

Onroad Vehicular Broadcast

Jesus Arango, Alon Efrat
Computer Science Department
University of Arizona

Srinivasan Ramasubramanian, Marwan Krunz
Electrical and Computer Engineering
University of Arizona

Abstract—

This paper presents a broadcasting algorithm that considerably reduces the number of retransmissions in applications such as onroad vehicular broadcasting where nodes are assumed to be arranged on a strip. Analysis and simulation results are presented to describe the overhead, coverage and latency characteristics of the algorithm.

I. INTRODUCTION

A whole new range of vehicular information services can be made possible by relaying information using vehicle-to-vehicle communications (V2VCOM). The first class of services that comes to mind is traffic alerts about different upcoming situations such as accidents, construction zones, or traffic jams.

With the exception of some cases such as caravans or convoys, communication between specific vehicles is rarely an issue. Broadcasting is a more natural communication primitive for this type of environment. Accordingly, the goal is to relay information between vehicles for a certain distance or for a given number of relay hops. Omnidirectional transmissions are used because the road and the vehicles it contains can have an arbitrary direction with respect to the transmitter's frame of reference.

The simplest broadcasting algorithm is known as *flooding*. An originator or *source* performs the first transmission. All nodes in the network must retransmit the message once. If all retransmissions are successful and the network is connected, then every node is guaranteed to receive the message. Multiple *duplicates* can be received and nodes must keep track of recent messages to avoid retransmitting more than once. Each distinct message has a unique label that consists of a source address (i.e., IP address or VIN number) and a unique *sequence number* generated by the source.

Other broadcasting algorithms use a *retransmission scheduling* scheme. When a broadcast message is received, its retransmission is scheduled for a future time that is randomly chosen, allowing for the reception of additional duplicates before the actual retransmission takes place. The scheduled retransmission can be canceled at any time during this short *backoff period* if it is deemed to be redundant. A retransmission is said to be *redundant* if the node is sufficiently confident that the area to be covered by its retransmission has already been covered by the retransmission of the previously received duplicates. The coverage of a retransmission is based on the assumption of omnidirectional communication. Different algorithms use different techniques for predicting redundant retransmissions.

Previous algorithms perform poorly in vehicular environments because they perform considerably more retransmissions than necessarily. For example, vehicle *b* shown in Figure 1 has received duplicates from both *a* and *c*. Existing broadcast algorithms would require a retransmission by *b* because the shaded areas in its range of transmission have not been covered. However, the strip of road in the figure is entirely covered by previous retransmissions.

This paper presents an algorithm for vehicular broadcasting which considerably reduces the number of retransmissions. Analysis and simulation results are presented to analyze the performance of the algorithm.

The rest of this paper is organized as follows. A description and analysis of the algorithm is presented in Section II. The simulation results are analyzed in Section III. The related work is summarized in Section IV and we offer some concluding remarks in Section V.

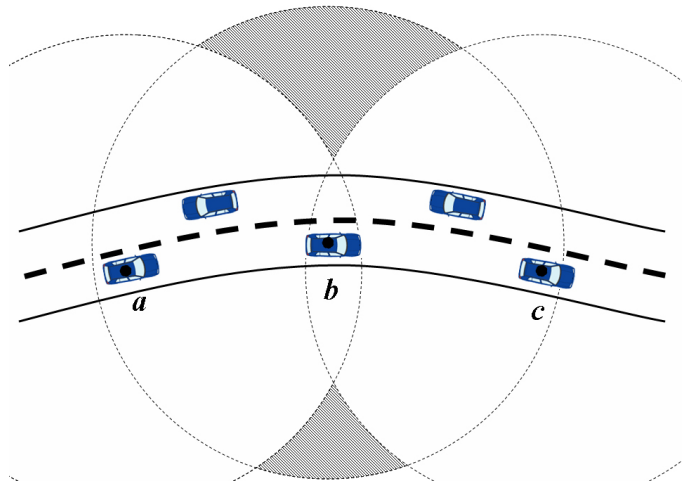


Fig. 1. Redundant retransmission.

II. STRIP BROADCASTING

If the transmission range is large enough in comparison to the width of the road, then in most situations a retransmission is deemed to be redundant once a message has been received from the front and back of the vehicle.

