Effective Delay-Optimal Broadcast for Multihop Wireless Networks Using Self-Interference Cancellation

1Manikandan.S, 2Rajesh. K

1,2Sree Sastha Institute Of Engineering and Technology Chennai, Tamilnadu, INDIA
Email: 1manikandhansekar@gmail.com, 2Kr.raham@gmail.com

Abstract- Conventional wireless broadcast protocols rely heavily on the 802.11-based CSMA/CA model, which avoids interference and collision by conservative scheduling of transmissions. While CSMA/CA is amenable to multiple concurrent unicasts, it tends to degrade broadcast performance significantly, especially in lossy and large-scale networks. In this paper, the propose a new protocol called Chorus that improves the efficiency and scalability of broadcast service with a MAC/PHY layer that allows packet collisions. Chorus is built upon the observation that packets carrying the same data can be effectively detected and decoded, even when they overlap with each other and have comparable signal strengths. It resolves collision using symbol-level interference cancellation, and then combines the resolved symbols to restore the packet. Such a collision-tolerant mechanism significantly improves the transmission diversity and spatial reuse in wireless broadcast. Chorus’ MAC-layer cognitive sensing and scheduling scheme further facilitates the realization of such an advantage, resulting in an asymptotic broadcast delay that is proportional to the network radius. This advantage is exploited further by Chorus’s MAC-layer cognitive sensing and scheduling scheme. The evaluate Chorus with symbol-level simulation, and validate its network-level performance via ns-2, in comparison with a typical CSMA/CR broadcast protocol.

1.1 SCOPE
A novel broadcast protocol, called Chorus, based on a MAC layer that adopts CSMA with collision resolution (CSMA/CR). Chorus is built upon the key insight that packets carrying the same data can be detected and decoded, even when they overlap at the receiver with comparable strength. With Chorus, collision of the same packets from different relays can be effectively resolved. In this section, The introduce the physical-layer collision resolution in Chorus. For clarity, The start with a simple case of two-packet collision, focusing on how to detect, decode, and combine the collided packets to achieve the diversity gain. Then, The deal with the general case of resolving more than two packets’ collision. Note that The have adopted a similar PHY layer in a separate paper which presents a more comprehensive introduction to the implementation of collision resolution in software radios. Its objective is to realize non-orthogonal cooperative communications without tight synchronization among relays.

It resolves collision using symbol-level interference cancellation, and then combines the resolved symbols to restore the packet. Such a collision-tolerant mechanism significantly improves the transmission diversity and spatial reuse in wireless broadcast, providing an asymptotic broadcast delay that is proportional to the network radius. This advantage is exploited further by Chorus’s MAC-layer cognitive sensing and scheduling scheme. The evaluate Chorus with symbol-level simulation, and validate its network-level performance via ns-2, in comparison with a typical CSMA/CR broadcast protocol.

1.2 Existing System:
The double-coverage broadcast (DCB), for example, reduces redundant transmissions by selecting nodes that cover more neighbors, while ensuring each node is covered at least twice, such that retransmission can be
exploited to improve delivery ratio. The fundamental difference between Chorus and such existing protocols lies in its MAC layer scheduling protocol. It simplifies the relay-selection problem by allowing random mixing of information at relays. However, network coding only applies to continuous broadcast, where a batch of source packets can be encoded. Network coding improves broadcast efficiency for lossy and mobile networks, but still cannot achieve the optimal latency in general.

**Disadvantages**

- The key insight that packets carrying the same data can be detected and decoded, even when they overlap at the receiver with comparable week.
- Therefore, only two time slots are required to deliver one packet over the entire network, due to the improved spatial reuse.
- In reality, it is difficult to synchronize the independent transmitters A and B at the symbol level.
- The asynchrony between them to identify collision-free symbols in the overlapping packets.

**1.3. Proposed System:**

A novel broadcast protocol, called Chorus, based on a MAC layer that adopts CSMA with collision resolution (CSMA/CR). Chorus is built upon the key insight that packets carrying the same data can be detected and decoded, even when they overlap at the receiver with comparable strength. With Chorus, collision of the same packets from different relays can be effectively resolved.

