# CSMA/CA

- IEEE 802.11 standard for WLAN defines a distributed coordination function (DCF) for sharing access to the medium based on the CSMA/CA protocol
- Collision detection is not used since a node is unable to detect the channel and transmit data simultaneously
- A node listens to the channel before transmission to determine whether some one else is transmitting
- The receiving node sends an acknowledge packet (ACK) a short time interval after receiving the packet
- If an ACK is not received, the packet is considered lost and a retransmission is arranged

# DCF basic access

- DCF consists of a basic access mode as well as an optional RTS/CTS access mode
- In basic access mode the node senses the channel to determine whether another node is transmitting before initiating a transmission
- If the medium is sensed to be free for a DCF inter-frame space (DIFS) time interval the transmission will proceed
- If the medium is busy the node defers its transmission until the end of the current transmission and then it will wait an additional DIFS interval and generate a random backoff delay uniformly chosen in the range $[0, W-1]$ where $W$ is called the backoff window or contention window (CW)

# DCF basic access

- The backoff timer is decreased as long as the medium is sensed to be idle for a DIFS, and frozen when a transmission is detected on the medium, and resumed when the channel is detected as idle again for a DIFS interval
- When the backoff reaches 0, the station transmits it packet
- For IEEE 802.11 time is slotted in a basic time unit which is the time needed to detect the transmission of a packet from any other station
- The initial CW is set to $W = 1$, if two or more nodes decrease their backoff timer to 0 at the same time a collision occur, at this situation the CW is doubled for each retransmission until it reaches a maximum value

# DCF basic access

- A short inter-frame space (SIFS) is used to give priority access to ACK packets
- When receiving a packet correctly, the destination node waits for a SIFS interval immediately after the reception has completed and transmits an ACK back to the source node confirming the correct reception
- If the source node does not receive an ACK due to collision or transmission errors it reactivates the backoff algorithm after the channel remains idle for an extended IFS (EIFS) interval

# WLAN carrier sensing

- Carrier sensing is done in two ways, physical carrier sensing by detecting activity on the radio interface, and virtual carrier sensing which is performed by the DCF RTS/CTS access mode
- To implement virtual carrier sensing each node sends duration information in the header of request-to-send (RTS) and clear-to-send (CTS) packets
- The duration information indicates the amount of time the medium is to be reserved for transmitting the data and returning ACK packets after the end of current frame

# WLAN carrier sensing

- The stations in the same basic service set (BSS) uses this information to update its network allocation vector (NAV) that represents the amount of time it has to defer in accessing the medium
- By using virtual carrier sending all nodes within the same BSS learn how long the channel will be used for this data transmission
- This solves the problem of a “hidden node”, a third node that may not be able to receive the RTS from the sending node will hear the CTS from the receiving node and the channel will be reserved for the transmission

# DCF RTS/CTS access mode

- In RTS/CTS access mode, prior to the data transmission the sending node will send a RTS packet to announce the upcoming transmission
- When the destination node receives the RTS it will send a CTS packet after a SIFS interval
- Both the RTS and CTS packets are short control packets
- The sending node is allowed to transmit its data packet only if it receives the CTS packet correctly
- The purpose of this RTS/CTS exchange is to avoid long collisions since we don’t have collision detection

# Exponential backoff scheme

- Whenever a backoff occurs the backoff time is uniformly chosen in the range $[0, W-1]$
- After each unsuccessful transmission the backoff windows size is doubled, up to a maximum value
- Once the backoff window size reaches its maximum value it will stay at that value until it is reset
- The value of $W$ will be reset after every successful transmission of a data or RTS packet, or when a retry counter reaches its limit
- Since $W$ is used to control the backoff counter its value will affect the performance of the DCF protocol, improvements can be made by choosing better update rules than in IEEE 802.11 standard

# WLAN operating modes

- The IEEE 802.11 standard specifies two operating modes, either ad hoc mode (peer-to-peer) or infrastructure mode (peer-to-AP)
- The former is used when connecting a number of wireless nodes, e.g. for a temporary network at a meeting or for connecting a few wireless appliances
- The latter have one special node, the Access point (AP) which is an Ethernet switch and provides access to a subnet with wire access
- In infrastructure mode all communication between nodes or between a node and the wired network go through the AP

# WLAN operating modes

- A basic service set (BSS) are a collection of nodes that have recognized each other and established communication
- It is also possible to have an extended service set (ESS) consisting of a series of overlapping BSSs (each containing an AP) and the APs are connected together by means of a Distribution System (DS) which in practice is an Ethernet LAN but could be any type of network
- All nodes within a BSS are synchronized by periodic transmission of time stamped beacon packets, in infrastructure mode the AP is the timing master and generates all timing beacons

# PCF access method

- In addition to the mandatory DCF access method there is an optional extension called the point coordination function (PCF)
- In PCF the nodes in a BSS is polled by the AP, providing a time division multiplexing mode for delay sensitive connection-oriented services

# Maximum throughput 802.11

- Assuming one peer communicating with an AP using TCP protocol (e.g. transferring file with ftp, fetching web-page with http).
- Assume that each TCP data packet is followed by a TCP ACK packet.
- To transfer the data segment there will be
  - Silence during at least one DIFS slot, signaling that the medium is available. More than one if the node is executing back-off.
  - The data frame containing the TCP data.
  - A SIFS gap between data frame and 802.11 ACK frame.
  - The 802.11 ACK frame.

