EEL 6591 : Project
Reliable Broadcast in Error-prone Multi-hop
Wireless Networks : Algorithms and Evaluation

Team Members
Bhavya Daya
Shu Jiang
Marisha Rawlins
# Table of Contents

I. Introduction: Overview of Paper ................................................................................................... 3  
   Research Paper’s Abstract and IEEE Citation.............................................................................. 4  
   Description of Research Paper Analysis ................................................................................... 4

II. Analysis of Paper’s Algorithm ................................................................................................... 5  
   Analysis of the Transmission Overhead Computation in the Paper ........................................... 6  
   NS-2 Simulation of the Original Algorithm ................................................................................. 7  
   Results ....................................................................................................................................... 9  
   Comparison of Team's Results and the Original Paper's Results ............................................. 10

III. Analysis of Team’s Optimized Algorithm .............................................................................. 11  
   NS-2 simulation of Team's New Algorithm ............................................................................... 12  
   Results .................................................................................................................................... 13  
   Comparison of New Results with the Original Paper. ............................................................. 14

IV. Conclusion ............................................................................................................................... 15
I. Introduction: Overview of Paper

Reliable broadcast is very important when deploying certain sensor networks. If the links of the network are error prone it is difficult to broadcast the message with strict reliability requirements, and still have a reasonable communication overhead. There are several approaches to achieving strict reliable broadcast. These approaches are source-initiated, destination-initiated, and sender-initiated. The first two approaches, source and destination initiated, are end to end solutions. These are not well suited for multi-hop wireless networks because of high probability of error when sending acknowledgements. The source-initiated is not a useful approach is because the source of the broadcast has to make sure that all the nodes received the message and the source retransmits if a certain time elapsed and the destination didn’t send an acknowledgement. With the sender-initiated approach, each node is a destination as well as a sender. The task of broadcast is spread across the nodes, so the nodes are responsible for sending reliable transmissions to the neighbors only. This method is used to propagate the message through the network.

The algorithm developed in the paper being analyzed, focuses on the sender-initiated approach for the multi-hop wireless networks. When there are error-prone links, error discovery and recovery need to be performed to ensure reliable broadcast. Strict reliable broadcast means that all the nodes will receive the message without error. This can be achieved using positive acknowledgements in the statics networks that are being considered. The goal for the algorithm is to provide reliable broadcast with less broadcast overhead in an error-prone network. The paper being analyzed performs an analysis of the transmission overhead that is incurred for a single broadcast message in a network with m nodes. This analysis is critiqued to determine the validity of the results. The performance analysis values are compared against simulations that were performed using the NS 2 simulator. An optimization to the algorithm is brainstormed and implemented. The results of the performance are also obtained.

The wireless network architecture to be considered is a multi-hop wireless ad-hoc network. The algorithm in the research paper uses other nodes to forward data, thus classifying the target architecture as a wireless ad-hoc network. Multi-hop wireless ad-hoc networks can be further classified by their application. The classifications are mobile ad-hoc networks (MANETs), wireless mesh networks, and wireless sensor networks (WSN). The algorithm can be applied to wireless mesh networks and static sensor networks. The mobility aspect is not contained in the algorithm and hence MANETs and mobile sensor networks are not considered. The wireless multi-hop mesh network is a possible target architecture for the algorithm. A wireless sensor network architecture is also shown. The architecture can be any multi-hop wireless ad-hoc network. The simulations require a wireless ad-hoc network containing 10 nodes and 50 nodes.
Research Paper’s Abstract and IEEE Citation


**Abstract**: Reliable broadcast is an essential functionality for disseminating messages to all nodes in a multi-hop wireless network. Ensuring broadcast reliability while minimizing communication overhead is a challenging task when error-prone links are considered. In this paper, we introduce and analyze techniques to address this challenge, and propose an overall algorithm encompassing a combination of the investigated techniques as an efficient solution for reliable broadcasting in multi-hop wireless networks. Simulations are conducted to validate the analytical results and to further evaluate the performance of the algorithm.

**Description of Research Paper Analysis**

The type of protocol to be considered is a routing protocol. The routing protocol will deliver reliable broadcast in an error-prone multi-hop wireless network. The routing protocol will be implemented as well as the MAC mechanism and channel error model. The MAC mechanism is similar to CSMA/CA. The steps of the broadcast algorithm are detailed below.