The advantage of such a collision-tolerant protocol is obvious. With collision resolution, A and B can now transmit packets immediately and independently after receiving them from the source. Node D exploits Chorus’ collision resolution to decode the two collided packets from A and B. Therefore, only two time slots are required to deliver one packet over the entire network, due to the improved spatial reuse. Moreover, when links are unreliable, the two decoded packets from A and B create transmit diversity for the common receiver D, without consuming any additional channel time.

**Advantages**

- This is because Chorus detects and decodes collided packets with a relatively high SNR, while treating undetectable packets as noise.
- The implement the cognitive sensing and broadcast scheduling protocols based on the 802.11b module in ns-2.
- Since it requires a strict definition of neighborhood, DCB assumes the existence of a transmission range, within which all nodes receive packets from the transmitter with the same probability.

**II. SURVEY**

**2.1 Minimum-Latency Broadcast Scheduling in Wireless Ad Hoc Networks**

Given by Scott C.-H. Huang, Peng-Jun Wan, Xiaohua Jia, Hongwei Du and Weiping Shang in May 2007 A wide range of applications for wireless ad hoc networks such as military surveillance, emergency disaster relief and environmental monitoring are time-critical. These applications impose stringent requirement on the communication latency. One major challenge in achieving fast communication is how to handle the intrinsic broadcasting nature of radio communications.

From the perspective of communication latency, the broadcasting nature of radio transmission is a double-edged sword. On one hand, it may speed up the communications since it enables a message to reach all neighbors of its transmitter simultaneously in a single transmission. On the other hand, it may also slow down the communications since the transmission by a node may interfere and disable nearby communications. In order to achieve fast communication, one has to magnify the speed-up impact while diluting the slowdown impact of the broadcasting nature. In this paper, the problem Minimum-Latency Broadcast Scheduling (MLBS) in wireless ad hoc networks in which communication proceeds in synchronous time-slots.

**2.2 Collision-Free and Low-Latency Scheduling Algorithm for Broadcast Operation in Wireless Ad Hoc Networks**

Given by Wei Wang, Student Member, IEEE, and Boon-Hee Soong in OCT 2007, BROADCAST operation, during which messages are destined to all network nodes, plays an important role in wireless ad hoc network such as network-wide dissemination of data messages and control messages. Sometimes it is critical for broadcast operation to minimize the latency of one round of dissemination, which is the time from when the message is initiated until it reaches all nodes in the network. Taking Mobile Ad-hoc Network (MANET) for example, low-latency broadcast operation could quickly disseminate an linkbreak message to the whole network so that the topology information needed by the routing module at each node could be updated in time. Besides, emerging real-time applications like spontaneous audio conferencing also require a low-latency broadcast operation to deliver their delay-sensitive traffic over wireless ad hoc networks. In literature related to broadcast operation, most work has been focused on how to minimize the number of transmissions in a broadcast operation, and collisions were prevented simply through a random backoff timer for messages forwarding without consideration in latency performance.

Among the limited existing work on collision-free low-latency broadcast operation, prove that finding optimal solution for minimizing latency in wireless ad hoc network is NP-hard, and propose a phase...
assignment (or scheduling) algorithm to avoid collision. A recent work adopts a 2-step approach (broadcast tree construction plus transmission scheduling), and proposes an algorithm that generates both bounded latency and bounded number of transmissions results.

Scheduled broadcast operation actually resembles a wave propagation process, and the number of “wave cycle” for completing broadcast operation can be minimized if in every cycle the “wave-front” nodes cover maximum number of new nodes. Following this kind of greedy strategy, it is possible to construct a low-latency and collision-free broadcast scheduling algorithm, which is able to cover maximum number of untouched nodes in each wave cycle without incurring collision.

