Robust Broadcast :
Improving the reliability of broadcast transmissions on CSMA/CA

Jean Tourrilhes

jt@hplb.hpl.hp.com
Hewlett Packard Laboratories, Filton Road, Bristol BS12 6QZ, U.K.

This paper presents a scheme to improve the efficiency of radio MAC protocols in the case of broadcast and multicast transmissions, like TCP/IP multicasting. First, the reliability problems with broadcast packets and their consequences are analysed. Then the Robust Broadcast scheme is presented, which decreases the probability of loss of broadcast packets over MAC protocols based on CSMA/CA. Finally, the new protocol is simulated against other simple solutions to show how it performs.

1 Introduction

Broadcasting has always been a controversial subject in networking. Broadcasting is by essence unreliable and difficult to manage by applications. Some networking technologies such as ATM which don't provide broadcasting have to include complex mechanisms to accommodate standard networking layers which require broadcasting.

Common Wireless LANs based on CSMA/CA MAC protocol do include broadcasting, directly translated from their wired counterpart. However transmission by radio is fundamentally different than on a wire, and unicast transmission by radio already includes specific mechanisms to adapt to the specific channel conditions, whereas broadcast transmission includes usually none.

2 Usage of broadcast and multicast messages

Most of the traffic on a network is unidirectional packets, because it is the most efficient and convenient way to transmit data between two computers. Broadcast (and multicast) is also used for specific applications.

The first main use of broadcast messages is for network management. Protocols such as TCP/IP try to minimise the use of broadcasting, but still requires it for functionalities related to discovery, such as ARP, DHCP and network autoconfiguration in IPv6. Netbios (Windows networking) over TCP/IP or Netbeui makes a very extensive use of broadcast messages to discover and keep track of the state of the network.

Broadcast and multicast messages are also used by some applications which need to distribute information to multiple nodes. These are mostly multimedia applications, such as gaming, audio/video multicasting or conferencing.

Of course, those applications using broadcast and multicast take into account the fact that broadcast transmissions are unreliable. Network management messages are repeated (for example, the number of ARP requests is 4) and multimedia coding accommodates data loss [2] (usually up to a few percent).

3 The Reliability problem

The transmission of packets on a wireless LAN is notoriously unreliable. This reliability problem creates some performance problems in the transport layer for unicast transfer [3]. The usual way to deal with that problem is to includes MAC level retransmissions, as in 802.11 [1] (see next section).

For broadcast and multicast packets, the problem is even worse than just performance. The transport layer can't include any acknowledgment and retransmission mechanism, due to the non defined number of recipient and the disparity of the reception conditions between them. The MAC level can't provide either any acknowledgment and retransmission scheme for the exact same reason.

On the other hand, the level of reliability expected for broadcast messages is not as high as for unicast (there is no performance problem). The data loss rate accommodated by the broadcast applications is usually in the order of one percent (see previous section).

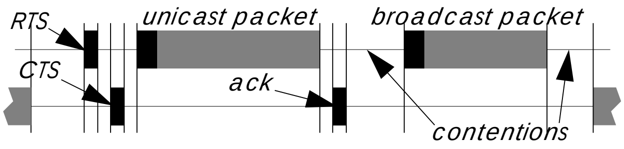
3.1 Unicast techniques

To overcome packet losses on the medium, the MAC protocols used on Wireless LANs use MAC level acknowledgments and retransmissions.

For each packet received correctly, the receiver immediately sends back a positive acknowledgment to the sender. This acknowledgment is embedded in the protocol, so guaranteed not to collide with any other transmission. The sender will keep retransmitting its packet until it receives the acknowledgment (or its timeout expires).

The main restriction is that this mechanism supposes a unique receiver, and can't work with an undefined number of receivers (none or multiple) as in the case of broadcast messages.

Another technique to detect collisions is RTS/CTS [4]. Each packet is preceded by a handshake between the sender and the receiver to ensure that the medium is free. The handshake is composed of a request transmitted by the sender (the RTS) and a reply by the receiver (the CTS) which confirm that it is able to receive. The information contained in the RTS and CTS packets performs medium reservation and solves the hidden node problem [4].



The same restriction as for acknowledgment applies : the receiver must be unique, so this scheme doesn't work with broadcast messages.

