission reception rate is $\eta_z \pi_{z,p} \pi_{p,z}$, where the first two terms $\eta_z \pi_{z,p}$ is the NAK reception rate at the parent. The retransmission reception rate is an approximation for the portion of $\zeta_p$ belonging to $z$.

Denoting the time derivative as $f'$, for a node $z$ with parent $p$,

$$\gamma(z,p) = (\lambda_p + \eta_z \pi_{z,p})\, \pi_{p,z} \tag{6}$$

$$\eta_z{}' = \lambda_p - \lambda_z \tag{7}$$

$$\zeta_z{}' = -\zeta_z + \sum_{i \in C(z)} \eta_i \pi_{i,z} \tag{8}$$

$$\lambda_z{}' = -\lambda_z + \gamma(z,p) + \sum_{i \in S(z)} \gamma(i,p) \pi_{i,z} \varepsilon_{i,z} \tag{9}$$

The NAK generation rate $\eta_z$ is the difference between the useful data reception rate at $p$ minus the useful reception rate at $z$. For the broadcast root, $\eta_0 \equiv 0$. Notice that $\eta$ does not drain on it's own accord. $\eta$ drives the coupled system. It forces parent and child nodes towards a common $\lambda$ rate.

The retransmission rate at node $z$, $\zeta_z$, increases by the total received NAK flow, $\sum \eta_i \pi_{i,z}$ from all children. Our model uses one NAK per packet. $\zeta$ drains to zero in the absence of NAK traffic.

Eq. 9 is the differential equation for the total useful inflow to node $z$, which drains at the previous useful in-flow rate. It adds the in-flow from $z$'s parent $p$ and from all of $z$'s siblings, $\gamma(i,p)\pi_{i,z}\varepsilon_{i,z}$. $\gamma(i,p)$ is the useful rate at $i$ and $\pi_{i,z}\varepsilon_{i,z}$ is the received useful portion at node $z$.

The broadcast root no longer has a time invariant transmission rate of $\mu_0$. We denote the driving rate as $m_0$, which is now the Poisson source rate. The root transmission rate $\mu_0(t)$ is subject to the differential equation

$$\frac{\mathrm{d}}{\mathrm{dt}}\mu_0 = m_0 - \mu_0 + \zeta_0 \tag{10}$$

The broadcast root has new packets to broadcast at a constant Poisson rate $m_0$ and drains at its previous rate $\mu_0$. The broadcast root increases its transmission rate with retransmissions to its children.

Eq. 11 is the differential equation for non-root node $z$'s transmission rate. The initial values for this equation affect the short-term behavior more than they do for the unreliable case. If we set $\mu(0) = \lambda(0) = 0$, then there is a high NAK rate, because it appears that all nodes have total loss. We find that setting $\mu(0) = \lambda(0) = m_0$ provides a stable initial condition.

$$\frac{\mathrm{d}}{\mathrm{dt}}\mu_z = \lambda_z - \mu_z + \eta_z + \zeta_z \tag{11}$$

## 3. Simulation and Numeric Results

We present numerical analysis and simulation results for our fluid model. Through simulation, we compare our

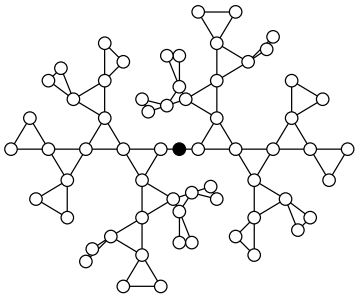

**Figure 3. BCT-5 Topology, source center node**

model to networks under constraints similar to our model assumptions. The results show that our model provides estimates similar to simulation. Although the absolute values of some metrics differ between simulation and the model, the differential equations produce curves with similar properties to those extracted from simulation. The section concludes with a discussion of our results and a comparison with a perfect TDMA schedule.

We begin with a description of our numerical and simulation techniques. We compare the long-run model behavior to simulation averages. We use simulation as a validation tool for the fluid model. Simulation scenarios try to match the assumptions of our model, such as no carrier sense and a perfect radio channel.

All simulations are on a binary cluster tree, described below. We have restricted our current work to one multihop network topology so that we may study in detail how various parameters affect network load and NAK rates. Studies on random graphs and networks with mobility are left for future research.