A greedy broadcast scheduling algorithm based on MWIS to minimize latency without incurring collision in wireless ad hoc network. Simulation results show that our approach is better than recent 2-step algorithm in term of both D-max and the number of transmissions. Furthermore, despite of its approximation, GWMIN2 can efficiently produce comparable performance result to that of BF method whose running time is exponential. It shall be noted that, similar like the algorithm, our algorithm relies on the knowledge of overall network topology.

2.3 Low-Latency Broadcast Scheduling in Ad Hoc Networks

Given by Scott C.-H. Huang, Peng-Jun Wan., Xiaohua Jia, and Hongwei Du in AUG 2009 Among many operations of mobile ad hoc networks, broadcast is probably the most fundamental yet challenging operation since [25] tells us naïve flooding is simply not practical. Our first objective is to find a good scheduling algorithm that can mitigate the impact of potential collision and have a low broadcast latency. In addition, this paper also want our algorithm efficient such that nodes only have to transmit the message very few times. Redundancy is measured by how many times a node has to retransmit in order to guarantee collision-free reception. This paper want to balance latency and redundancy in this work.

It is known that broadcast in ad hoc networks has a constant approximation algorithm. However, it is still not practical because the approximation ratio is overwhelmingly large (it was estimated to be near 648). In this paper, we present two basic broadcast scheduling algorithms that significantly reduce this ratio. One of our algorithms has ratio 51 and another has 24, and, more importantly, they do not increase redundancy much. The two algorithms guarantee that each node only has to retransmit and 4 times to guarantee proper reception respectively. Moreover, this paper also present a highly efficient algorithm whose latency is $R*O(\sqrt{Rlog1.5} \ R)$ (where $R$ is the network radius) and each node only has to transmit up to 5 times. This result, in a sense of approximation, is nearly optimal since $O(\sqrt{Rlog1.5} \ R)$ is negligible when $R$ is large. Moreover, $R$ is itself a lowerbound for latency, so the approximation ratio is nearly 1 and this algorithm is nearly optimal.

III. PROBLEM DESCRIPTION

This chapter deals with the problem definition of the paper and methodology used in this paper. Problem definition describes the detailed information about the paper.

The source node to all other nodes in the network, with high packet-delivery ratio (PDR) and low latency. To improve PDR in a lossy network, multiple relay nodes can forward and retransmit each packet, thereby creating retransmission diversity. To reduce latency and resource usage, however, the number of transmissions must be kept to minimum, since redundant retransmissions waste channel time, slowing down the packet’s delivery to the edge of the network. Therefore, a delicate balance needs to be maintained between PDR and delay. A typical scenario where CSMA/CA restricts the broadcast efficiency.

- The key insight that packets carrying the same data can be detected and decoded, even when they overlap at the receiver with comparable week.
- Therefore, only two time slots are required to deliver one packet over the entire network, due to the improved spatial reuse.
- In reality, it is difficult to synchronize the independent transmitters A and B at the symbol level.
- The asynchrony between them to identify collision-free symbols in the overlapping packets.

IV. ALGORITHM

4.1 Chorus:

Chorus’ physical-layer collision resolution must be integrated with the MAC layer, in order to reduce unresolvable collisions occurring when packets with different data collide. In addition, Chorus’ network layer must ensure broadcast packets can reach the network edge.

4.2 Chorus Packet Format

Below Fig. illustrates the broadcast packet format in Chorus. First, a known random sequence is attached to facilitate packet detection and offset identification.
Second, a Chorus header field is added, which informs the receiver of the packet’s identity, including the broadcast source’s ID and the packet’s sequence number. A 16-bit Cyclic Redundancy Check (CRC) [15] is included in this header. In case of CRC failure, this packet is discarded as it conveys wrong identity information.

When the headers of two packets collide, Chorus proceeds with the iterative decoding, assuming they have the same identity. After the decoding, it performs CRC over the header of each packet to ensure they are identical. If not, a decoding failure occurs, and both packets will be discarded. A decoding failure also occurs when the CRC check over the payload fails.

4.3 Scheduling of Sensing and Transmissions

With the collision-resolution capability, each transmitter calls a SEND procedure to perform cognitive sensing, as shown in Fig. 2. Transmitters make scheduling decision following three rules:

R1. Forward a packet immediately if the channel is idle.

