Optimal Broadcast for Multihop Wireless Network through Collision Resolution

NisturniMuthu.M[1],N.Ishwareya[2], G.Vijayalakshmi[3]
Agni College of Technology, Chennai, India
nishllad@gmail.com[1],ishwareya@gmail.com[2],vijayalakshmi7g@gmail.com[3]

International Conference on Green Technologies for Power Generation and Communications-ICGPC 2014

Abstract- A Conventional wireless network uses a 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. But there is a major disadvantage in it. We could not able to recognize the nodes which is in out of range (ie hidden terminals).In order to bring those hidden nodes inside a clustered network we break the mesh network into a Random network. So that we can reach all the nodes which is beyond the limit.

Keywords: *Optimal broadcast, wireless ad hoc and mesh networks, collision resolution, multipacket reception, self-interference cancellation, analog network coding,Hidden Terminals,clustered network.*

INTRODUCTION

Network wide broadcast is a fundamental primitive for many communication protocols in multihop wireless networks, such as route discovery and information dis-semination. An efficient broadcast protocol needs to delivers packet (or a continuous stream of packets) from 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.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. 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, we introduce the physical-layer collision resolution in Chorus. For clarity, we 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. 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 [5]. Unlike traditional transmit diversity schemes such as beam forming ,Chorus requires neither symbol time synchronization nor instantaneous channel state information. In reality, it is difficult to synchronize the independent transmitters A and B at the symbol level . The decoding succeeds as long as one packet has sufficient SNR, hence realizing the diversity offered by multiple transmitters.

At the MAC layer, Chorus adds a cognitive sensing and scheduling module to the 802.11 CSMA mechanism. Specifically, senders back off only when they sense a packet on the air that has a different identity from what they intend to transmit. Such a cognitive MAC allows Chorus to fully exploit the advantage of collision resolution, while main-taining friendliness to background traffic. In addition, the collision-resolution capability enables anonymous broad-cast at the network layer, without any topology or neighborhood information.

To quantify the effectiveness of Chorus, we establish an analytical framework for its achievable SNR and bit error rate (BER), which takes into account the error-propagation effects in iterative collision resolution. We further analyze its network-level performance in terms of latency and throughput. With a joint design of CSMA/CR and broad-cast, Chorus achieves Âorp latency (r is the network radius), which is asymptotically optimal and unachievable in existing CSMA/CA-based broadcast protocols. The performance gain is relatively insensitive to network size, source rate, and link quality, and is observed for both static and mobile topologies, and in both single- and multisource broadcast scenarios. These salient properties are important, especially for information dissemination in large-scale wireless net-works, and represent the importance of exploiting PHY-layer signal processing to improve application performance.

RELATED WORK

Efficient broadcast in multihop wireless networks has beenstudied extensively, from both theoretical and practical perspectives. From the theoretical perspective, it is well known that scheduling a minimum-latency broadcast is NP-hard, either in a general undirected graph [3] or in a unit disk graph (UDG) [1]. Without the minimum latency constraint, analytical solutions demonstrated the feasibility of scheduling with time complexity ðr log nÞ in a distributed anonymous broadcast, and r þ Oðlog rÞ [2] in centralized broadcast with a known topology, where r and n denote the network radius and number of nodes. More recent work has improved the efficiency, and adopted more realistic models, such as the interference graph.

The above algorithmic solutions generally assume perfect MAC-layer scheduling. In reality, scheduling in wireless networks is mostly based on distributed CSMA/CA. The widely used 802.11 standards provide best- effort service broadcast, using CSMA/CA without any ACK or retransmissions. Practical broadcast protocols have mostly adopted the 802.11 CSMA/CA and extended it to multihop networks. A main mechanism is to prune the topology, leaving only a backbone that covers the entire topology. The double-coverage broadcast (DCB) [4], 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. With a joint design of CSMA/CR and network level broadcast, Chorus can achieve the ÂðrÞ latency bound, hence it has both theoretical and practical relevance.

The PHY layer of Chorus shares similar spirits with the ZigZag protocol , which exploits the signal processing capability of SRs to solve the hidden terminal problem in WLANs. ZigZag extracts symbols from collided packets by identifying repeated collisions of two hidden terminals. It treats each collided packet as a sum over two packets. The two original packets are recovered from two known sums, similar to solving a linear system of equations. In the PHY layer, Chorus uses a collision resolution mechanism similar to ZigZag, but it resolves multiple packets from a single collision, given that the packets are identical. In addition, Chorus aims to improve broadcast efficiency in wireless mesh networks, where it exploits transmit diversity and spatial reuse, using MAC-layer cognitive sensing and broadcast scheduling. The feasibility of allowing concurrent transmissions to create diversity has also been explored in communications. Concurrent cooperative communication, for example, allows collocated wireless nodes to transmit at the same time, thus forming a virtual antenna array that increases signal strength at the common receiver. Beam forming protocols [6] synchronize the transmitters, such that their signals can combine coherently at the receiver. These techniques require strict frequency, phase, and time synchronization at the symbol level, among distributed transmitters. Such fined- grained synchronization remains an open problem [6], due to the limited time resolution at the wireless nodes, and the variation of the wireless channels.

