

# **Wirless Networks And Mobile Applications**

Year 2022-2023

Authors:

*Brugnolaro Filippo*

*Mister X*

# Disclaimer

Hello guys!

These notes contain all the concepts and arguments which are explained during professor's lessons. However they are not intended in any sense as a replacement for professor's lessons, but as a help for studying and preparing the exam. There could be also the presence of some errors and we don't take any responsibility for them. If you like to contribute for any correction, here is the link to the repository:

<https://github.com/filippobrugnolaro/WNMA-notes>

You can create a new branch with all modifications and create a pull request. We'll be pleased for any correction in order to improve the quality of the document. Hope it could be useful.

Cheers :)

## Contents

<b>1 Introduction</b>	<b>4</b>
1.1 Wireless Development	4
1.2 Wireless Systems	5
1.2.1 Cellular Systems	5
1.2.2 Wireless Local Area Networks (WLANs)	6
1.2.3 Satellite Systems	7
1.2.4 Bluetooth	7
1.2.5 Ad Hoc Networks	7
1.2.6 Mesh Networks	8
1.2.7 Sensor Networks	8
1.2.8 Distributed Control over Wireless Links	8
1.2.9 Mobile Ad Hoc Networks (MANET)	9
1.2.10 Opportunistic Ad Hoc Networks	9
1.2.11 Vehicular Ad Hoc Networks (VANET)	9
1.2.12 Flying Ad Hoc Networks (FANET)	9
1.2.13 Underwater Sensor Networks	9
1.2.14 Radio Frequency Identification (RFID)	10
<b>2 Radio Frequency</b>	<b>11</b>
2.1 Properties	11
2.2 Wireless Transmission	12
2.3 Antennas	13
2.3.1 Omnidirectional antennas	13
2.3.2 Semi-directional antennas	13
2.3.3 Highly-directional antennas	14
2.3.4 Sectorised-directional antennas	14
<b>3 Wireless Physical Layer</b>	<b>15</b>
3.1 Characteristics	16
3.2 Wireless vs Wired	20
<b>4 MAC Layer</b>	<b>21</b>
4.1 Introduction	21
4.2 MAC Protocols	22
4.2.1 Slotted Aloha	23
4.2.2 Pure Aloha	23
4.2.3 Considerations Pure & Slotted Aloha	24
4.2.4 Carrier Sense Multiple Access (CSMA)	24
4.2.4.1 1-persistent CSMA	24
4.2.4.2 Non-persistent CSMA	25
4.2.4.3 P-persistent CSMA	25
4.2.4.4 CSMA with Collision Detection (CSMA/CD)	26
4.2.5 Wireless Medium Access Control	26
4.3 802.11 Protocol	27
4.3.1 CSMA version	28
4.3.2 CSMA/CA version	28
4.3.3 Point Coordinating Function (PCF)/Polling version	30
4.3.4 Considerations CSMA	30

4.3.5	Synchronization in 802.11 . . . . .	31
4.3.6	Congestion Avoidance (DCF) . . . . .	31
4.3.7	MILD Algorithm MACAW . . . . .	32
4.3.8	Fairness Issue . . . . .	32
<b>5</b>	<b>Network Layer . . . . .</b>	<b>34</b>
5.1	Routing algorithm . . . . .	34
5.2	Infrastructure Network vs MANET . . . . .	35
5.3	Routing Procols Classification . . . . .	36
5.3.1	Proactive Routing Approach . . . . .	36
5.3.1.1	Destination Sequenced Distance Vector (DSDV) . . . . .	37
5.3.1.2	Optimized Link State Routing (OLSR) . . . . .	38
5.3.1.3	Clusterhead Gateway Switch Routing (CGSR) . . . . .	38
5.3.2	Reactive Routing Approach . . . . .	38
5.3.2.1	Ad Hoc On-Demand Distance Vector Routing (AODV) . . . . .	39
5.3.2.2	Dynamic Source Routing (DSR) . . . . .	41
5.3.2.3	Associativity-Based Routing (ABR) . . . . .	42
5.3.2.4	Signal Stability Routing (SSR) . . . . .	42
5.3.2.5	Other metrics definition . . . . .	42
5.3.2.6	Flooding . . . . .	42
5.3.3	DSDV vs AODV . . . . .	43
5.3.4	DSR vs AODV . . . . .	43
5.3.5	Other Routing Protocols . . . . .	43
5.3.5.1	Greedy Perimeter Stateless Routing (GPSR) . . . . .	43
<b>6</b>	<b>Transport Layer . . . . .</b>	<b>45</b>
6.1	Overview . . . . .	45
6.2	TCP Protocol . . . . .	46
6.2.1	TCP Reliability . . . . .	46
6.3	Legacy TCP Versions . . . . .	49
6.3.1	TCP Tahoe . . . . .	49
6.3.2	TCP Reno . . . . .	49
6.3.3	TCP New Reno . . . . .	49
6.3.4	TCP SACK . . . . .	49
6.3.5	TCP Vegas . . . . .	49
6.4	Wireless TCP . . . . .	53
6.4.1	Wireless TCP Protocols . . . . .	54
6.4.1.1	SNOOP Protocol . . . . .	54
6.4.1.2	Satellites . . . . .	55
6.4.1.3	Slow Start and Congestion Avoidance Models . . . . .	55
6.4.1.4	RTT Unfairness . . . . .	56
6.4.1.5	TCP Hybla . . . . .	56
6.4.1.6	TCP Westwood & TCP Westwood Plus . . . . .	57
6.4.1.7	TCP Adaptive Selection . . . . .	58
6.4.1.8	TCP Cubic . . . . .	58
<b>7</b>	<b>Vehicular Ad Hoc Networks (VANET) . . . . .</b>	<b>60</b>
7.1	IEEE 802.11p . . . . .	60
7.2	Vehicular Networks: System Model . . . . .	60
7.3	Fast Broadcasting . . . . .	61

# 1 Introduction

## 1.1 Wireless Development

### Present

it is constantly growing due to higher use of laptops or devices which can connect to internet. This implied an important growth of WiFi and n-G (3G, 4G, 5G) technologies also thanks to the emerging of apps with both low and high data demand. Smartphones open to new wireless scenarios such as AR, VR, MR, tele-presence. . . Other topics are Tactile Internet (combination of low latency, high availability, reliability and security) and Web Squared (integration of web 2.0 with technologies of sensing).

### Future

it is based on ubiquitous communication among people and devices. So this implies to take into account some requirements such as bandwidth, delay, energy and connectivity.

### Challenges

- Wireless channels are a difficult and capacity-limited broadcast communications medium (with respect to the wired counterpart);
- Traffic patterns, user locations, and network conditions are constantly changing;
- Applications are heterogeneous with hard constraints required by the network;
- Energy and delay constraints change design principles across all layers of the stack.

### Multimedia requirements

	Voice	Data	Video	Game
Delay	low	irrelevant	low	low
Packet Loss	low	no	low	low
Bit Error Rate	$10^{-3}$	$10^{-6}$	$10^{-6}$	$10^{-3}$
Data Rate	8-32 Kbps	1-100 Mbps	1-20 Mbps	32-100 Kbps
Traffic	Continuous	Bursty	Continuous	Continuous

One-size-fits-all protocols and design

- are used by wired networks → poor results;
- do not work well → Crosslayer design.

**Crosslayer Design**

It's made of 5 layers:

Application	→ Meet delay, rate and energy constraints
Network	→ Adapt across design layers
Access	→ Reduce uncertainty through scheduling
Link	→ Provide robustness via diversity
Hardware	

**1.2 Wireless Systems**

There are different types of current wireless systems:

- Cellular Systems;
- Wireless LANs;
- Satellite Systems;
- Bluetooth;
- ...

And others which are emerging:

- Ad hoc Wireless Network;
- Mesh Network;
- Sensor Network;
- Distributed Control Network;
- MANET/VANET/FANET;
- Underwater Networks;
- RFID;
- Nano-networks;
- ...

**1.2.1 Cellular Systems**

Characteristics:

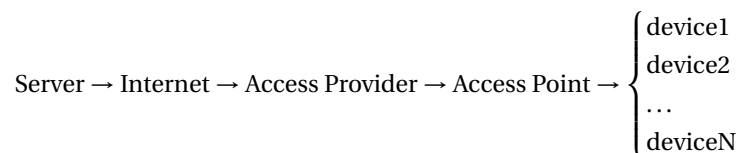
- every geographic region is divided into cells
  - more transmission distance  $\Rightarrow$  more power;
- frequency/timeslots/codes are reused at separated locations;
- co-channels interference between same color cells;

- base stations has control of functions and handoff;
- it can be shrunk to increase capacity and relax networking burden.
- it supports both voice (continuous) and data (bursty) requiring different:
  - access
  - routing strategies
- About connectivity:
  - 3G: packet-based switching for both voice and data (up to 7.2 Mbps)
  - 4G - 5G: are more focused on data (high bandwidth, high reliability, low latency)

### 1.2.2 Wireless Local Area Networks (WLANs)

Characteristics:

- devices are connected (wireless) to an AP<sup>1</sup>
  - it is wired-connected to internet;
- breaks data into packets ( $\approx 1500$  B) → AP<sup>1</sup> in even smaller size (500 B);
- MAC layer control access to shared channel (random access);
- backbone internet provides best-effort service
  - bandwidth cannot be determined!
  - users pay subscription only for home-access provider distance
    - ⇒ it can be bottleneck if the backbone is faster
  - having QoS (subscription) here can increase digital gap



There are different versions (802.11):

- b (old gen): only 2.4 GHz, speed 1-11 Mbps, range 100m
- g (legacy std): 2.4-5 GHz, speed up to 54 Mbps
- n (current gen): 2.4-5 GHz, speed up to 300 Mbps, multiple I/O
- ac (emerging gen): 2.4-5 GHz, speed up to 500 Mbps, multiple I/O
- s: used for mesh networks
- p: used for vehicular networks

---

<sup>1</sup>Access Point

### 1.2.3 Satellite Systems

Satellites haven't been used so much until starlink which is gaining popularity because, even if they make light pollution, they are very lightweight and easy to wake up. There are many types of satellites:

- GEO (Geostationary Earth Orbit);
- MEO (Medium Earth Orbit);
- LEO (Low Earth Orbit).

In particular satellites:

- can cover large areas depending on their height in the space:
  - $> \text{height} \Rightarrow > \text{covered area}, > \text{latency}, < \text{bandwidth}$
  - $< \text{height} \Rightarrow < \text{covered area}, < \text{latency}, > \text{bandwidth}$
- for one-way transmission are optimised (i.e. radio and movie broadcasting);
- for two-way transmission are given up because of costs and few ambitions.

### 1.2.4 Bluetooth

Characteristics:

- it is a low cost replacement for cables;
- it covers a short range up to 100m with multihop
  - it requires exponential energy as distance grows
- frequency 2.4 GHz
- 4 channels (3 for voice, 1 for data up to 700 Kbps)
- Widely supported by telecommunications, PC...
  - it is a standard de facto (also BLE...)

