Networks and Systems Security

Dr Sudipta Saha

https://www.iitbbs.ac.in/profile.php/sudipta/

Decentralized and Smart Systems Research Group (DSSRG)

https://sites.google.com/iitbbs.ac.in/dssrg

Computer Science & Engineering, School of Electrical Sciences
Indian Institute of Technology Bhubaneswar

IEEE 802.11 Standard

- WiFi
- **Different *variants*** of
  the **standards** are **differentiated** with a **letter**
- **Example:**
- **802.11a**
- **802.11b**
- **802.11g**
- **802.11n**
- **802.11ac**

- And so on... they keep **changing** the **standard (higher speed)**....

Brief history

- **802.11a**: (1999):
- Operates in the **5 GHz band** with a maximum net data rate of **54 Mbit/s**
- *Not* **popular....** (equipment was **expensive**)
- Does not **inter-operate** with **802.11b**

- **802.11b**: (1999)
- operates in the **2.4 GHz band** with a maximum net data rate of **11 Mbit/s**
- ***Very* popular** (maybe because **equipment** was **cheap** :))

- You should know that **microwave ovens** also operates in the same **2.4 GHz** range !!!

- **802.11g**: (2003)
- operates in the **2.4 GHz band** with a maximum net data rate of **54 Mbit/s**
- It is fully **backwards compatible** with **802.11b**.

Brief history

- **802.11n**: (2009)
- Not really a new standard....
- Is an **amendment** to the **existing 802.11** standards....
- Maximum data rate of **600 Mbit/s !!!**
- ***Most* important amendment**:
-

MIMO (Multiple Input, Multiple Output -- multiple antennas)

MIMO *Signal Processing* uses **multiple *antennas*** that **send** and **receive data** at the ***same* time** to improve signal coherence.

Significant increase in the transmission rate…

IEEE 802.11 Structure

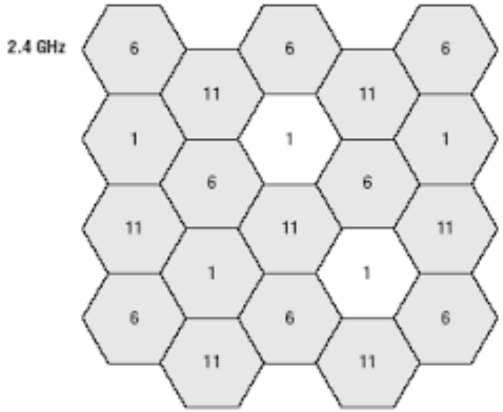
An **802.11 Lan** is subdivided into **cells**

Each **cell** is called a **Basic Service Set (BSS)**

A **BSS** is controlled by an **Access Point (AP)**

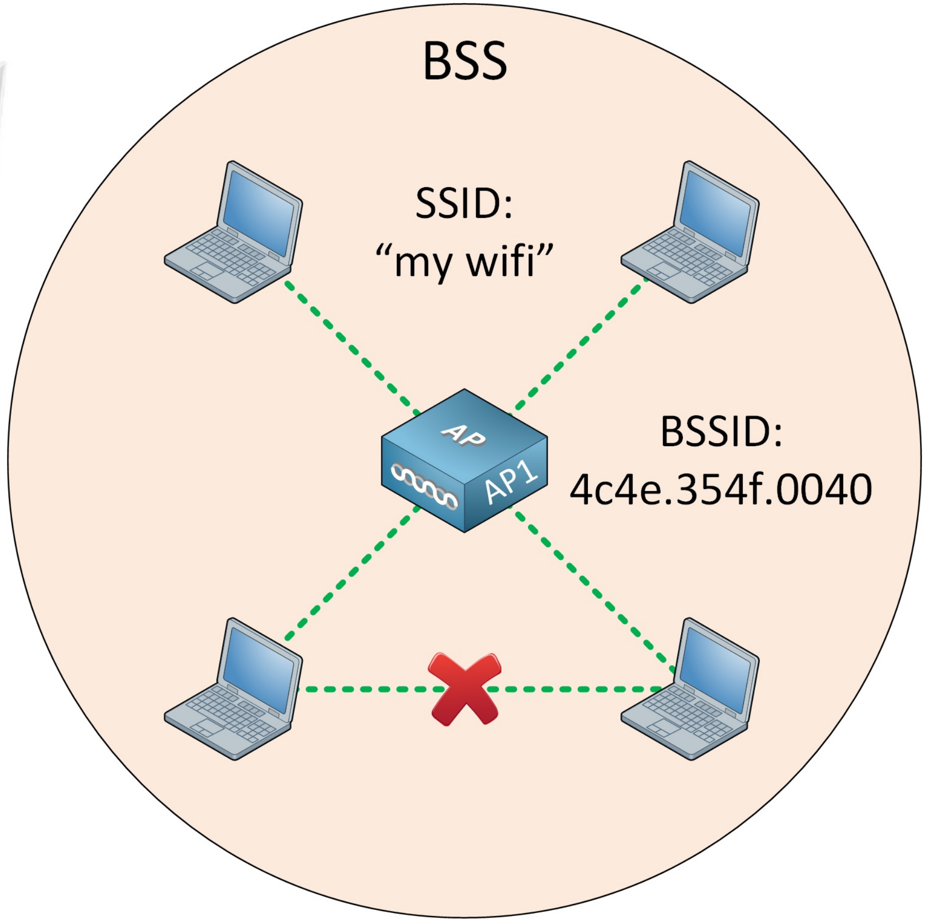
Most **802.11 networks** consist of **multiple APs**

The **APs** are **interconnected** together by a **backbone network**.



Each cell uses a ***different*** **transmission frequency** !!!

Signals of ***different*** **frequency** do **not interfere** with **each other**

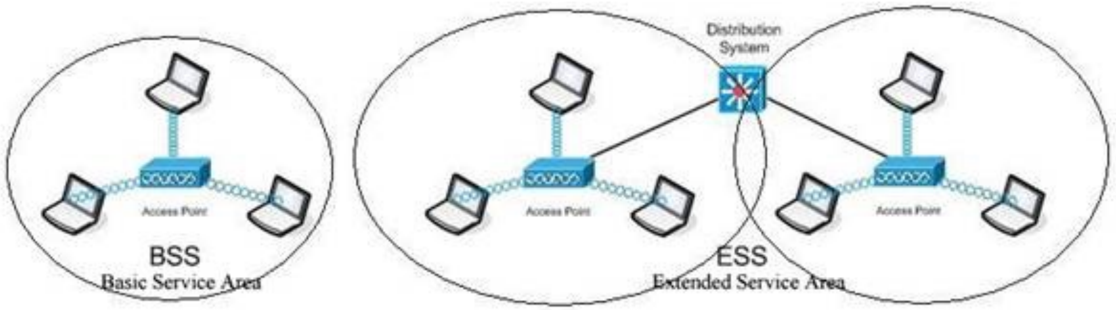


Each **cell** is called a **Basic Service Set (BSS)**

A **BSS** is **serviced** by an **Access Point (AP)** (**a.k.a.** a **base station**)

ESS and BSS

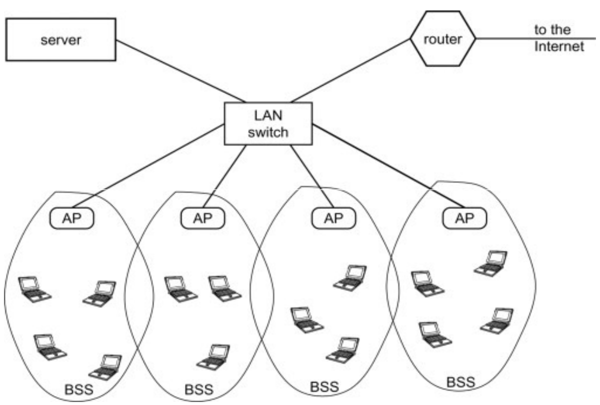
Most **802.11 networks** consist of ***multiple*** **Aps**
The **Access Points** are **interconnected** together using a **"backbone" network**.



The ***entire*** **interconnected wireless LAN** is called:

***Extended* Service Set (ESS)**

•

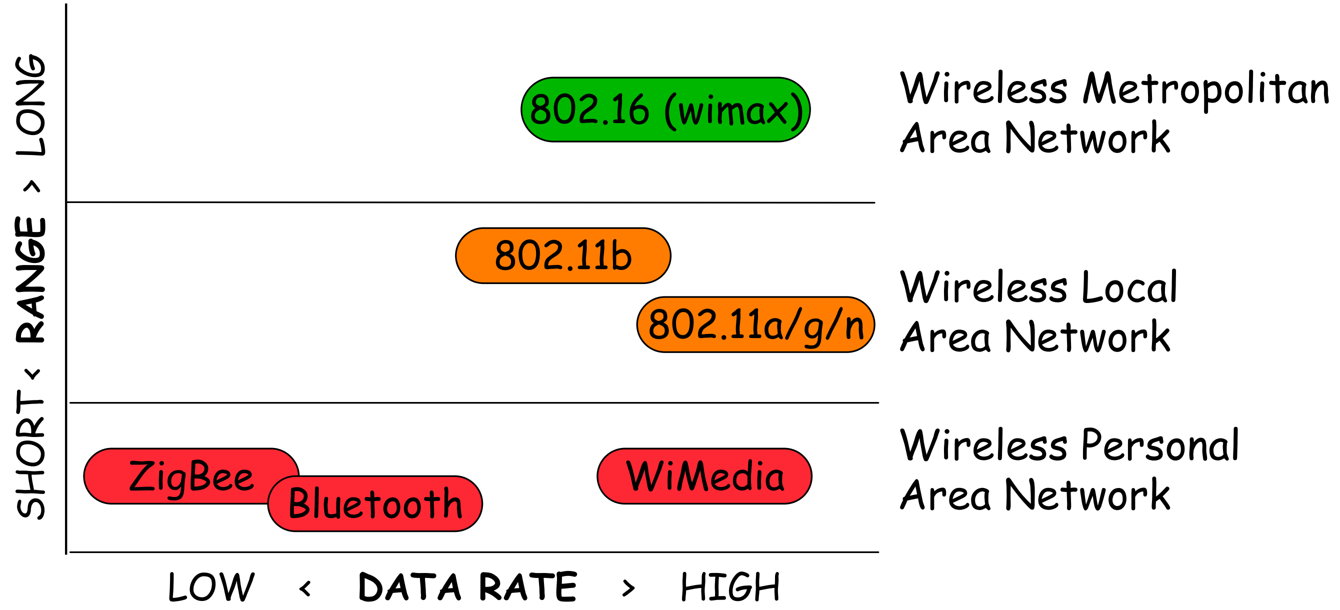


•The **backbone network** is **usually** an ***Ethernet*** **network**

The **backbone network** is called:

The **distribution system** in **802.11 literature**

Other wireless standards -



Standards typically define the Medium Access Control (MAC) and the Physical layers

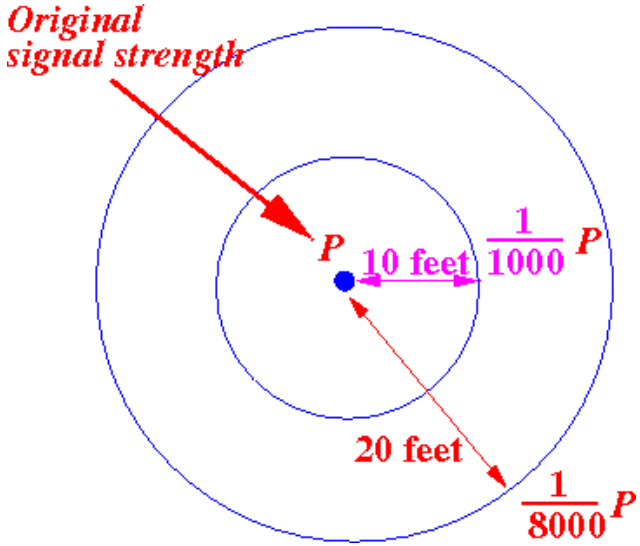
	Bluetooth	WiFi (802.11)	WiMax (802.16)
Data rate	2.1 Mbps	54 Mbps	70 Mbps
Link length	10 meters	100 meters	10 km
application	Peripheral devices	LAN	Access

The Wireless Networking environment (Physics...)

- The **signal strength** of wireless transmission **attenuates (= weakens) very rapidly**:
- **Signal strength *decreases*** at a rate of **$1/r^3$** where **r is the distance** to the **source**

- ***Rapid* signal strength attenuation**

results in: a ***limited* range** of **reception**.



-

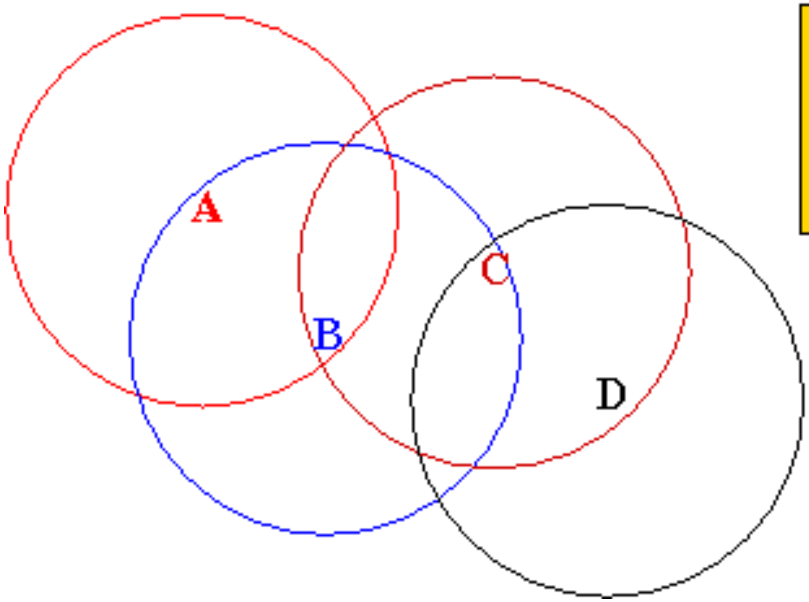
Basics

- **The Wireless Networking environment**

- **Physics:** The **signal strength** of wireless transmission **attenuates very rapidly** - at a rate of **$1/r^3$** where **r is the distance**)

This results in a **limited range** of reception.

Example



Reachability: A -> B
B -> A, C
C -> B, D
D -> C

- The letters indicate the physical location of each node
- The **circles** indicate the **transmission range** of each node
    - Node **B** is in **A**'s range.
    - Nodes **A** and **C** are in **B**'s range.
    - Nodes **B** and **D** are in **C**'s range.
    - Node **C** is in **D**'s range.

Range of **802.11 devices:**

- The **range** is **relatively short**:   100 feet
- **Because** of the **short distance**:
- A **node's *transmissions*** will **reach other nodes almost *instantaneously***
- **Nodes** will ***still* be transmitting** (a **frame**) when ***other* nodes detect** the **transmission**

- **Apparently it seems that Channel sensing can** help you ***avoid* collisions** !!! – Like Ethernet….

Question

- Since the propagation delay is so short, can a wireless node use carrier sensing to **avoid collisions ?**

Yes and No

Let us understand the difference between Wireless and Wired Networks

Career Sensing in Wireless Medium

***Wired* LAN:**

A **transmission** from **any node** will be **"heard"** by ***all* nodes** in the network

***Wireless* LAN:**

A **transmission** from **a node** will **only** be **"heard"** by **nodes within the *range*** of the **transmitting node**

The **limited range** makes a **significant difference** in the **ability to prevent collision** by **sensing the channel**

Issues caused by Limited Range

- There are **two** well-known **problems** caused by the **limited range** of transmissions:

The **Hidden node** problem

The **Exposed node** problem

Hidden Node Problem

Hidden node:

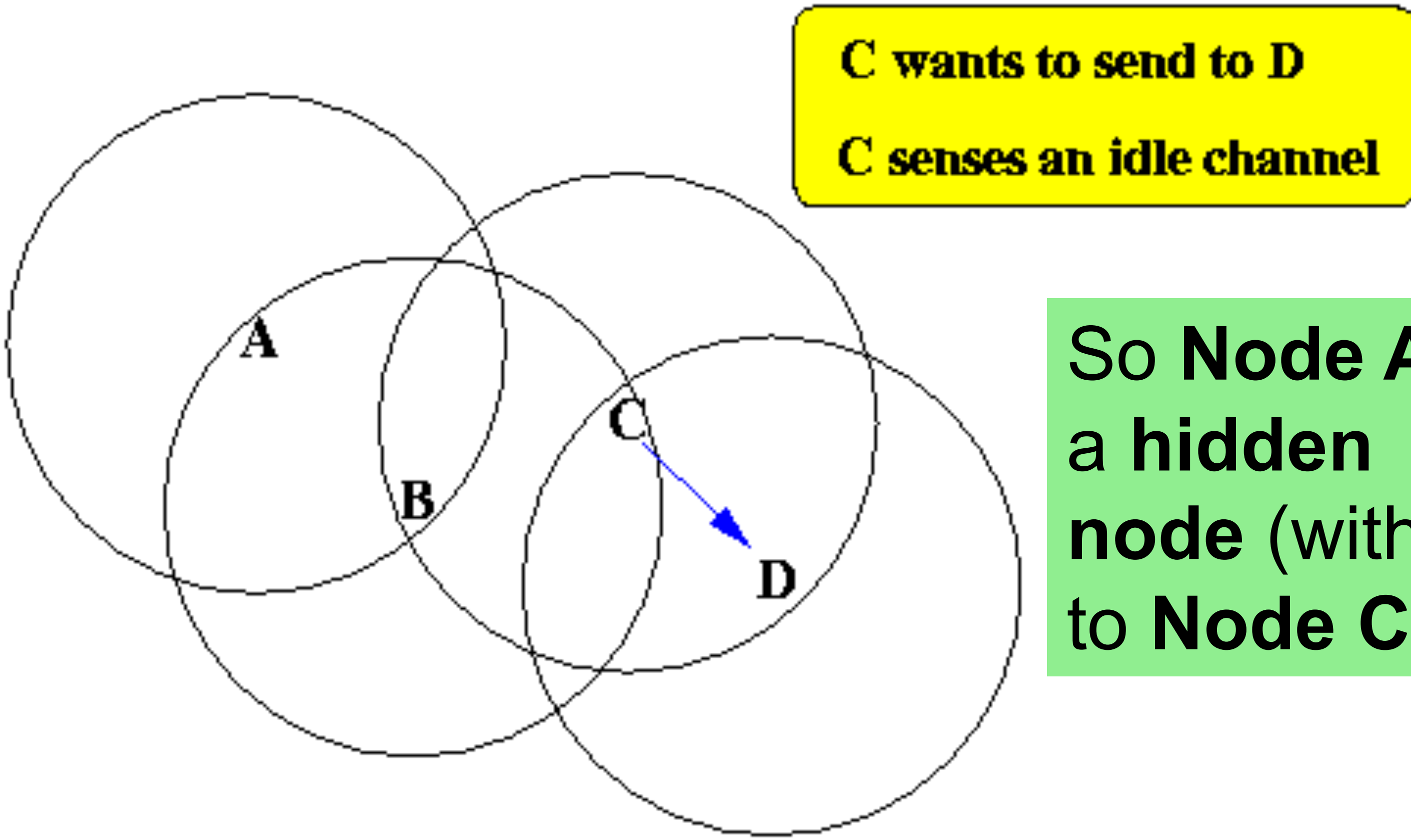
A **hidden node** is a node that is **outside your range**

You **cannot** be **aware** of **its existence**

Because you cannot transmit to nor receive anything from that node

Hidden Node

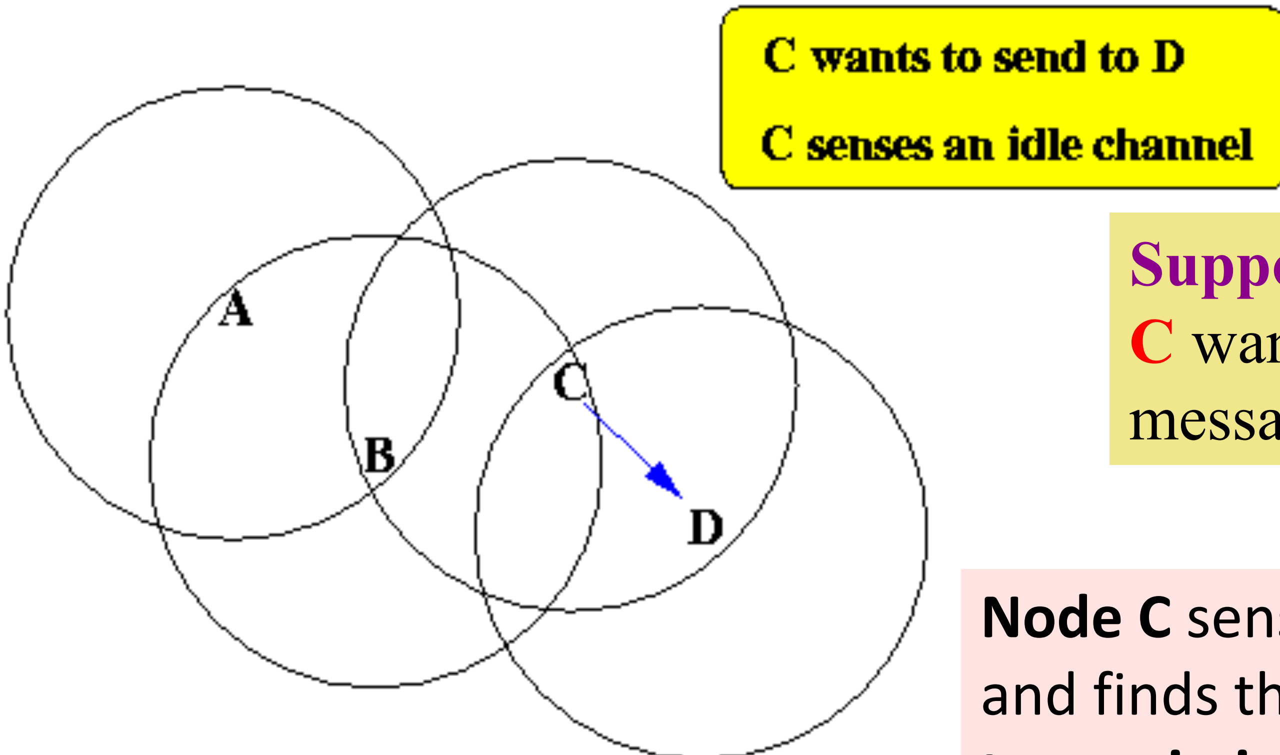
Node A is **outside the range** of **node C**