We propose the *Strip Broadcasting* (SB) algorithm shown in Algorithm 1 for nodes located on a strip whose width is considerably smaller than the range of the omnidirectional retransmissions.

Algorithm 1 Strip Broadcasting Algorithm

Require: packet p

- 1: **if** $\text{duplicates}(p) = 1$ **then**
- 2: $\text{schedule.tx}(p)$ \triangleright Schedule retransmission
- 3: **end if**
- 4: **if** p arrived from the rear **then**
- 5: $\text{rear}(p) \leftarrow \text{TRUE}$
- 6: **else**
- 7: $\text{front}(p) \leftarrow \text{TRUE}$
- 8: **end if**
- 9: **if** $\text{rear}(p) = \text{TRUE} \wedge \text{front}(p) = \text{TRUE}$ **then**
- 10: $\text{cancel.tx}(p)$ \triangleright Cancel scheduled retransmission
- 11: **end if**

A. Definitions and Analytical Model

The SB algorithm can be analyzed by modeling the problem as a one dimensional arrangement of N nodes on a segment of length L , as shown in Figure 2. All transmissions are omnidirectional and every node has a range of transmission of length R . A node that receives a duplicate from the left and from the right does not need to retransmit. The density ρ represents the expected number of nodes in a segment of length R .

$$\rho = \frac{NR}{L}$$

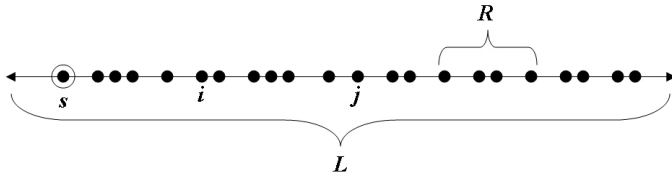


Fig. 2. Analytical Model

Let s be the *source* of the broadcast message, meaning it is the first node to transmit the message. The broadcasting process is symmetrical on both sides of s , thus it is assumed that s is the leftmost node and the model only considers the broadcast process to the right of s . If node x is closer to s than node y , then node x is called a *predecessor* of y , and node y is called a *successor* of x . In Figure 2, j is a successor of i .

For analytical purposes, the communication model is based on ideal network conditions. Transmission errors and collisions are not considered.

Both discrete-time and continuous-time analysis are considered. In the discrete model, time advances in discrete slots of length Δ , and the time necessary for a node to retransmit the message to all its neighbors is assumed to be less than Δ . When a node receives the first duplicate, there is a probability p that the node will retransmit the message in the next time slot. With probability $1 - p$ it will wait for the next slot, and the process is repeated until the packet

is sent or the scheduled retransmission is canceled. In the continuous time model, the back-off period is exponentially distributed, and the transmission time is assumed to be short enough such that there is a negligible probability that two retransmissions overlap in time. Because of the memoryless property of both time models, the waiting time for all nodes is equally distributed regardless of how long ago they received the first message.

The following metrics are defined to analyze the performance of a broadcast operation.

- 1) *Overhead* (ω): The fraction of the nodes in the network that retransmit the message.
- 2) *depth* (δ): The expected number of duplicates heard on a single point in the network:

$$\delta = \int_0^L \frac{1}{L} \cdot \eta(x) \cdot dx$$

where $\eta(x)$ is the number of duplicates heard at point x . Note that the depth and the density are related as follows:

$$\delta = \frac{2 \times R \times \omega \times N}{L}$$

- 3) *coverage* (c): The fraction of nodes that receive the message.
- 4) *delay* (χ): The amount of time necessary to relay the message for a certain distance.

B. Retransmission Analysis

This section presents some interesting analytical results about the number of retransmissions required to perform a broadcast operation.

Lemma 1: The expected number of retransmissions in a segment of length R for a broadcast operation using the LB algorithm is upper-bounded by:

$$\ln(\rho) + 1.$$

Proof: Consider an interval of size R like the interval $[y, z]$ shown in Figure 3. This interval has an arbitrary node distribution, but to its right is an empty interval $[z, w]$ also of size R . Consider a broadcast operation taking place from left to right. The *expected* number of retransmissions on $[y, z]$ is an upper bound for the general case. This is because there can't be retransmissions on $[z, w]$ that would otherwise silence some of the nodes on $[y, z]$.

