Implementing a Reliable Local Broadcast Primitive in Wireless Ad Hoc Networks

Technical Report (July 2005)

Vartika Bhandari
Dept. of Computer Science, and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
vbhandar@uiuc.edu

Nitin H. Vaidya
Dept. of Electrical and Computer Eng., and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
nhv@uiuc.edu

***Abstract*— Reliable broadcast is an operation of great relevance for distributed applications. Much work in the past has focused on achievability of reliable broadcast in the presence of crash-stop or Byzantine failures. More recently, the issue has been addressed in the context of radio networks [1], [2] that are characterized by a shared channel, and where a transmission is heard by all nodes within the sender's *neighborhood*. This basic defining feature of the radio network model is what we term as the *reliable local broadcast* primitive. However, in actuality, wireless networks do not exhibit such perfect and predictable behavior. Factors such as fading, physical obstacles, and interference can lead to a situation where certain nodes in the theoretical *neighborhood* of a node do not receive the transmitted message. Thus any attempt at building systems for multi-hop wireless networks, based on the radio network model, requires the availability of a reliable local broadcast primitive that can provide guarantees of radio-network-like behavior. We describe a possible implementation of such a reliable local broadcast primitive, that relies on time synchronization to provide probabilistic guarantees. The probability that a node makes an error converges to 0 as the network density increases to infinity. The protocol exploits sender diversity, and hence can also be useful in achieving reliable broadcast when some links exhibit significant continuous periods of down-time (as when a transient obstacle moves in).**

I. A Reliable Local Broadcast Primitive

Theoretical work on Byzantine fault-tolerant broadcast in wireless ad hoc networks has invoked the radio network model [1], [3], [4] wherein if a node transmits a message it is received by each and every node within a designated neighborhood in its spatial vicinity. This ensures local agreement as follows: when the sender is non-faulty, agreement is trivial, since all non-faulty neighbors of a non-faulty sender will receive the message directly. If the sender is faulty and sends multiple conflicting copies of the message, all non-faulty neighbors will receive all messages in the same order, and can agree on a one (say the first). While this model reflects the characteristics of wireless transmissions in terms of the shared nature of the medium, it fails to capture the unreliability that marks wireless transmissions. The wireless medium can be extremely unreliable, and show highly variable channel quality over time, due to various effects such as fading, interference etc. This leads to an error probability that is not merely non-negligible, but can be fairly large. Thus, any attempt at designing reliable broadcast protocols based on theroetical radio network results must begin with an effort to implement a *reliable local broadcast* primitive. Such a primitive would ensure that if a node transmits a message, all non-faulty nodes in its neighborhood are able to agree on a single value for that message, and if the sender was non-faulty, the agreed-upon value is the one transmitted by the sender. This paper considers the possibility of realizing a probabilistic local broadcast primitive, and describes a simple *proof-of-concept* approach with which the probability that any given node makes an error diminishes as the network density increases (when a certain fraction of nodes in each neighborhood may exhibit Byzantine faults). However, as density increases, the delay also increases (due to increased load on a channel of fixed capacity). However, we envision that networks of moderate density would be suitable for such an approach, by allowing one to achieve fairly small error probabilities with acceptable delay.

This research is supported in part by the National Science Foundation.

II. Network and Communication Model

The minimum degree of any node v in the network is d, and the minimum number of common neighbors shared by two neighbors u and v is d_o. Accidental collisions and interference are possible, but deliberate collisions by faulty nodes are not allowed. Address-spoofing is also assumed not to occur, i.e., for each message/packet, the node that put that packet on the air is uniquely and correctly identifiable. If a node transmits a message, the probability that a neighbor successfully receives it is p_s. Thus possible errors due to fading, interference etc. are subsumed in the error probability $(1-p_s)$. We define a timeout T and a probability p_a such that if a packet was put into a node's outgoing queue at time t, then with probability at least p_a, it gets a chance to transmit it by time $t+T$. The choice of T is such that p_a may be large. It may be obtained via loose estimates on network density (and hence contention). All nodes possess a single transceiver and operate on a single channel. They also use a single transmission rate,

and all valid messages are of a predetermined (and equal) size.