So **Node A** is a **hidden node** (with respect to **Node C**)

Problem due to hidden node

Due to **limited range**, ***your*** **transmissions** may **collide** with **transmissions** from **hidden nodes** without you being **aware of them**



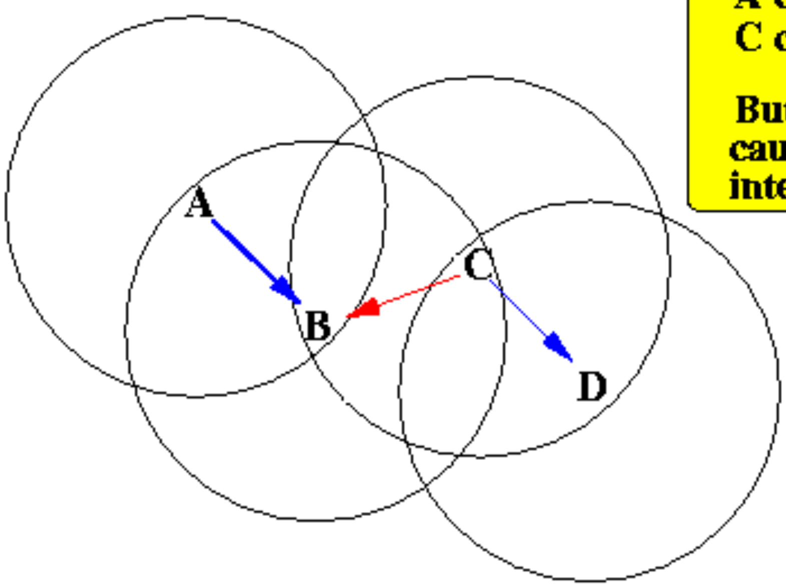
Suppose: Node C wants to send a message to D...

Node C senses the channel and finds that it is **idle (no transmission)** ....

Problem due to hidden node

- **Node C** can be **misled** by the **result** because the **hidden node A** can be **transmitting**:

The hidden node problem:



A wants to send to B
C wants to send to D

A can't hear C
C can't hear A

But C's transmission will cause errors at B due to its interference with A.

C cannot hear A's transmission

But, if **Node C does transmit**

It will cause an **collision** at node **B** !!!

Why this is not possible in wired medium ?

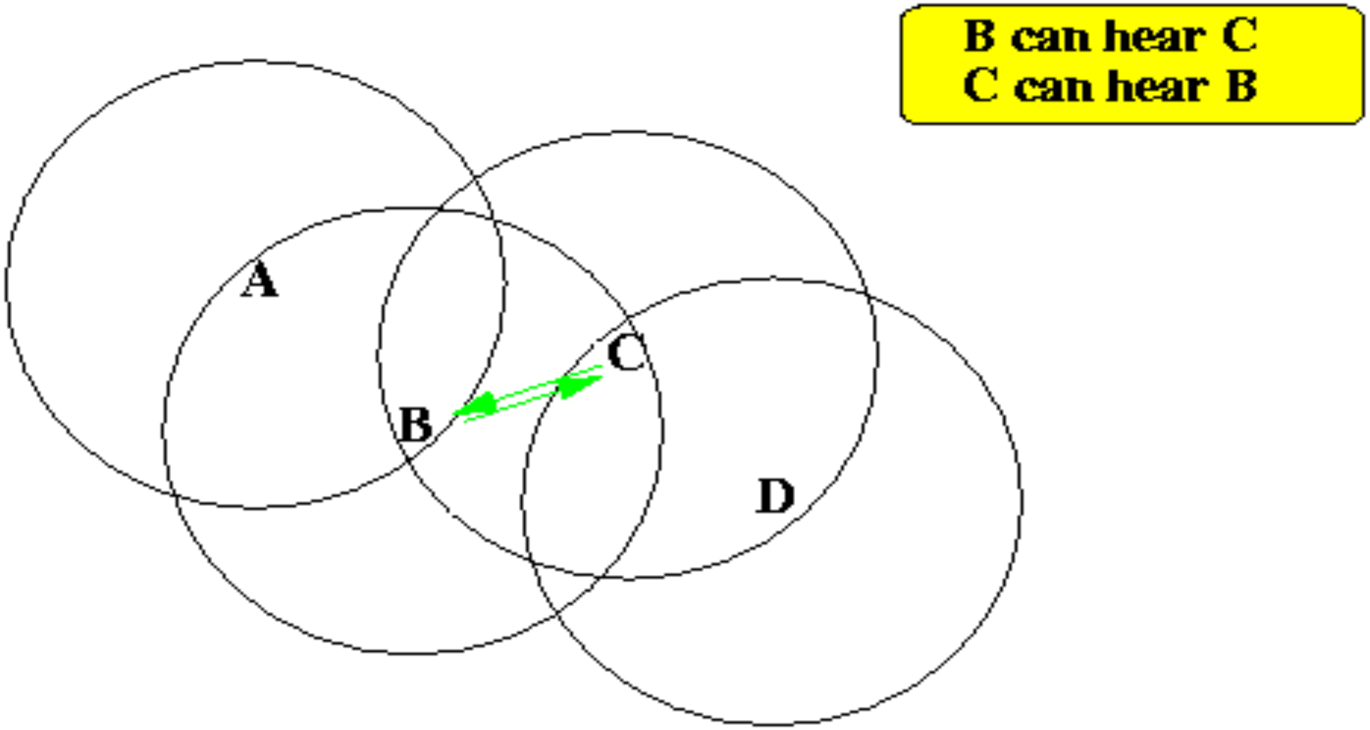
- When C will start CSMA – it will listen to the Carrier
- At that time C will sense A is transmitting
- If C starts first then A will get the idea that C is transmitting and hence wait
- CS in Wired medium prevents many possibilities of collision

Exposed Terminal / Node Problem

Exposed node:

A **exposed node** is a node that is ***within*** **your range**

Exposed nodes:



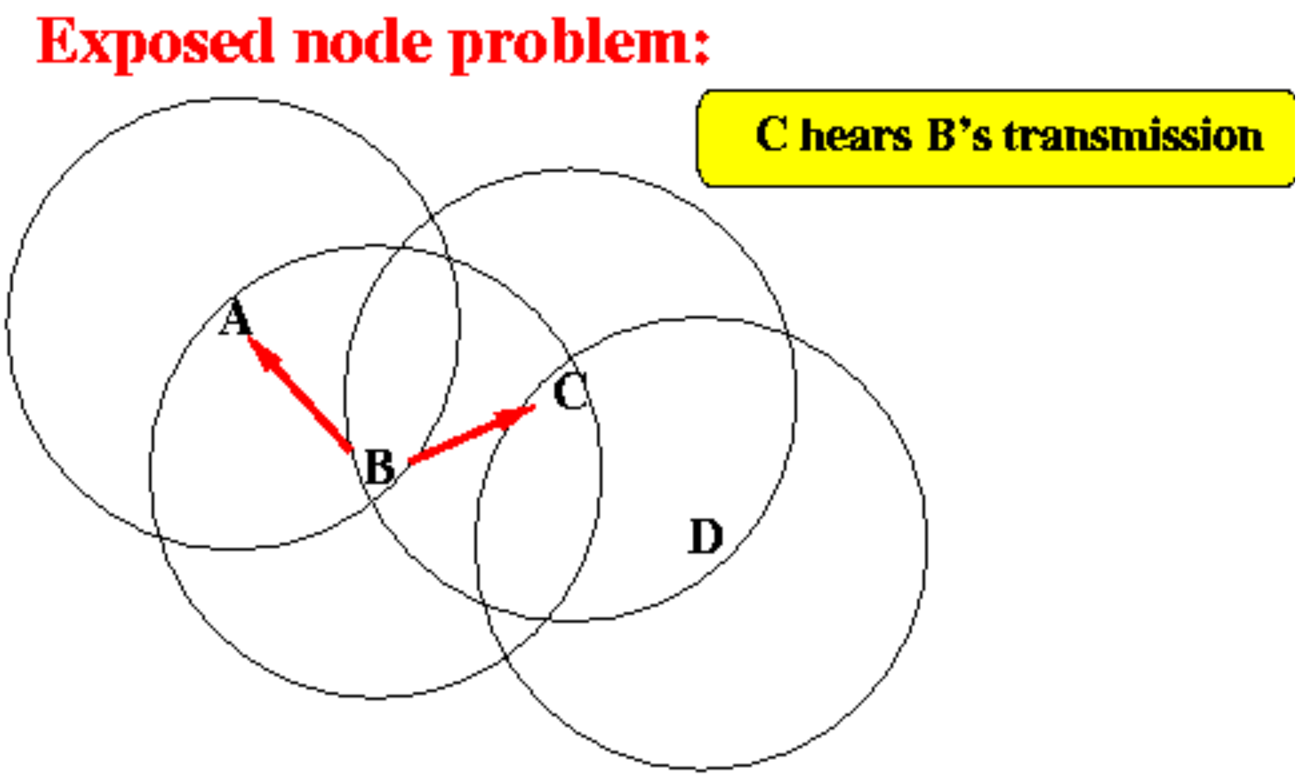
Problem caused by Exposed Nodes

Suppose

Node B is currently **transmitting** to **Node A**:

Now,

Node C wants to send a message to **D**...