### 1.2.5 Ad Hoc Networks

Characteristics:

- it is a peer-to-peer communications (born for military purposes)
- there isn't any backbone infrastructure
- routing is very hard because of:
  - dynamic topology;
  - typically multihop → to extend coverage area or reduce interferences

Problems:

- |                        |                         |
|------------------------|-------------------------|
| • hops;                | • energy consumption;   |
| • bandwidth;           | • topology;             |
| • collisions handling; | • dependency on device. |



### 1.2.6 Mesh Networks

Characteristics:

- Ad hoc opportunistic extension of a fixed urban infrastructure  
→ full of wireless access point which can connect to other ones
- it is easier than ANET because of almost static topology;
- creation of wireless coverage which is:
  - low-cost
  - easily deployable
  - high performing
- Challenges to face:
  - QoS
  - routing protocols optimisation for fairness and load balancing
  - automatic setup on infrastructure's failures

### 1.2.7 Sensor Networks

Characteristics:

- there is at least one sensor as device in the network;
- energy is the principal constraint (low or no battery)
- data flows to centralised locations;
- low per-node rate → up to 100K nodes and they can cooperate in:
  - ★ transmission
  - ★ reception
  - ★ compression
  - ★ signal processing

### 1.2.8 Distributed Control over Wireless Links

Characteristics:

- it is a possible scenario where there is control over something;
- it has to be robust to failures;
- Packet loss and delays impact controller performance;
- used mainly on automated vehicles such as cars, UAVs...

**1.2.9 Mobile Ad Hoc Networks (MANET)**

Characteristics:

- ANET with a dynamic topology using:
  - Infrastructure Network (WiFi or 3G/4G)
  - Ad Hoc Multihop wireless Network
- Instantly deployable and re-configurable (for temporary needs);
- Portable (i.e. sensors) and mobile (i.e. cars);

**1.2.10 Opportunistic Ad Hoc Networks**

Characteristics:

- they are created when needed;
- Driven by “commercial” application needs:
  - Indoor WLAN extended coverage
  - Bluetooth sharing
  - Peer-to-Peer networking on vehicles
- Access to internet available
  - BUT if too costly or inadequate ⇒ replacement with Ad Hoc Network

**1.2.11 Vehicular Ad Hoc Networks (VANET)**

Characteristics:

- ANET for vehicles
- it has 1000m range
- it supports 5.9 GHz
- it has 6-27 Mbps data rate depending on range
- it is more predictable → it may deduce infos ⇒ useful for crosslayers

**1.2.12 Flying Ad Hoc Networks (FANET)**

Characteristics:

- ANET for flying objects (i.e drone, mixed vehicles...)
- there is a 3D topology → protocols needs to be redesigned

**1.2.13 Underwater Sensor Networks**

Characteristics:

- communication happens by sound → messages propagate in circles;
- important to compute when message arrives → avoid collisions.

**1.2.14 Radio Frequency Identification (RFID)**

Characteristics:

- it is based on tags (low cost), readers (high cost) and eventually a server;
- tags can have:
  - ★ no battery → emitter charges the tag with energy (steal control,...)
  - ★ battery → tag periodically emits its ID (check of product history, control with sensors,...)
- systems can be built:
  - lot of tags + one emitter ⇒ cheap
  - lot of emitters + one tag ⇒ expensive
- it can identify specific instance of a product! (not only type like barcode)

## 2 Radio Frequency

Most wireless communications are based on this technology.

### 2.1 Properties

Here is some characteristics and properties of radio frequency:

- Antenna:
  - it has high frequency alternate current  $\Rightarrow$  generates electromagnetic energy
  - it converts wired current to radio frequency and viceversa
  - it can produce radio frequency with different frequency/amplitude
    - as signal propagates  $\Rightarrow$  it becomes weaker and weaker
- Frequency  $\rightarrow$  it is the number of waves in a second:
  - there is a wireless spectrum (regulated and free areas)
  - wavelength =  $\frac{c}{\text{freq}}$   $\Rightarrow$  distance between spikes
    - it gives antenna's recommended length
    - it works better if size is  $\frac{1}{2\pi}$  length of wavelength
- Amplitude:
  - higher amplitude signals  $\Rightarrow$  it goes further
  - transmission power =  $\frac{\text{energy}}{\text{time}} \rightarrow \frac{\text{joule}}{\text{s}}$
- Coverage:
  - as distance grows  $\Rightarrow$  signal becomes weaker in an exponential decline
    - you can detect a weak signal  $\rightarrow$  but you can't really use it  
(weak for exchanging messages)
  - problems:
    - \* obstacles  $\rightarrow$  can reflect or absorb waves
      - it depends on material and frequency
      - rules of thumbs
        - high frequency  $\rightarrow$  short distances, more affected by obstacles
        - low frequency  $\rightarrow$  long distances, less affected by obstacles
    - \* phase shifting  $\rightarrow$  positive/negative aspects  $\rightarrow$  early/late wavefront
      - signals can be null and overlap each other
  - polarisation  $\rightarrow$  physical orientation of antenna
    - \* radio frequency is made up of 2 perpendicular fields (electric/magnetic)  $\Rightarrow$  the presence of:
      - Horizontal polarisation  $\rightarrow$  electric field parallel to ground
      - Vertical polarisation  $\rightarrow$  electric field perpendicular to ground  $\rightarrow$  if 2 antennas are perpendicular to ground  $\Rightarrow$  better transmission

## 2.2 Wireless Transmission

It happens through electromagnetic waves. There is a dependency on amplitude, frequency and phase values → each combination produces a new signal

Characteristics:

- Range:
  - Transmission: communication possible, low error rate
  - Detection: detection of signal, no exchanging messages
  - Interference: no detection for too much noise depending from many factors (distance, environment...)

Detection requires more energy than communication

- Propagation:
  - it is at the light speed in free spaces
  - receiving power depends from distance between sender/receiver  
 $rp = \frac{1}{d^2} \rightarrow rp$  influenced by:
    - \* fading (dependent on frequency)
    - \* shadowing (obstacles)
    - \* reflection (large obstacles)
    - \* refraction (density of obstacles)
    - \* scattering (small obstacles)
    - \* diffraction (at edges)
  - signal can follow different paths due to refraction, scattering, diffraction.  
So there is:
    - \* Time dispersion → signal is dispersed over time
    - \* Phase shifting → signal is distorted
- Power measurement
  - It is the Decibel (dB) → expression power loss
  - It is more practical to use logarithmic decay → easy calculations
  - Decibel measures the logarithmic relative strength between 2 signals
  - Values of power measurements:
    - \* positive → power gain
    - \* negative → power loss

## 2.3 Antennas

Characteristics:

- it converts electrical energy in radio frequency waves (transmission) and viceversa (reception)
- its size → depends on radio frequency of transimission/reception
- its shape → depends on radio frequency radiation pattern
- position important to have max coverage

There are different types of antennas:

- Omnidirectional antennas
- Semi-directional antennas
- Highly-directional antennas
- Sectorised-directional antennas

### 2.3.1 Omnidirectional antennas

Characteristics:

- radio frequency power is equally distributed in all direction around Y-axis
- used when:
  - need of uniform radio coverage
  - point-to-multipoint connections (star topology)
- Tilt → it is degree of inclination of antenna with respect to Y-axis
- Example of dipole antenna
  - passive gain due to concentration of radiations
  - active gain obtained with power amplifiers
  - signal is weak near the dipole
  - there is also:
    - \* low gain → high signal near antenna, low far
    - \* high gain → low signal near antenna, high far

### 2.3.2 Semi-directional antennas

Characteristics:

- radio frequency power is equally distributed only on  $\frac{1}{2}$  direction (also few goes behind that direction)
- Types:
  - Patch → flat antennas mounted on walls
  - Panel → flat antennas mounted on walls
  - Yagi → rod with tines sticking out

### 2.3.3 Highly-directional antennas

Characteristics:

- radio frequency power is distributed on a specific direction and antenna could be as:
  - parabolic dish
  - grid
- it is used for long distances → point-to-point link
- there is what is called LoS (Line of Sight):
  - straight line between sender and receiver
  - needs no obstruction
- there is also the Freshnel Zone:
  - it is an area which is centered on LoS axis
  - most additive radio frequency signal is concentrated here
  - there is the need of no obstacles  
(useless increasing power if Freshnel Zone is not free)
  - it depends on distance and frequency
    - ⇒ there is no dependency from type, degree, gain of antennas

### 2.3.4 Sectorised-directional antennas

Characteristics:

- there are multiple antennas → each one points to a direction
- it is applied the space multiplexing (channel reuse)
  - ⇒ assigned the same frequency for antennas which do not collide each others

### 3 Wireless Physical Layer

There are different frequency areas which can be regulated or free.

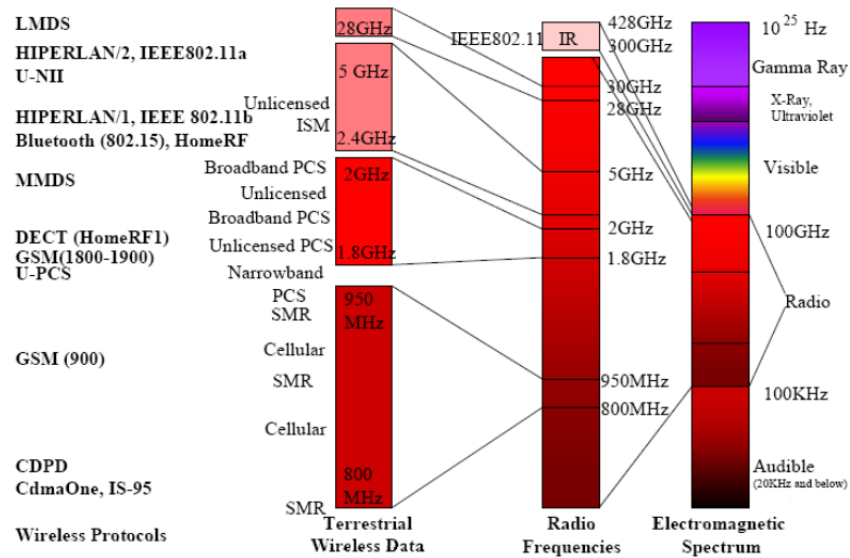


Figure 1: Wireless spectrum

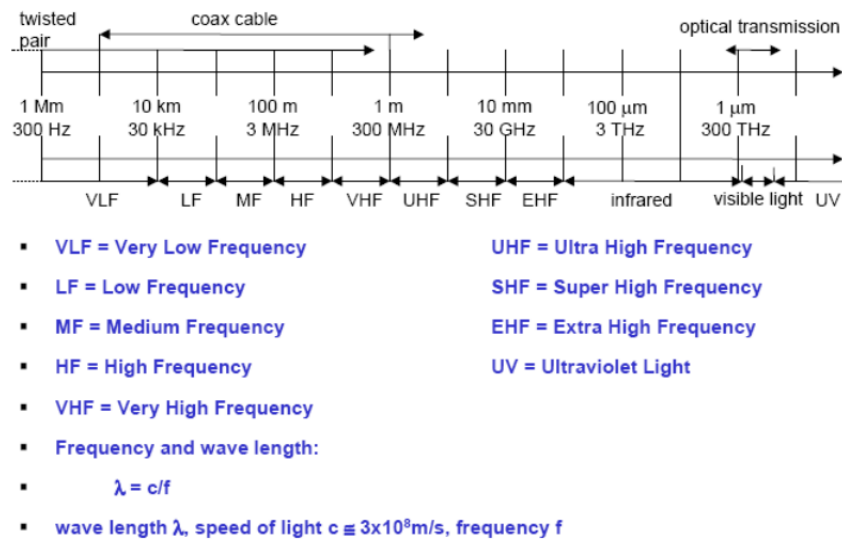


Figure 2: Wireless frequency



### 3.1 Characteristics

In this section there is some description about the main concepts of wireless physical layer

**Bandwidth** → maximum transfer capacity

- it can vary between each wireless channel
- bits go at the same speed (light physical limit) → gain in encoding/decoding
- spectrum can be bigger → more space ⇒ more risks (errors, interferences, ...)
- time to accomodate (less time, ...)