1. **Source node begins with a broadcast message that can only reach a few neighbor nodes within transmission range.**
   a. Transmitter sets a timer at which the node shall retransmit the message if an acknowledgement (ACK or ack) from any neighbor(s) was not received.
2. **The neighbor nodes that receive the message, send an ACK to the transmitter**
3. **If this is the first reception of the data message**
   a. Forward the received data message to neighbors within transmission range.
   b. Transmitter sets a timer at which the node shall retransmit the message if an ACK from any neighbor(s) was not received.
4. **Else**
   a. Send an ACK to the transmitter but do not forward the data message because it has been forwarded before.
5. **Repeat steps until no more nodes are available to forward the message to.**
The research paper analyzes the transmission overheard in the form of data and acknowledgement transmissions. The paper performs a written analysis of the approximate value of the number of data transmissions and number of ack transmissions. The analysis was performed again in order to verify the same result. The paper performs simulations for a ten node network and a fifty node network, calculating the number of data and ack transmissions. The paper didn’t specify the topology that was used, but a topology was created in order to verify the same trend seen in the paper’s simulation results.

II. Analysis of Paper’s Algorithm

The paper being studied proposed many variants of the baseline algorithm to improve the performance of the reliable broadcast algorithm. However, only the baseline broadcast algorithm is analyzed and implemented. The baseline algorithm achieved reliable broadcast by having each node in the network acknowledge each received packet, relay the data to its immediate neighbors and wait for ACKs to come back. A retransmission is initiated by the transmitting node when any of the expected ACKs are not received. While the algorithm has the advantage of reliable broadcast, it also incurs large overhead due to the fact that each node has to send an ACK for each data reception. An analysis of the overhead of the baseline algorithm is presented below.
Analysis of the Transmission Overhead Computation in the Paper

The analysis of the baseline algorithm in the paper is performed by assuming that a node can receive a broadcast data message and its corresponding ACK with packet error probabilities of $p_d$ and $p_a$, respectively. The packet error probabilities can be defined as follows.

1. $P_d$: packet error rate or probability for a node to receive a broadcast data message
2. $P_a$: packet error rate or probability for a node to send an acknowledgement

The number of transmissions of a message, $T_i$, is sent by an arbitrary node $i$. The number of nodes in node $i$’s one-hop neighbors is $n_i$. The total number of nodes in the network is ‘m’. The probability distribution for data message transmissions is obtained as follows:

Assume the number of transmissions equals $k$

All neighbors receive the data message without error $= (1 - p_d)^{n_i}$

All neighbors send an acknowledgement without error $= (1 - p_a)^{n_i}$

The transmission and(or) the ack is not received $= (1 - (1 - p_d)^{n_i} * (1 - p_a)^{n_i})$

$P_r(T_i = k)$

$= \text{one transmission and ack by node i is completed successfully and the rest of the transmissions fail}$

$P_r(T_i = k) = [1 - (1 - p_d)^{n_i} * (1 - p_a)^{n_i}]^{k-1} * ((1 - p_d)^{n_i} * (1 - p_a)^{n_i})$

The expected number of data and ACK transmission per message of Node $i$ is as follows. The expected number of data transmissions is equal to the probability of one successful transmission multiplied by $k$ to obtain the number of transmissions total. These values for summed for values of $k$ from 1 to infinity. The expected number is determined by where the summation converges. The number of ACK transmissions depends on the expected number of data transmissions. The expected number of data transmissions times the probability the transmission is successful is the time when the ACK is transmitted. The ACK is transmitted from the neighbors, $n_i$. The likelihood that the ACK is transmitted back is measured for each neighbor and the successful ones are counted as the expected number of ACK transmissions.

$$E[T_i] = \sum_{k=1}^{\infty} P_r(T_i = k) \ast k$$

$$E[A_i] = \sum_{j=1}^{n_i} (1 - p_d) \ast E[T_j]$$

MathCAD was used to obtain the symbolic form of the expected number of data transmission per message of Node $i$. The MathCAD result is shown below.