Node B transmitting to A

Node C wants to send to D

Node C senses the channel and find that it is ***busy*** ....

Node C will be **misled** by the **result** because -

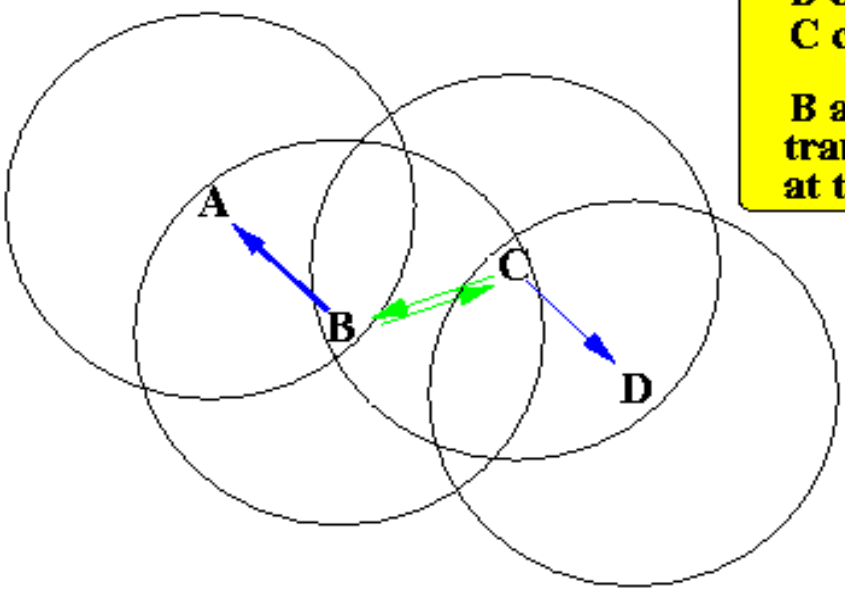
Although B is transmitting,

The **transmission** from **B** ⇒ **A** is harmless

It will ***not*** **interfere** with the **transmission** from **C** ⇒ **D**

Problem due to exposed nodes

The exposed node problem:



B wants to send to A
C wants to send to D

B can hear C
C can hear B

B and C will think their transmission will collide at the intended destinations

C hear B's transmission

But, if **Node C does transmit** (to **D**), it will **not** cause a **collision** !!!

Can this issue be there in Wired Medium ?

- When C senses B is transmitting – Indeed C should not transmit – as it will be colliding with B's transmission

- So, wireless medium brings the possibility of in parallel transmission – which is limited by the protocol CSMA

- Exposed transmission Limits the possibility of in parallel transmissions

Effectiveness of channel sensing in Wireless network

- **Summary of the above results:**

- A **wireless node** finds the channel **idle** → There is possibility **that** its transmission will **collide** with a current transmission from a **hidden node**
  - A **node** that is ***out of range*** can **interfere** with **your transmissions**
- If a **wireless node** finds the channel **busy** → There is the **possibility** than the **current transmission** will ***not*** **collide** with its own transmission...
  - A ***transmitting*** **node** that is ***in range*** but **do *not* interfere** with **your transmissions**

CSMA in Wireless Medium

Carrier sensing is **not very effective** in **wireless networks**...

Information that a node get from sensing the carrier does not tell him whether his transmission will or will not interfere with another transmission

But... it is ***not*** **completely useless** either -

Usually, there are **some hidden nodes** but **most** of the nodes are **not hidden**

So **carrier sensing still** makes sense...

But: **carrier sensing *alone*** is ***not*** **sufficient**

In addition to carrier sensing, we need **other strategies to avoid collision**...

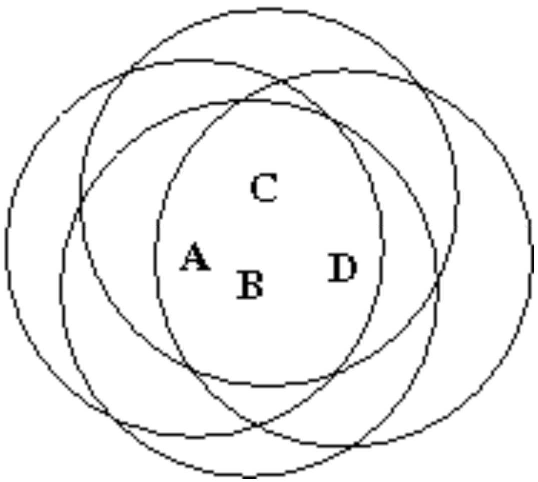
Why Collision detection does not work?

- **The 802.11 Medium Access Protocol - Channel sensing and no collision detection**
- **Channel sensing** is ***useful***:

  If there are **no hidden node**:

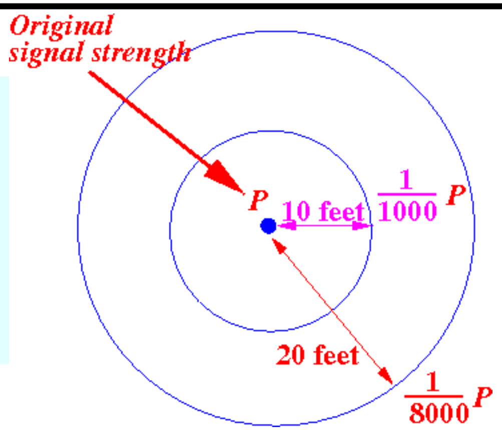
  -
- **carrier sensing** will ***avoid*** a **collision**

NO hidden node problem:



Collision detection

It is ***technically* not possible** to **detect collision** in **wireless transmissions** due to the **Hugh *difference*** in **signal strength**



Signal *strength* attenuates *very* rapidly in **wireless transmissions** (proportional to ***distance***^{3}):

The **signal level** of the **node's *own* transmission** is ***extremely* high**:

But: the **signal level** of ***another* (colliding) transmission** is ***extremely* weak → Because** the ***send* antenna** of the ***other* node** is ***far* away** from the **node's *receive* antenna**..

- Result:

- The **node's *own* transmission** will ***overwhelm*** the **signal** from the ***other*** **node(s)**

- **Thus, Collision detection** in **802.11** is **not technically feasible**

Thus

- Channel sensing is not foolproof
- Collision Detection does not work

- → FRAMEs may get lost.

- NEED MECHANISM FOR **Faster recovery of lost frames**

To **recover** *lost* **frame** as **quickly** as possible:

- **ACK**

- The **receiver** must send an **ACK** for each **correctly received frame**
- If **sender** does **not receive** an **ACK** for the **transmitted frame** within a **timeout period**:
- The **sender** will **retransmit** the **frame**

- **MAC level Acknowledgement** : **Protocol specification** is **defined** in the **Medium Access Control (MAC) layer(IEEE 802.11** is **just like Aloha)**

The **MAC level ACK protocol** used in **IEEE 802.11**:

- **Sender**

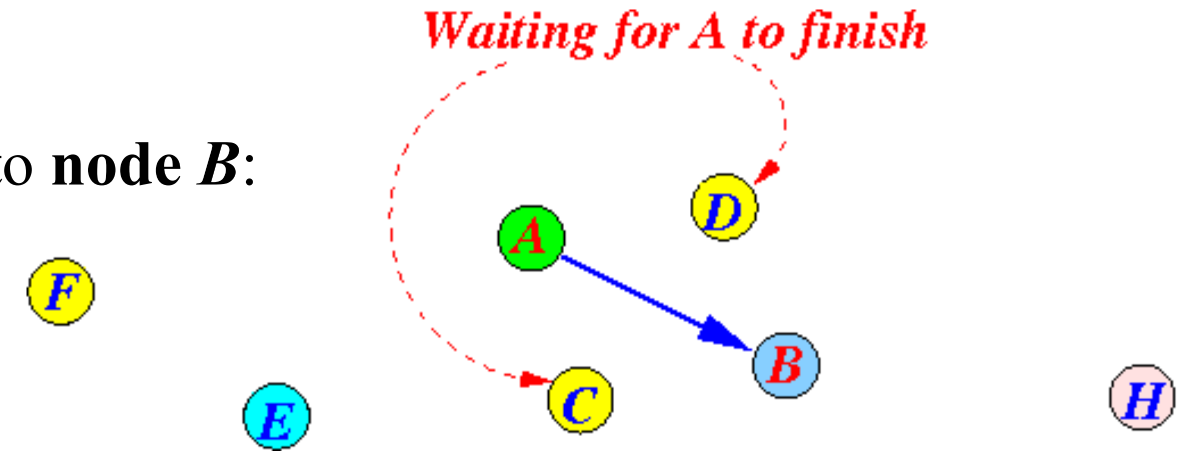
- Transmits a **frame** to **receiver**
- **Wait** for an **ACK frame** within a **time out period**
- If **time out** expires, **retransmit** the **frame**

- **Receiver**
- **Receives** a **frame**
- **Send** an **ACK frame** back to the **sender**

Problem implementing the MAC level ACK scheme

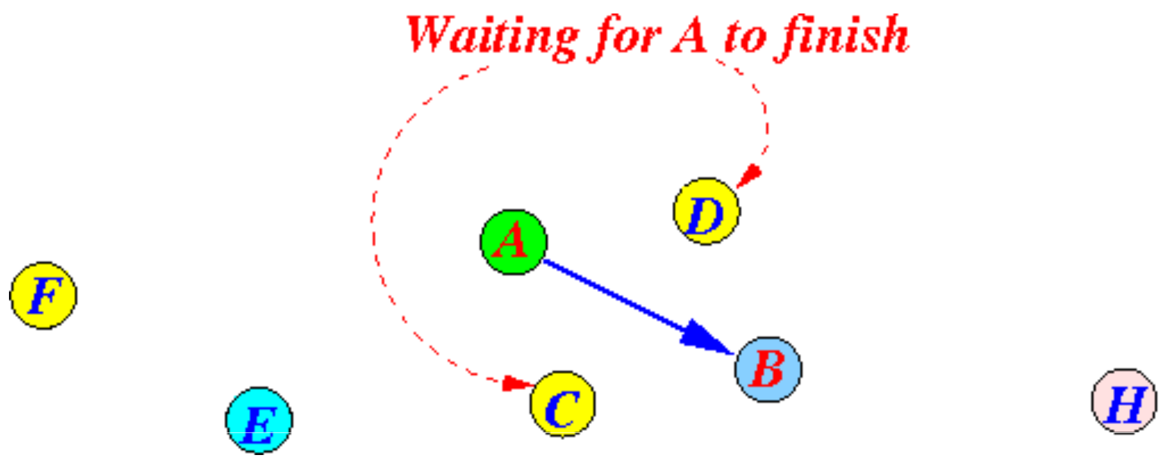
Consider the following scenario

Node ***A*** is currently **transmitting** to **node** ***B***:



Nodes ***C*** and ***D*** are ***waiting*** for **node** ***A*** to finish.