We assume that the minimum time between two successive packet transmissions (transmission time + contention resolution etc.) is t_δ. Note that t_δ can be no less than the transmission time of a packet. The maximum propagation delay is ${d_{prop}}^{max}$ (note that ${d_{prop}}^{min} > 0$). Nodes are externally synchronized with bound $D < \frac{1}{2}(t_\delta - {d_{prop}}^{max})$. Such high-precision synchronization may be feasible in the near future with the advent of on-chip atomic clocks [5]. Also observe that given the maximum clock skew D in the network, it is possible to ensure that the condition holds (albeit at the expense of inefficient bandwidth usage) by padding all messages with extra bits to increase the transmission time so that $t_\delta > 2D + {d_{prop}}^{max}$, and the required condition holds. This condition ensures that if a node sends out two different messages on the same channel, then the node-local time at which any non-faulty node receives the later message shall always be greater than the node-local time at which any non-faulty node receives the first message.

Distinct messages sent by a particular source are distinguished via *identifiers*, that we shall denote as *id*. The *id* is a number in some range $[0, MAX]$. Individual nodes choose the sequence of *id*s for their messages in some privately determined pseudo-random manner (such that *id*s are re-used only after large intervals of time). Thus, if a node sends two conflicting versions of the same message, it implies that they both have the same *id*, but different values. Messages are represented as $m(src, id, value)$. We assume that *value* can take values 0 or 1. If a message m is repeated by a neighbor, it is represented as $REPEAT(m, orig\_src, timestamp)$.

III. The Protocol

We present a simple protocol that assumes external time synchronization. Nodes are assumed to have an estimate of who their neighbors are. This is basically required because given a local broadcast source v, non-faulty nodes that not direct neighbors of v, but are 2-hop neighbors, shall receive some protocol messages (the *REPEAT* messages mentioned in the previous section) sent by neighbors of v, but should not erroneously consider themselves to be neighbors and attempt to make a decision about the locally broadcast value. Such an estimate is obtainable by considering only those nodes as neighbors from whom at least one message was directly received in the recent past (over some pre-defined window). A periodic HELLO exchange can help ensure that most valid neighbors satisfy this condition.

The goal of the protocol is that if a local broadcast source v sends a message, then all its non-faulty neighbors should agree on a single value for this message. If v is non-faulty, this agreed-upon value should be the one actually sent by v. If v is faulty and sends multiple conflicting versions of the message, the protocol is designed to enable nodes to choose the *first* value that v sent.

Suppose we have sender v. We follow the following protocol:

- On receipt of a message $m(v, i, p)$ from v directly at (local) time t (the time at which receipt started), if no other earlier version of this message (i.e. of form $m(v, i, q)$) was received from v, make note of p as a candidate message value, and re-broadcast a copy of m as $REPEAT(m, v, t)$. If an earlier version of the same message was received directly from v, discard this message.
- On receipt of a message $REPEAT(m, v, t)$, make note of m as a candidate message with timestamp t. Keep track of all copies of m received from different repeaters along with their timestamps.
- Suppose first copy of message with identifier i from v (direct or repeated) was received at time t. At time $t + T$ (where T is a pre-defined timeout), perform a filtration procedure on the received messages, and determine the version of m for which the highest number of copies were received. Commit to this message version. The majority determination involves application of the following procedure:

  Let us refer to the value with highest count as c_1, and the other one as c_2. If the number of copies of $c_2 \leq b$, choose c_1 as the correct value. If the number of copies of $c_2 > b$: discard any messages with value c_1 whose timestamp t is greater than the timestamps of more than b copies of c_2. Find majority value from amongst the remaining copies of c_1 and c_2.