R2. If the channel is busy, and the packet on the air is exactly one of the packets in the transmit queue, then start transmission of the pending packet.

R3. If the channel is busy, but a preamble cannot be detected, or the header field of the packet on the air cannot be decoded, or a different packet is on the air, then start the back off procedure according to the 802.11.

R1 is typical of all CSMA protocols. R2 is unique to Chorus’s CSMA/CR. It enforces the principle of Chorus, i.e., overlapping packets carrying the same data may not cause collisions. Instead, by collision resolution, these packets offer transmit diversity to the receiver. Therefore, a sender node, such as node B in Fig. 1, can transmit its pending packet if it has the same identity as the one on the air (e.g., the one that A is transmitting). In contrast, CSMA/CA transmitters stall and back off whenever the channel is busy.

R3 ensures friendliness to alien traffic, and is relevant for multisource broadcast and coexistence with CSMA/CA-based unicast traffic. To prevent unresolvable collisions between different packets, Chorus starts the normal 802.11 back off if it senses that the channel is occupied by such alien traffic. To reduce interference to coexisting traffic, it also backs off conservatively if the identity of the packet on the air cannot be decoded.

The advantages of cognitive sensing and scheduling come at the expense of additional overhead. In 802.11b, the sensing time slot is 20 $\mu$s, equivalent to the channel time of 20 bits in the broadcast mode. In contrast, Chorus needs to sense over the entire preamble and the header (80 bits in total, as indicated in Fig. 1). However, this overhead is negligible compared to the packet length (a similar result holds for 802.11a/g/n).

V. MODULE DESCRIPTION

A modular design reduces complexity, facilities change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different part of system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirement.

Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comps ability, Modular Understandability, Modular continuity.

The following are the modules of the paper, which is planned in aid to complete the paper with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.
5.1 MULTIHOP WIRELESS NETWORKS

Wireless Mesh Network (WMN) has become an important edge network to provide Internet access to remote areas and wireless connections in a metropolitan scale. In this paper, the study the problem of identifying the maximum available bandwidth path, a fundamental issue in supporting quality-of-service in WMNs. Due to interference among links, bandwidth, a well-known bottleneck metric in wired networks, is neither concave nor additive in wireless networks. We propose a new path weight which captures the available path bandwidth information. We formally prove that our hop-by-hop routing protocol based on the new path weight satisfies the consistency and loop-freeness requirements. The consistency property guarantees that each node makes a proper packet forwarding decision, so that a data packet does traverse over the intended path. Our extensive simulation experiments also show that our proposed path weight outperforms existing path metrics in identifying high-throughput paths.

5.2 DOUBLE COVERAGE BROADCAST

Double-covered broadcast (DCB) that takes advantage of broadcast redundancy to improve the delivery ratio in the environment that has rather high transmission error rate. Among 1-hop neighbors of the sender, only selected forward nodes retransmit the broadcast message. Forward nodes are selected in such a way the sender's 2-hop neighbors are covered and the sender's 1-hop neighbors are either a forward node, or a non-forward node but covered by at least two forwarding neighbors. The retransmissions of the forward nodes are received by the sender as confirmation of their receiving the packet. The non-forward 1-hop neighbors of the sender do not acknowledge the reception of the broadcast. If the sender does not detect all its forward nodes' retransmissions, it will resend the packet until the maximum times of retry is reached. Simulation results show that the algorithm provides good performance for a broadcast operation under high transmission error rate environment.

5.3 COLLISION RESOLUTION IN CHORUS

Chorus is built upon the key insight that packets carrying the same data can be detected and decoded, even when they overlap at the receiver with comparable strength. With Chorus, collision of the same packets from different relays can be effectively resolved. Both the spatial reuse and the transmit diversity gain in Chorus are realized via its collision resolution scheme, which is based on self-interference cancellation. Chorus exploits the asynchrony between them to identify collision-free symbols in the overlapping packets. It then initiates an iterative cancellation process that subtracts clean and known symbols from the collided ones, and obtains estimates of unknown symbols. The decoding succeeds as long as one packet has sufficient SNR, hence realizing the diversity offered by multiple transmitters. Chorus exploits the asynchrony between them to identify collision-free symbols in the overlapping packets. It then initiates an iterative cancellation process that subtracts clean and known symbols from the collided ones, and obtains estimates of unknown symbols. The decoding succeeds as long as one packet has sufficient SNR, hence realizing the diversity offered by Multiple transmitters.