3.2 Channel errors

The Bit Error Rate (BER) on the radio waves is much higher that on a wire, due to propagation phenomena (attenuation and fading). The normal condition on a wire is a BER lower than 10^{-9}. The BER on a radio is usually much higher and goes up under certain conditions (large range, obstacles, noise...).

The BER experienced over the radio depends on the transmission and reception techniques, but is closely linked to the attenuation between the sender and the receiver. This relation between BER and attenuation is generally of exponential form, which implies that when the attenuation is lower than the value corresponding to the sensitivity, the packet losses are marginal, and that they tend to be high otherwise (the system works well, or poorly).

Section 6.1 details simulation results showing channel errors for broadcast and unicast transmissions.

To overcome channel errors for broadcast packets, the protocol can use some very classical techniques like FEC or multiple transmissions, or limit the range of the device.

3.3 Protocol collisions

The attenuation and fading is not the only source of packet losses on top of the MAC layer. MAC protocols such as CSMA/CA are contention based, so generate collisions between the different transmitters trying to access the medium (on a per packet basis).

In CSMA/CD (Ethernet), the physical layer is able to detect collisions in the transmitter and so is able to retransmit the failed packets. The radio hardware doesn't have this ability, so the protocol relies on the MAC level acknowledgments to detect collisions. As broadcast packets are not acknowledged, the protocol can't detect collisions.

Most wireless data MAC protocols use CSMA/CA, with a slotted contention. The probability of collision derive from the probability of two nodes choosing the same slot for transmission. 802.11 has a 16 slots contention window, so in theory a 1/16 probability of having two nodes contending for the medium to collide (in fact, as the node having chosen the lowest slot number transmits and contends immediately for the next packet before the other has elapsed its count of slots, this probability is higher). Of course, a higher number of nodes contending will yield a higher probability of collision, and a higher contention window a lower collision rate.

Section 6.2 details simulations showing collision rate for broadcast and unicast messages.

4 Robust Broadcast

Unicast techniques can't overcome the high failure rate of broadcast transmission on radio. Channel errors are usually low in most of the usable range of the device or can be dealt with classical techniques (FEC). However, protocol collisions are high when there is traffic on the network and can not be avoided by usual techniques.

Robust Broadcast is a technique trying to overcome those protocol collisions with minimal impact on the network performance.

4.1 The principle

The goal of the scheme is to detect collisions for broadcast transmissions, in order to perform retransmission in those cases, and only in those cases (to save bandwidth). The principle of Robust Broadcast is to use another node of the network to detect collision (the collision detector) and to feedback this information to the transmitter.

4.2 How to detect collision

CSMA/CA protocols such as 802.11 provide already 2 ways to detect collisions for unicast packets, through packet acknowledgment and through RTS/CTS (see section 3.1).

Using RTS/CTS is the most interesting solution, because of its transparency and the medium reservation feature of the mechanism (giving an advantage towards hidden nodes). Using RTS/CTS for broadcast messages requires almost no change of the MAC protocol. No new field is added in the packet header and no change is required in the receiver or in the collision detector, which ensures backward compatibility. The only change is that before each broadcast, the sender has to choose a collision detector and to perform the RTS/CTS handshake with it.

Because of the possibility of hidden nodes, this scheme can't avoid all collisions in every nodes, but the range extension given by the CTS should help in that respect.

Another way to detect collisions would be to use packet acknowledgment for broadcast : the collision detector has just to acknowledge each broadcast packet on the medium. It would require a few modifications to the protocol : the broadcast packet must include a new field for the address of the collision detector and the collision detector must be able to read that field and act upon it. This might also lead to packets repeated on the medium and in the network stacks of the receivers.

Packet acknowledgment in theory is more powerful than RTS/CTS, because it guarantees that the message was correctly received by the sender of the acknowledgment, whereas RTS/CTS doesn't detect channel errors. In practice, for broadcasting, this doesn't make any difference because the medium condition between each pair of nodes is totally uncorrelated, so the correct or bad reception of the message by one node doesn't give us any hint on the state at the other nodes.