Theorem 1: Given that the nodes are externally synchronized such that the following holds: $D < \frac{1}{2}(t_\delta - {d_{prop}}^{max})$, if the minimum number of common neighbors shared by any two neighbors is d_o, and less than $\frac{\alpha}{1+\alpha} d_o$ nodes in the neighborhood of any node are faulty (where $\alpha \leq p_a p_s^2 - \varepsilon$, and $\varepsilon > 0$), then the above protocol ensures that all non-faulty neighbors of v shall be able to agree on a common value for v's message with error probability $\to 0$ as $d_o \to \infty$. If v is non-faulty, the agreed value is the one sent by v, else it is the first value sent by v.

Proof: For the sake of simplicity and w.l.o.g., we assume that the message m may take one of two values 0 or 1. There are two cases: v is non-faulty or v is faulty:

- *v is non-faulty:* v transmits exactly one copy of message m. Since any neighbor of v may receive up to b spurious repeats of v's message, we require that the number of received copies of the correct message be $> b$ in order to distinguish a legitimate value from a spurious one.
- *v is faulty:* By assumption, all nodes are externally synchronized to some time source with synchronization bound D. Then, in the protocol described ealier, one may filter out copies of the second message received from non-faulty neighbors. Suppose the sender v sends the two messages at time t_1 and t_2 respectively (according

to the external clock). Then all neighbors of v that received the first message would have received it within $(t_1 - D, t_1 + D + {d_{prop}}^{max}]$ by their local time. Similarly the second message is received by nodes within $(t_2 - D, t_2 + D + {d_{prop}}^{max}]$ by their local time. Since the two messages are sent by v on the same medium, they are temporally ordered i.e. $t_2 > t_1 + t_\delta$. Thus, the receipt time observed by any node for first message $<$ receipt time for second message. All non-faulty nodes attach the correct time at which they started receipt to any *REPEAT* messages they send, and nodes that receive *REPEAT* messages, record the timestamp along with the message.

Let us denote the first message sent out by v as m_1 and the second by m_2. Recall that the message with highest count is also referred to as c_1 and the next highest is referred to as c_2. Then the following cases may arise:

- If $c_1 = m_1$ i.e. m_1 has highest count and $> b$ copies of m_2 were received, then no more than b copies of $c_2 = m_2$ can bear a false earlier timestamp. Thus no copy of m_1 will get filtered out erroneously, and it will win the majority vote.
- If $c_1 = m_2$ i.e. m_2 has the highest count initially, and $> b$ copies of $c_2 = m_1$ were received from non-faulty nodes: then any copy of m_2 sent by a non-faulty node shall have a timestamp later than the timestamps on the $> b$ correct copies of m_1, and the timestamp filtration rule ensures that all copies of m_2 sent by non-faulty nodes will get filtered out, and only upto b copies sent by faulty nodes will remain. Thus if the correct copies of m_1 are $> b$, it will win the majority vote.
- If m_1 had the highest count initially, and $\leq b$ copies of m_2 were received, m_1 will get chosen immediately.
- If m_2 had the highest count initially, and $\leq b$ copies of m_1 were received, an error will be made.

Thus, the algorithm definitely makes the correct decision if more than b correct copies of m_1 were received. This is the same as the requirement for correct decision with a non-faulty source. When less than b copies of m_1 are received, the decision may be correct or wrong. We assume the worst, i.e., it is always wrong.

