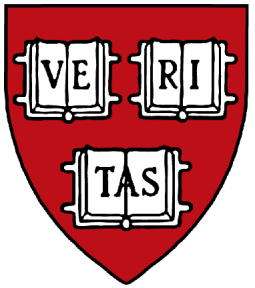
CS263: Wireless Communications and Sensor Networks

Matt Welsh



Lecture 5: The 802.11 Standard
October 4, 2005

Today's Lecture

All about 802.11

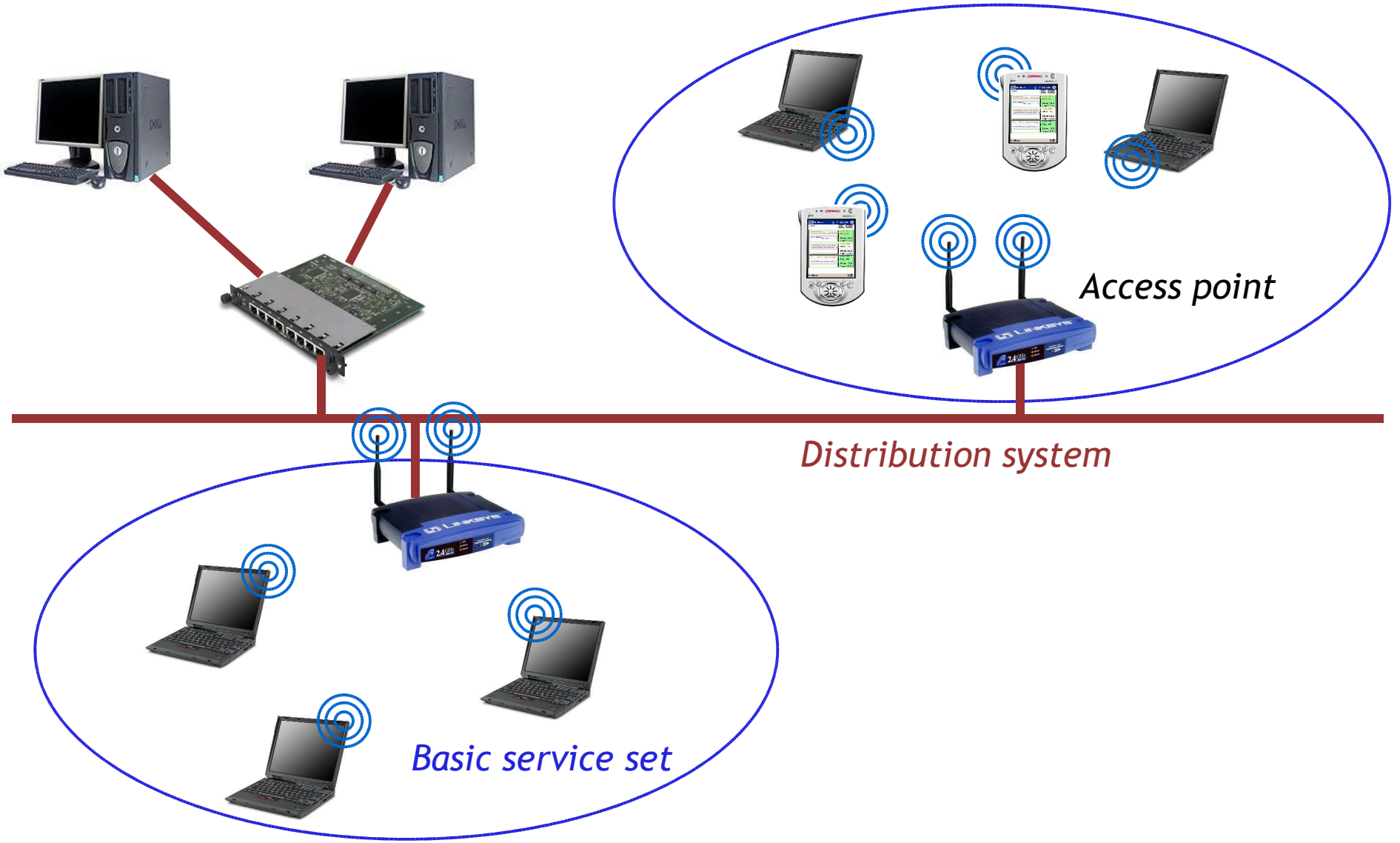
CSMA/CD MAC and DCF

WEP and 802.1x Security