$$P_{r(k)} = [1 - ((1 - pd)^n \ast (1 - pa)^n)]^{k-1} \ast (1 - pd)^n \ast (1 - pa)^n$$
The result obtained is different from the research paper result. It is unclear at this point as to error in the paper’s observations or the team’s observations. It seems that the mathematical result is incorrect in the paper. The concepts used to obtain the expected number of transmissions for node i, matches the research paper findings. The total number of data transmissions and ACK transmissions is equal to the total number of expected transmissions for each node in the network.

\[ E[T] = \sum_{i=1}^{m} E[T_i] \]

\[ E[A] = \sum_{i=1}^{m} E[A_i] \]

The analysis of the transmission overhead of the baseline algorithm consisted of finding the total number of data and ack transmissions in the network. The analysis performed manually for the most part matched the research paper's analysis. The only discrepancy found was in the computation of the expected number of data transmission for node i, \( E[T_i] \).

**NS-2 Simulation of the Original Algorithm**

To verify the correctness of the paper's result, the broadcast algorithm is simulated using NS2. The paper’s simulator consists of 3 basic components: a channel error model, a MAC, and the broadcast algorithm. However, since the paper doesn’t specify the details of each component, assumptions were made in our simulator implementation to mimic the simulator used by the paper. Our goal was to simulate a broadcast over the chosen network, accumulate statistics, and plot graphs corresponding to those produced in the original paper.

The error model in NS2 simulates the link level error, with the probabilities of error events occurring on each link independent of each other. In the paper, the packet error probabilities \( P_d \) and \( P_a \) are set to be equaled for simplicity. Our simulation adapted the same choice for error model implementation. In NS2, the error model is added to the simulated wireless network with the following segment of code:
The paper implemented a CSMA/CA-alike MAC mechanism, in which the transmission time is divided into time slots. This implementation mimics the CSMA/CA based MAC by adding time slots where only one node in a two-hop neighborhood can transmit in a specific slot. The choice of CSMA/CA was deduced to be 802.15.4. Research was performed to compare the MAC supported by NS2 and the time slotted CSMA/CA specified by the research paper.

Since no topology was specified in the paper, the team chose an arbitrary topology as mentioned below. A wireless TCP network with 10 nodes was created and the nodes were given specific coordinates. The network topology implemented for the simulation is shown in Figure-2. The topology consists of 10 wireless nodes. Our simulation used static nodes instead of mobile nodes because the focus of the paper is reliable broadcast rather than mobility. Using static nodes also simplifies both analysis and implementation of the broadcast algorithm. Since the network is static, the broadcast algorithm can have prior knowledge of the neighbors of each node.

Lastly, the means of broadcast was also not specified. TCP was chosen because it is reliable and retransmissions of data are performed when an acknowledgement isn’t received within a time limit. FTP over TCP traffic flow was added to the network to analyze the performance of the broadcast algorithm. For TCP networks in NS 2, explicit links must be set up between the nodes. We made a simplification by taking advantage of the fact that the topology was known beforehand so we could set up the links as

```
proc UniformErr () {
    set err [new ErrorModel]
    $err unit packet
    $err set rate_ 0.0
    return $err
}

$ns_ node-config -IncomingErrProc UniformErr -OutgoingErrProc UniformErr
```

![Figure 2: Network topology](image)
needed. Transmissions were explicitly made as needed until each node had received the message at least once.

One of the more difficult aspects of implementing broadcast using TCP in NS 2 was achieving the correct timing. In NS 2, TCP is set up using FTP to stream packets in the appropriate link. The start and stop times for each node had to be adjusted in a way to minimize the number of packets in the stream (ideally we only need 1 packet) while allowing enough time to allow retransmissions as error rates increased. However, since we were not trying to minimize the time taken to complete the broadcast, continuously adjusting the timing did not affect our overall results. Another difficulty encountered was analyzing the trace files produced. This is because, as mentioned before, FTP generates a stream of packets and we were looking for the situation where every node has received the packet at least once. The “grep” command was used to select only the transmissions we were interested in (i.e. ACKs or data) then further analysis was done to ensure that we were counting only the transmissions we needed.

Results

Because TCP needs explicit link set up, we were not able to implement the case of 50 nodes as this would have been a time consuming task. The results for 10 modes are presented here.