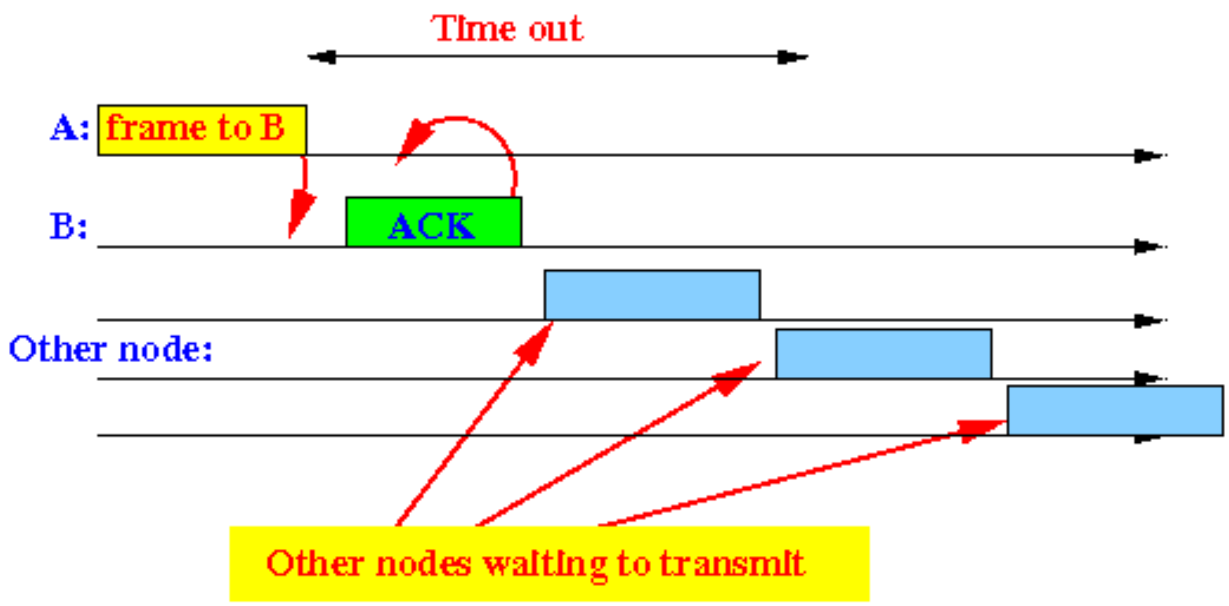
- We have the following ***interesting*** **scenario**:



- **Node** ***B*** will **transmit** the **ACK frame** (to *A*) as soon as **node** ***A*** is ***finished***

- **But**: **nodes** ***C*** and ***D*** will ***also*** **transmit as soon as node** ***A*** is ***finished*** !!!

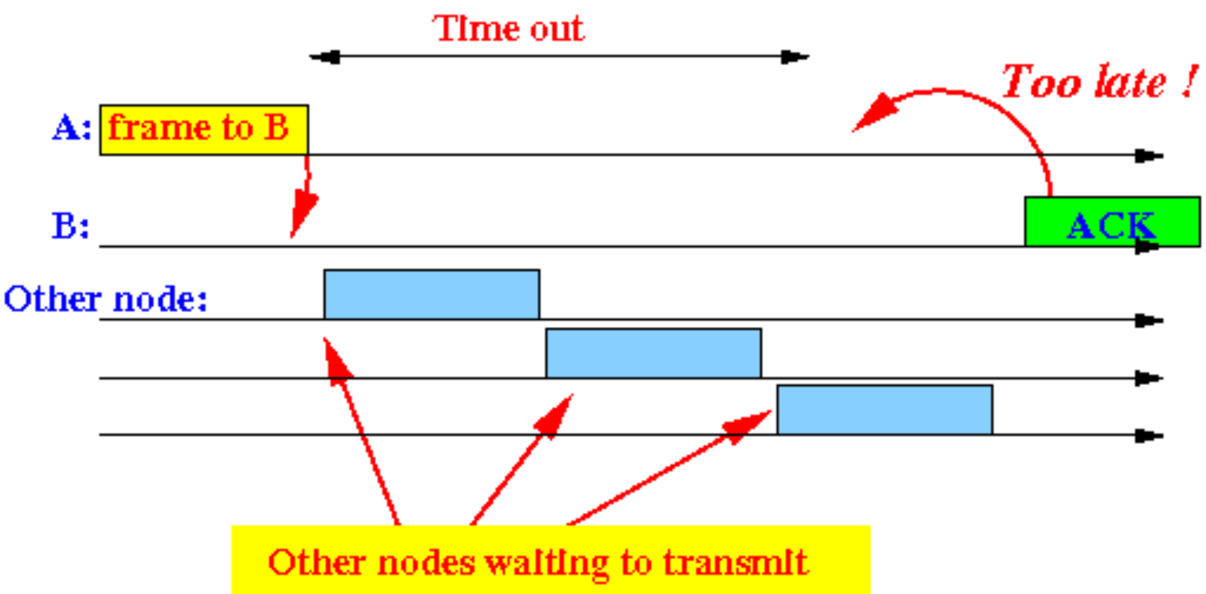
- We **need** to **ensure** that the **ACK transmission** goes **before** all ***other*** **transmissions**:



- We expect the scenario as above.

However in practice the following may happen

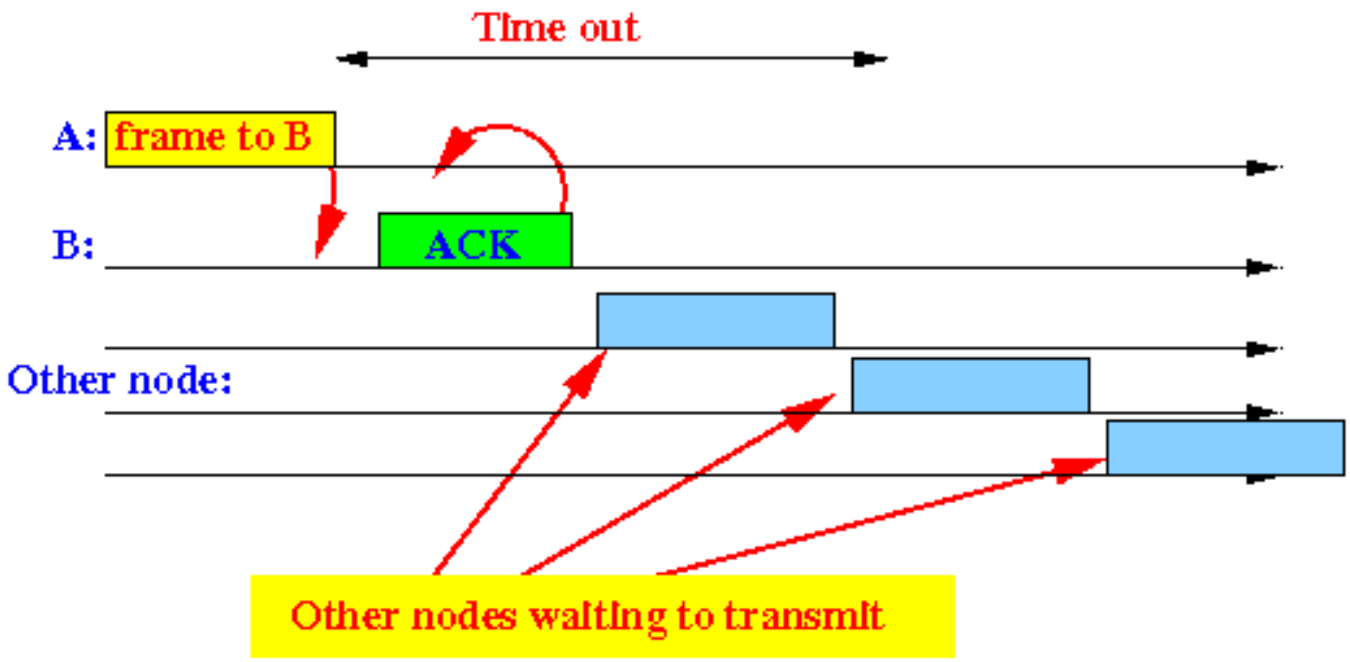
- **Transmissions** from ***other* nodes** gets ***ahead*** of **node *B***:



- The **ACK frame** of **B** will be **too late** !!!
- **Node *A*** will ***timed out*** !!!!

Question

- **How** can we **make sure** that an **ACK frame** is **transmitted** ***first*** ???



Problem

- The ***main*** **difficulty** of this **problem** is the **fact** that:

- **Nodes** that are ***ready*** **to transmit** are ***not*** **aware** of **each other** !!!

- **Furthermore**:

- The **nodes** must **not send messages** to **each other** to **solve** this **problem** !!!

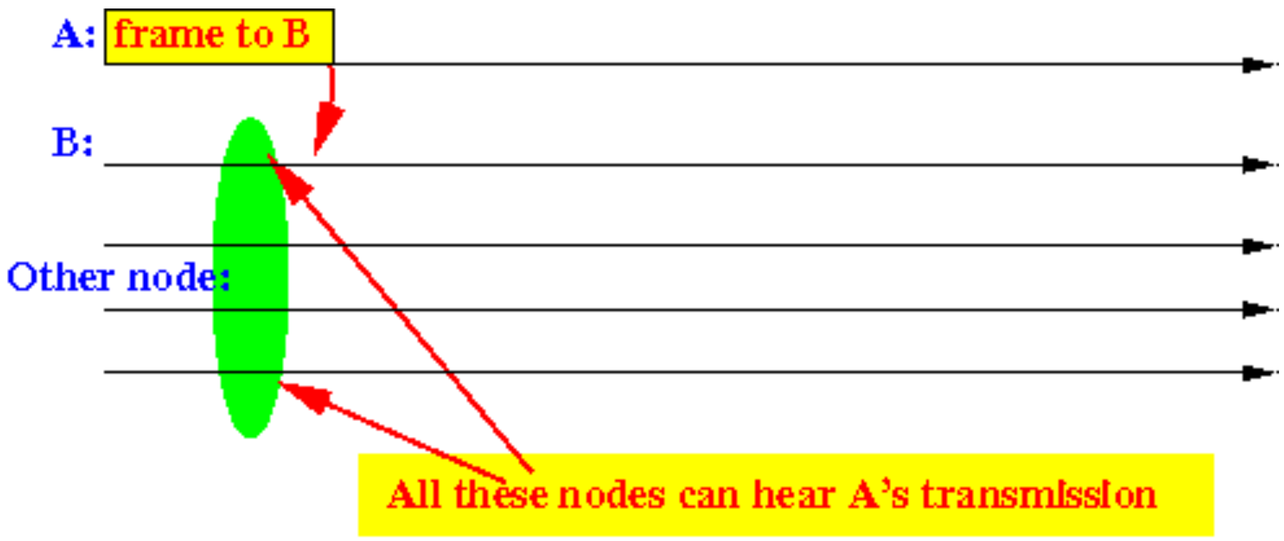
- **(Because the time it takes to send messages to each other will increase the delay --- the ACK frame will get too late !!!)**