COLLISION RESOLUTION IN CHORUS

In this section, we introduce the physical-layer collision resolution in Chorus. For clarity, we 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, we deal with the general case of resolving the collision of more than two packets.

Detecting Collided Packets

In Chorus, a transmitter attaches a known random sequence to the beginning of each packet as a preamble. The receiver then uses a matched filter to detect the exact arrival time of this preamble. A matched filter is an optimal linear correlator that maximizes the SNR when correlating unknown signals with a known sequence [1]. It outputs a peak value whenever the packet preamble is detected, even if the preamble is hidden in a strong noise. It operates continuously, so that those preambles overlapping with other packets can still be identified. The number of preambles detected in a run indicates the number of overlapping packets at the receiver.

The peak output grows linearly with the number of bits in the preamble, and with the RSS of the packet [6]. Therefore, the detection threshold is also a linear function of these two factors . In has been observed that using a 32-bit pseudorandom preamble, the collision detection probability is higher than 98 percent under practical wireless settings [3]. Hence, the preamble introduces negligible overhead to the packet.

Iterative Resolution of Collision

Since a packet usually consists of thousands of symbols, the probability of two collided packets being aligned perfectly is close to 0. In practice, the higher layer operations at transmitters introduce further randomness, resulting in asynchronous arrivals. We identify the natural offset between the two packets by detecting their preambles. Within the offset region, no collision occurs. We first decode the clean symbols therein, and then iteratively subtract such known symbols from the collided ones, thereby obtaining the desired symbol.

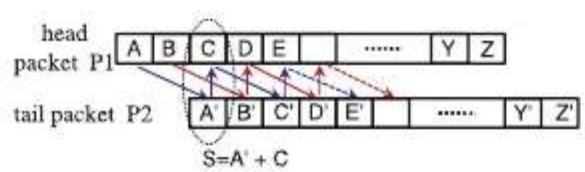

Fig. 1. Iterative decoding of two collided packets carrying the same content.

For instance, Fig. 2 shows collision of two packets (headpacket P1 and tail packet P2) from different transmitters. We first decode the two clean symbols A and B in P1. Symbol C is corrupted as it collides with A0 in P2, resulting in a combined symbol S. To recover C, note that symbols A0 and A carry the same bit, but the analog forms are different because of channel distortion There fore, we need to reconstruct an image of A0 by emulating the channel distortion over the corresponding bit that is already known via A. The channel distortion effects, including amplitude attenuation, phase shift, frequency offset, and timing offset, can be accurately estimated using standard communication techniques, as demonstrated in a realistic experimental setting.

After reconstruction, we subtract the emulated A0 from S, obtaining a decision symbol for C. Then, the decision symbol is normalized using the channel estimation for P1, and a slicer decides if the bit in C is 0 or 1. For BPSK, the slicer outputs 0 if the normalized decision symbol has negative real part, and 1 otherwise. The decoded bit in C is then used to reconstruct C 0 and decode E. This process iterates until the end of the packet is reached. The iteration for other collided symbols proceeds similarly. The estimation, reconstruction, and cancellation for higher order modulation schemes, such as M-PSK (M ¼ 4, 8, 16,64), can be realized similarly, except that the signal constellation is mapped to different complex numbers . Also, note that the above procedure has linear complexity with respect to packet length, which is similar to ZigZag [6] and interference cancellation [5].

Multipacket Collision Resolution

Since Chorus allows concurrent transmissions, multiple versions of a packet can collide, especially when the network has high density. The resolution of multipacket collision is complicated by the fact that intermediate packets no longer have clean symbols at the beginning or end.

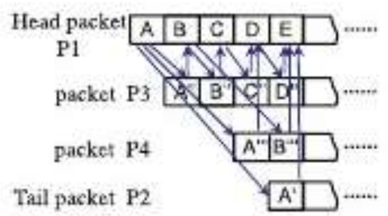

Fig. 2. Collision resolution: The multipacket collision case.

COGNITIVE SENSING AND BROADCAST SCHEDULING

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. Next, we detail both the MAC- and network-layer support for broadcast.