**Coverage**

- both isolated ⇒ they can't hear each others
- if A receives B, but B don't receive A ⇒ unidirectional link
- if A receives B and viceversa ⇒ bidirectional link
- Bidirectional links can be:
  - symmetric: A & B communicate with same speed
  - asymmetric: A & B communicate with different speed

**Technology**

There are different types of technologies used for wireless networks:

- Narrowband Radio System
  - used for long distance, LoS needed
  - send/receive using a single, licensed, narrowing radio frequency
  - cross-talks require coordination/license for each site (low rate)
- Spread Spectrum it can be of 2 types:
  - Frequency Hopping Spread Spectrum
    - \* it can changes frequency in the way which is known by the receiver/transmitter
    - \* unintended receivers may listen to FHSS<sup>2</sup> as impulse noise
    - \* lower power/cost/throughput than DSSS<sup>3</sup>
  - Direct Sequence Spread Spectrum
    - \* redundant bit pattern spreaded over a large spectrum
      - long chips can increase the possibility to recover the original bits
      - ⇒ it may avoid retransmission

---

<sup>2</sup>Frequency Hopping Spread Spectrum

<sup>3</sup>Direct Sequence Spread Spectrum

- \* unintended receivers may listen to DSSS<sup>4</sup> as low power wideband noise
- \* high performance, low interferences, good security, more expensive
- Infrared
  - it is just below visible light ⇒ it can't go beyond obstacles
  - LoS is the key (it limitates mobility) → short range (indoor, LANs, ...)
  - high data-rate potential
  - high bandwidth, easily obstructed, inexpensive

	PROS	CONS
Frequency Hopping Spread Spectrum (FHSS)	<ul style="list-style-type: none"> <li>• Use less power than DSSS</li> <li>• Lower cost</li> <li>• Increased security due to frequency switching</li> </ul>	<ul style="list-style-type: none"> <li>• Lower throughput than DSSS</li> </ul>
Direct Sequence Spread Spectrum (DSSS)	<ul style="list-style-type: none"> <li>• High performance</li> <li>• Low interference</li> <li>• Increased security due to chip coding</li> </ul>	<ul style="list-style-type: none"> <li>• Expensive</li> </ul>
Narrowband Microwave	<ul style="list-style-type: none"> <li>• Long distance</li> </ul>	<ul style="list-style-type: none"> <li>• Line-of-sight with satellite dish</li> <li>• Requires FCC license</li> <li>• Not designed for WLAN use</li> </ul>
Infrared	<ul style="list-style-type: none"> <li>• High bandwidth</li> </ul>	<ul style="list-style-type: none"> <li>• Easily obstructed</li> <li>• Inexpensive</li> </ul>

Figure 3: Wireless technologies comparison

### Coverage Areas

There are different coverage areas:

- Wireless Wide/Metropolitan Area Network (WWAN & WMAN)
 

It is characterised by the use of:

  - satellites
    - \* GEO → 3 of them cover the entire world → 500ms Round Trip Time
    - \* LEO → more mobility, low coverage → nodes have to switch between them
  - cellular/multistructure WLAN
    - \* lots of Access Point all connected to local Mobile terminals
    - \* local Mobile terminals connected to internet backbone

<sup>4</sup>Direct Sequence Spread Spectrum

- Wireless Local Area Network (WLAN)

It can be of 2 types:

→ Ad-Hoc

- \* it is a Peer-to-Peer "on the fly" communication
- \* there is no administration, no setup, no costs

→ Infrastructure

- \* it is a centralised control unit (Access Point + Local Server)
- \* there is roaming between cells
- \* there is resource sharing and backbone connection

- Wireless Personal Area Network (WPAN)

- it is used for alternative cable connection for in-home/offices
- common protocols are HomeRF, Bluetooth, ...

### Environment

There are some challenges to take into account:

- capability to maintain needs for apps/services
- limited resources such as bandwidth, energy (battery constraints) ...
- device limits (I/O, keyboards, mouse, ...)
- mobility (number of users in the system, ...)
- QoS<sup>5</sup> problems, reliability, negotiation...

### Multiplexing

- Goal → to reach the multiple use of a shared channel  
⇒ bandwidth to a large amount of devices
- There are multiple options and each one needs to have a guard spaces  
⇒ avoid interferences, ...
- Types:
  - Space Multiplexing:
    - \* devices are far away from each other
    - \* devices have all the same frequency → no interference
    - \* guard → safety physical space
  - Frequency Multiplexing:
    - \* channel's spectrum is divided into smaller bands
    - \* host use a single piece for the whole time

---

<sup>5</sup>Quality of Service → guarantee a certain amount of bandwidth

- \* guard → safety frequency between bands
- \* Pros:
  - no dynamic coordination
  - it works also for analog systems
- \* Cons:
  - inflexibility → traffic unbalanced ⇒ bandwidth waste

→ Time Multiplexing:

- \* one carrier (round-robin) at a time uses the whole bandwidth
- \* guard → time between transitions
- \* Pros:
  - high throughput for many users
- \* Cons:
  - require precise synchronization

→ Code Multiplexing:

- \* how it works:
  1. each channel has a unique code
  2. each medium transmits at the same time
  3. messages overlapping
  4. signal combination
  5. receiver decode only what of interest
- \* Pros:
  - no synchronization
  - more bandwidth
  - good protection in security/interferences
- \* Cons:
  - lower data rates
  - more expensive → it needs to regenerate the signal (receiver)

### 3.2 Wireless vs Wired

Here there is a comparison between wireless and wired networks.

Attribute	Wireless PAN/LAN	Wired PAN/LAN
Throughput	10-100 Mbps	10-100 Mbps (and more)
Integrity & Reliability	Subject to interference	Highly reliable
Simplicity/ Ease of Use	<ul style="list-style-type: none"><li>• No need to pull cable</li><li>• Set up time is significantly lower</li><li>• Moves, additions &amp; changes much simpler</li></ul>	<ul style="list-style-type: none"><li>• Cable required</li><li>• Set up time is significantly higher</li></ul>
Security	<ul style="list-style-type: none"><li>• Susceptible to interception</li><li>• Encryption</li></ul>	<ul style="list-style-type: none"><li>• Not as susceptible to interception</li></ul>
Cost	<ul style="list-style-type: none"><li>• Initial HW investment high</li><li>• Installation expenses and maintenance low</li></ul>	<ul style="list-style-type: none"><li>• Initial HW investment low</li><li>• Installation expenses and maintenance high</li></ul>
Scalability	Simple to complex networks	Simple to complex networks
Safety	Little exposure to radio frequency energy	No exposure to radio frequency energy
Mobility	Provides access to real-time info anywhere	Does not support mobility

Figure 4: Wireless vs wired comparison

## 4 MAC Layer

### 4.1 Introduction

Multiple Access Control (MAC) layer:

- it is a media access control protocol in which there is:
  - coordination and scheduling of transmissions
  - hosts competing for having the channel
- Access control
  - it is referred to shared channel
  - broadcast of wireless transmission (at the light of speed)
  - who can transmit when/where
  - collisions → avoid/recover from them with detection or not  
⇒ the problem is receiving at the same time (NOT SENDING)
- Goals:
  - low latency
  - good channel utilization (no collisions → using it as much as possible)
  - best effort + real time support

As in a human conversation:

- Everybody should have the chance to talk
- Do not speak until it is your turn
- Do not monopolize the conversation
- Raise your hand if you have to ask for something
- Do not interrupt while somebody is talking
- Do not fall asleep while somebody is talking

So the most important concepts are:

- efficiency in the bandwidth use → the maximum possible
  - resilience → avoid collisions
  - fairness → given  $n$  nodes and a bandwidth  $b$ , each one should have a bandwidth  $b_n = \frac{b_{tot}}{n}$
  - robustness → decentralised, no single point of failure
  - simplicity → easy to implement
- Channel Access Problem
    - there is a multiple nodes share channel  
⇒ simultaneous communication is not possible

- MAC protocols give schemes to schedule communication
  - \* maximise number of communication → avoid collisions
  - \* guarantee fairness among all transmitters
- trivial solution is Transmit and Pray
  - ⇒ plenty of collisions → poor throughput at high
- Carrier Sense Multiple Access (CSMA):
  - \* it provides a fix to Transmit and Pray
  - \* transmitters listen to the channel before sending → waiting when signal on channel
  - \* collisions:
    - can still occur due to propagation delay
    - when it happens the entire packet could be lost → time wasted

## 4.2 MAC Protocols

MAC protocol → coordinates transmissions from different stations  
⇒ minimize or avoid collisions

There are 3 different types of protocols:

- Channel partitioning (TDMA, FDMA, CDMA)
- Random Access (CSMA, MACA)
- Taking turns (polling)

Approaches to MAC layer are:

- Random Access:
  - Without carrier sensing → Pure Aloha, Slotted Aloha
  - With carrier sensing → CSMA, CSMA/CD, MACAW
- Controlled Access:
  - Centralized → entity regulate channel's access (FDMA, TDMA, CDMA)
  - Distributed → distributed apps with peer nodes regulate channel's access (Token ring)

### Random Access Protocols

Characteristics:

- node transmits at random at full channel data rate
- if nodes collide then they retransmit at random times
- each one detects/recovers from collision in a different way

Here there is the description of the most important protocols.