- **We need a different solution**

Formulation of the problem -

- **Assigning *priority* in a *distributed* manner**

- ***Multiple* nodes** are ***waiting*** for a **transmission (node *A*)** to **finish**:



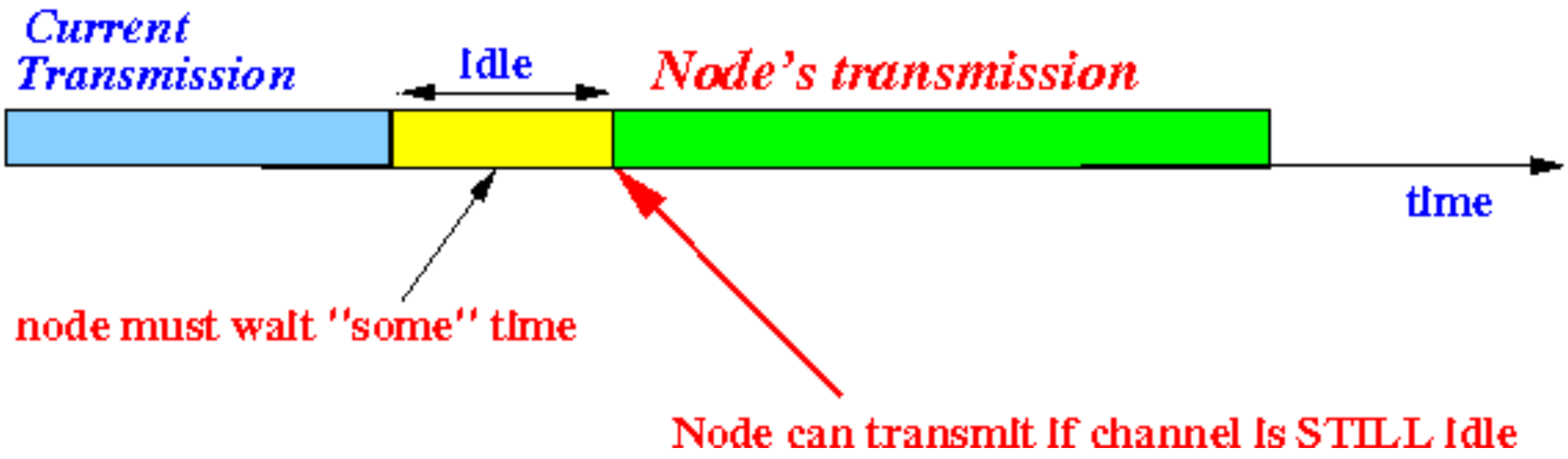
- **(multiple) nodes** will **transmit** when the **transmission** is **complete**

Problem statement …

- These **nodes** are **not aware** of **each other**
- We must **make sure** that **node *B*** transmits ***first***
- We must do
  so ***without*** using ***any*** **messages** communicated between the **nodes**

Solution

- When a **node** want to **transmit** a **frame**:
- The **node** must ***first*** **waits ("defer")** a ***predefined*** **amount of time** before the **node** can **transmit** its **frame**
- If the **channel** is **idle** after **waiting** the **predefined time**:
- The **node** will **transmit** the **frame** (immediately)

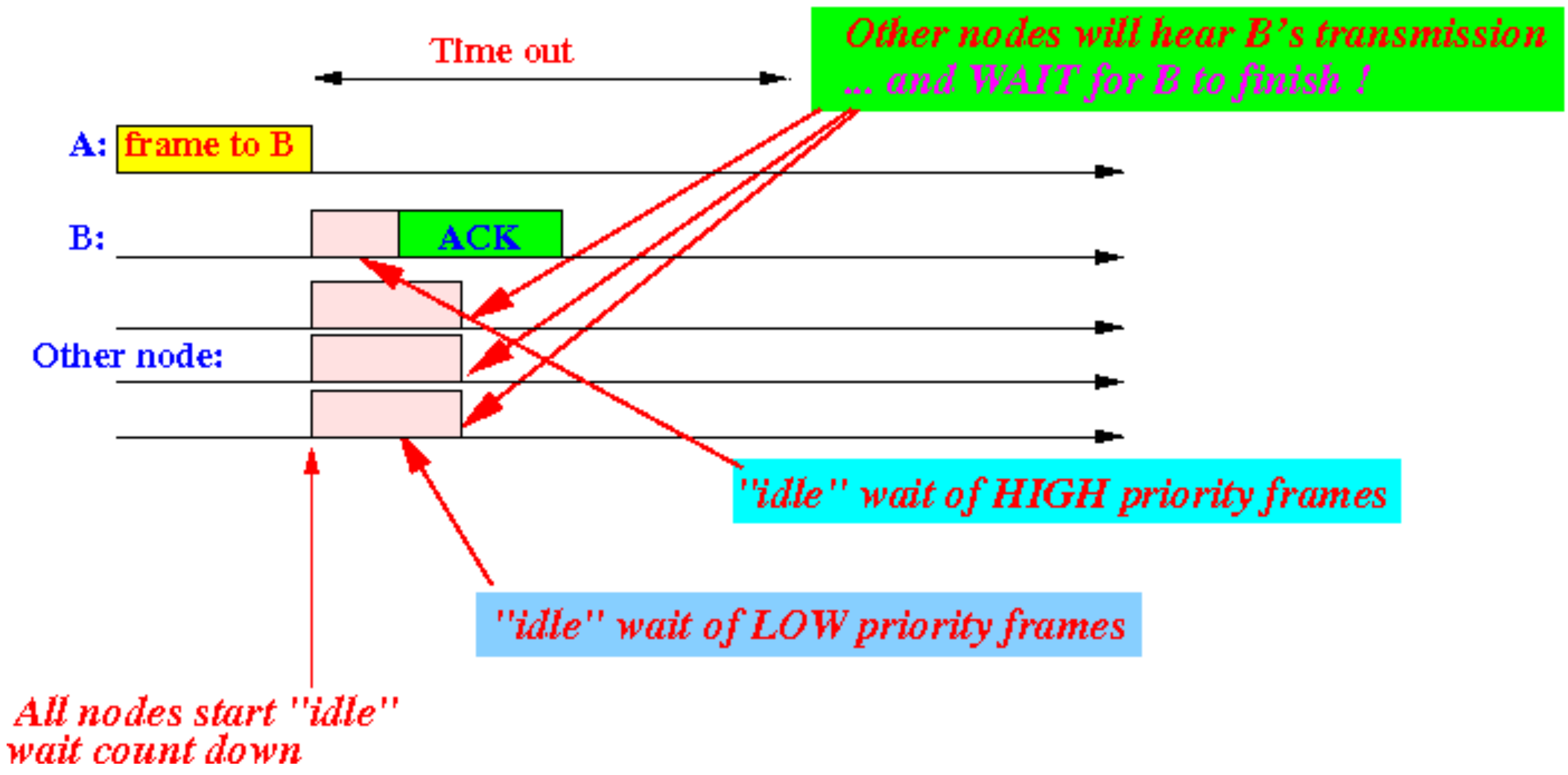


Solution

- Otherwise – [If some other transmission starts]
- The **node** must **wait** until the **current transmission** is **over**
- **Next it should *defer*** the **predefined wait time** again before **transmitting** !!!

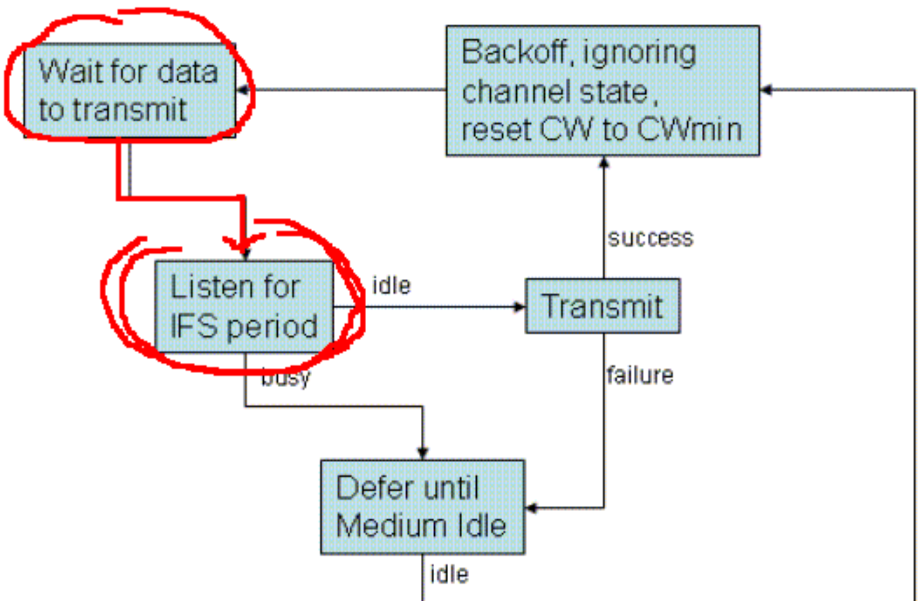
- **Key** to **prioritizing** the **ACK transmission**:
  - ***Different* types** of **frames** will use ***different* waiting (defer) time** !!!
  - ***Higher* priority frames** will use a ***shorter* waiting time** !!!

- Example



Time out
Other nodes will hear B's transmission
... and WAIT for B to finish !
A:
frame to B
B:
ACK
Other node:
"idle" wait of HIGH priority frames
"idle" wait of LOW priority frames
All nodes start "idle"
wait count down

Interframe Spacing

- **Interframe Spacing (IFS)** = the **period** of time that a **(transmitting) node** must ***wait*** **(and listen)** before the **node** can ***start*** **transmitting** a **frame**
- *Partial* **flow chart** of the **802.11 Medium Access Protocol**:
- 

Explanation:

- A **(transmitting) node** must **listen (monitor)** the **transmission medium** for ***IFS* amount of time**
- The **IFS** for ***different*** **types** of **frames** are ***different*** !
- If there are ***no*** **transmissions** for ***IFS*** **amount of time**, the **node** will ***transmit*** its **frame**
- If the **node** detect a **transmission during** the ***IFS*** **wait time**:
- The **node** will ***back off*** and **try again** later

What is the length of IFS?

