Lecture 11
Wireless Channel Access Protocols

This lecture introduces the problems and challenges in the design of wireless networks and discusses the area of channel access protocols (aka "medium access", or MAC protocols) in detail.

■ 11.1 Wireless Networks are Different

Communicating over radio is an old idea, and people have been doing it for a long time. Data networks have been around for about four decades. Combining the two is attractive because wireless networks offer important potential benefits over their wired counterparts: they enable mobile communications, they don't require as much investment in and upkeep of a wired infrastructure, and they enable broadcasts in a natural way.

For many years now people have been building data networks that operate over radio channels. One lesson that has been learned is that radios are different from wired links and affect the design of networks in important ways: radios make the design of data networks both interesting and challenging. This lecture and the three following it discuss these issues in detail.

Realizing the potential benefits of wireless networks in practical, working systems is difficult for one deceptively simple reason: *radios are not wires*. To understand the implications of this statement, consider resource sharing in a wired network.[1] Placing a "network layer" over the "link" and "physical" layers works well in wired networks because the network designer has to worry about sharing only at the network layer (and higher): links themselves (and therefore the link and physical layers) are shielded from one another and concurrent data transmissions on two or more wired links have no influence on each other unless those transmissions interfere at a switch (i.e., at the network layer).

In contrast, wireless transmissions are over a shared medium with very little shielding, especially when omni-directional antennas are used. The problem is that two radio transmitters that are near each other will interact adversely with each other, especially if their intended receivers can hear both transmissions. Such "link-layer" and "physical-layer"

[1] Sharing, as we have seen time and again in this course, is a fundamental property of all communication networks.

interactions don't usually occur with wired links (except for shared media such as Ethernet, but as we'll see in a bit, coping with the problem in Ethernet is much easier because packet collisions can be detected reliably in an Ethernet).

One approach to designing a robust radio-based wireless network is to make each communicating pair look like a robust point-to-point link. If that can be achieved, then the radio channel between a pair of nodes starts to look like a "link". The underlying principle here is to ignore all other transmissions and focus on just one pair, and attempt to achieve the Shannon capacity for that communicating pair:

$$C \leq B \cdot \log_2(1 + \frac{S}{N}), \tag{11.1}$$

where B is the bandwidth of the communication channel in Hz, S is the signal level at the intended receiver, and N is the background noise. The signal level, S, itself attenuates with distance, so if the transmitter sends data at a power level P, then S usually drops off with distance d as d^{-n}, where n is a small number (2 in free space, and perhaps 4 indoors). In the formula above, the log term is the maximum number of bits that can be sent; practical systems attempt to achieve that information-carrying capacity using various modulation and coding schemes.

The idea of making each pair of radio transmissions look like a "link" is a reasonable way to manage the complexity of the system, but does not eliminate the channel-sharing issues that must be solved. In particular, the goal of many wireless network designers is to maximize the aggregate data delivery capacity of the network, while allocating that capacity in a reasonably fair way (e.g., to prevent gross unfairness, avoid starvation, etc.). Achieving this goal is hard for the following reasons:

- While engineering a single communicating pair to come close to the Shannon capacity is possible, engineering S/N on a *network-wide* basis is very hard. In fact, it is quite hard to answer the question, "What is the maximum capacity of a wireless network (as opposed to "wireless link")?"

- Radio channels vary in quality with time and depend on location: various studies have shown that channel variations occur across multiple time scales, from a few bit-times to much longer. The result is that channel bit-error rates vary with time and space, and errors often occur in bursts. Coping with these variations to maximize capacity is not easy.

- A graph is not the best abstract model of a wireless network. Because radios are inherently broadcast media, packet reception is often probabilistic. The number of nodes that will successfully receive a packet is hard to predict, and determining a graph among the nodes to model likely collisions is also very hard. As a result, the traditional approach of solving the routing problem by finding paths in a graph (topology) is not the best approach to maximizing capacity.

In addition, designing good routing protocols, coping with mobility, achieving good TCP performance, and conserving energy on battery-operated devices are all problems that are non-trivial in wireless networks.

This course has four lectures on wireless networks. The rest of this lecture discusses wireless channel access protocols. The next lecture will focus on principles for designing high-capacity wireless networks. Then, we will discuss wireless routing protocols. Finally, we discuss how to take advantage of wireless broadcast and collaborating multiple radios to design better wireless protocols.