Via numerical methods, we solved the system of differential equations for binary cluster trees (BCT) using the XPPaut [6] package. The results show both reliable and unreliable behavior (RBCT, UBCT). We examined a tree of height five, which has a total of 63 nodes. We denote a tree and it's hight as, for example, "RBCT-5", for a reliable binary cluster tree of height five.

We programmed an equivalent network in GloMoSim [3]. We created a Poisson source server that generated 5000 fixed size packets with exponential inter-packet times. The packets were 500 byte UDP datagrams sent over a 2 Mbps wireless channel without capture. We ran the simulation for several tens of seconds after the source node transmitted the 5000th packet. Repeater nodes rescheduled packets in the future from an exponential variate with mean equal to half the broadcast data period. Nodes keep a history of broadcast packet sequence numbers and only rebroadcast unique packets.

The reliable broadcast protocol uses either end-to-end sequence numbers or per-hop sequence numbers. In the end-to-end model, a node will NAK anytime it receives out of order packets. NAKs are broadcast, but only the cluster root will respond to the NAK. In the per-hop NAK model, nodes only NAK out of order packets from their cluster root. NAKs are transmitted as broadcast packets to avoid introducing two classes of traffic. Nodes schedule NAKs and NAK timeouts from an exponential variate with mean $10m_0$. In the per-hop simulations, a packets carry an end-to-end sequence number, a host specific sequence number, and the last host sequence number (two-hop sequence number). Including the two-hop sequence number allows rebroadcast packets to repair losses within their 1-hop cluster.

We modified the radio to use a deterministic propagation model. If a receiver is within the specified radio range (in grid spacing), the receive power equals the transmit power. If a node is outside the radio range, there is complete loss.

### 3.1. Unreliable Broadcast

Fig. 4 shows the steady-state model behavior of a UBCT-5 network for various source rates. The model predicts minimal loss under about 4% channel capacity. Figg. 5 shows the benefit of using appropriate packet scheduling and pacing. Using a better value for the retransmit timer, we may about double the allowable data rate.

Fig. 6 shows the steady state transmission rates from Eq. 5 and a simulated ALOHA-type scheme over various source node loads. Fig. 7 shows the results for an 802.11 MAC layer in the simulation. The results have the same general trend between the 802.11 and ALOHA MAC layers. The 802.11 MAC performs better because of carrier sense, and thus diverges more quickly from our model. In the discussion below, we will focus on the ALOHA MAC.

At very low load (not shown), being under 5% channel capacity, there is good agreement between model and simulation. At low load, below a source rate of about 15% of the channel capacity, our model underestimates the success probability towards the extremities of the BCT, but is close to the simulation results towards the core. Above that rate, our model overestimates the success rate at the core of the tree, but is close to the simulation results at the extremities. Between 5% and 20% source channel utilization, our model is within 20% of the simulation results at the worst points and overall is under 10% difference from the simulation results.

At loads greater than about 50% of the channel capacity, there is significant divergence of simulation to model. This is caused by channel access contention, finite queuing, and vulnerability periods caused by packet lengths.

The poor performance predicted and observed for unreliable broadcast is caused largely by the selection of timers. The fluid model assumes that nodes adjust their output rate

to match their input rate. In the simulation, we used a fixed output rate of one half the source rate. These choices yield a high collision rate. As shown in Fig. 5, proper timer selection and scheduling yields significant improvements. Determining these rates, however, is not a simple proposition, though usually around two to three times the reciprocal packet time seems to work well.

### 3.2. Reliable Broadcast

We begin with several graphs of the transient behavior of a RBCT-5 network at 2% load. The 2% load is stable and shows how the equations converge. After a discussion of these results, we compare our reliable broadcast model to the simulation results. We set the initial conditions for $\lambda$ and $\mu$ to $m_0$. $\eta$ and $\zeta$ begin at 0. Fig. 8 shows $\mu$ rates, Fig. 9 shows $\lambda$ rates, and Fig. 10 shows $\eta$ rates.