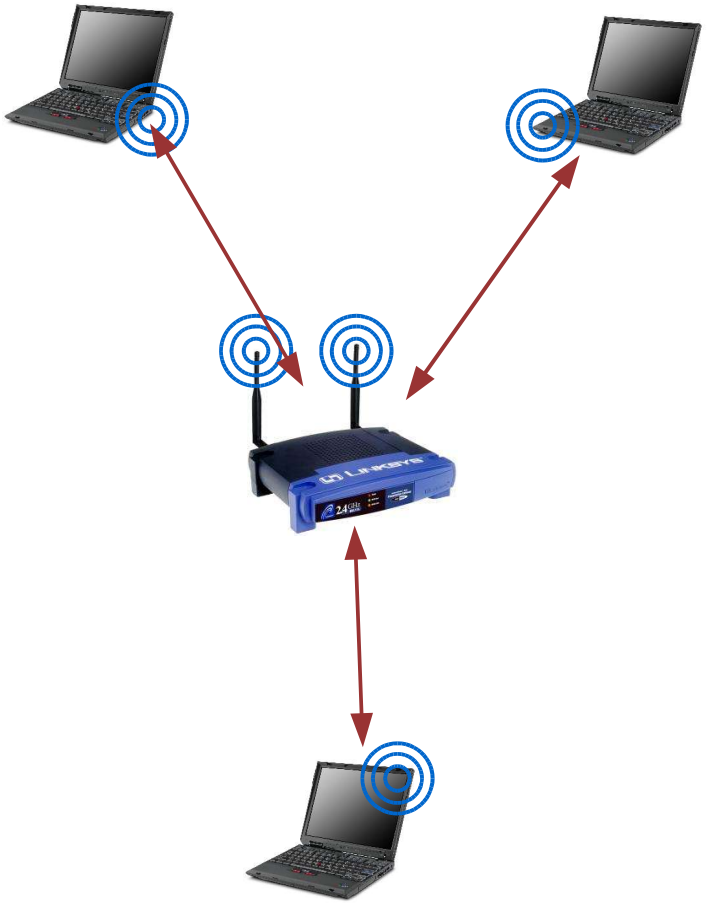
802.11 / WiFi

IEEE working group 802.11 formed in 1990

- Now the most popular and pervasive Wireless LAN standard



Infrastructure vs. Independent Mode

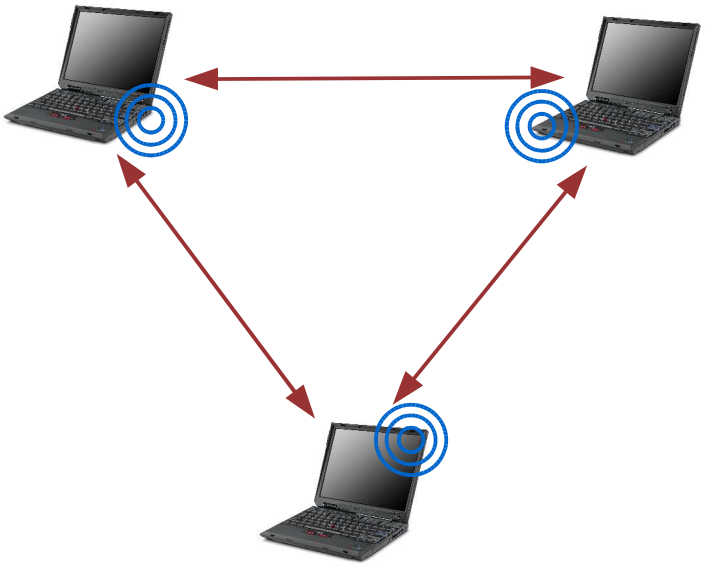


Infrastructure mode:

All communications must be relayed by access point

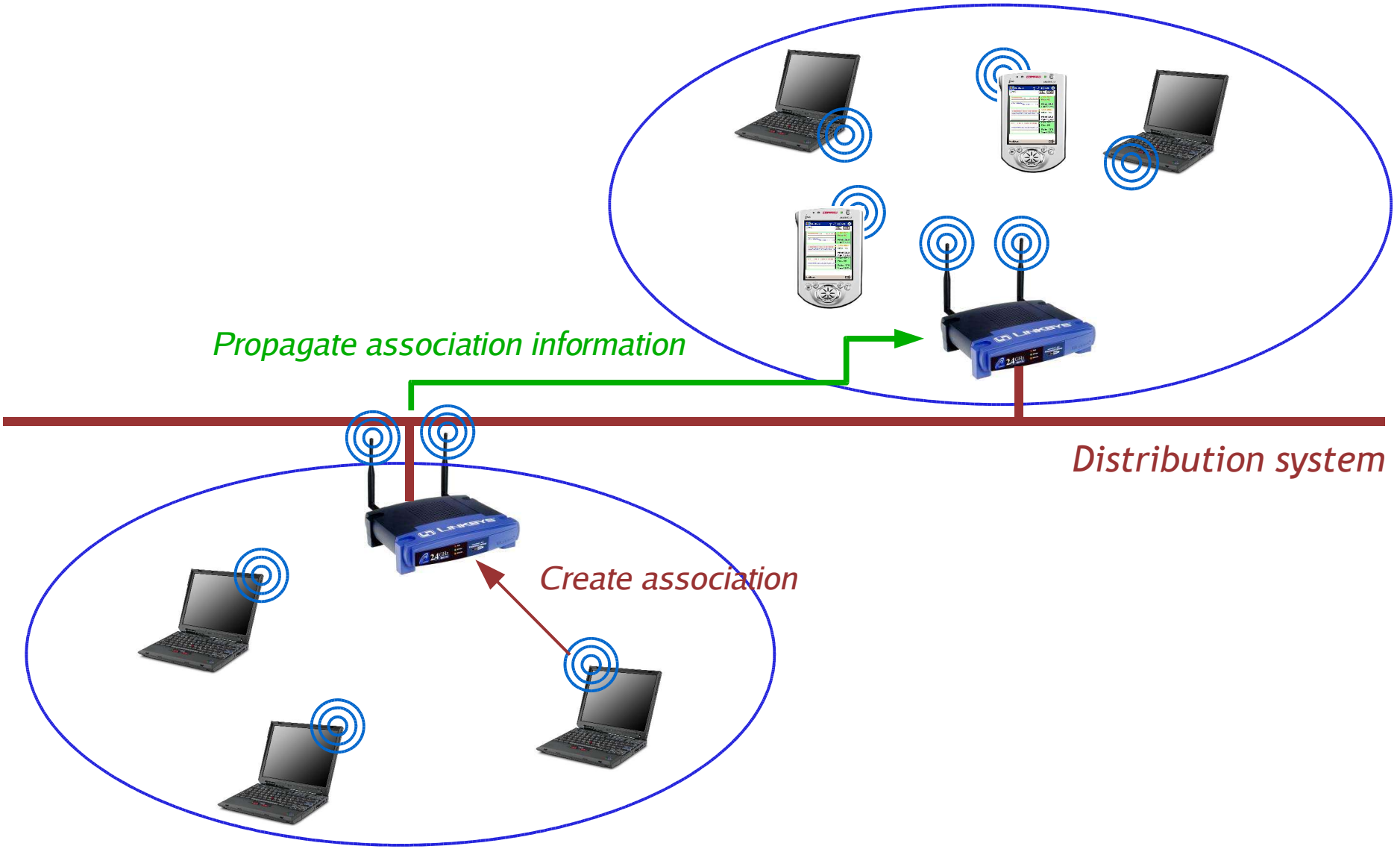
Independent mode:

Nodes communicate directly with each other



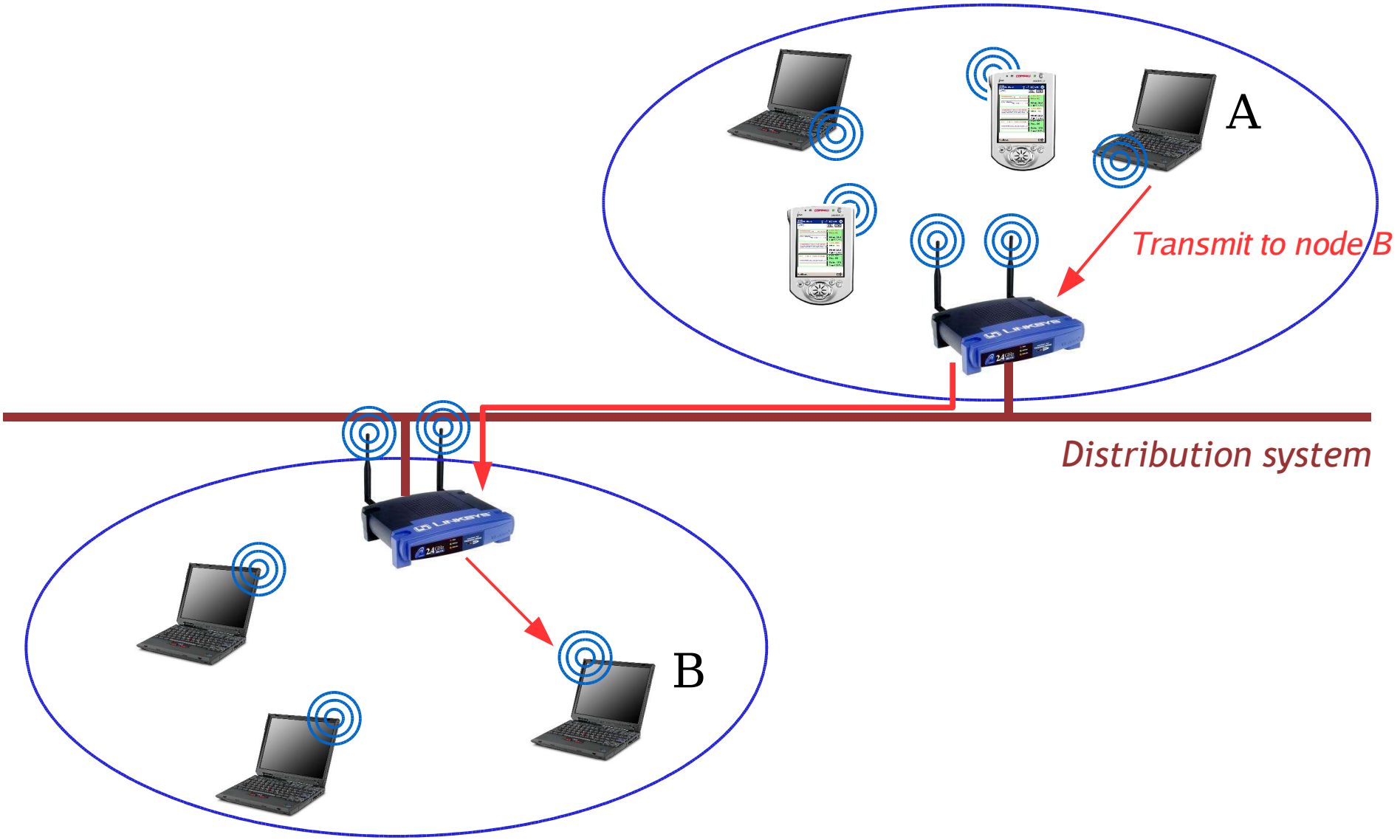
Extended Service Set Model

Entire ESS looks like a single virtual LAN!



Extended Service Set Model

Entire ESS looks like a single virtual LAN!



CSMA/CD

A better approach to contention-based access: *Carrier Sensing*

- Each station listens on channel before transmitting
- If channel is busy, waits before transmission

What happens as soon as the channel is clear?

- Transmit immediately?
  - **Pros and cons?**
- Wait for some random period of time?
  - **Pros and cons?**

CSMA/CD

A better approach to contention-based access: *Carrier Sensing*

- Each station listens on channel before transmitting
- If channel is busy, waits before transmission

What happens as soon as the channel is clear?