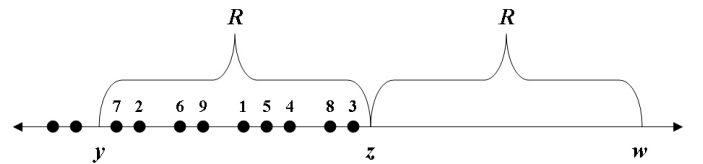


Fig. 3.

Enumerate the nodes in $[y, z]$ according to the order in which they complete the backoff period. Number 1 is assigned

to the first node to complete the backoff period, number 2 is assigned to the second node to complete the backoff period, and so on. Such an assignment is shown in Figure 3. Let X_i be a random variable associated with node i and defined as follows:

$$X_i = \begin{cases} 0 & \text{if node } i \text{ does not retransmit} \\ 1 & \text{if node } i \text{ retransmits} \end{cases}$$

Note that node i will only retransmit if it's physical location is the rightmost among the first i nodes to complete the backoff period. Node i has equal probability $1/i$ of being in any of the possible i locations. This is because the length of the backoff period is randomly and independently selected by each node, so all node enumerations are equally possible. As a result, the probability $\Pr(X_i = 1)$ that node i retransmits is:

$$\Pr(X_i = 1) = \frac{1}{i}$$

and the expected value of X_i is:

$$E[X_i] = \Pr(X_i = 0) \cdot 0 + \Pr(X_i = 1) \cdot 1 = \frac{1}{i}$$

Let $T([y, z], n)$ be a function of random variables representing the number of retransmissions among n nodes located inside the interval $[y, z]$.

$$T([y, z], n) = \sum_{i=1}^n X_i$$

Since $E[n] = \rho$, the expected number of retransmissions in interval $[y, z]$ is:

$$\begin{aligned} E[T([y, z], \rho)] &= E\left[\sum_{i=1}^{\rho} X_i\right] \\ &= \sum_{i=1}^{\rho} E[X_i] \\ &= \sum_{i=1}^{\rho} \frac{1}{i} \\ &\leq \ln(\rho) + 1 \end{aligned}$$

Corollary 1: A broadcast operation using the LB algorithm on a network with density ρ has a depth that is upper-bounded by:

$$\delta(\rho) \leq 2(\ln(\rho) + 1)$$

Corollary 2: A broadcast operation using the LB algorithm on a network with N nodes, range R , and length L has an overhead that is upper-bounded by:

$$\omega(N, L, R) \leq \frac{L}{R} \left[\ln\left(\frac{NR}{L}\right) + 1 \right]$$

Lemma 2: If nodes are uniformly distributed and backoff times are geometrically distributed, then the expected depth using broadcast algorithm ?? is 4.

Proof: ■

A fundamental assumption of these theoretical results is that no two nodes decide to retransmit at the same time. A practical interpretation of this assumption is that there are no concurrent retransmissions. Two retransmissions are said to be *concurrent* if the nodes involved are within range of each other and if the difference between their backoff periods is less than the retransmission delay. The retransmission delay is the time interval from the moment the sender completes the backoff period to the moment the receiver verifies the reception of previous duplicates.

If concurrent retransmissions are considered, a retransmission by an upstream node is possible even if a downstream node completes the backoff period first. Once the backoff period is completed by the downstream node, it takes some time for the message to travel down the protocol stack¹ and contend for access to the channel. Depending on the system load, this time can be substantial and highly random. Therefore, it is quite possible that the upstream node transmits the packet first even if the first one to complete the backoff period was the downstream node. Another possibility is where both duplicates cross paths on the upstream node, simultaneously moving in opposite directions of the protocol stack. In this case, the upstream node has not physically transmitted the packet but it is too late to stop the transmission. The broadcast protocol cannot detect or control this situation without requiring inelegant interactions between layers.