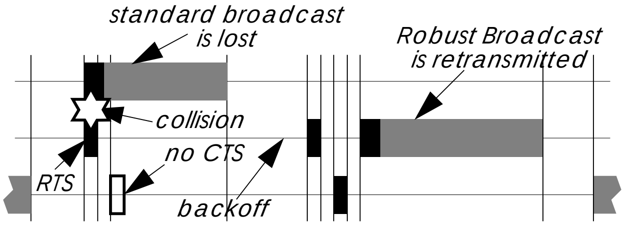
4.3 Description of the scheme

Each time a node wants to send a broadcast (or multicast) packet, it must first get the address of the collision detector (using for example one of the two methods described in the next section). Note that in some cases such an address is not available (initialisation, single node network, timeout), and the whole scheme is disabled.

Then, the sender transmits a RTS addressed to the collision detector. If the node receives the corresponding CTS returned by the collision detector, the node sends the broadcast packet (to the broadcast address). If no CTS is received, the node performs a backoff and retries later (as it would do for an unicast packet).

No change is needed in the receiver behaviour, and any node can be a collision detector.

A simple optimisation of the scheme is to send the packet without RTS/CTS for the last attempt (before to timeout), avoiding problems due to a bad collision detector.



4.4 Choice of the collision detector

The node used for detecting collisions must be unique and active in hearing range at the time of the transmission.

In networks having a base station (or a central coordinator), the scheme can use the base station for that purpose. But, this strategy doesn't cover the base station itself (as a broadcast sender) and the case of ad-hoc networks.

Another strategy is to use the source of the last message sent over the medium that wasn't sent by the node itself. The scheme knows for sure that this node exists and is active.

An advantage of this second solution is that each attempt to send the broadcast message is likely to use a different collision detector. This second strategy should also provide a timeout to discard the address used as the collision detector, and if there hasn't been a message on the medium for a period of time, not to try to detect collisions at all. In this case, the traffic is low, so the probability of collision is also low.

5 Simulation model

The models used for these simulations have been carefully chosen to be simple and realistic, to illustrate the Robust Broadcast scheme and to avoid side effects leading to invalid results.

5.1 MAC model

The MAC model includes a fairly complete 802.11 channel access mechanism. This model is based on an 802.11 backoff (slotted exponential contention). All management functionalities have been removed to keep the model simple.

The model implements MAC level acknowledgments and retransmissions (up to 4), and RTS/CTS (for packets larger than 250 B).

Robust Broadcast (when selected) applies to every broadcast packet regardless of its size except for the last attempt.

The maximum packet size is 1500 B (non fragmented). All other parameters conform to 802.11 [1] (CWmin = 16 ; SIFS = 28 µs ; Slot = 50 µs ; Headers = 50 B).

5.2 Channel model

The channel model is a simple radio channel model, including node to node attenuation (80 dB by default), Rayleigh fading (calculated on a per packet basis) and antenna diversity. The bit rate is 2 Mb/s, and there is no hidden nodes and no interferers. The transmitted power is +20 dBm, and the sensitivity is -80 dBm (in a Gaussian channel).

5.3 Traffic models

Two traffic models were used for the different nodes of the network. The node broadcasting data uses a voice traffic, whereas the other nodes of the network use a data traffic.

The voice traffic model simulates a voice stream over IP. It generates constant size packet at fixed interval. The load is 32 kb/s, the interval between packets 20 ms and the IP overhead 32 B.

We have taken voice traffic as an example, because it is easy to characterise, but all the results will apply to other multimedia applications (like gaming), and to a large extent to management messages.

The data traffic model simulates a large data transfer over TCP. It is a simple bimodal distribution. Each packet is either big (maximum size) or small (40 B), the probability of being small is 1/3. Packets are sent as fast as the link can manage (fully loaded).

More information on the traffic models and their behaviour may be found in [5].

6 Simulation results

Some simulations have been performed to study how well Robust Broadcast performs. All the simulations have been implemented under the Bones® Designer™ environment.

6.1 Medium errors

Figure 6.1 shows the packet losses on a radio medium due to attenuation and fading (one sender, one receiver). This is the failure rate that broadcast transmissions (and Robust Broadcast) will experience. For comparison, the packet failure rate for unicast messages protected by MAC retransmissions (up to 4) has been plotted.