Different type of frames uses different IFS durations !!!

SIFS = **Short** Interframe Spacing (has the **shortest duration**)
SIFS = 28 μsec

PIFS = **Point Coordination Function (PCF)** Interframe Spacing (has the ***second*** **shortest duration**)

PIFS = SIFS + 1 Slot Time = 78 μsec

PIFS is used as sensing delay by a **base station** (= coordination point) that operate in a special **transmission coordinator role**

Different IFS values

DIFS = **Distributed Coordination Function (DCF)** Interframe Spacing (has the ***"normal"*** **duration**)

DIFS = SIFS + 2 × Slot Time = 128 μsec = 28 + 2 X 50 = 128

DIFS is used as sensing delay for transmitting ***ordinary priority*** **data frames**

EIFS = **Extended** Interframe Spacing (has the **longest duration**)

The EIFS is derived from the SIFS and the DIFS and the length of time it takes to transmit an ACK frame at 1 Mbit/s by the following equation:

EIFS = a SIFS Time + (8 x ACK Size) + a Preamble Length + a PLCP Header Length + DIFS

EIFS is used as sensing delay for transmitting ***ordinary priority*** **data frames** when it has ***recently*** **received a** ***corrupted*** **frame**

IFS and Frame Priority

- **Interframe Spacing IFS defines *Priority* of the frames**

- The **length** of the **interframe space** determines the **priority** of a frame

The **shorter** the **duration (length)** of an **interframe space**, the **higher** the **assigned priority** of that frame will be

Example: **ACKs frames have the highest priority**

Various types of IFS in 802.11

- **SIFS** = **Short** Interframe Spacing (has the ***shortest*** **duration**): **SIFS = 28 µsec**
- **SIFS** is used as **sensing delay** for transmitting **ACK frames**

- **PIFS** = **Point Coordination Function (PCF)** Interframe Spacing (has the ***second*** **shortest duration**)
- **PIFS = SIFS + 1 Slot Time = 78 µsec**
- **PIFS** is used as **sensing delay** by a **base station** (= ***coordination point***) that ***operates*** in a ***special*** **coordination mod**

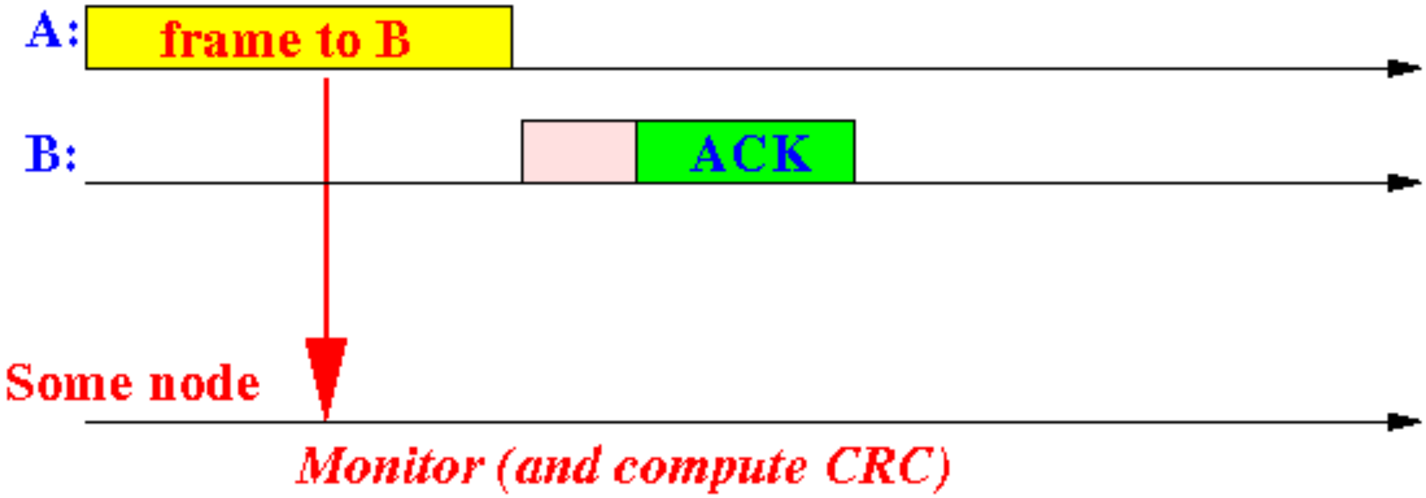
Various types of IFS in 802.11

- **DIFS = Distributed Coordination Function (DCF)** Interframe Spacing (has the ***"normal"*** **duration**)
- **PIFS = SIFS + 2 × Slot Time = 128 μsec**
- **DIFS** is used as **sensing delay** for transmitting **data frames**

- **EIFS = Extended** Interframe Spacing (has the **longest duration**)
- **EIFS = ACK frame duration + SIFS + DIFS**
  EIFS is used by a **transmitting node** that **received** a ***corrupted*** **data frame**
-

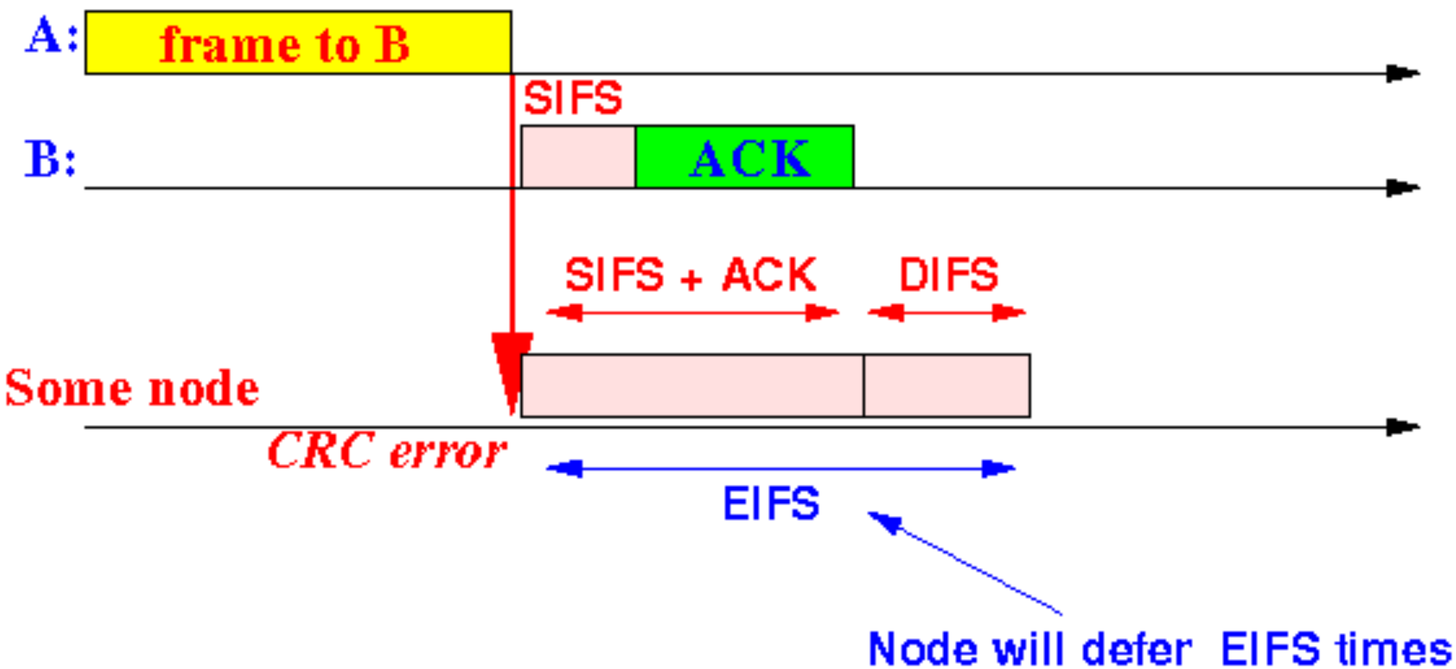
Use of EIFS

- *Every* **node** in **IEEE 802.11** will **continuously receive** and **CRC check** ***all*** **frames** !!!
-



Use of EIFS

- If a **received frame** contains an **(CRC) error** then:
- A **node** will defer **EIFS** duration **instead** of **DIFS** before **transmitting** a **data frame**



Reason

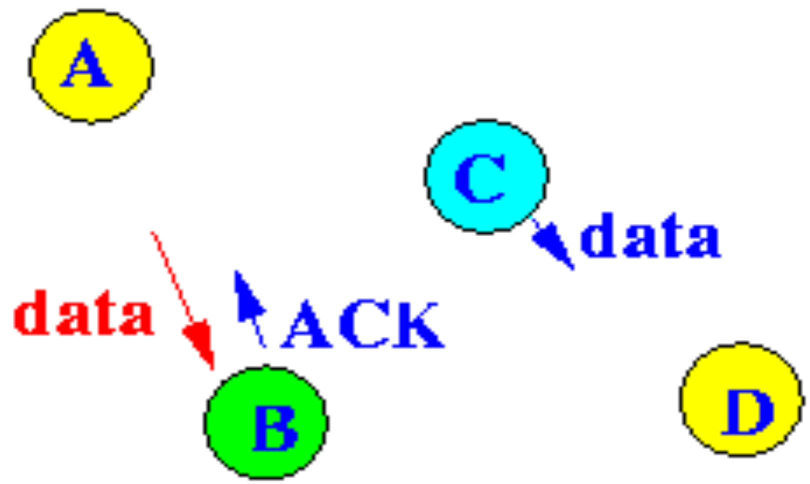
- We think of the following possibility
- The ***corrupt*** **frame** may be a **data frame** for ***another*** **node** (far away -- that's why it was corrupt)
- ***That*** **(other) node** may have **received** the **data frame** ***correctly*** !
- By ***waiting*** **EIFS**, the **node** will let the ***other*** **node** transmit the **ACK frame without collision** !!!
-

Examples

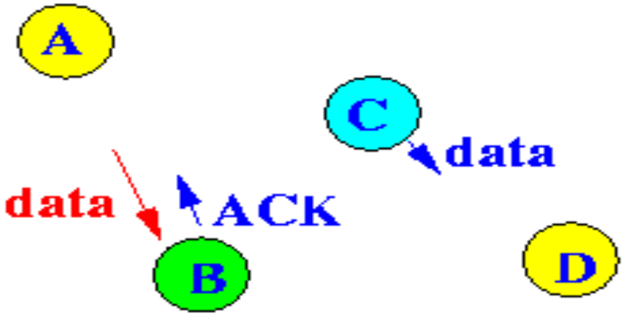
Suppose **node A** just finished sending a **data frame** to **B**