In the stable network, all the rates stabilize. For $\mu$, we see that leaf nodes have the lowest network load, because they do not retransmit packets. Nodes in the middle of the tree, at heights 1 to 4, have the highest transmission rates. They generate both NAK and retransmission packets. Looking at the $\lambda$ graph, we see that all nodes converge to rate $m_0$. Nodes further away from the source have wider and wider oscillations. In the $\eta$ graph for NAKs, the curves first dip below 0. This is an artifact of starting the process with $\eta(0) = 0$, which is not physically realizable.

To compare our model to the simulation, we graphed the overall network load, $\mu$ and the NAK generation rate $\eta$. These graphs only compare stable network loads up to $m_0 = 0.03$. Both simulation and the models showed loads at 0.04 or higher to be unstable. Fig. 11 shows transmit rates. Fig. 12 shows the NAK rates using a per-hop sequence number scheme. Our model has very good agreement with simulation results for the overall network load. Using per-hop sequence numbers generates about one third the NAKs of an end-to-end protocol.

Fig. 13 shows model against simulation results when we used only end-to-end sequence numbers with NAK snooping. If a node hears a NAK for a packet it is about to NAK, the node will skip the first transmission of the NAK and reschedule it for a timeout. The figure shows that as nodes propagate out of order packets, there is a ripple of NAKs through the network. As nodes generate more NAKs, there is more packet loss, and more out of order packets. Note that in Fig. 12 for per-hop sequence numbers, the NAK rate drops away from the core while for Fig. 13, the NAK rate is about equal over the whole tree, since most all nodes NAK the same packets. We did not use cumulative NAKs, so one could slightly decrease the NAK storm. Cumulative NAKs, however, cannot solve the problem. Cumulative NAKs would reduce the overhead, but nodes would still generate unnecessary NAKs for packets lost outside of their 1-hop neighborhood.

From the simulation, we found that there were about 1.1 missing packets for each out-of-order packet when the load was 0.01. A cumulative acknowledgment protocol would not have the dramatic performance gain exhibited by our hybrid end-to-end and per-hop NAK scheme. The exact number of missing packets varies by tree height. The mean number of missing packets for heights 1 to 5 are 1.05, 1.09, 1.13, 1.09, and 1.13. These numbers included NAK snooping. The snooping leads to the oscillating missing packet rates. The snooping only works on a 1-hop radius, so every 2-hops, nodes generate unnecessary extra NAKs.

### 3.3. TDMA schedules

In previous sections, we have presented predicted and simulated throughputs for a binary cluster tree under different conditions. This section discusses the maximum possible throughput if we were able to realize perfect scheduling with slots sized exactly to packet lengths.

We can devise a five-slot schedule for the binary cluster tree with a two-hop collision free zone. Each node gets a chance at the channel. The network capacity would be 20% of the channel capacity. Nodes would not need to use NAKs (assuming a perfect channel). This represents a four-fold improvement over randomized scheduling, not counting any overhead for determining the schedule.

Because all our topologies are essentially trees, we may always construct a "rude" broadcast schedule where siblings transmit over each other. This does not result in data loss and will always give a schedule of three time slots. Thus, the absolute maximum TDMA capacity is 33% of the channel capacity, about ten fold higher than predicted for contention networks.

If slots were sized to 1500 bytes and we were using 500 byte packets, there could be wasted channel capacity (assuming no data piggy-backing). Our five slot coloring of the binary cluster tree would only have about a 7% efficiency, because $2/3$ of the air time is wasted. It would then have about twice the channel efficiency as our randomized methods.

## 4. Conclusion

We have presented a novel analysis of broadcast communication in wireless ad-hoc networks over a contention channel. Our differential flow equations allow us to model a class of networks made up of cluster trees. Cluster trees are an approximation of real wireless networks. They include both rich cluster connectivity and hidden terminal topologies.