### 4.2.1 Slotted Aloha

Characteristics:

- time is divided into equal size slots → equal to full packet size
- newly arriving station transmits at the beginning of the next slot
- if collision occurs:
  - assumption of the presence of channel feedback
  - retransmission of packet at each slot with probability  $P$ , until successful
- Successful of transmission:
 

given:

  - $N$  = number of stations
  - $P$  = probability that each station transmits in the slot
  - $S$  = probability of successful of transmission

the value of  $S$  is:

  - $S = p(1-p)^{(N-1)}$  by a single node
  - $S = Np(1-p)^{(N-1)}$  by any of  $N$  nodes
- throughput efficiency is about  $\frac{1}{e}$  → and:
  1. obtaining  $p = \frac{1}{N}$  ( $p$  should be tailored based on  $N$ )
  2. substituting  $p$  to  $S = Np(1-p)^{(N-1)} \Rightarrow S = N \frac{1}{N} (1 - \frac{1}{N})^{(N-1)}$
  3. solving  $S$  at the limit obtaining  $S = \frac{1}{e}$
- it is fully decentralised

### 4.2.2 Pure Aloha

Characteristics:

- it doesn't require time slots → no synchronization
- nodes can transmit at any time  $\Rightarrow$  collision may increase
- Successful of transmission:
 

given:

  - $N$  = number of stations
  - $P$  = probability that each station transmits in the slot
  - $S$  = probability of successful of transmission

the value of  $S$  is:

- $S = p(1-p)^{2(N-1)}$  by a single node
- $S = Np(1-p)^{2(N-1)}$  by any of  $N$  nodes



- throughput efficiency is about  $\frac{1}{2e}$  → every transmission can occupy 2 slots  
→ and:
  1. obtaining  $p = \frac{1}{2(N-1)}$  (p should be tailored based on N)
  2. substituting p to  $S = Np(1-p)^{2(N-1)} \Rightarrow S = N \frac{1}{2(N-1)} (1 - \frac{1}{2(N-1)})^{2(N-1)}$
  3. solving S at the limit obtaining  $S = \frac{1}{2e}$

#### 4.2.3 Considerations Pure & Slotted Aloha

Both are:

- not efficient at all → a lot of retransmissions:
  - ★ Pure Aloha throughput → 18.4 %
  - ★ Slotted Aloha throughput → 36.8 %
- unfair → aggressive senders can capture the channel
- robust → decentralized
- simple:
  - ★ Pure Aloha → no coordination
  - ★ Slotted Aloha → just synchronization

#### 4.2.4 Carrier Sense Multiple Access (CSMA)

Characteristics:

- Aloha protocols are less performing → lack of coordination among nodes
- Each node continuously listens to channel → awareness of channel's freedom  
⇒ improve in efficiency

There are different types of CSMA:

- 1-persistent CSMA
- non-persistent CSMA
- p-persistent CSMA

##### 4.2.4.1 1-persistent CSMA

Characteristics:

- how it works:
  1. nodes listen to the channel
  2. the channel can be:
    - free → immediate transmission
    - busy → waiting until channel is free →  $P_R = 1$

$\Rightarrow P_R$  = probability of retransmission  
(if there is a collision  $\rightarrow$  node waits for a random time and  
retries  $\Rightarrow$  desynchronization)

- propagation time
  - $\rightarrow$  impact on performance
  - $\rightarrow$  more time  $\Rightarrow$  more collisions
  - Example:  
A can't hear B  $\rightarrow$  B is transmitting for so much time and A want to transmit  
 $\rightarrow$  channel is free for A but it is not  $\Rightarrow$  collision
  - $\rightarrow$  even with no propagation time
  - Example:  
if two nodes transmit and a third is occupying the channel  
 $\rightarrow$  when channel is free  $\rightarrow$  all 2 transmit at same time  $\Rightarrow$  collision

#### 4.2.4.2 Non-persistent CSMA

Characteristics:

- how it works:
  1. nodes listen to the channel
  2. the channel can be:
    - $\rightarrow$  free  $\rightarrow$  immediate transmission
    - $\rightarrow$  busy  $\rightarrow$  waiting a random time and then retry to listen
- it is less aggressive than 1-persistent CSMA

#### 4.2.4.3 P-persistent CSMA

Characteristics:

- it is slot based
- how it works:
  1. nodes listen to the channel
  2. the channel can be:
    - $\rightarrow$  free  $\rightarrow$  transmission with probability  $p$
    - $\rightarrow$  busy  $\rightarrow$  wait with probability  $(1-p)$  and then retry to listen
- Aggressiveness:
  - $\rightarrow$  it depends on  $p$
  - $\rightarrow$   $p$  can be choose depending to the number of nodes:
    - \* many  $\rightarrow$  it may be conservative  
 $\Rightarrow$  bandwidth waste depending on number of collisions
    - \* few  $\rightarrow$  it may be aggressive  
 $\Rightarrow$  bandwidth waste depending on time of channel not used

#### 4.2.4.4 CSMA with Collision Detection (CSMA/CD)

Characteristics:

- it is like CSMA → but collisions are detected within few bit times
- when it is detected → transmission aborted ⇒ reduction of channel wastage
- transmission is typically implemented persistently
- collision detection can approach channel utilization = 1 in LANs  
→ it can detect immediately if something is wrong
- easy detection in wired LANs → it can measure signal strength  
→ on the line, or code violations, ...
- collision detection can't be done in wireless LANs  
Example:  
the receiver shut off while transmitting → avoid damaging it with excess power

#### 4.2.5 Wireless Medium Access Control

Some basics:

- transmission strength drops exponentially as the distance grows
- there is the SINR (⇒ signal interface noise ratio)  
→ calculation of how strong is signal compared to interference
- SINR threshold → bound in which signal can't be detected anymore  
→ C can hear but not receive correctly the message  
→ A can't send/listen at the same time  
Example:  
collision with A → D may not; B/C maybe
- Collision detection:
  - A is out of C range (and vicerversa), both have B on range  
→ they send together ⇒ collision at receiver B
  - Both:
    - \* don't know their position
    - \* can only hear themselves → no listening while transmitting
    - \* can't determine the signal quality at receiver
  - Two problems:
    - \* Hidden Terminal Problem → nodes don't know the topology  
→ they transmit to same node without knowing other's transmission  
⇒ collision
    - \* Exposed Terminal Problem → nodes are in each other range  
→ transmission to different receivers → even if channel is free  
→ one don't transmit thinking channel is busy

### 4.3 802.11 Protocol

Characteristics:

- it is a standard ratified in 1999
- aim is to have a common way to allow communication between nodes
- it is used in WLAN indoors with various version (n/ac/...)
  - frequency used are unlicensed → 2.4/5 GHz, 900 MHz
- WLAN configuration with access point or ad-hoc network
- it defines both MAC and physical layer (radio frequency, header, size, ack, ...)
- definition of some terms:
  - DIFS (Distributed Inter Frame Space): time waiting for channel to be free
    - if channel is occupied ⇒ timer restarts from the beginning (low priority → for asynchronous data service)
  - SIFS (Short Inter Frame Space): time useful to process some procedures like:
    - \* CTS (Clear To Send)
    - \* Datas
    - \* ACK (Acknowledgement)
  - ⇒ high priority
  - CW (Contention Window):
    - slots to be waited after a successful DIFS → 1 slot = 1 SIFS
    - if channel is occupied ⇒ timer stops and restart from there
    - if there is a collision → CW is increased at max 1024 slots
    - when succeeding → CW reset to the value
  - NAV (Network Allocation Vector): time the sender declares to hold the medium → other nodes can go to sleep ⇒ sparing energy

There are different versions of 802.11 protocol and for accessing to MAC layer which are explained in the next sections.

Access methods are:

- MAC-DCF CSMA/CA:
  - collision avoidance via randomized back-off mechanism
  - minimum distance between consecutive packets
  - ACK packet for acknowledgements (not for broadcasts)
- MAC-DCF CSMA/CA with RTS/CTS
  - Distributed Foundation Wireless MAC
  - avoids hidden terminal problem
- MAC- PCF
  - access point polls terminals according to a list

#### 4.3.1 CSMA version

- nodes listen before transmitting
- if the channel is:
  - free → node begins to transmit datas
  - busy → NAV defer access to medium
    - if it was in:
      - \* DIFS → it restarts the timer
      - \* CW → it stops and restarts at same timer position when newly free
- receiver returns to emitter ACK after SIFS amount of time
- how it works:
  1. transmitter waits a DIFS time ( $\approx 16\mu s$ )
  2. if the channel is:
    - busy → it restarts the DIFS counter (point 1.)
    - free → it goes to CW (point 3.)
  3. transmitter waits in the CW
  4. if the channel is:
    - busy → it stops the timer and restarts at the same time of CW
    - free → it begins to transmit (point 5.)
  5. transmitter sends datas (for a max of  $33\mu s$ )
  6. receiver sends back an ACK after SIFS time ( $9\mu s$ )
    - automatic retransmission of packets in case of transmission errors
- it is subjected to the hidden terminal problem  $\Rightarrow$  use of CSMA/CA

#### 4.3.2 CSMA/CA version

It works similarly as CSMA, but it improves hidden terminal problem (not solved).

- it uses the collision avoidance (CA).

In particular it adds:

  - RTS (Request To Send) → it freezes stations near the transmitter
  - CTS (Clear To Send) → freezes stations near the receiver
    - station could be possibly hidden from transmitter
    - $\Rightarrow$  this prevents collisions by hidden station during data transfer

$\Rightarrow$  RTS and CTS are very short → collisions during data phase are very unlikely
- nodes listen before transmitting

- if the channel is:
  - free → nodes begin to send the RTS/CTS and datas
  - busy → NAV defer access to medium
    - if it was in:
      - \* DIFS → it restarts the timer
      - \* CW → it stops and restarts at same timer position when newly free
- receiver returns to emitter ACK after SIFS amount of time
- how it works:
  1. transmitter waits a DIFS time
  2. if the channel is:
    - busy → it restarts the DIFS counter (point 1.)
    - free → it goes to CW (point 3.)
  3. transmitter waits in the CW
  4. if the channel is:
    - busy → it stops the timer and restarts at the same time of CW
    - free → it begins to transmit (point 5.)
  5. transmitter sends a RTS to receiver
  6. receiver sends back a CTS to transmitter after SIFS time
  7. transmitter sends data after SIFS time
  8. receiver sends back an ACK after SIFS time
    - automatic retransmission of packets in case of transmission errors

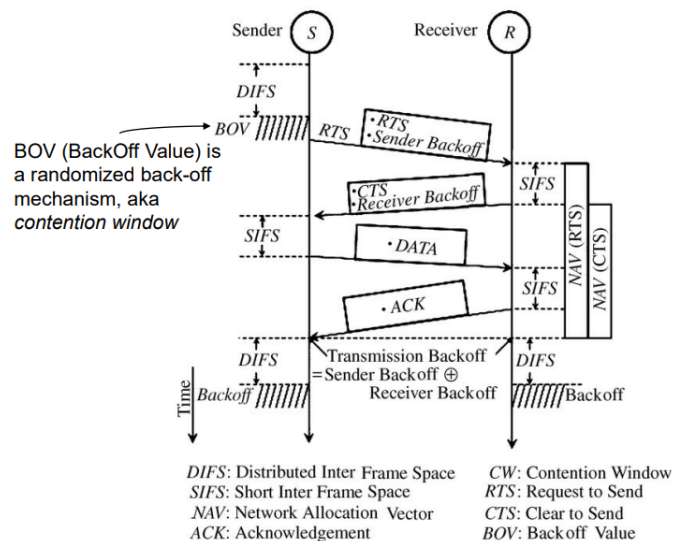


Figure 5: CSMA/CA

### 4.3.3 Point Coordinating Function (PCF)/Polling version

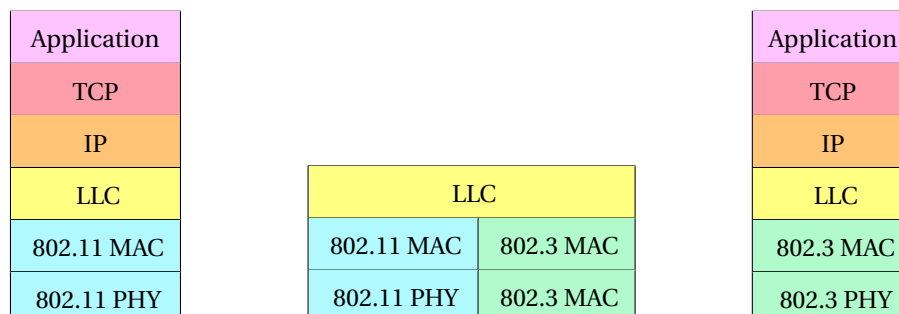
Characteristics:

- Access Point has complete control over transmissions
- it asks to each node if it has something to transmit → managed by round-robin  
⇒ no collisions, ok with many nodes
- how it works:
  1. AP announces if it supports PCF in the beacon
  2. The AP periodically broadcasts beacons
  3. Nodes use these beacons to learn about AP
  4. The node and the AP authenticate each other:
    - (a) node associates with that AP
    - (b) node sends an association request management frame  
→ here node announces to the AP if:
      - ★ pollable
      - ★ capable to transmit during the contention free period (CFP)
    - (c) AP replies with an association response

### 4.3.4 Considerations CSMA

Some considerations about CSMA:

- exposed terminal problem → it is not improved → with CSMA is even worse  
→ enlarge the detection ⇒ there are potentially more nodes
- it is always better to use RTS/CTS on CSMA/CA and polling versions because:
  - more bandwidth consumed → but probability of collision is smaller  
⇒ it is an improvement at the cost of limited overhead of transmission
  - if data packets are very small (as RTS ones) → it is worse
- positioning → can deal with different MAC layers (802.11, 802.3)



#### 4.3.5 Synchronization in 802.11

Characteristics:

- AP send beacons in infrastructure networks
- beacons are scheduled in beacon interval → transmission may be delayed by CSMA deferral
- timestamp contains timer value at transmit time
- Power Management approach:
  - allow idle stations to go to sleep → save mode stored in AP
  - AP buffers packets for sleeping stations → AP announces which station has packets buffered → message is sent with TIM<sup>6</sup> interval
  - power saving station wake up periodically to listen to beacons
  - TSF<sup>7</sup> assures AP and power saving stations are synchronized
  - there is also dTIM → less frequently → stations give priority to dTIM → used for broadcasting/multicasting
- Scanning:
  - used for
    - \* finding/joining networks
    - \* finding a new AP while roaming
    - \* initialising a new ad hoc network
  - MAC layer uses common mechanism for all physical
  - it could be:
    - \* active → it looks explicitly for AP  
On each channel ⇒  
send a probe → wait for probe response  
send an association request → wait for association request response
    - \* passive → only listen for beacons

#### 4.3.6 Congestion Avoidance (DCF)

Characteristics:

- how it works actually:
  - after DIFS → randomly choose a backoff time interval
  - Countdown the backoff interval when medium is free
    - it goes stand-by if medium becomes busy in the range [0,CW]
  - When backoff time interval reaches 0 → transmit packet (or RTS)

---

<sup>6</sup>Traffic Indicator Map

<sup>7</sup>Timing Synchronization Function



- Congestion control → dynamically adjusting the CW
- Counting down backoff time intervals contributes to MAC overhead
- Binary Exponential Backoff → a node fails to receive CTS  
⇒ it double up the CW (typically max size is 1024)  
→ when node successfully completes transfer ⇒ it restores CW to  $CW_{min}$
- So about the dimension of CW:
  - Large CW ⇒ large backoff time intervals ⇒ can result in larger overhead
  - Small CW ⇒ probabilistically to a larger number of collisions

#### 4.3.7 MILD Algorithm MACAW

Characteristics:

- node fails to receive CTS → it multiplies CW by 1.5  
⇒ less aggressive than 802.11 which multiplies by 2
- node successfully completes a transfer → it reduces CW by 1  
⇒ more conservative than 802.11 where CW is restored to  $CW_{min}$
- 802.11 reduces cw much faster than it increases it
- MACAW: cw reduction slower than the increase  
⇒ Exponential Increase and Linear Decrease
- MACAW can avoid wild oscillations of CW when congestion is high

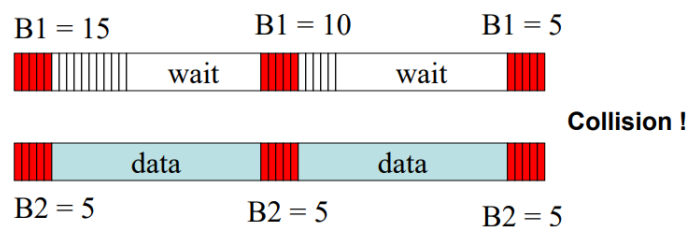
#### 4.3.8 Fairness Issue

Characteristics:

- Definition:
  - nodes should receive equal bandwidth
  - bandwidth should be tailored to how much they want to transmit  
⇒ otherwise waste of time/bandwidth
- unfairness → one node has backed off much more than some other node  
( $\approx$  channel dominance)  
→ A transmits many packets before B is transmitting its first
- how MACAW tries to solve:
  1. a node transmits a packet → it appends on packet its current CW value
  2. All nodes hear CW value → use it for their future transmission attempts
  3. The effect is to reset all competing nodes to the same level
- Weighted Fair Queuing → it is assigned a weight to each node  
⇒ bandwidth used by each node → proportional to the weight assigned

- Distributed Fair Scheduling (DFS)

- it is fully distributed algorithm for achieving weighted fair queueing
- it works well on a LAN
- how it works:
  1. Chooses backoff intervals proportional to packet size/weight
  2. DFS attempts to follow the centralized Self-Clocked Fair Queueing



Weight of node 1 = 1      B1 = 15 (DFS actually picks a random value with mean 15)  
 Weight of node 2 = 3

Assume equal packet size      B2 = 5 (DFS picks a value with mean 5)

Figure 6: Distributed Fair Scheduling (DFS)

## 5 Network Layer

It is difficult to have a direct path to destination (in ad hoc networks)

Routing protocol → goal is having a good path from source to destination

⇒ good can be:

- the shortest
- the less expensive
- the fastest

### 5.1 Routing algorithm

Characteristics:

- Graph abstraction for algorithms:
  - Nodes → routers
  - Edges → physical links
- Classification:
  - information can be:
    - \* Global → all router have complete topology  
⇒ Ex: link state algorithm
    - \* Decentralised → routers know topology only of their neighbours  
→ routing table build when needed ⇒ exchange of information  
⇒ Ex: Distance vector algorithm
  - algorithm can be:
    - \* Static → topology fixed all the time → it changes slowly
    - \* Dynamic → topology changes frequently → periodic update of nodes
- Example of link state algorithm ⇒ Dijkstra algorithm:
  - net topology → link costs known to all nodes
    - \* accomplished via link state broadcasting
    - \* all nodes have same info
  - it computes least cost paths from one node to all the others  
⇒ it gives forwarding table for that node
  - how it works:

Notation:

    - \*  $c(x, y)$ : link cost from node x to y; initially =  $+\infty$
    - \*  $D(v)$ : current value of cost of path from source to destination v
    - \*  $p(v)$ : predecessor node along path from source to destination v
    - \*  $N'$ : set of node whose least cost path is known

Algorithm:

```

1: Initialization:
2:  $N' = \{u\}$ 
3: for all nodes  $v$  do
4:   if  $v$  is adjacent to  $u$  then
5:      $D(v) = c(u, v)$ 
6:   else
7:      $D(v) = +\infty$ 
8:   end if
9: end for
10:
11: Iterations:
12: repeat
13:   find  $w \notin N' \mid D(w)$  is a minimum
14:   add  $w$  to  $N'$ 
15:   for all  $v$  adjacent to  $w$  &  $v \notin N'$  do
16:      $D(v) = \min(D(v), D(w) + c(w, v))$  //update  $D(v)$ 
17:   end for
18: until all nodes  $\in N'$ 

```

## 5.2 Infrastructure Network vs MANET

Differences:

<p>Infrastructure Network</p> <ul style="list-style-type: none"> <li>• AP/base stations define cells/areas</li> <li>• simple routing → one hop from AP to wireless node</li> </ul> <p>Wired</p> <ul style="list-style-type: none"> <li>• symmetric link → bidirectional</li> <li>• limited redundancy → for reliability and load sharing</li> <li>• planned links → high QoS, fixed topology</li> </ul>	<p>MANET</p> <ul style="list-style-type: none"> <li>• no infrastructure → no static network</li> <li>• hard routing → some nodes are not close ⇒ someone must forward data</li> </ul> <p>Wireless</p> <ul style="list-style-type: none"> <li>• asymmetric link → unidirectional</li> <li>• random redundancy → in connectivity between nodes</li> <li>• dynamic links → dynamic topology, interference</li> </ul>
---	---

⇒ Traditional Routing Algorithms → doesn't work well for MANET

This is because MANETs:

- have dynamic topology → nodes leave and enter quickly
- have minor performance due:
  - energy consumption → incompatibility with periodic updates
  - limited bandwidth → even more to exchange routing info

- asymmetric links → info about link + quality of link  
→ different costs depending on the direction
- are inefficient → slow convergence time
- are not robust enough
- are non-functional → large amount of data not dealing with asymmetric links
- could have path length (hop count) → which may not be the best metric
- need routing which rely on data link (not just only network layer updates)  
→ this determines connectivity + quality of links

So for a good Unicast Routing Protocol these are all goals:

- minimal control/processing overhead
- multi-hop path routing
- self-starting
- energy constraints
- traffic patterns → inform nodes on what a node is going to do
- dynamic topology maintenance
- no loops
- no central authority

### 5.3 Routing Procols Classification

Routing protocols can be classified into 3 different categories:

- Proactive (table-driven):
  - routing information always up-to-date
  - routing overhead independent of route usage
  - used for not so much dynamic networks
- Reactive (Source-initiated):
  - routes maintained only for routes in use ⇒ less overhead
  - explicit route discovery mechanism
- Hybrid
  - combination of proactive and reactive

#### 5.3.1 Proactive Routing Approach

Characteristics:

- based on link-state/distance-vector protocols
- each node knows to path to reach another node thanks to consistent tables
- there is periodic/event-driven routing updates
- routing updates ⇒ more overhead
- if ad-hoc network is really dynamic ⇒ most of updates are not used at all
- low latency → because route is known
- protocols differ on the method to propagate informations  
→ every node use a unique ID

- high route convergence time
- Examples: DSDV, WRP ...

Here there are some proactive routing protocols.