HIDDEN TERMINAL

The hidden-node problem has been shown to be a major source of Quality-of-Service (QoS) degradation in Wireless Sensor Networks (WSNs) due to factors such as the limited communication range of sensor nodes, link asymmetry and the characteristics of the physical environment. In wireless contention-based Medium Access Control protocols, if two nodes that are not visible to each other transmit to a third node that is visible to the formers, there will be a collision – usually called hidden-node or blind collision. This problem greatly affects network throughput, energy-efficiency and message transfer delays, which might be particularly dramatic in large-scale WSNs. This paper tackles the hidden- node problem in WSNs and proposes H-NAMe, a simple yet efficient distributed mechanism to overcome it. H-NAMe relies on a grouping strategy that splits each cluster of a WSN into disjoint groups of non-hidden nodes and then scales to multiple clusters via a cluster grouping strategy that guarantees no transmission interference between overlapping clusters. We also show that the H-NAMe mechanism can be easily applied to the IEEE 802.15.4/ZigBee protocols with only minor add-ons and ensuring backward compatibility with the standard specifications.

TRIPLE HIDDEN PROBLEM

RTS/CTS based multi-channel MAC protocols have been extensively researched for terrestrial radio networks [7], in these studies, the communication link is divided into one control channel and multiple data channels and channel assignment is integrated into the RTS/CTS handshaking process on the control channel. For single-transceiver multi-channel long-delay underwater networks, however, these approaches are not efficient, because, except for the traditional multi-hop hidden terminal problem for the single channel network, they will suffer from two new hidden terminal problems that are inherent in the new network scenario: multi-channel and long-delay hidden terminal problems.

A. Multi-channel Hidden Terminal Problem

Multi-channel hidden terminal problem was firstly discovered in [2] for nodes with single transceivers. If each node has only one transceiver, it can work either on the control channel or on a data channel, but not on both. This essentially causes the multi-channel hidden terminal problem. As shown in Fig. 1, when node and node b do handshaking on the control channel, node c and node d are communicating with each other on data channel 2. Thus,

node c and node d do not know the channel that is selected by node and node b (i.e. data channel 1). Later, when node c wants to send a data packet to node d, it initiates a handshaking process on the control channel. Since node d does not know that channel 1 has been used by others at this time, it may select the same channel and thus create a collision. Multi-channel hidden terminal problem can be solved by having one dedicated transceiver listening on the control channel continuously and thus, at least two transceivers are needed on every node.

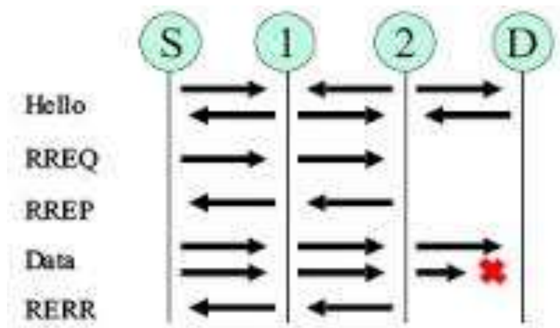

Fig 3:Adhoc On demand distance vector routing Messages

LONG-DELAY HIDDEN TERMINAL PROBLEM

The long propagation delays of the underwater acoustic channel introduce another kind of hidden terminal problem. As shown in Fig. 2, at the beginning, all nodes are listening to the control channel. Node a starts its handshaking process with node b on the control channel and then selects channel 1 for communication. Later, node c and node d also negotiate on the control channel for their data transmission. Let us assume the CTS message from node b arrives at node d after it selects its own data channel and sends its CTS message back to node c. In this case, node d does not know that the channel has already been used by node b. It may select channel 1 to communicate with node c and thus create a collision. We call this delay-related hidden terminal problem as “long-delay hidden terminal problem”. This problem is usually negligible in terrestrial radio networks due to the high propagation speed of radio signal. For long-delay underwater acoustic networks, however, this problem has to be well addressed.

7 Ad Hoc On demand Distance Vector Routing In AODV, the network is silent until a connection is needed. At that point the network node that needs a connection broadcasts a request for connection. Other AODV nodes forward this message (fig 3)(fig 4), and record the node that they heard it from, creating an explosion of temporary routes back to the needy node. When a node receives such a message and already has a route to the desired node, it sends a message backwards through a temporary route to the requesting node. The needy node then begins using the route that has the least number of hops through other nodes. Unused entries in the routing tables are recycled after a time. When a link fails, a routing error is passed back to a transmitting node, and the process repeats. Much of the complexity of the protocol is to lower the number of messages to conserve the capacity of the network. For example, each request for a route has a sequence number.