- Transmit immediately?
  - *Good for minimizing delays....*
- Wait for some random period of time?
  - *Avoids collisions from multiple stations detecting clear channel at the same time*

How do you determine whether the channel is clear?

How do you determine if a collision has occurred?

CSMA/CD

A better approach to contention-based access: *Carrier Sensing*

- Each station listens on channel before transmitting
- If channel is busy, waits before transmission

What happens as soon as the channel is clear?

- Transmit immediately?
  - *Good for minimizing delays....*
- Wait for some random period of time?
  - *Avoids collisions from multiple stations detecting clear channel at the same time*

How do you determine whether the channel is clear?

- Must estimate *noise floor* or actively decode incoming data

How do you determine if a collision has occurred?

- Transmitter listens for another packet immediately after its own transmission
- Or, wait to receive an ACK from the receiver, which implies no collision

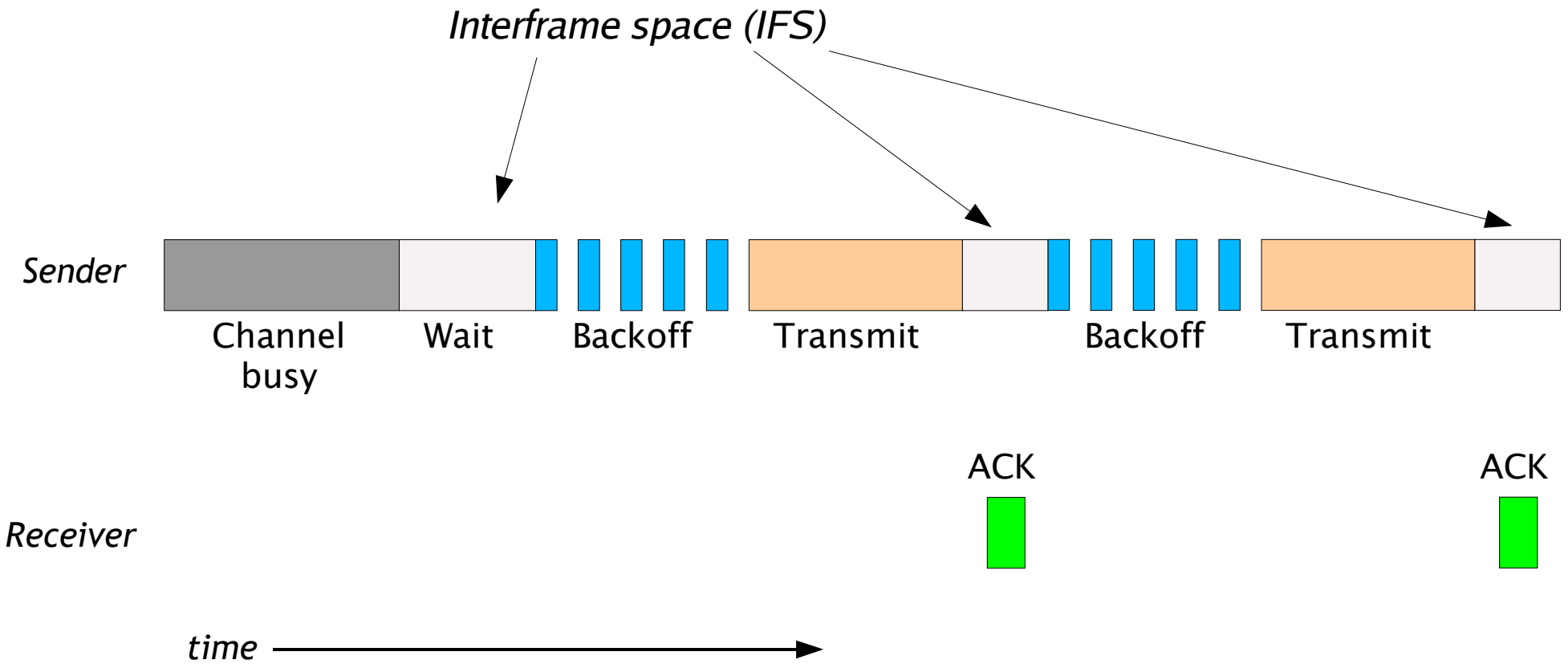
Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