#### 5.3.1.1 Destination Sequenced Distance Vector (DSDV)

Characteristics:

- based on Bellman-Ford algorithm
- use sequence number to avoid loops → new nodes ⇒ higher sequence number
- optimizations → as incremental data exchange, delayed exchange of updates
- packets are transmitted according to routing table
- each node:
  - maintains a routing table with entry for each node in the network  
→ <dest\_addr; dest\_seq\_number; next\_hop; hop\_count; install\_time>
  - maintains its own sequence number ⇒ so it:
    - \* updates if neighbour informations are changed
    - \* avoid loops
    - \* distinguish new routers
- updates:
  - are periodic to keep consistency → done with the inclusion of  
<dest\_addr; dest\_seq\_num; hop\_count> ⇒ lot of overhead
  - nodes send routing tables for important link changes → ex: link broken
  - if node receive routes from 2 different neighbours → it can be chosen:
    - \* the one with higher dest\_seq\_num
    - \* the one with lower hop count
  - lots of control traffic is created → so there are 2 types of routing update packets:
    - \* Full dumps:
      - it carries all routing table information
      - transmitted relatively infrequently
    - \* Incremental updates:
      - it carries only information changed since last full dump
      - it fits within one network protocol data unit (NPDU)
      - when it doesn't fit ⇒ full dump

### 5.3.1.2 Optimized Link State Routing (OLSR)

Characteristics:

- based on state-link algorithm
- good for large and dense networks
- all links with neighboring Mobile Hosts → declared and flooded in entire network
- it is done periodic control of messages sent
- use of traffic patterns
- it minimizes flooding of control traffic using only selected Mobile Hosts
  - ⇒ Multipoint Relays:
    - it minimizes the flooding of broadcast packets in the network
    - each Mobile Host select a set of neighboring Mobile Host to retransmit
      - ⇒ reduction of duplicate retransmissions in the same region
    - this set can change over time → indicated by Hello messages
    - typically are selected neighbour at 1 hop

### 5.3.1.3 Clusterhead Gateway Switch Routing (CGSR)

Characteristics:

- nodes organised in hierarchy:
  - Clusterhead → selected by election
  - Gateway → it connects clusterhead
  - how it works:
    1. Nodes send packet through clusterheads
    2. Clusterheads communicate among themselves using DSDV
      - two clusters are connected through a gateway node

### 5.3.2 Reactive Routing Approach

Characteristics:

- source builds routing on demand by flooding
  - nodes receive and broadcast again → route discovery cycle
- it maintains only active routes
- pro → less control overhead ⇒ better scaling
- cons → latency or long delay in finding route  
RFC<sup>8</sup> 3561 not suitable for real-time traffic

---

<sup>8</sup>Request For Comments

- route discovery by flooding → every node propagates the message  
⇒ assumption of algorithms → sender need to know the existence of receiver
- route maintenance procedure is used to repair routes
- Example: AODV, DRS ...

### 5.3.2.1 Ad Hoc On-Demand Distance Vector Routing (AODV)

Characteristics:

- RFC 3561 → based on DSDV
- dest\_seq\_num avoids loops
- routing table only exchanges with nodes of a given route
- how it finds a route:
  1. source sends Route Request Packet (RREQ)  
→ many routes can be found, but one implemented
  2. nodes floods it to neighbours
  3. Route Reply Packet (RREP) is sent back by destination  
⇒ but nodes:
    - respond the first time they receive the request
    - reply only if they have contact/valid route with destination
  4. Route Error messages update routes
- if routes are not used ⇒ they expire and get discarded
  - it reduces obsolete routes ⇒ it doesn't require explicit route maintenance
  - it minimizes routes from source to destination
- it discovers routes as and when necessary  
→ path maintained only in involved nodes (routing table)
- routes are maintained as long as necessary
- nodes have a unique dest\_seq\_num  
⇒ it is increased for every change in neighbourhood topology
- it utilizes a routing table to store routing informations → there are 2 types:
  - for unicast routes → path from source to destination
  - for multicast routes → flooding  
→ it is a problem if a node receives all at once ⇒ collision and congestion
- route table stores → <dest\_add; next\_hop\_add; dest\_seq\_num; life\_time> →  
life\_time → used as expiring time → updated each time route is used
- each node maintains list of precursor nodes for each destination to route to  
⇒ help in route maintenance



- how route discovery works ( $\Rightarrow$  node wishes to send a packet):
  1. it checks if destination is in the routing table:
    - $\rightarrow$  Yes  $\rightarrow$  it forwards the packet to the next hop;
    - $\rightarrow$  No  $\rightarrow$  it initiates a route discovery process;
  2. source node creates a new Route Request (RREQ) packet
    - $\rightarrow$  which contains:
      - ★ source\_IP\_add
      - ★ source\_curr\_seq\_num
      - ★ dest\_IP\_add
      - ★ dest\_seq\_num
      - ★ broad\_ID
    - $\rightarrow$  broad\_ID incremented each time a source node uses RREQ
    - $\rightarrow$  broad\_ID + dest\_IP\_add = unique ID for RREQ
  3. Broadcasting is done via flooding
  4. when intermediate node receives RREQ
    - $\Rightarrow$  node sets up reverse route entry for source node in its routing table
    - $\rightarrow$  Reverse Route Entry consists of:
      - ★ source\_IP\_add
      - ★ source\_seq\_num
      - ★ num\_hops\_to\_source
      - ★ pred\_IP\_add
      - ★ life\_time
    - $\rightarrow$  nodes can send RREP using reverse route
  5. when RREQ reaches destination  $\Rightarrow$  to respond it should have:
    - $\rightarrow$  unexpired entry for destination
    - $\rightarrow$  dest\_seq\_num  $\geq$  seq\_num\_RREQ (for loop prevention)
  6. there are 2 scenarios depending on satisfaction of conditions in last point. If they are:
    - $\rightarrow$  satisfied  $\Rightarrow$  node responds sending a RREP back to source using unicasting and reverse path
    - $\rightarrow$  not satisfied  $\Rightarrow$  node increments hop count in RREQ and floods to neighbours

#### Observation

Also intermediate node can also RREP

- $\rightarrow$  if it knows a more recent path  $\Rightarrow$  dest\_seq\_num newer
- $\rightarrow$  not used so much  $\Rightarrow$  because: new RREQ  $\Rightarrow$  new dest\_seq\_num
- $\Rightarrow$  intermediate dest\_seq\_num < new dest\_seq\_num  $\Rightarrow$  it can't send RREP

- Timeouts:

Routing table entry keeping a reverse path is deleted:

- after a timeout  $\Rightarrow$  it should be long enough to allow RREP to come back
- if it is not used for `active_route_timeout` interval
  - if no data is being sent  $\Rightarrow$  that entry will be deleted even if it is valid

- Link Failure Detection:

- neighboring nodes periodically exchange "Hello" messages
- absence of message  $\Rightarrow$  link failure
- MAC level ACKs missing  $\Rightarrow$  link failure

- Optimizations:

- RREQ are sent with a TTL<sup>9</sup> with hop count  $\Rightarrow$  limit flooding propagation
- initially small TTL  $\Rightarrow$  then larger if RREP is not received by source
- Expanding Ring Search:
  - \* it prevents flooding of network during route discovery
  - \* it controls TTL of RREQ to search incrementally larger areas
  - \* Advantages  $\rightarrow$  less overhead when successful
  - \* Disadvantages  $\rightarrow$  longer delay if route isn't found immediately:
    - there could be many possible destinations
    - it would be better without optimization
    - maybe it doesn't find the optimal path

### 5.3.2.2 Dynamic Source Routing (DSR)

Characteristics:

- each data packet has full source route and provokes overhead
- RREQ has attached full source-route and is sent back in RREP
- route table overhead is only at source node
- route discovery similar to AODV:
  1. sender initiate request
  2. intermediate nodes add their address onto request
  3. when request reaches destination  $\Rightarrow$  it includes the full path
- it is better in terms of mobility

---

<sup>9</sup>Time To Leave

### 5.3.2.3 Associativity-Based Routing (ABR)

Characteristics:

- it defines degree of association stability as metric instead of number of hops
- nodes with less mobility/better links → they have higher stability value
- it is DSR-like protocol for routing

### 5.3.2.4 Signal Stability Routing (SSR)

Characteristics:

- it defines signal strength of links as metric
- RREQ is forwarded only if packet is received over a link with good signal
- it is DSR-like protocol for routing

### 5.3.2.5 Other metrics definition

- Expected Transmission Time (ETT) → time to reach a node ⇒ easy to compute  
→ more useful than signal strength
- Weighted Cumulative Expected Transmission Time (WCETT)  
→ better for multi-radio/asymmetric links

### 5.3.2.6 Flooding

Characteristics:

- used for Control Packet Delivering
- how it works:
  1. sender sends a broadcast packet to all its neighbours → to find a route
  2. neighbours forward it  
→ if seq\_num no already seen ⇒ it avoids packet resending
  3. destination doesn't forward it  
→ if destination receives packet ⇒ sender can reach the destination
- Pro:
  - simple and more efficient when there is:
    - \* high dynamic topology
    - \* small data packet are sent infrequently
  - higher reliability of data delivery → multiple paths to reach destination
- Cons:
  - potential high overhead → nodes can receive packets multiple times
  - lower reliability of data delivery → broadcasting hard to implement without overhead increasing
  - potential packet loss → more nodes send same packet to same destination simultaneously

### 5.3.3 DSDV vs AODV

Differences:

<b>DSDV</b>	<b>AODV</b>
<ul style="list-style-type: none"> <li>• every change is broadcasted</li> <li>• new neighbours link <math>\Rightarrow</math> news is broadcasted</li> <li>• new broken link <math>\Rightarrow</math> news is broadcasted</li> <li>• more control overhead (high)</li> <li>• local movements <math>\Rightarrow</math> global effects</li> </ul>	<ul style="list-style-type: none"> <li>• no broadcasting necessary</li> <li>• only affected nodes are informed</li> <li>• new broken link <math>\Rightarrow</math> no global broadcasting</li> <li>• less control overhead (low)</li> <li>• local movements <math>\Rightarrow</math> local effects</li> </ul>

### 5.3.4 DSR vs AODV

Differences:

<b>DSR</b>	<b>AODV</b>
<ul style="list-style-type: none"> <li>• source route in packet headers <math>\Rightarrow</math> especially for small packets</li> <li>• routing table maintained only on source</li> <li>• lifetime for routes discovery</li> </ul>	<ul style="list-style-type: none"> <li>• only predecessor on each packet</li> <li>• routing tables maintained also on nodes</li> <li>• no lifetime for routes discovery</li> </ul>

### 5.3.5 Other Routing Protocols

There are other protocols:

- Hybrid:
  - $\rightarrow$  Zone Routing Protocol (ZRP)
- Geographic Routing Protocols:
  - $\rightarrow$  Location Aided Routing (LAR)
  - $\rightarrow$  Distance Routing Effect Algorithm for Mobility (DREAM)
  - $\rightarrow$  Greedy Perimeter Stateless Routing (GPSR)

#### 5.3.5.1 Greedy Perimeter Stateless Routing (GPSR)

Characteristics:

- nodes know the destination/neighbours location
- each node forwards a packet to its neighbor closest to the destination
  - $\rightarrow$  using some greedy, local optima algorithms
- There is a problem  $\rightarrow$  the risk of local minima ( $\Rightarrow$  two nodes keep exchanging the same packet)

- if routing holes are found  $\Rightarrow$  it uses perimeter routing (right hand rule)  
→ to find local minimum
- Right Hand Rule:
  - use of first 3 fingers
  - it works only in 2D topology
  - how it works:
    1. there is an imaginary line from source to destination
    2. the packet is sent to the first node counterclockwise