One of our contributions is to show that “reasonable” backoff times produce an excessive amount of overhead as a result of concurrent retransmissions. Concurrent retransmissions cannot be avoided. They can only be reduced by using

$$\Pr(X = 0) = \prod_{i=1}^{\rho} \frac{t - i\delta + \delta}{t}$$

C. Delay Analysis

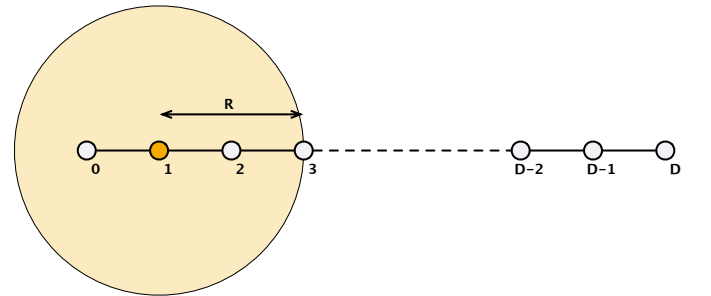


Fig. 4. An example linear network where nodes are separated by 1 unit.

¹Multi-hop broadcast operations are usually not supported below the application level. In particular, it is not supported by the IP protocol suite. Therefore, are algorithm is likely to be implemented at the application level.

TABLE I
LIST OF NOTATIONS EMPLOYED.

Variables	Comments
X_i	Random variable that indicates the farthest node that has seen the message at the end of time step i
χ_D	Delay for node at a distance D to receive the message.

Discrete-time analysis. Let p be the probability of transmitting in a time slot. Let X_k denote the maximum distance the message has traveled at the end of time step k . Clearly, $X_0 = R$.

The conditional probability that $X_{k+1} = j$ given that $X_k = i$ may be computed as:

$$P(X_{k+1} = j | X_k = i) = \begin{cases} (1-p)^R & \text{if } j = i \\ p(1-p)^{R-(j-i)} & 1 \leq j - i \leq R \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We are interested in the first time step at which the distance covered by the message is D .

Continuous-time analysis. Consider an exponentially distributed random backoff period with mean μ^{-1} . Let $X(t)$ denote the maximum distance traveled by the message at time t . Clearly, $X(0) = R$. The distance traveled by the message with time is modeled as a discrete-state continuous-time Markov chain as shown in Figure 5.

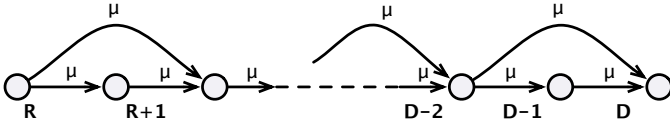


Fig. 5. The distance traveled by a message modeled as a discrete-state continuous-time Markov chain.

The transition rate from state i to state j is given by μ if $1 \leq j - i \leq R$.

The expected time to first visit to state D is computed as follows. The waiting time in a state follows an exponential distribution with mean $\bar{W} = (R\mu)^{-1}$. When the system exits any state, the expected distance covered is given by $\bar{H} = \frac{R+1}{2}$. The expected time by which the message would have traveled a distance $D - R$, denoted by Δ_D , is given by:

$$E[\chi_D] = \frac{2(D-R)}{R+1} \times \frac{1}{R\mu}$$

Note: The above assumes that every node has R neighbors on the right who will transmit the packet. We can also make this as an average case; and we can derive expressions if the number of nodes to the right (reachable through the transmission radius) is randomly distributed with some mean.

Other modifications. You can decrease the expected reception time of the message χ_D by two means - decrease the waiting time or increase the expected distance traveled or both. The latter is achieved by making the waiting time inversely proportional to the distance over which the message was received by a node.

Note: There is a small catch here, if you want to retain the Markovian property. A node receives a message from another node that is distance r_1 units to its left. Assume that its waiting time is exponentially distributed with a mean of $[f(r_1)]^{-1}$. During its waiting time, it hears another message from its left from a node at a distance r_2 (obviously $r_2 < r_1$) from itself. Now, if the node of interest simply wants to wait until the waiting time, then it cannot be modeled as a Markov chain! Instead, we allow the node to re-evaluate its waiting time as a random value that is exponentially distributed with a mean $[f(r_2)]^{-1}$. A node re-evaluates its waiting time every time it hears a message from its left based on the distance from the last node that transmitted the message. In such a case, we can model the above as a continuous time Markov chain.