■ 11.2 Wireless MAC Protocols

Wireless channel access (aka MAC, for media access) protocols attempt to share the wireless channel amongst multiple contending transmitter-receiver pairs in the same neighborhood. The goal of these schemes is to maximize capacity within the "local neighborhood" by attempting to make every transmission count. Because concurrent transmissions in the same neighborhood might collide and cause packet corruption, the goal is to manage which nodes are allowed to send at the same time. These protocols don't take a network-wide view of the sharing problem, focusing instead on just "local" radio regions. Such protocols are also called *multiple access* protocols because they arbitrate transmissions amongst multiple concurrent users.

A popular MAC protocol design, CSMA, uses the *carrier sense* mechanism. Before transmitting a packet, the sender *listens* on the channel to determine if any other transmission is in progress. If it is, then the sender *defers*, waiting until the channel becomes idle. There is a rich literature in CSMA protocols: schemes differ based on how persistently they try when the channel is idle (*e.g.*, a node may send with probability p when the channel is idle), and in how nodes detect collisions. The latter problem is easy on a wired Ethernet because a sender can detect a collision reliably on an Ethernet.

In contrast, a radio transmitter cannot detect collisions reliably, because the receiver and sender don't "share fate" with respect to successful packet delivery. For example, the receiver may be near a source of noise or near other interfering transmitters that cause the packet to be corrupted because the signal strength, which attenuates with distance, is not high enough (cf. Eq. 11.1). The approach of just listening after a radio transmission to look for evidence of collision will not work without additional machinery (*e.g.*, explicit ACKs or NACKs from the intended receiver). Hence, CSMA protocols for wireless channels cannot use collision detection mechanisms as in Ethernet, but require other *collision avoidance* mechanisms to reduce the likelihood of repeated collisions.

Wireless MAC protocols must handle the following problems:

1. Use suitable collision avoidance schemes to reduce the number of wasted transmissions.

2. Provide reasonable fairness among contending nodes.

3. Cope with *hidden terminals*: a hidden terminal situation occurs when two nodes, A and B, cannot hear each other (when they each sense carrier when the other transmits, not much energy is detected), but a node C hears both of them. The problem is that both A and B may end up transmitting at the same time, and if one (or both) of those transmissions is destined for C, C may not be able to receive the packet

successfully.[2]

4. Take advantage of *exposed terminals*: Consider four nodes, *A*, *B*, *C*, and *D*, where *A* and *B* are near each other, *C* can receive packets from *A* but not *B* (even when *B* transmits at the same time as *A*), and *D* can receive packets from *B* but not *A* (even when *A* transmits at the same time as *B*). With carrier sense in place, *A* and *B* will detect each others' transmissions, and only one of them would transmit at a time, even though it might have been possible for both of them to concurrently send data to *C* and *D* respectively.

No MAC protocol successfully solves all these problems in all situations today. Most practical MAC protocols favor reducing collisions over maximizing every bit of available capacity. The rest of this lecture discusses the ideas behind three kinds of MAC protocols: CSMA with collision avoidance (CSMA/CA), reservation-based protocols, and time-division multiple access (TDMA) protocols.

■ 11.3 Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

The best current example of CSMA/CA is in the 802.11 (WiFi) family. The general idea behind such protocols is as follows:

Each node maintains a *contention window*, CW, and before transmitting data, picks a random "slot" in the range [0, CW]. Each slot is a fixed (small) amount of time, and all transmissions must start at slot boundaries. (Years ago, the Aloha system showed that a slotted distributed multiple access system has higher capacity than an unslotted one, because it forced all collisions to occur at slot boundaries rather than spread them out so they can happen anywhere through the entire packet transmission).

In 802.11, when a node has data to transmit, it picks a random number in [0, CW]. The node *listens* by sensing the carrier. On each idle slot, the node decrements a countdown timer by 1. When that timer reaches 0, and the channel is sensed as "idle", the node sends data.

During the countdown process, if the node senses the carrier as "busy" (usually done by comparing the energy with the "background noise" level gathered when the channel was believed to be "idle"), then the node *defers* transmission. *At this time, it "holds" its countdown timer, until the carrier is sensed as "idle" once again.* We will return to discussing this "holding" step in a bit.

But how does a node know when there are a large number of collisions occuring? One approach is for the protocol to include a *link-layer* ACK on each frame. This is the approach used in 802.11 for unicast transmissions. The absence of an ACK, which is sent "synchronously" and with a very short time delay from the receiver, signals a packet loss, and the sender attributes that loss to a collision. In response, the sender backs-off exponentially by doubling CW.

[2] In some cases, when both nodes are sending to *C*, one of them may be able to "capture" the channel and have its data received successfully. We don't worry about this capture effect further in this lecture, although it is important in some networks in practice.