VI. IMPLEMENTATION

Chorus Algorithm can be implemented using Network Simulator 2.31. NS is a discrete event simulator targeted at networking research. It provides substantial support for TCP routing and multicast protocols over wired and wireless networks. Using Xgraph (A plotting program) we can create graphical representation of simulation results. All the work is done under Linux platform, preferably ubuntu.

6.1 IMPLEMENTATION PROCESS:

In NS-2, the frontend of the program is written in TCL (Tool Command Language). The backend of NS-2 simulator is written in C++ and when the tcl program is compiled, a trace file and namfile are created which define the movement pattern of the nodes and keeps track of the number of packets sent, number of hops between 2 nodes, connection type etc at each instance of time. In addition to these, a scenario file defining the destination of mobile nodes along with their speeds and a connection pattern file (CBR file) defining the connection pattern, topology and packet type are also used to create the trace files and nam files which are then used by the simulator to simulate the network. Also the network parameters can be explicitly mentioned during the creation of the scenario and connection-pattern files using the library functions of the simulator.

6.2 RESULTS AND DISCUSSION

To detect the node collision in mobile sensor networks, Chorus algorithms, are proposed. Chorus is built upon the key insight that packets carrying the same data can be detected and decoded, even when they overlap at the receiver with comparable strength. With Chorus, collision of the same packets from different relays can be effectively resolved. The techniques developed in our solutions, challenge-and-response and encounter-number, are fundamentally different from the others.

6.3 Screenshot:

- Using the terminal to run the pro_nam.tcl programing file. Then the network animator will open and show the pro_nam window.
Fig 9.1 Run TCL in Terminal
➢ Delay optimal broadcast in multihop wireless mesh networks.

Fig 9.2 Mesh Network
➢ Delay optimal broadcast in multihop wireless mesh networks. After that self information cancellation.

Fig 9.3. Self-Interference Cancellation
➢ Now self information cancellation. Then spatial going to reuse.

Fig 9.4. Spatial Reuse
➢ Spatial going to reuse then transmit diversity.

Fig 9.5. Transmit Diversity
➢ Now source node scalability of quality service. Than multiple groups forming.
➢ Now multiple groups forming after that source node helps to multiple concurrent unicasts.

9.2 XGRAPH
Congestion window size is a major factor in predicting the performance of a TCP Network. NS2 handles the Congestion window like a variable cwnd, and it can be printed to a graph and as well printed as values at various intervals of time.

The tcl script will let you understand the working of the network (TCP Reno Protocol) and also helps you in plotting the characteristics of Congestion window value. Xgraph is used to plot the congestion window.

Below the figures shows the performance between the Chorus and DCB algorithm in graph format
➢ Packet delivery ratio is defined as the ratio of data packets received by the destinations to those generated by the sources.
➢ Y-axis indicate the packet ratio
➢ X-axis indicate the time
➢ Channel measurements are indispensable for wireless system design. It is the wireless channel that determines the ultimate performance limits of any communication system. In the beginnings of cellular communications, fading and path loss of the narrowband channel were the key figures of merit.
➢ Y-axis indicate the packet ratio
➢ X-axis indicate the time

VII. CONCLUSION
This paper the Chorus scheduling with collisions was studied. The first provided a model and gave a throughput formula that takes into account the cost of collisions and overhead. The formula has a simple product form. Next, distributed algorithms is designed by where each link adaptively updates its mean transmission length to approach the throughput-optimality and provided sufficient conditions to ensure the convergence and stability of the algorithms.
REFERENCES


△△△