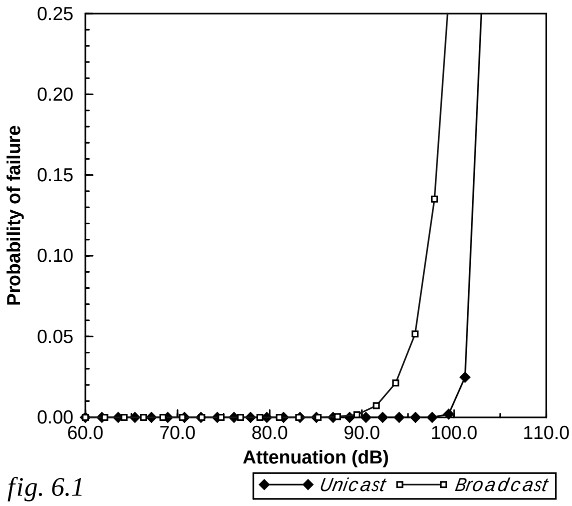


fig. 6.1

This confirms the analysis of section 3.1. Retransmission, as any diversity technique, allows a gain of a few dB in the usable attenuation, adding in this case nearly 10 dB of attenuation to unicast packets compared to broadcast packets (however, unicast transmissions need a much better reliability than broadcast).

At the same time, the simulation shows that in most of the usable range of the device, the rate of failure of broadcast packets remains low (below 0.5 %).

6.2 Packet failures

This set of simulations compares Robust Broadcast with using standard broadcast and unicast techniques (for reference). We have also evaluated two other solutions to reduce the collision losses, the first is to transmit each broadcast packet on the medium twice (*x2*) and the second is to simply increase the size of the contention window for every node of the network to 64 slots instead of 16 (*CW64*).

The network is composed of one broadcast (or unicast) node loaded with a voice traffic and an increasing number of unicast nodes (using the data traffic model) contending for the medium. Figure 6.2 shows the packet failure rate of these different solutions. As we measure only the packet failure rate, these results are quite independent of the traffic nature (arrival rate, packet size...).

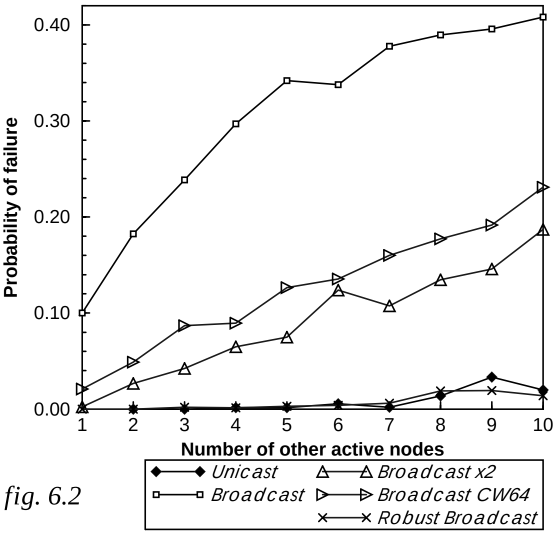


fig. 6.2

With only one unicast node fully loaded, the probability of loosing a broadcast packet is already 10 %, and increases with a higher contention rate. The four retransmission of unicast packets is usually enough to bring that value to reasonable level (but due to the unicast TCP performance problem [3], this parameter should be set to a higher value for such highly loaded networks).

Robust Broadcast reduces the level of failures to the one experienced by unicast packets (and again with only 4 retransmissions). On the other hand, duplicating each broadcast message (*x2*) or increasing the contention window to 64 (*CW64*) give a significant improvement, but still leave a significant packet failure rate.

6.3 Network overhead

The addition of any scheme to improve the reliability of transmission has very often some adverse effects on the network performance. To see the impact of Robust Broadcast on the network performance, the throughput of the whole network has been measured in the same condition as in the previous section. The results are shown in figure 6.3.