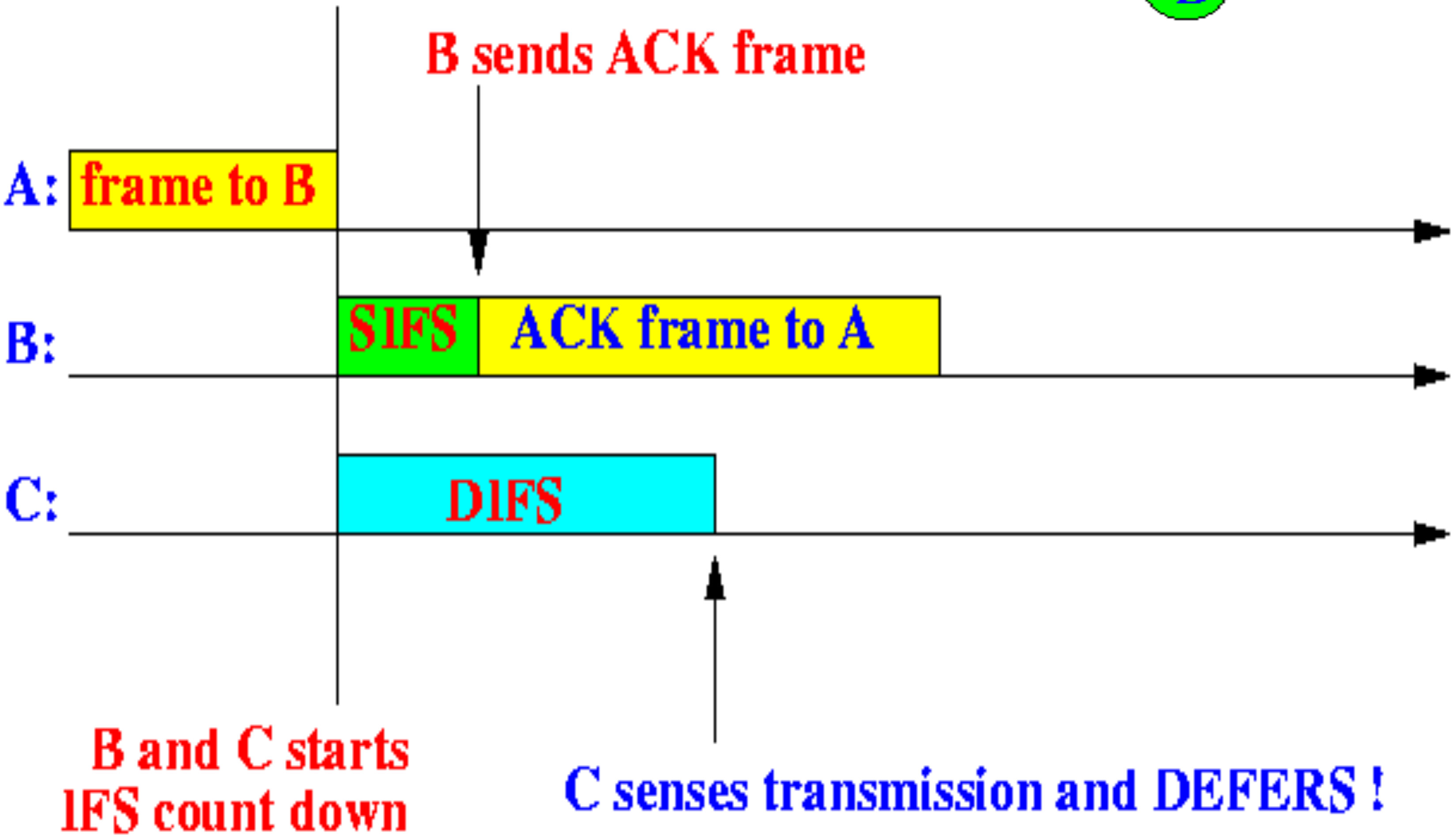
Node B wants to send an **ACK frame** back to **node A**

But in the mean time, **node C** wants to send a **data frame** to a **node D**



Timing diagram





Use of EIFS

- The **EIFS** is a **very long waiting duration** in **802.11**

- It is **not** intended for regulating **priority** but to overcome a **problem**

Background

- Stations **monitor frames transmissions** on the transmission medium **continuously**

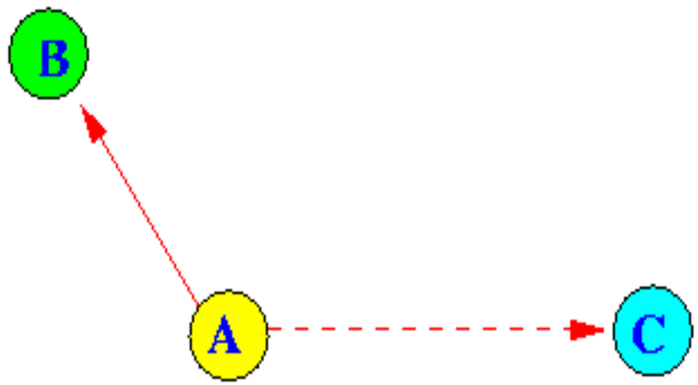
- This **monitoring** includes **computing the CRC** on the frames

- The **EIFS** is used in **DCF** when a station wants to send a **data frame after** it has **monitored an erroneous frame** i.e., the **previous** MAC frame was **incomplete** or has an **incorrect CRC**.

Intuition

- When I receive a corrupted packet
  - I infer that some is trying to send a packet to someone else – (if not intended to me)
  - Now, since I am not receiving the actual packet, obviously I am not able to see the ACK packet also
  - Therefore, I should wait until the sender receives the ACK packet or complete any back off too
  - Here if I wait only for a DIFS period then – with high chance when I will do CS – the node from which the corrupted packet came would be receiving ACK – which will get corrupted due to my transmission
  - So wait for EIFS

EIFS



A sends a frame to **B**

C receives the frame with an error (it might be interference at C or it's slightly out of range)

B receives the frame **correctly**

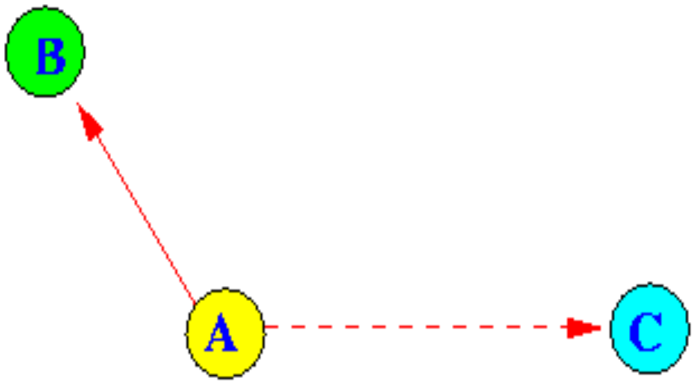
B will send the **ACK** frame
C will **not** hear it

Under normal operation, **C** will transmit **after delaying DIFS**

C will sense an **idle channel** after **DIFS** and transmit...

Result: collision an **important ACK frame** at **node A**

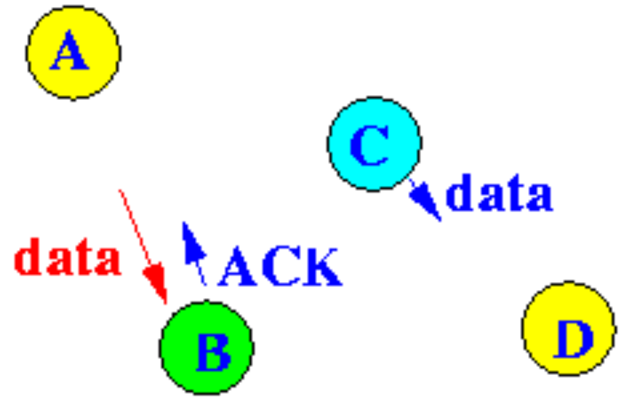
Compare the two Scenarios



Here C can see A – roughly – but cant sense transmissions from B

So ACK will not been seen by C

So should wait longer if gets an indication that B may send an ACK

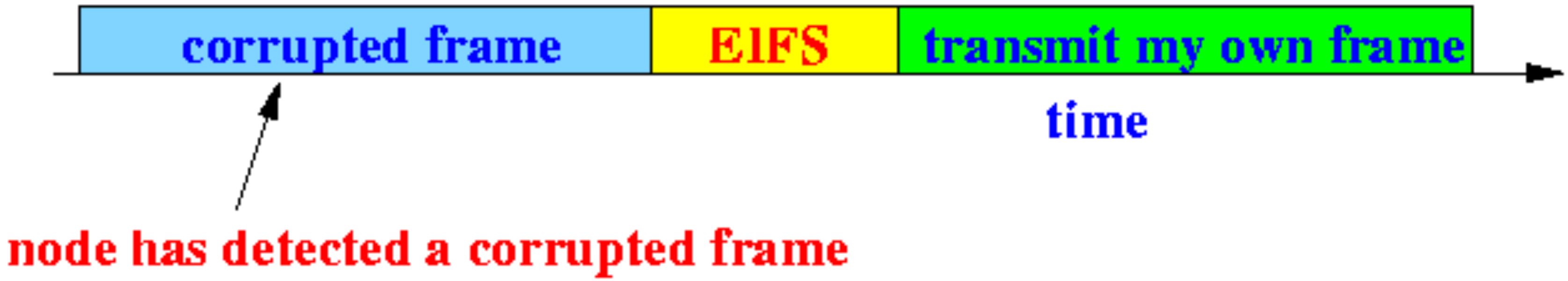


Here C can see A and B both

So, may wait for DIFS and then start

In case there is ACK transmission, C can sense it and wait more

EIFS based solution



Solution:

Make **C** wait **long enough** for the **(short) ACK frame** from **B** to **finish**

This **longer waiting time** is **EIFS**

Overview of 802.11 Medium Access Control protocol

- **IEEE 802.11 uses slotted transmission**
- **Slotted transmissions** (in general) will **reduce** the **likelihood** of **collisions**
- Since **nodes** in the **802.11** network ***cannot*** detect **collisions**, there is no **"wasting"** of slots when **collision** is **detected**....
- **Duration** of a **slot** in **802.11** is:

- **σ = 20 μsec**
- (A message can be longer than one slot)

- **Modes of MAC protocol operations**
- The **802.11 MAC protocol** can operate in **2 different modes**:
- **Centralized mode without contention**
- **Distributed mode with contention**