A different approach, used in some wireless LANs before the 802.11 standard emerged, is not to use link-layer ACKs, but for a sender to infer the likelihood of a collision if it finds the channel "busy" each time it wants to transmit. In response, the sender backs-off its CW exponentially. The benefit of this approach is that it does not require ACKs, but the problem is that it turns out to be quite unfair. Nodes that successfully transmit data end up picking small CW values, while nodes that detect a busy channel back-off. 802.11 avoids this problem by *holding* the countdown timer when a sending node detects a busy channel, and by using a reactive, rather than proactive, method to back-off its CW. Thus, while the 802.11 approach may end up with more lost data, it has better fairness properties. Because the absence of a link-layer ACK causes the sender to retransmit the corresponding data at the link layer, these losses can often be shielded from higher layers.

As discussed thus far, CSMA/CA does not handle hidden or exposed terminals satisfactorily. In practice, however, carrier sense mechanisms are based on a set of heuristics that often cause a "busy" carrier to be sensed even when the other transmitting node's (or nodes') transmission is not properly decodable. That is, the "carrier sense range" is often larger than the "largest reception range" (we use these terms in quotes because they are not fixed quantities, varying in both time and space). It should be easy to see that in a very ideal (and impractical) world of circular or spherical "ranges", a "carrier sense range" that is two or more times bigger than the "reception range" will not have any hidden terminals.

■ 11.4 Reservation-Based Protocols

A different approach to orchestrating wireless data transmissions is to use explicit reservations, as described in the MACAW paper [3]. MACAW is based on the previously proposed MACA scheme [12], which advocated that each transmission be preceded by a handshake between the sender and receiver to "reserve" the channel for a period of time. In one instantiation of such a protocol, before sending data, a node sends an RTS ("request to send") message, as long as it has not heard any CTS message in the recent past. If the intended receiver hears this message and has not heard any other RTS, then it responds with a CTS. The original RTS message includes the amount of time that the sender wishes to reserve the channel, as does the CTS (subtracting out the RTS time). A node that does not get a response to its RTS sets an exponential backoff timer, and retries the RTS after a random period of time chosen from the backoff interval.

This "RTS-CTS-Data" approach reduces the number of collisions because (1) a sender can send only if it has not heard any recent CTS (so it's own transmission can't interfere with another node's reception), and (2) a sender can send only if the receiver has not heard any RTS (so there's no other sender in the receiver's vicinity). Of course, in practice, RF data reception may not be symmetric, and other factors could conspire to corrupt packet delivery, but the approach does have merit. Moreover, it does not require any carrier sensing support.

When link-layer ACKs are used, the protocol as described thus far does not suffice, because ACKs are as important as the data packets themselves (the absence of an ACK causes a retransmission, so if packets are big enough, lost ACKs are quite wasteful). The solution to this problem is to enhance the protocol by having the sender not send an RTS *either* when it has heard another CTS *or* a RTS in the recent past, and similarly, for a receiver to not send a CTS if it has heard either an RTS or a CTS recently. Thus, all communication channels between pairs of nodes are treated as if they are bidirectional, which makes sense because we want to avoid collisions of both data and ACKs.

It is easy to see that this protocol handles hidden terminals, and because it defines all communication to be bidirectional, exposed terminal-based transmission opportunities are effectively eliminated from consideration.

The 802.11 committee standardized both CSMA/CA and RTS/CTS for 802.11 devices. In practice, both are supported, but network operators typically turn of RTS/CTS. I believe there are three reasons why:

1. RTS/CTS has very high overhead, particularly for small packets and in *rate-diverse* networks. 802.11 devices support a variety of modulations and associated transmission rates, adaptively picking a suitable scheme. RTS/CTS has to be sent at the lowest supported rate, because the goal is to avoid collisions regardless of transmission rate (because RTS/CTS bits must be decoded, a low transmission rate implies a higher probability that all nodes that are likely collide will hear the RTS or CTS). Because the data transfer itself may happen at a much higher rate than the RTS/CTS exchange, the overhead is prohibitively expensive.

2. Most current deployments of 802.11 are based on a cellular infrastructure, and are not *ad hoc*. Neighboring cells are usually configured to operate on different channels (frequencies) by explicit provisioning, so hidden terminal problems on the downlink (to the wireless LAN clients) are rare. On the uplink, hidden terminals could occur, but often these packets are small (e.g., TCP ACKs) and the RTS/CTS overhead is then significant.

3. In practice, many commercial WiFi cards can sense carrier as “busy” even when they can’t decode the bits, reducing the need for explicit reservations.