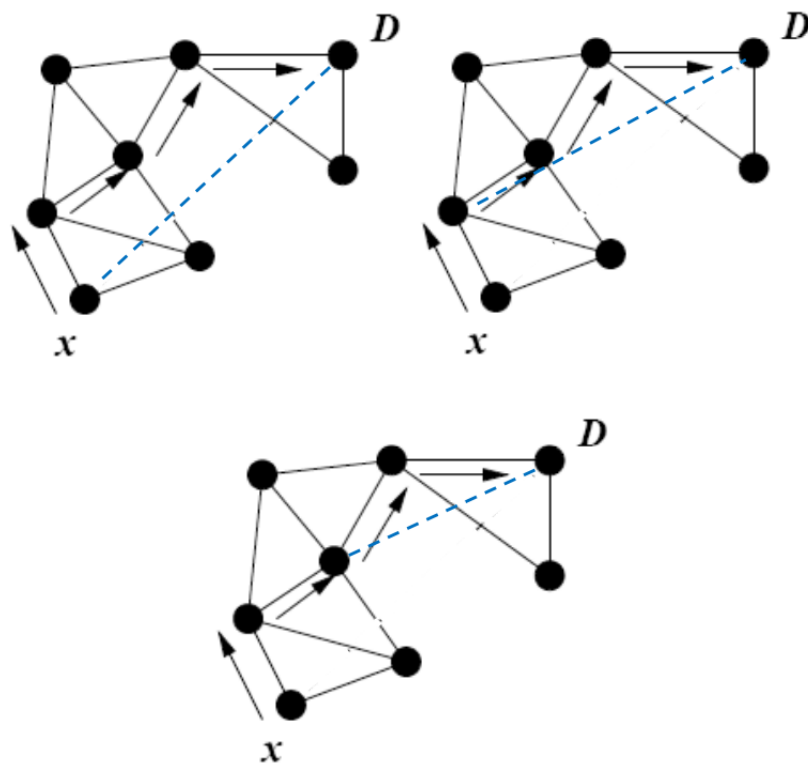


Figure 7: Example: Right Hand Rule

## 6 Transport Layer

### 6.1 Overview

- it is after the network layer → receiver transport layer communicates with the analogue sender transport layer
- it sends data of application layer → as fast as it decides  
→ in order not to create congestions
- it is the last layer unpacked from the layers' stack  
→ used to know how to send packets

There are 3 main types of protocols for transport layer:

- UDP:
  - it is light, fast, unreliable → used for streaming and online games
  - it just forwards what it is received from application layer  
⇒ this implies:
    - \* no ACK
    - \* no congestion control
    - \* no care about order in receiver
- TCP:
  - it is reliable, end-to-end, two-way protocol → used for sending files
  - it assures that receiver gets packets correctly and ordered → waiting for ACK
  - there is control over flow and congestion → on receiver and on internet  
⇒ this implies:
    - \* it cares about speed → not overwhelming receiver
    - \* it prevents bottleneck → it prevents packet-loss
    - \* if packets aren't received correctly ⇒ they won't go up in the layers' stack of receiver
  - it is closed to IP → TCP/IP
  - there are many protocols → new ones must be backwards-compatible
- QUIC:
  - it is faster and simpler than TCP ⇒ no 3-way handshake, ...
  - transport ⇒ UDP + new layer → to emulate only TCP good things

Some definitions:

- Capacity → total data transfer available
- Bandwidth → total data transfer available right now
- Throughput → what is sent out

- Goodput → what is received in

## 6.2 TCP Protocol

Characteristics:

- it is byte-stream connection-oriented, reliable, full-duplex
- byte-stream:
  - app writes bytes
  - TCP sends segments  $\Rightarrow \approx 1.5\text{KB}$
  - app reads bytes
- it has flow and congestion control
- it is tied to the Internet Protocol (IP)

### 6.2.1 TCP Reliability

Characteristics:

- it is used checksum → to detect bit level errors
- it is used sequence numbers → to detect sequencing errors  $\Rightarrow$  so:
  - duplicates packets are discarded
  - packets can be reordered
  - lost packets can be retransmitted
- how to detect lost packets:
  - Timeout-based Recovery:
    - \* it requires sender to maintain data until it is ACKed
    - \* based on RTT (Round Trip Time) → waiting before retransmitting
    - \* it requires RTO (Retransmission TimeOut) calculation
      - accurate RTT estimators:
        - low RTO → unneeded retransmissions
        - high RTO → poor throughput
  - 3 Dupacks:
    - \* Ack → packet received correctly, in order until packet  $x$
    - \* Dupack → same ack is received 2 times  $\Rightarrow$  it happens:
      - packet loss
      - packet reordering
      - AWND update

- \* Problem → TCP timeouts lead to inactivity periods
- \* Proposal → use 3 duplicate ACKs to trigger retransmission
- Speed:
  - There are various types of increasing speed → packets for RTT:
    - linear (1-1, 2-2, 3-3, 4-4, ...) → increase sliding windows (active packets)
    - exponential (1-1, 2-2, 4-4, 8-8, ...) → slide on every packet
    - Ssthresh → it doesn't start from first every retransmission:
      - \* additive
      - \* multiplicative
- Flow Control:
  - it blocks the sender from overwhelming the receiver
  - Receiving side → AWND in returning ACKs is set by receiver  
⇒ how much space is left in its buffer
  - Sending side → Sending Window represents the actual bytes sent out  
 $SW = \min(AW, CW)$  → min between advertised and congestion window
- Congestion Control:
  - Delay-Bandwidth Product:
    - \* delay → time passed from sender to receiver → known as ping
    - \* bandwidth → how much is sent simultaneously on the channel
    - \* RTT is twice the Delay
    - \* bandwidth is distributed like:
      - half the traffic is travelling
      - half reached the receiver and is sending ACKs back
  - it blocks the sender from overwhelming the network
  - Idea:
    - \* each source determines network capacity for itself
    - \* it is used implicit feedback
- feedback algorithms are used for congestion control → AIMD
  - Context:
    - \* in the past people didn't think about wireless
    - \* any loss in wired links ⇒ caused by congestion (no error loss)  
→ creation of algorithm to regulate this
  - AIMD (Additive Increase Multiplicative Decrease):
    - \* adjust to changes in the available capacity



- \* CWND (Congestion Window):
  - increase when congestion goes down
  - decrease when congestion goes up
- \* Congestion detection → assumption that lost packet ⇒ congestion:
  - timeout → for serious problems
  - 3 dupacks → for minor problems
- \* how it works basically:
  - if timeout:
    - occurs ⇒ increment CWND by one packet per RTT  
→ linear increment
    - doesn't occur ⇒ divide CWND by 2  
→ multiplicative decrement
  - ⇒ transmission goes up and down
- \* Slow Start Threshold:
  - it is used:
    - ★ when first starting a connection
    - ★ when connection goes dead waiting for timeout
  - threshold is fixed → very large at the beginning
  - how it works:
    1. beginning with CWND = 1 packet
    2. before threshold → exponential increase ⇒ 2x CWND
    3. after threshold → linear increase
    4. if congestion level is reached ⇒ new threshold becomes  $\frac{1}{2}$   
and if it is indicated by:
      - ★ 3 dupacks ⇒ CWND = new threshold  
⇒ linear increase
      - ★ timeout ⇒ CWND = 1 ⇒ use of slow start  
⇒ exponential increase

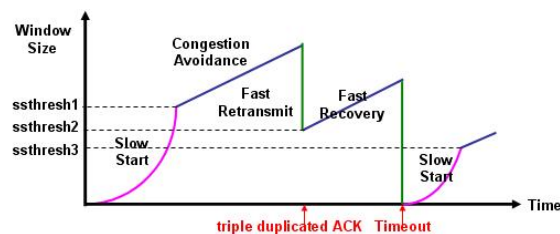


Figure 8: AIMD and SStThreshold

### 6.3 Legacy TCP Versions

Here there are a description of some TCP versions.

#### 6.3.1 TCP Tahoe

Characteristics:

- it has all TCP features previously described
- congestion control with AIMD + slow start + timeouts only for losses detection

#### 6.3.2 TCP Reno

Characteristics:

- it has both timeouts and 3 dupacks
- 3 dupacks is used to quickly recover from light congestion (1 packet loss)  
→ without having a timeout

#### 6.3.3 TCP New Reno

Characteristics:

- it is like TCP Reno
- it introduces partial ACKs to recover more packets → without using timeouts  
→ one recovery every RTT

#### 6.3.4 TCP SACK

Characteristics:

- it is the acronym for Selective ACK
- returning acks declares which packets were received
- all non received packets (no ACK) can be retransmitted  
→ recover from multiple losses in just one RTT

#### 6.3.5 TCP Vegas

Characteristics:

- it is based on the assumption on throughput →  $\text{actual} \leq \text{expected}$   
where:
  - $\text{actual} = \text{acks} / \text{round trip time}$
  - $\text{expected} = \text{window size} / \text{round trip time}$
- reaction happens per congestion episode not per loss
- it includes some modification from basic TCP:
  - modified Congestion Avoidance
  - aggressive Retransmission

- aggressive Window Adaptation
- modified Slow-Start
- Modified Congestion Avoidance:
  - throughput → actual  $\leq$  expected
  - expected throughput → it is transmission rate with no other traffic/queue
  - Monitor transmission rate (throughput, goodput):
    - \* Given static parameters  $\alpha, \beta$  as values representing how many packets TCP Vegas can have in queues ( $\alpha = 3, \beta = 1$ )
    - if  $\alpha < \beta \Rightarrow$  expected  $- \beta < \text{expected} - \alpha < \text{expected}$
    - there are different scenarios:
      - if expected  $- \alpha < \text{actual} < \text{expected}$ 
        - $\Rightarrow$  decrease queues  $\rightarrow$  increase rate
        - $\Rightarrow$  low congestion  $\rightarrow$  closer to expected
      - if expected  $- \beta < \text{actual} < \text{expected} - \alpha$ 
        - $\Rightarrow$  don't do anything
        - $\Rightarrow$  maybe congestion
      - if actual  $< \text{expected} - \beta$ 
        - $\Rightarrow$  increase queues  $\rightarrow$  decrease rate before packet drop
        - $\Rightarrow$  high congestion  $\rightarrow$  prevent packet loss
  - CWND is updated every RTT  $\rightarrow$  if:
    - \* expected  $- \text{actual} < \alpha$ 
      - $\Rightarrow \text{CWND} = \frac{\text{CWND} + 1}{\text{CWND}}$
    - \* expected  $- \text{actual} > \beta$ 
      - $\Rightarrow \text{CWND} = \frac{\text{CWND} - 1}{\text{CWND}}$
    - \*  $\alpha < \text{expected} - \text{actual} < \beta$ 
      - $\Rightarrow \text{CWND} = \text{CWND}$

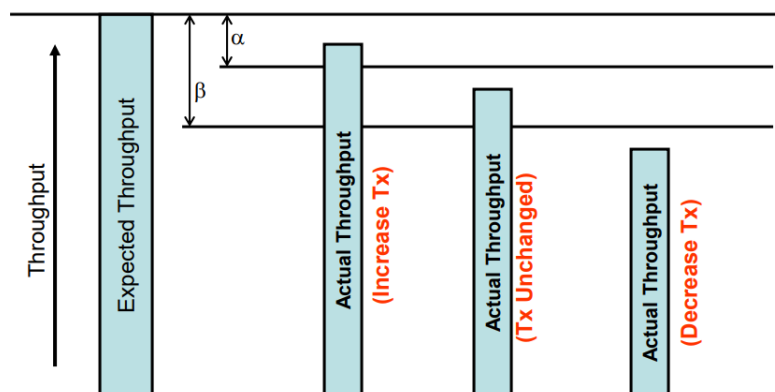


Figure 9: Modified Congestion Avoidance in TCP Vegas

- Aggressive Retransmission → with 3 dupacks:
  - 1<sup>st</sup> and 2<sup>nd</sup> packets → it checks timeout
  - if timeout expires ⇒ immediately retransmission
- Aggressive CWND update → 3 different types of update:
  - recovery → CWND becomes  $\frac{3}{4}$  when it enters into recovery
  - instead of  $\frac{1}{2}$
  - multiple loss → CWND is reduced by 1 size
  - initial setting → CWND is set on dimension 2 → instead of 1
- Modified Slow Start:
  - TCP keeps the congestion window fixed in every other RTT
  - it measures the throughput
  - Given:
    - \* static parameter  $\gamma$  as value  $\frac{1\text{packet}}{RTT}$
    - \* actual and expected throughput defined as backwards (§6.3.5)
  - on every next RTT, it does the followings:
    - \* if  $\text{expected} - \text{actual} < \gamma$  → continue Slow Start:
      - $\text{CWND} = 2 \cdot \text{CWND}$  for each RTT
      - $\text{CWND} = \text{CWND} + 1$  for each ACK
    - ⇒ Exponential Increase
    - \* if  $\text{expected} - \text{actual} > \gamma$  → switch to Congestion Avoidance:
      - Set  $\text{SSThreshold} = \text{CWND}$
      - follow Congestion Avoidance's rules written before