![Figure 3: Number of Data Transmissions and ACKs vs. Error Rate for 10 nodes](link)
Comparison of Team's Results and the Original Paper's Results

As was expected, the results of our implementation and that of the original authors’ were not identical. This can be explained by the fact that assumptions were made for our implementation. For example our resulting topology was likely very different from their unspecified topology. Although the number of nodes was the same, differently positioned nodes could create more hops required for transmissions where each hop uses a link with error added to it. This is one possible reason for why we observed that the numbers recorded by our team were greater than those in the original paper. We still consider our implementation to be successful since our graphs show the same trends presented in the paper – i.e. with increasing packet error rate, we observe an increase in the number of data and ACK transmissions required to implement the broadcast as shown above.

The paper compares the written analysis results and the simulation results for ten nodes. The number of transmissions versus the packet error rate is measured. Since emphasis is placed on reliable broadcast, the number of transmissions increases as the packet error rate worsens. Figure 4 below displays the research paper’s simulation results. The paper was vague with regard to the exact implementation and many assumptions had to be made in order to perform the simulation.

Figure 4: Research Paper's Results
The algorithm that was implemented contained the following assumptions and choices. Since the network is static, the broadcast algorithm can have prior knowledge of the neighbors of each node. The choice of CSMA/CA was deduced to be 802.15.4. Research was performed to compare the MAC supported by NS2 and the time slotted CSMA/CA specified by the research paper. The means of broadcast was also not specified. TCP was chosen because it is reliable and retransmissions of data are performed when an acknowledgement isn’t received within a time limit. An arbitrary topology was chosen because the paper does not mention a specific topology

**III. Analysis of Team’s Optimized Algorithm**

The original algorithm requires each node along the multi-hop network to respond to the sender with an acknowledgment. When the links are error prone, this is a good method to ensure reliable broadcast. The optimization would not be linked to achieving the goal of reliable broadcast but instead for efficiency by decreasing the number of data and acknowledgement transmissions that occur. This is achieved by assuming the multi-hop broadcast is transmitted reliably; therefore when a node receives a data message an acknowledgement isn’t transmitted. Since an acknowledgement isn’t received by the sender, data retransmissions aren’t performed. The multi-hop broadcast is still performed and the MAC is the source of the algorithm optimization. The new algorithm is specified as follows.

1. Source node begins with a broadcast message that can only reach a few neighbor nodes within transmission range.
2. The neighbor nodes that receive the message, do not send an ACK to the sender
3. If this is the first reception of the data message
   a. Forward the received data message to neighbors within transmission range.
4. Else
   a. Do not send an ACK to the transmitter and do not forward the data message because it has been forwarded before.
5. Repeat steps until no more nodes are available to forward the message to.

The optimization eliminates the RTS/CTS/ACK overhead that occurs and assumes the broadcast messages reach the destination. The topology used for the program is the same as the one used for the simulation of the original algorithm. The efficiency of the two algorithms couldn’t be compared because NS-2 requires the timing of the TCP to be specified by the user. If we compared the efficiency, the data would be accurate. Theoretically, the new algorithm is more efficient due to the reduced overhead in the
MAC and the disregard for retransmissions. Although the efficiency couldn’t be measured in time, the simulation was compared by the number of data and ack transmissions. The comparison also shows that the optimization doesn’t consider reliable broadcast, because that wasn’t the goal of the optimization. The goal of the optimization was to increase the efficiency and decrease the overhead of the broadcast procedure. In a network where many broadcasts are occurring, efficiency is important. The new algorithm’s implementation is very scalable to larger architectures and to mobile networks. The original algorithm only focuses on static networks. The broadcast information is determined during runtime for the new algorithm. Figure 5 shows the flowchart of the optimization. The same high level approach is used with some minor changes.

**Figure 5: Diagram of Optimization Algorithm**

**NS-2 simulation of Team's New Algorithm**

The NS-2 simulation consisted of developing the broadcast algorithm and analyzing the MAC in order to make the adjustments. It was observed that the 802.11 MAC in NS-2 already had the reduced overhead for broadcast messages. The algorithm was written to utilize the MAC and specify the messages as broadcast messages. Difficulties were met when trying to have each message keep a record that it has
seen the message and only forward it if it hasn’t seen it before. When performing the optimization, adjustments to the 802.11 MAC were attempted to make the original algorithm more dynamic and scalable in order to compare the data from both algorithms more accurately. Adjustments to the 802.11 MAC was difficult and the adjusted original algorithm code wouldn’t work as intended.