Distributed Coordination Function

802.11 uses a variant of CSMA

- Called the *Distributed Coordination Function (DCF)*
- Access point controls when nodes can transmit.
- No collision detection – rather, *collision avoidance (CSMA/**CA**)*

DCF Illustrated



Exponential Backoff

ACK-based scheme for reliability

- Receiver sends ACK after each successful transmission
- Sender will retransmit if no ACK is heard, *after waiting for a **random** interval*

Binary exponential backoff

- First backoff interval between [0 ... 31] time slots
- If collision occurs, new backoff interval chosen between [0 ... 63] slots
- Repeat until backoff interval reaches [0...1023] slots.

Why increase the backoff interval each time???

SIFS and DIFS

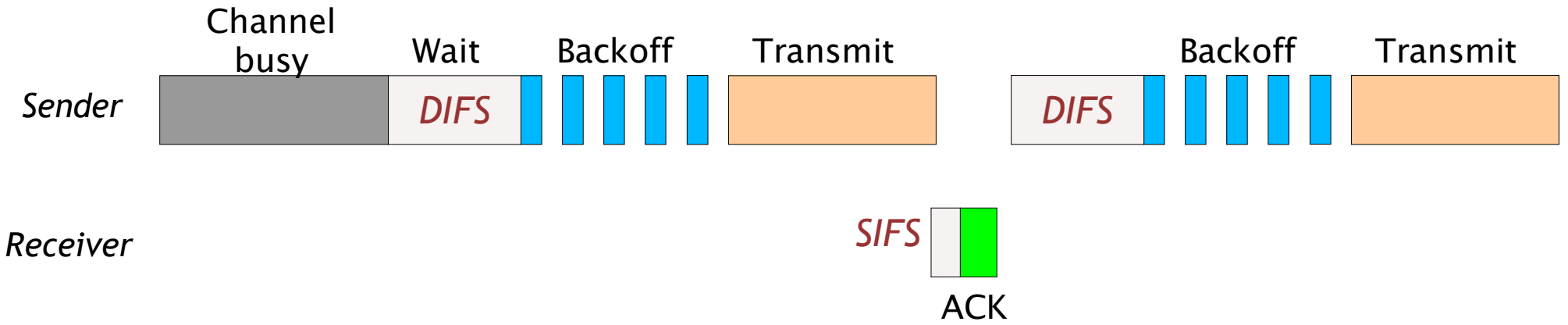
802.11 provides four different interframe spacing times

- Provide different traffic "priorities"

Standard IFS time is the "Distributed IFS" (DIFS)

"Short IFS" (SIFS) used for higher priority frames

- e.g., ACK packets from AP back to a node
  - *Allows ACKs to "sneak in" before contention period begins*



Fragmentation

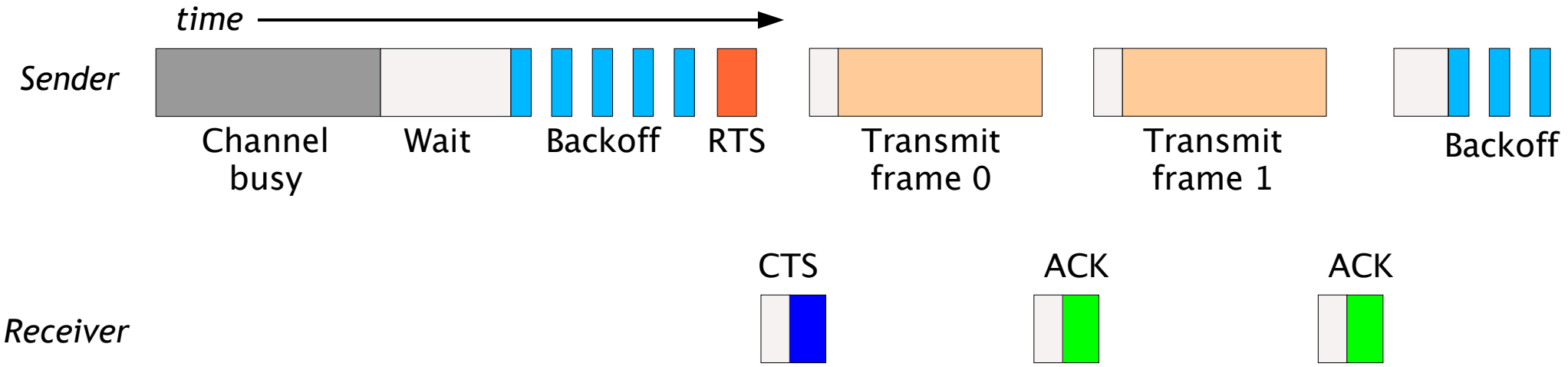
This is all fine for messages that can fit into a single transmission.