■ 11.5 Time-Division Multiple Access (TDMA)

An entirely different approach to sharing wireless channels is to allocate access by time. This approach is used in some cellular telephone networks, where the base station determines a transmission time-schedule for clients. A form of TDMA is used in Bluetooth, where nodes form a subnetwork with a master and one or more slaves. The master divides time into odd and even slots, with the convention that in each odd slot, the master gets to send data to some slave device. The even slot that follows an odd one is reserved for the device that received data in the previous slot from the master. This approach is called “time division duplex” (TDD).

In general, at high loads, time division makes sense; otherwise, slots are wasted. Avoiding this waste in TDMA usually makes protocols more complex. At the same time, CSMA-based approaches don’t perform too well when there is heavy, persistent load from a large number of nodes. Much work has been done in the community on MAC protocols, including on hybrid CSMA/TDMA protocols.

■ 11.6 Summary

This lecture introduced the area of wireless channel access protocols and discussed the principles underlying their design, with examples. The next lecture will focus on the issues facing the designer of a high-capacity wireless network.

References

[1] T. Bates, R. Chandra, and E. Chen. *BGP Route Reflection - An Alternative to Full Mesh IBGP*. Internet Engineering Task Force, Apr. 2000. RFC 2796. (Cited on page 51.)

[2] I. V. Beijnum. *BGP*. O'Reilly and Associates, Sept. 2002. (Cited on page 48.)

[3] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A Media-Access Protocol for Packet Radio. In *Proc. ACM SIGCOMM*, London, England, Aug. 1994. (Cited on page 5.)

[4] D. Clark and D. Tennenhouse. Architectural Consideration for a New Generation of Protocols. In *Proc. ACM SIGCOMM*, pages 200–208, Philadelphia, PA, Sept. 1990. (Cited on page 31.)

[5] R. Dube. A Comparison of Scaling Techniques for BGP. *ACM Computer Communications Review*, 29(3):44–46, July 1999. (Cited on page 49.)

[6] Cisco IOS IP Command Reference, ebgp-multihop. `http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_command_reference_chapter09186a00800ca79d.html`, 2005. (Cited on page 52.)

[7] K. Fall and S. Floyd. Simulation-based Comparisons of Tahoe, Reno, and Sack TCP. *ACM Computer Communications Review*, 26(3):5–21, July 1996. (Cited on page 37.)

[8] N. Feamster and H. Balakrishnan. Detecting BGP Configuration Faults with Static Analysis. In *Proc. 2nd Symposium on Networked Systems Design and Implementation (NSDI)*, pages 43–56, Boston, MA, May 2005. (Cited on page 52.)

[9] T. Griffin and G. Wilfong. On the Correctness of IBGP Configuration. In *Proc. ACM SIGCOMM*, pages 17–29, Pittsburgh, PA, Aug. 2002. (Cited on pages 49 and 52.)

[10] C. Hedrick. *Routing Information Protocol*. Internet Engineering Task Force, June 1988. RFC 1058. (Cited on page 42.)

[11] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM*, pages 314–329, Stanford, CA, Aug. 1988. (Cited on pages 33 and 61.)

[12] P. Karn. MACA – A New Channel Access Method for Packet Radio. In *Proc. 9th ARRL Computer Networking Conference*, 1990. (Cited on page 5.)

[13] P. Karn and C. Partridge. Improving Round-Trip Time Estimates in Reliable Transport Protocols. *ACM Transactions on Computer Systems*, 9(4):364–373, Nov. 1991. (Cited on pages 33 and 35.)

[14] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. *TCP Selective Acknowledgment Options*. Internet Engineering Task Force, 1996. RFC 2018. (Cited on page 34.)

[15] J. Moy. *OSPF Version 2*, Mar. 1994. RFC 1583. (Cited on page 42.)

[16] D. Oran. *OSI IS-IS intra-domain routing protocol*. Internet Engineering Task Force, Feb. 1990. RFC 1142. (Cited on page 42.)

[17] Y. Rekhter and T. Li. *A Border Gateway Protocol 4 (BGP-4)*. Internet Engineering Task Force, Mar. 1995. RFC 1771. (Cited on pages 42 and 48.)

[18] Y. Rekhter, T. Li, and S. Hares. *A Border Gateway Protocol 4 (BGP-4)*. Internet Engineering Task Force, Oct. 2004. `http://www.ietf.org/internet-drafts/draft-ietf-idr-bgp4-26.txt` Work in progress, expired April 2005. (Cited on page 42.)

[19] P. Traina, D. McPherson, and J. Scudder. *Autonomous System Confederations for BGP*. Internet Engineering Task Force, Feb. 2001. RFC 3065. (Cited on page 51.)