We represent the correct copies of m_1 received by a node u as a random variable Z. Then, the requirement is that $Z > b$. Let the number of non-faulty common neighbors of the source v and node u be g. Then Z is a binomial random variable, and $Z = Binomial(g, p_a p_s^2)$. This allows us to apply the following special form of the Chernoff bound:

$$Pr[Z \leq (1-\delta)E[Z]] \leq \exp(\frac{-\delta^2 E[Z]}{2}) \qquad (1)$$

Thus, if we set $b = \alpha g$, and $\delta = 1 - \frac{\alpha}{p_a p_s^2}$, application of Chernoff bound yields:

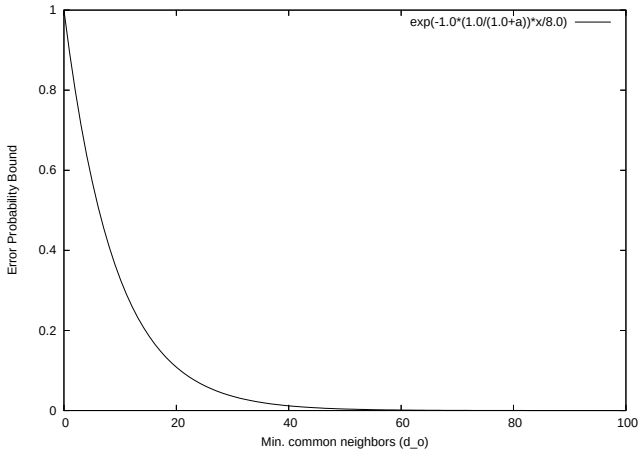

Fig. 1. Upper bound on error probability ($p_a p_s^2 = \frac{1}{2}$, and $\alpha = \frac{1}{8}$)

$$Pr[Z > b] \geq 1 - \exp(-\frac{(1-\frac{\alpha}{p_a p_s^2})^2 p_a p_s^2 g}{2}) \qquad (2)$$

The constraint on α is that $\alpha \leq p_a p_s^2 - \varepsilon$ with $\varepsilon > 0$. Thus α can be large when the probability of successful receipt ($p_a p_s^2$) is large, and becomes very small when it is small. Given a particular value of $p_a p_s^2$, when α is much smaller than the maximum allowable value, the convergence is fairly fast, e.g., if $\alpha = \frac{1}{4} p_a p_s^2$, the probability that a particular node makes an error is $< \exp(-\frac{9}{32} p_a p_s^2 g) < \exp(-\frac{1}{4} p_a p_s^2 g)$. If $p_a p_s^2$ is not very small, e.g. is $\geq \frac{1}{2}$, the error probability $< \exp(-\frac{g}{8}) < \exp(-\frac{d_o}{9})$ (note that in the worst case, $g = \frac{d_o}{1+\alpha}$), which converges very fast (Fig. 1), and hence may be useful in many real-world practical networks.

Note that, as $d_o \to \infty$, one would need to suitably increase the timeout T, in order to maintain a sufficiently high value of p_a (due to increased contention for the channel), i.e., *delay would also increase to infinity*. However, in most cases of practical interest, d_o will not be unduly large, and a moderate value for T can suffice.

Also note that the error probability we have determined is a conservative estimate. We have assumed that whenever the number of received copies of m_1 is less than b, a wrong decision is made. In actuality, if the number of copies of m_1 is less than b but still has the highest count (it is possible that the number of received copies of m_2 be much less than b, since these transmissions are also subject to the same channel conditions), the correct decision will be made. Thus the error probability would actually be somewhat less. ■

REFERENCES

[1] C.-Y. Koo, "Broadcast in radio networks tolerating byzantine adversarial behavior," in *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*. ACM Press, 2004, pp. 275–282.

[2] V. Bhandari and N. H. Vaidya, "On reliable broadcast in a radio network," Technical Report, CSL, UIUC, May 2005.

[3] A. Pelc and D. Peleg, "Broadcasting with locally bounded byzantine faults," *Information Processing Letters*, vol. 93, no. 3, pp. 109–115, Feb 2005.

[4] V. Bhandari and N. H. Vaidya, "On reliable broadcast in a radio network," in *Proceedings of the 24th annual ACM symposium on Principles of distributed computing (to appear)*, 2005.

[5] S. Knappe, L. Liew, V. Shah, P. Schwindt, J. Moreland, L. Hollberg, and J. Kitching, "A microfabricated atomic clock," *Appl. Phys. Lett.*, vol. 85, 2004.