Our fluid model analysis exposes several key metrics for unreliable and reliable broadcast networks. These include

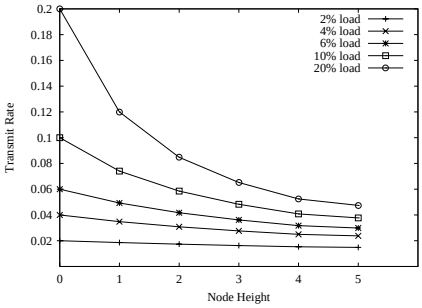


**Figure 4. Unreliable broadcast model**

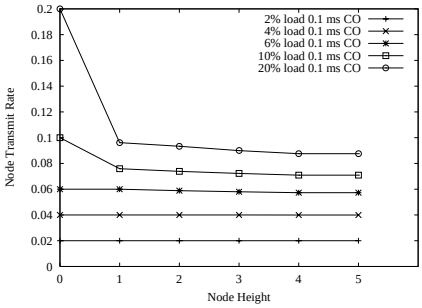


**Figure 5. Unreliable Broadcast with pacing**

loss rates, NAK generation rates, and network load. We compared our results for unreliable and reliable broadcast with simulations using GloMoSim. For unreliable broadcast, our model generates curves that match simulation results, although the absolute values may vary by 10% to 20%. For reliable broadcast, our curves for total network load match the simulation results very well.

From our simulation and analysis work, we found that NAKs based solely on end-to-end sequence numbers do not work well in an ad-hoc wireless environment. Out-of-order packets propagate through the network and create chains of NAKs. We found that a protocol combining end-to-end and one-hop sequence numbers is more stable.

Future work on the fluid model includes changing from instantaneous differential equations to delay equations that model transport and processing time. Delay equations would also allow for longer repair paths, so we could model mesh protocols such as ODMRP or CAMP. We would also like to model cluster repair through multi-hop retransmission. The current model limits the repair to only one hop.

A NAK scheme based on per-hop sequence numbers introduces a large amount of state in the network and most likely would not work with node mobility. A promising line of investigation consists of using forward NAK notifications, with which a node adds information about NAKs it is sending to a packet before repeating it. This would allow, in effect, multihop NAK snooping.

## References


[1] D. Anick, D. Mitra, and M. Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell System Technical Journal*, 61(8):1871 – 1894, 1982.

[2] ANSI/IEEE. *Std 802.11 1999 Edition [ISO/IEC 8802-11:1999(E)], Part 11: Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications*. IEEE, New York, 1999.

[3] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. GloMoSim: A scalable network simulation environment. Technical Report 990027, UCLA Computer Science Department, 1999.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–16, 2001.

[5] C.-C. Chiang, M. Gerla, and L. Zhang. Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks. *Cluster Computing*, 1(2):187–96, 1998.

[6] G. Ermentrout. Xppaut 5.0. http://www.pitt.edu/˜phase.

[7] I. Gaber and Y. Mansour. Broadcast in radio networks. In *Proc. 6th Annual ACM-SIAM Symp. Discrete Algorithms*, pages 577–82, Jan. 1995.

[8] J. Jetcheva, Y.-C. Hu, D. Maltz, and D. Johnson. A simple protocol for multicast and braodcast in mobile ad hoc networks. draft-ietf-manet-simple-mbcast-01.txt, July 2001.

[9] J. Jetcheva and D. Johnson. The adaptive demand-driven multicast routing protocol for mobile ad hoc networks (ADMR). draft-ietf-manet-admr-00.txt, July 2001.

[10] M. Jiang, J. Li, and Y. Tay. Cluster based routing protocol (CBRP). draft-ietf-manet-cbrp-spec-01.txt, July 1999.

[11] J. Kuri and S. K. Kasera. Reliable multicast in multi-access wireless LANs. *Wireless Networks*, 7(4):359–69, 2001.

[12] H. Lim and C. Kim. Flooding in wireless ad hoc networks. *Computer Communications*, 24(3-4):353 – 363, Feb 2001.

[13] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *MobiCom'99*, Aug 1999.

[14] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report 3898, INRIA, Mar 2000.

[15] W. J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton University Press, Princeton, NJ, USA, 1994.

[16] I. Stojmenovic and X. Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–32, Oct. 2001.

[17] I. Stojmenovic, M. Seddigh, and J. Zunic. Internal nodes based broadcasting in wireless networks. In R. H. Edited by: Sprague, editor, *Proc. 34th Annual Hawaii International Conf. System Sciences. HICSS-34*, page 10 pp., Jan. 2001.

[18] K. Tang and M. Gerla. MAC layer broadcast support in 802.11 wireless networks. In *IEEE MILCOM: Military Communications Conference*, pages 544–8 vol.1. IEEE, Oct. 2000.