What should we do if we have longer messages to send?

Can clearly fragment long message into multiple packets.

- What is the possible problem with this?

Fragmentation and Reassembly



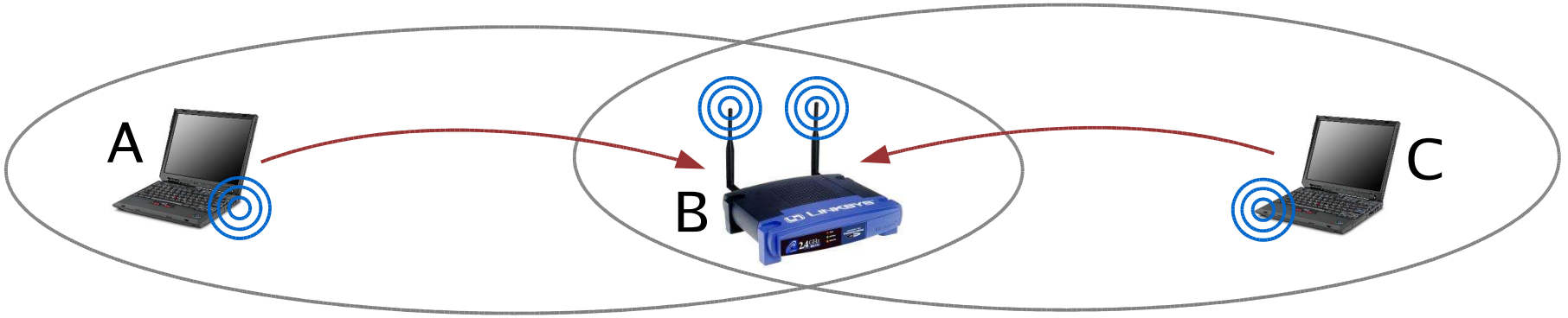
Long messages broken into multiple *frames*

- Node can transmit next frame in a sequence **immediately** after receiving ACK
- But, must do backoff before sending next **message**

Transmitter "reserves the channel" using request to send (*RTS*)

- Receiver transmits clear to send (*CTS*) to initiate transmission of long message

Hidden Terminal Problem



Node C is not aware of Node A's transmissions!

- Collisions can occur at Node B

Solution: Network Allocation Vector (NAV)

- Each message includes length of time other nodes must wait to send
- Node B's CTS to Node A can be heard by Node C
  - *CTS will prevent Node C from transmitting before Node A is done*

802.11 Standards

Standard	*Frequency*	*Data rate*	*Range*
802.11b	2.4 Ghz, DSSS	11 Mbps	~300 feet, ~100' indoors
Widely deployed and inexpensive			
802.11g	2.4 Ghz, O-FDM	54 Mbps	< 802.11b
Backwards compatible with 802.11b			
802.11a	5 Ghz, O-FDM	54 Mbps	~80 feet
Uses UNII band, products emerging now			