Nodes use this sequence number so that they do not repeat route requests that they have already passed on. Another such feature is that the route requests have a "time to live" number that limits how many times they can be retransmitted. Another such feature is that if a route request fails, another route request may not be sent until twice as much time has passed as the timeout of the previous route request. The AODV Routing Protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path. The major difference between AODV and Dynamic Source Routing (DSR) stems out from the fact that DSR uses source routing in which a data packet carries the complete path to be traversed. However, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission. In an on-demand routing protocol, the source node floods the Route Request packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single Route Request. The major difference between AODV and other on-demand routing protocols is that it uses a destination sequence number (DestSeqNum) to determine an up-to-date path to the destination. A node updates its path information only if the DestSeqNum of the current packet received is greater or equal than the last DestSeqNum stored at the node with smaller hopcount.The main advantage of this protocol is having routes established on demand and that destination sequence numbers are applied to find the latest route to the destination. The connection setup delay is lower.

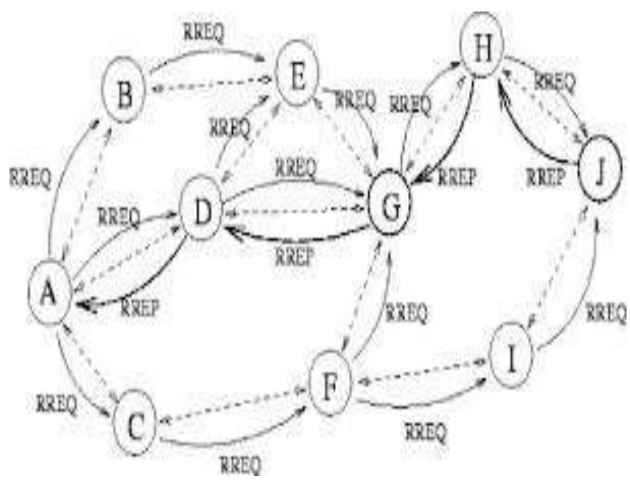

Fig 4 Request response and error messages in AdHoc On demand distance vector Routing

CONCLUSION.

In this paper, we provide theoretical and practical results that demonstrate the feasibility and advantages of a collision-resolution protocol for wireless broadcast. Existing model Chorus, which allows forwarders with the same outgoing packets to transmit roughly at the same time, and then employs physical-layer iterative decoding to resolve collisions at the receiver. By decoding multiple versions of a packet at once, Chorus achieves transmit diversity and improves loss resilience without any retransmission. More importantly, with its collision-tolerant MAC, Chorus significantly simplifies the CSMA scheduling and improves its spatial reuse. Even though the Chorus protocol is advantageous, Hidden Terminals cannot be reached. This is a Major Disadvantage when considering Chorus Protocol. For this we provide a solution in this paper. The solution is that, we break the mesh, but the nodes configuration remains the same as how it is present in Chorus protocol. So we can easily sense the hidden nodes through the Cognitive Sensing node (the node which receives the Information first from the Source Node).For this, here we are using the Algorithm AdHoc On demand Distance vector Routing. This Proposed Model maintains the efficiency, Low Latency and also improves Network Accessibility.

REFRENCES

[1]Xinyu Zhang and Kang G.Shin,"Delay-optimal Broadcast for Multihop Wireless Networks Using Self-Inteference Cancellation

[2]S.-H.Huang,P.-J.Wan,X.Jia,H.Du,and W.Shang,"Minimum latency Broadcast Scheduling in Wireless Adhoc Networks,"Proc IEEE INFOCOM 2007.

[3]R.Gandhi ,S.Parthasarathy,and A.Mishra,"Minimizing Broadcast latency and Redundancy in Adhoc Networks",Proc ACM MobiHoc,2003.

[4]Rajiv Gandhi,Yoo-Ah Kim,Seungjoon lee,Jiho Ryu,and Peng-Jun Wan,"Approximation Algorithm for databroadcast in Wireless Networks",Proc.ACM mobiCom,2008

[5]Wei Wang and Boon –Hee Soong,"Collision-Free and low-Latency Schedulling Algorithm for Broadcast Operation in Wireless Ad Hoc Networks",Proc ACM mobiHoc 2007.

[6]Scott C.-H Huang,Peng-Jun Wan,Xiaohua Jia,and Hongwei Du,"Low Latency Broadcast Scheduling in Ad Hoc Networks,Proc ACM MobiHoc 2006.

[7]Hrishikesh gossain,carlos cordeiro and Dharma P.Agarwal"Minimizing the Effect of Deafness and Hidden Terminal Problem in Wireless Ad Hoc Networks using Directional Antennas"