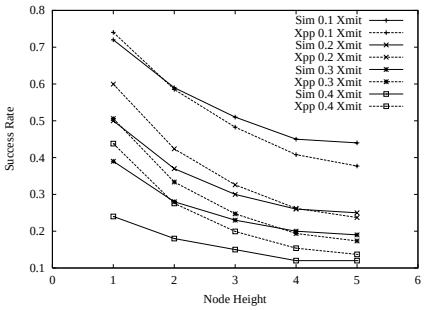


**Figure 6. Model compared to ALOHA**

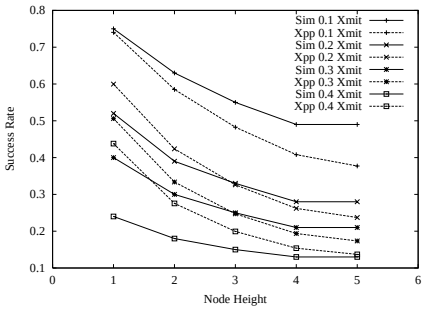


**Figure 7. Model compared to 802.11**

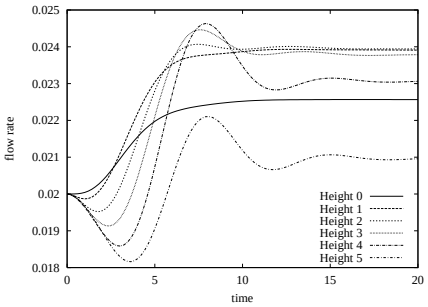


**Figure 8. RBCT-5 $\mu$ rate, $m_0 = 0.02$**

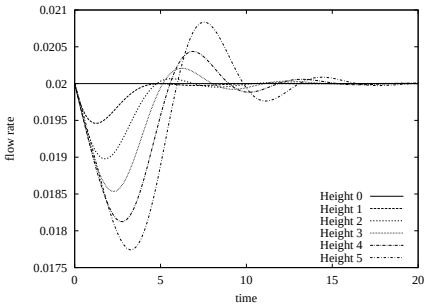


**Figure 9. RBCT-5 $\lambda$ rate, $m_0 = 0.02$**

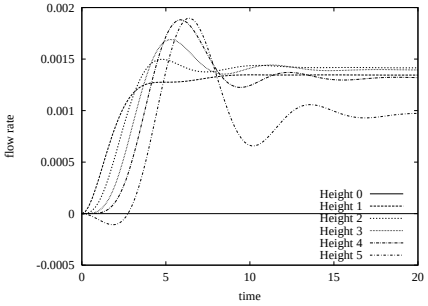


**Figure 10. RBCT-5 $\eta$ rate, $m_0 = 0.02$**

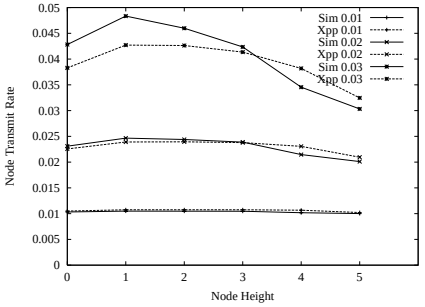


**Figure 11. Network Load: Model & Simulation**

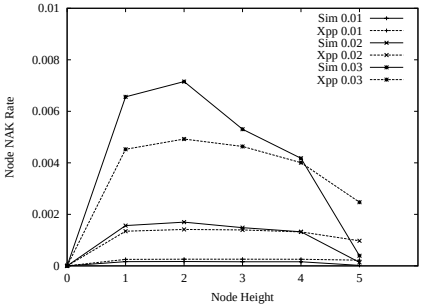


**Figure 12. Nak Rate: Model & Simulation**

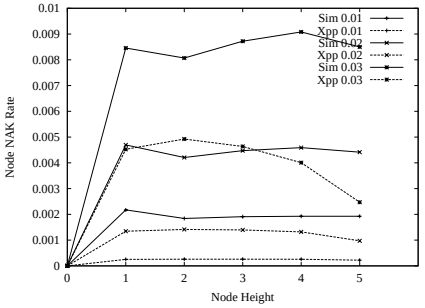


**Figure 13. Nak Rate for End-to-end SeqNo**