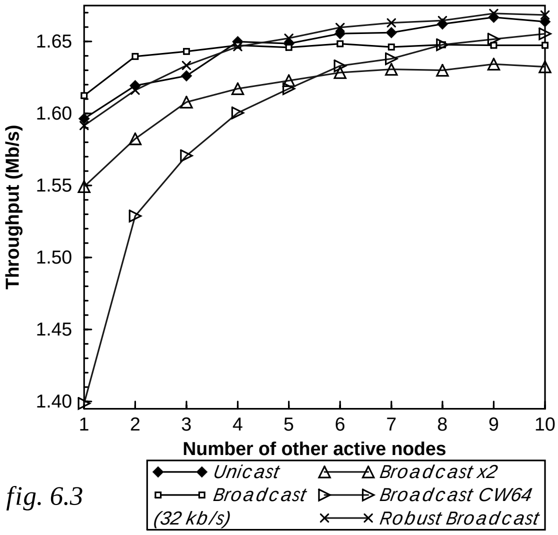


fig. 6.3

To analyse how those results scale with a different traffic, the same simulations have been performed with the throughput of the broadcast node being 128 kb/s instead of 32 kb/s (quadruple the size of each broadcast packet). The results are shown in figure 6.4 (the failure rates in this case are the same as in figure 6.2).

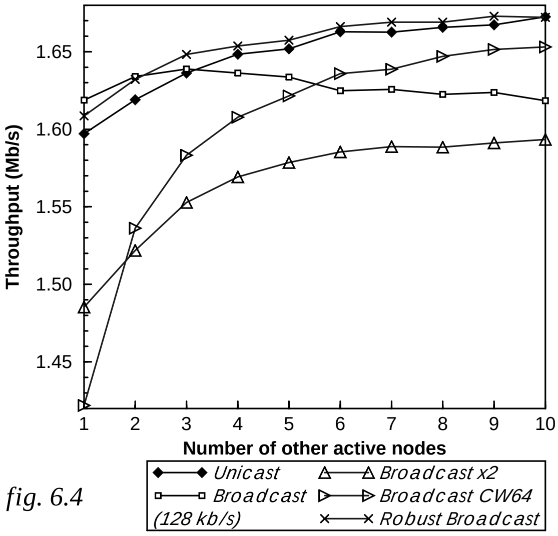


fig. 6.4

Duplicating each broadcast message (*x2*) reduces the available network throughput by the amount of information duplicated (in this case, 32 kb/s or 128 kb/s, plus the overhead, minus the packets discarded due to timeout). Increasing the contention window to 64 (*CW64*) has also a significant impact on the network performance, especially for small number of contending nodes.

On the other hand, the impact of using Robust Broadcast is very small (comparable to the difference between using broadcast and unicast). The curve of unicast and Robust Broadcast are very similar because the sender in both case has exactly the same behaviour (backoff and then retransmit). The added overhead per packet of RTS/CTS in the case of Robust Broadcast is compensated by the reduced penalty of collisions over the unicast and broadcast solutions.

An interesting feature of RTS/CTS shown on those graph is that when the number of nodes increase, the throughput increase. This is because the overhead of collision with RTS/CTS is low, and increasing the number of nodes decrease the average number of slots between packets. The additional reason is that the available bandwidth is fairly shared between all the nodes contending, and, with a higher number of nodes, the portion of the broadcast traffic is proportionally reduced (and it uses a less efficient packet size than the data nodes).

7 Conclusions

Transmission on a radio network are by the nature of the medium and by design of the MAC protocol unreliable. Standard MAC protocols include techniques to overcome this problem in the case of unicast transmissions. However, if broadcast transmissions are infrequent, they are often necessary, and they suffer in many cases from this high failure rate. The most important cause of packet losses is, in general, the protocol packet collisions.

Robust Broadcast fits totally transparently in a 802.11 network and allows it to overcome most of the collision losses for broadcast transmissions. Simulations have shown that it reduces those losses to very low level, as low as for unicast transmissions, without adding any significant overhead to the network.

8 References

[1] *IEEE 802.11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications.* IEEE.

[2] V. Varma, M. Thomas, S. Konish, L. Seltzer and D. Goodman. *Performance of 32 kb/s ADPCM in Frame Erasures.* Proc. of IEEE VTC ‘94.

[3] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan and Randy H. Katz. *A comparison of mechanisms for improving TCP performance over wireless links*. Proc. of ACM SIGCOM ‘96.

[4] Phil Karn. *MACA - A new channel access method for packet radio*. Proc. of the 9th ARRL/CRRL amateur radio computer networking conference.

[5] Jean Tourrilhes. *Packet Frame Grouping : Improving IP multimedia performance over CSMA/CA*. Submitted to ICUPC ‘98.