Let $[f(y)]^{-1}$ denote the mean of the (exponential) waiting time distribution at a node that is distance y away from the last transmitted node. The transition rate from state i to state j is given by $f(j-i)$, $1 \leq j - i \leq R$.

Given the above, the expected waiting time in a state is given by

$$\bar{W} = \left[\sum_{y=1}^R f(y) \right]^{-1} \quad (2)$$

and the average distance when the system exits a state is given by

$$\bar{H} = \frac{\sum_{y=1}^R y f(y)}{\sum_{y=1}^R f(y)} \quad (3)$$

$$E[\chi_D] = \left(\frac{D-R}{\bar{H}} \right) \bar{W} \quad (4)$$

$$= \frac{D-R}{\sum_{y=1}^R y f(y)} \quad (5)$$

Some additional stuff. Assume that users transmit a packet like in an ALOHA channel. Adjusting the back-off timer defines the arrival rate of packets! So, decreasing the waiting time increases traffic rate, hence the probability of collision (e^{-2G} for a simple ALOHA system, where G is the load that is given by $G = R\mu$). Given this, we can then analyze the impact of the number of attempts that a node can make to transmit its packets; or the impact of collision.

The problem with collision is as follows. Assume that a node is waiting to transmit a message. It receives two messages from two users that collide (so it does not know what message was transmitted). In order to be conservative, it assumes that in the worst case both the transmissions were from the

left, therefore, it waits for the random time and transmits the message. Clearly, the transmission of some nodes would be redundant as in reality the collision could have involved transmission from at least one node on the right.

III. SIMULATION RESULTS

This section presents the results of the simulations that were conducted to analyze the overhead and latency of broadcast operations performed with the SB algorithm.

The SB algorithm was implemented in ns-2 [1], a discrete event simulator with extensive support for wireless networks. The protocol was implemented at the application level, meaning that the backoff period of the SB algorithm takes place outside of the MAC layer. Once the backoff period is complete, the packet travels down the protocol stack and is handed to the MAC layer, where an additional backoff period is used as a collision avoidance mechanism. The MAC protocol used in our simulations is 802.11. For broadcast messages, 802.11 is reduced to a simple slotted CSMA protocol.

Figure 6 shows the depth as a function of the node density when the backoff period is exponentially distributed. Results are shown for four different expected values of the backoff period (150 ms, 250 ms, 500 ms, 16 sec). The exponential regression of each dataset is also shown.

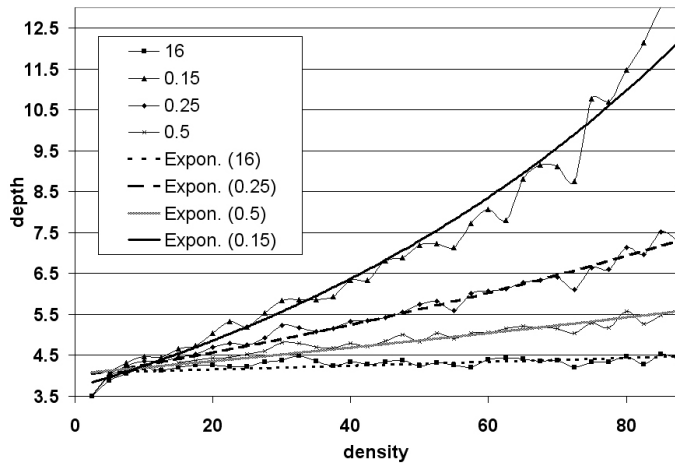


Fig. 6. Depth vs. density for different mean values of exponentially distributed backoff time.

According to Lemma 2, if there are no overlapping retransmissions then the depth is 4 and does not depend on the density or the mean backoff period. However, the probability of concurrent retransmissions increases with shorter mean backoff periods or higher densities. This results in expected depth values that are considerably higher than our theoretical result of 4. Note that the results for the mean backoff period of 16 seconds confirms that the depth approaches four when the probability of concurrent retransmissions is small.

IV. RELATED WORK

V. CONCLUSIONS

REFERENCES

- [1] The VINT Project. The *ns* Manual. Available in html, postscript and PDF, December 2003.