# Maximum throughput 802.11

- To transfer the TCP ACK packet there will be
  - Silence during at least one DIFS slot, signaling that the medium is available. More than one if the node is executing back-off.
  - The data frame containing the TCP ACK.
  - A SIFS gap between data frame and 802.11 ACK frame.
  - 802.11 ACK frame.
- In addition to the payload data, the data frame has additional 36 bytes of data, out of those 28 bytes are 802.11 MAC header for various control and management, error detection and addressing. The other 8 bytes are a header to identify network layer protocol.

# Maximum throughput 802.11

- To transfer 1460 bytes of payload, we have an packet with 1500 bytes of data including the TCP/IP headers, and 1536 bytes including the MAC and SNAP header.
- For the TCP ACK packet of 40 bytes TCP/IP header, there is also an additional 36 bytes for MAC and SNAP header, so total 76 bytes for the TCP ACK.
- We neglect media contention and backoff times and retransmissions

# Maximum throughput 802.11b

- SIFS = $10\mu s$, Slot time = $20\mu s$, DIFS = 2 Slot time + SIFS = $50\mu s$
- Every data frame has a preamble of $192\mu s$
- 802.11 ACK packet is 14 bytes
- Data rate is 1.375Msymbols/s, where each symbol is 8 bits, i.e. 11Mbit/s.
- TCP data packet is thus = DIFS + 802.11 data + SIFS + 802.11 ACK =
  $50\mu s + 192\mu s + 1536/1.375 + 10\mu s + 192\mu s + 14/1.375 =$
  $50\mu s + 1310\mu s + 10\mu s + 203\mu s = 1573\mu s$
- TCP ACK packet is = DIFS + data + SIFS + 802.11 ACK = $50\mu s + 192\mu s + 76/1.375 + 10\mu s + 203\mu s =$
  $50\mu s + 248\mu s + 10\mu s + 203\mu s = 511\mu s$

# Maximum throughput 802.11b

- In total we have $2084\mu s$ to transmit 1460 bytes of payload, which gives us a throughput of $5.6$Mbit/s.

# Maximum throughput 802.11g

- SIFS = $10\mu s$, Short slot time (802.11g-only) = $9\mu s$, Long slot time = $20\mu s$, DIFS = 2 Slot time + SIFS
- Every data frame has a preamble of $20\mu s$ and a signal extension of $6\mu s$ at the end of each packet.
- 802.11 ACK packet is 14 bytes
- Data rate is 0.25Msymbols/s, where each symbol is 216 bits, i.e. 54Mbit/s, each symbol takes $4\mu s$.
- TCP data packet is 1536 bytes = 12288 bits so we need 57 symbols to transmit it, the 802.11 ACK of 14 bytes = 112 bits needs one symbol, TCP ACK of 76 bytes = 608 bits needs 3 symbols.

# Maximum throughput 802.11g-only

- If we only have 802.11g-nodes we can use fast slot time.
- TCP data is then DIFS + 802.11 data + SIFS + 802.11 ACK =
  $28\mu s + 20\mu s + 57 \cdot 4\mu s + 6\mu s + 10\mu s + 20\mu s + 1 \cdot 4\mu s + 6\mu s =$
  $28\mu s + 254\mu s + 10\mu s + 30\mu s = 322\mu s$
- TCP ACK packet is = DIFS + data + SIFS + 802.11 ACK = $28\mu s + 20\mu s + 3 \cdot 4\mu s + 6\mu s + 10\mu s + 30\mu s =$
  $28\mu s + 38\mu s + 10\mu s + 30\mu s = 106\mu s$
- In total we have $428\mu s$ to transmit 1460 bytes of payload, which gives us a throughput of $27$Mbit/s.

# Maximum throughput 802.11g CTS-to-self

- The fast data frame will be protected by an 802.11b-compatible CTS consisting of preamble of $192\mu s$ and 14 byte of data transmitted at 11Mbit/s, i.e. same as 802.11b ACK packet, $203\mu s$.
- Also fast slots are no longer allowed so DIFS = $50\mu s$
- TCP data is then DIFS + CTS + SIFS + 802.11 data + SIFS + 802.11g ACK =
  $50\mu s + 203\mu s + 10\mu s + 254\mu s + 10\mu s + 30\mu s = 557\mu s$
- TCP ACK packet is = DIFS + CTS + SIFS + data + SIFS + 802.11 ACK =
  $50\mu s + 203\mu s + 10\mu s + 38\mu s + 10\mu s + 30\mu s = 341\mu s$
- In total we have $898\mu s$ to transmit 1460 bytes of payload, which gives us a throughput of $13$Mbit/s.

# Maximum throughput 802.11g RTS-CTS

- To guard against the hidden node problem we need a full RTS-CTS handshake. The RTS packet is slightly larger than the CTS since it contains 20 bytes of data, and takes $207\mu s$.
- TCP data is now DIFS + RTS + SIFS + CTS + SIFS + 802.11 data + SIFS + 802.11g ACK = $50\mu s207\mu s + 10\mu s + 203\mu s + 10\mu s + 254\mu s + 10\mu s + 30\mu s = 774\mu s$

# Maximum throughput 802.11g RTS-CTS

- TCP ACK packet is = DIFS + RTS + SIFS + CTS + SIFS + data + SIFS + 802.11 ACK = $50\mu s + 207\mu s + 10\mu s + 203\mu s + 10\mu s + 38\mu s + 10\mu s + 30\mu s = 558\mu s$
- This is however longer time than it takes to send TCP ACK with 802.11b so we can use that instead resulting in $511\mu s$
- In total we have $1285\mu s$ to transmit 1460 bytes of payload, which gives us a throughput of $9$Mbit/s.