**Results**

The results from the optimization show that the number of ACK transmissions is equal to zero because acknowledgements aren’t sent in the new algorithm. The number of data transmissions is dependent on the topology. Each node broadcasts the message to all its neighbors. The node doesn’t consider whether or not it is sending the message to the node it received it from. This would be a good optimization to further reduce the number of data transmissions. The results for the ten node and fifty node networks are shown below.

![Optimization Results: 10 Node Network](image1)

![Optimization Results: 50 Node Network](image2)
The figures display the number of data transmission, number of ack transmissions and number of nodes that receive the data. The time it took to run the ten node network simulation was 11.56 milliseconds and it took 14.6 milliseconds to run the fifty node network simulation. The data shows that as the packet error rate increases, the larger network is impacted more than the smaller network. The number of nodes that receive the message decrease tremendously for the fifty node network. The reason is mainly due to the network topology used. Different topologies weren’t attempted, but it would be a good analysis to see how the topology affects the broadcast algorithm.

The number of data transmission would definitely increase with the number of nodes. The figures show that the number of data transmissions remains constant as packet error rate is varied. Limited number of acknowledgements could be used to make the optimized algorithm more reliable. This can achieve both the reliability and efficiency goals.

Comparison of New Results with the Original Paper

The team’s algorithm is more scalable and performs the broadcast even if the network isn’t static. The algorithm is dynamic and the network topology doesn’t have to be known beforehand. Routing tables are not used to keep track of neighbors. The number of data transmissions and acknowledgement transmissions are measured to compare to the original algorithm results. The new algorithm is better for networks that aren’t static because the broadcast procedure is determined during runtime. Error-prone networks should not use the new algorithm.

The new algorithm contains a reduction in the number of data transmissions when the packet error rate is greater than ten percent in the links. The number of data transmissions per node for the optimization stays fixed to 2.6 for the ten node network. The number of acks is much less than the original algorithm. In the original algorithm the number of ack transmissions is very large and creates a lot of communication in the network. At approximately fifteen percent error rate, the new algorithm is better than the old algorithm in terms of efficiency. At the fifteen percent error rate, the new algorithm is also reliable for small networks. The overhead of the new algorithm is much less than the original algorithm, which is the goal of the optimization. This value could not be quantified, but it makes the new algorithm more efficient.
IV. Conclusion

The NS-2 tool was used to simulate and compare results to the research paper chosen. The disappointment with NS-2 was the lack of documentation as to what certain functions within the library do and how to create one of your own, if necessary. Another issue was that NS-2 prevented us from comparing the efficiency, in terms of time, of the two algorithms. TCP usage in NS-2 requires the programmer to specifically time when the messages should be sent, an automatic timing could not be created. Since the user times the transmissions, it isn’t accurate as to minimum time the broadcast algorithm takes to distribute the message to all nodes in the network.

The optimization improved upon the original algorithm by placing emphasis on efficiency and reduction in transmissions overhead, including the MAC overhead. The division of labor that occurred was as follows.

1. Downloading NS-2 and getting it running – Bhavya Daya, Marisha Rawlins, Shu Jiang
   - We all downloaded NS-2 on our own computers so that we can work on it separately.

2. Simulating and obtaining results for the research paper’s algorithm – Marisha Rawlins, Shu Jiang
   - Researching and implementing the MAC – Shu Jiang
   - Implementing the algorithm – Marisha Rawlins, Shu Jiang

3. Simulating and obtaining results for the new algorithm – Bhavya Daya

The number of data transmission would definitely increase with the number of nodes. Limited number of acknowledgements could be used to make the optimized algorithm more reliable. This can achieve both the reliability and efficiency goals. Future experiments could include changing the topology and analyzing both algorithms, adjusting the original algorithm to be more scalable, and adjusting the new algorithm to use a limited number of acknowledgements.