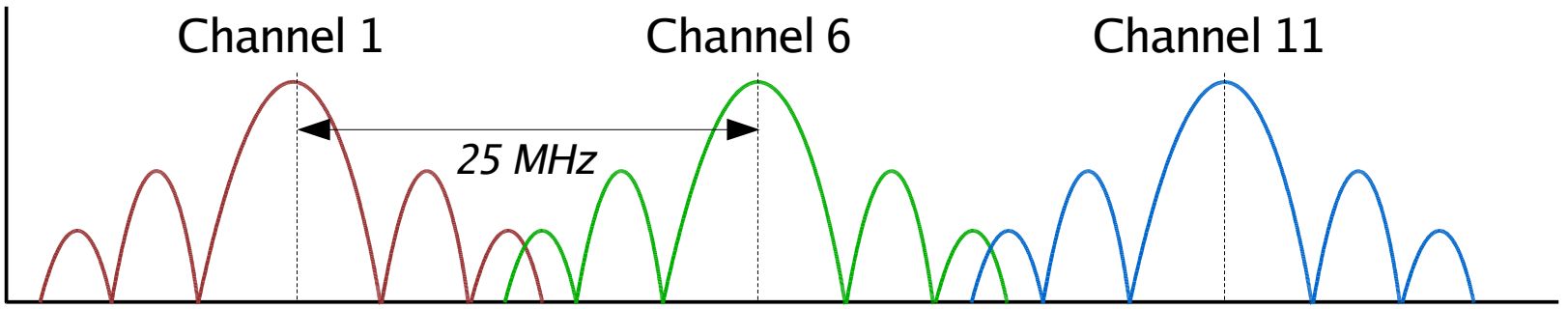
802.11b PHY

Original 802.11 standard used Frequency Hopping, G-FSK

- Divide 2.4 Ghz band into 78 channels, 1 MHz wide
- Dwell time of 390 ms per channel
- 26 different, fixed (globally known) hop sequences

802.11b standardized on DSSS with Q-PSK modulation

- 8-bit Complementary Code Keying (previous lecture)
- Band divided into 14 channels, 5 MHz wide each
  - *However, DSSS energy spread over a 22 MHz band!!!*



- This means that not all channels can be used simultaneously.

802.11 Security: WEP

Wireless networks are inherently a broadcast medium!

- It is easy to intercept transmissions between end hosts
- Compare to wired systems: Must physically tap into the wires
  - *Nightmare for companies: Hacker in the parking lot with a laptop*

Wired Equivalent Privacy (WEP)

- Rather than provide 802.11 with a truly robust security solution, goal was to prevent “casual” snooping
- Problem: WEP was developed from scratch by a closed committee, standard not readily accessible for review by researchers

WEP relies on a secret key being shared by end hosts and APs

- Traffic between nodes is encrypted using this key
- Requires key to be distributed in some fashion by system admins
  - *Makes it very difficult to change the key later!*

WEP Weaknesses

In 2001, researchers at UC Berkeley demonstrated that WEP was vulnerable to a range of attacks

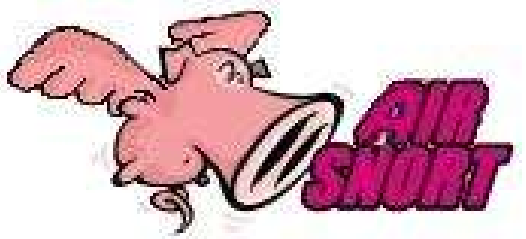
- 40-bit encryption keys are susceptible to brute force attacks
- WEP reuses portions of the random “keystring” making analysis possible
- Attackers can modify contents of frames without necessarily decrypting them

Not long afterwards, WEP cracking software was demonstrated

- Adam Stubblefield, Rice undergrad doing internship at AT&T, wrote the code in less than a week on a Linux laptop
- Open source AirSnort software now widely available
  - *Can recover a WEP key after intercepting 5-10 million packets*

Bottom line: Don't depend on WEP!

- “WEP is so flawed that it is not worth using in many cases.” -- Matthew S. Gast, *802.11 Wireless Networks: The Definitive Guide*



What to do?

Industry is working on solutions based on new 802.1x standard

- This is not without its problems, however

Better solution: End-to-end security

- Don't depend on underlying network infrastructure to ensure security
- Rather, perform authentication and encryption at the application level

Common solution: SSL/TLS protocol

- Same protocol used by Web browsers to talk to secure Web servers
- Provides a range of authentication and encryption options
- No assumptions about security of the underlying network

Next Lecture

Bluetooth and 802.15.4

Reading: Stallings Chapter 15

- (No required reading on 802.15.4)