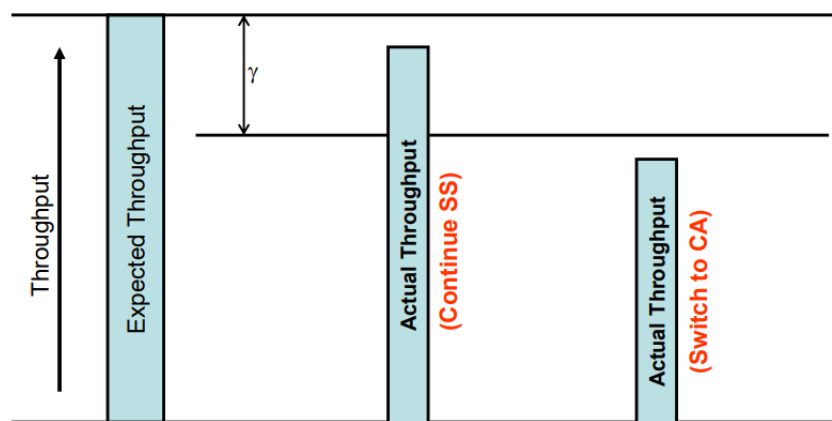


Figure 10: Modified Slow Start in TCP Vegas

- Retrocompatibility:

- it is sensible to delay variations
- it can't coexists with other versions
- Example:

When a TCP Vegas flow shares the same bottleneck with a TCP New Reno

→ as soon as the pipe is full and packets get buffered

⇒ TCP Vegas reduces its data rate

⇒ more space to TCP New Reno

→ it continues its growth till congestion

→ for the example this makes vegas not used → but it inspires other protocols

## 6.4 Wireless TCP

Characteristics:

- Problems:
  - there are error losses → they are assumed as congestion ⇒ Shrunked CWND ⇒ drop data rate
  - difficult to distinguish congestion from error losses
  - there are burst losses ⇒ multiple CWND shrinking ⇒ drop data rate
  - there is latency → it can be:
    - \* variable
    - \* hard to estimate →  $\frac{RTT}{RTO}$
    - \* high → protocol no more fair
  - there can be disconnections/handoff/fading → for weaker signal
  - bandwidth can be variable ⇒ wastage and loss bursts
- Multi-Hop Wireless Paths:
  - there is an exponential decrease of throughput/ increased delay
    - from 1 to 3 hops
  - packets/transmissions are in competition
    - contention also between datas and ACKs → if channel is not free ⇒ waiting
  - lareger number of hops ⇒ throughput stabilizes → effective pipelining
- Throughput decrease:
  - generally true when increasing speed
  - why → because of link breakage and repair latency:
    1. sender doesn't know that the link is broken
      - ⇒ it may continue transmitting (packet loss)
    2. sender doesn't know when it is all available again
      - it will retransmit when timeout occurs
  - how to improve it:
    - \* network feedback → if AWND of receiver has size 0, network can notify sender ⇒ sender stops immediately
    - \* TCP failure → sender is notified with a message about it
    - \* TCP callback → let sender knows when link is broken/repared
    - \* dynamic TCP timeout → alleviate TCP timeouts/backoffs

### 6.4.1 Wireless TCP Protocols

There are different types of protocols for transport layers in wireless TCP:

- Connection split:
  - local retransmission
  - quick action on wireless link
  - TCP specific for wireless link
  - Examples:
 

* I-TCP	* M-TCP
* PROXY	* SNOOP
- Pure End-to-End:
  - new protocol → better than old version, retrocompatibility ...
  - sender is aware of wireless link
  - Examples:
 

* Delayed Dupacks	* TCP-Aware
* Freeze-TCP	* TCP Probing
* WTCP	* TCP Westwood
* TCP Hybla	* TCP CUBIC
* TCP High Speed	* TCP Compound
* TCP Fast	* ...

Here there are a description of some of them.

#### 6.4.1.1 SNOOP Protocol

Characteristics:

- it is designed to address high BER
- Base Station implements a Snoop Agent
  - it monitors all packets of sender and receiver (also ACKs)
  - it caches packets not acked yet into base station
  - it intercepts dupacks:
    - \*  $\frac{1}{2}$  → immediate retransmission (sender not notified)
    - \* 3 → let sender knows about it (CWND shrinkage)
- Path: sender ↔ snoop agent ↔ receiver
- it needs low latency between sender and snoop agent
  - ⇒ otherwise traditional TCP is better
- Pro:
  - ✓ End-to-End preservation → no ACK created by snoop agent
  - ✓ local retransmission
  - ✓ high BER address

- Cons:
  - ✗ requirement of little latency on the wireless link
  - ✗ same performances on long disconnections
  - ✗ immediately after a handoff  $\Rightarrow$  Slow Start  $\rightarrow$  no packets in new cache

#### 6.4.1.2 Satellites

There are:

- GEO  $\rightarrow$  have backbone configuration  $\rightarrow$  so there is  $\Rightarrow$  bridge between terrestrial antenna
- LEO  $\rightarrow$  have direct to home configuration  $\rightarrow$  so it works as  $\Rightarrow$  (terrestrial antenna  $\rightarrow$  satellite  $\rightarrow$  home)

All of them have:

- high RTT  $\rightarrow$  it can be  $\approx 600\text{ms}$  for GEO
- important PER<sup>10</sup>  $\rightarrow$  due to radio channels  $\rightarrow$  interferences, wheather ...

#### 6.4.1.3 Slow Start and Congestion Avoidance Models

Characteristics:

- also referred as Van Jacobson algorithm
- In Slow Start phase:
  - $\rightarrow$  CWND = CWND + 1 for every new ACK received
  - $\rightarrow$  CWND = 2 · CWND for every RTT
- In Congestion Avoidance phase:
  - $\rightarrow$  CWND = CWND +  $\frac{1}{\text{CWND}}$  for every new ACK received
  - $\rightarrow$  CWND = 2 · CWND for every RTT
- RTT changes dynamically

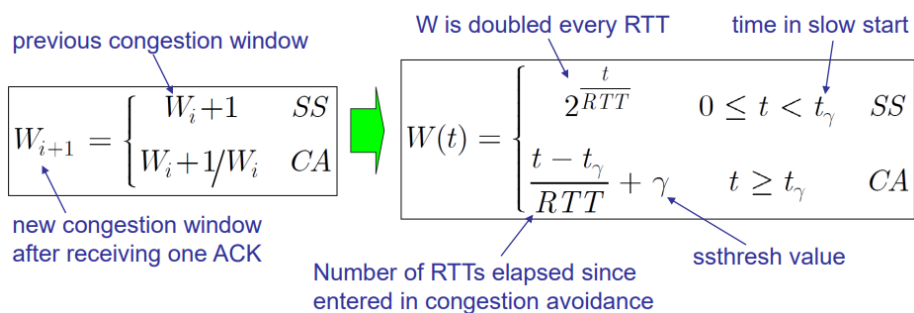


Figure 11: Slow Start & Congestion Avoidance Models

<sup>10</sup>Packet Error Rate



#### 6.4.1.4 RTT Unfairness


It is one of the problems that protocols have to solve:

- $B(t) = \frac{W(t)}{RTT} \Rightarrow$  where  $B(t)$  = bandwidth and  $W(t)$  = window growth rate
- longer RTT  $\Rightarrow$  slower phase growth rate
- smaller RTT  $\Rightarrow$  bigger bandwidth

#### 6.4.1.5 TCP Hybla

Characteristics:

- it equalizes transmission rate against RTT  $\Rightarrow$  fair RTT
- a longer RTT  $\Rightarrow$  compensate by sending twice on every ACK ( $\neq$  every RTT)
- there is the introduction of parameter  $\rho = \frac{RTT}{RTT_0}$  where:
  - $\rightarrow$  RTT is actual Round Trip Time
  - $\rightarrow$   $RTT_0$  is reference Round Trip Time (for example  $RTT_0 = 25\text{ms}$ )

$$W^H(t) = \begin{cases} \rho 2^{\rho \frac{t}{RTT}} & 0 \leq t < t_{\gamma,0} \quad \text{SS} \\ \rho \left[ \rho \frac{t - t_{\gamma,0}}{RTT} + \gamma \right] & t \geq t_{\gamma,0} \quad \text{CA} \end{cases}$$


$$B^H(t) = \frac{W^H(t)}{RTT} = \begin{cases} \frac{2^{\rho \frac{t}{RTT}}}{RTT_0} & 0 \leq t < t_{\gamma,0} \quad \text{SS} \\ \frac{1}{RTT_0} \left[ \rho \frac{t - t_{\gamma,0}}{RTT} + \gamma \right] & t \geq t_{\gamma,0} \quad \text{CA} \end{cases}$$

Figure 12: Slow Start & Congestion Avoidance for TCP Hybla

- Pros:
  - $\rightarrow$  End-to-End solution
  - $\rightarrow$  it changes only on sender side  $\Rightarrow$  easily deployable
    - $\rightarrow$  no damage for the entire system
  - $\rightarrow$  it has RTT fairness
- Cons:
  - $\rightarrow$  it is so aggressive  $\Rightarrow$  it can lead to multiple losses
  - $\rightarrow$  measured RTT is sensitive to buffer size  $\rightarrow$  limited and not sustainable anymore at a certain point
  - $\rightarrow$  no handling on BER/disconnections (as most of TCPs)
  - $\rightarrow$  doubts about friendliness and fairness

What is friendliness:

- how different flows of TCPs cooperate
  - $\rightarrow$  total amount of bandwidth increased

- average amount of bandwidth
- new version shouldn't steal bandwidth of older versions
  - it can take all new available bandwidth

#### 6.4.1.6 TCP Westwood & TCP Westwood Plus

Characteristics:

- it is pure End-to-End
- flow control is based on estimation of eligible bandwidth (BWE)
  - monitoring of acks' arrival rate at sender side
  - use of BWE to set CWND and SSThresh after a loss
    - \* 3 Dupacks:
      - $SSThresh = BWE \cdot RTT_{min}$  ( $\neq$  TCP New Reno  $\rightarrow SSThresh = \frac{CWND}{2}$ )
      - if  $CWND > SSThresh \Rightarrow CWND = SSThresh$
    - \* Timeout expiration:
      - $SSThresh = BWE \cdot RTT_{min}$  ( $\neq$  TCP New Reno  $\rightarrow SSThresh = \frac{CWND}{2}$ )
      - $CWND = 1$
  - sending more than what you receive won't affect BWE

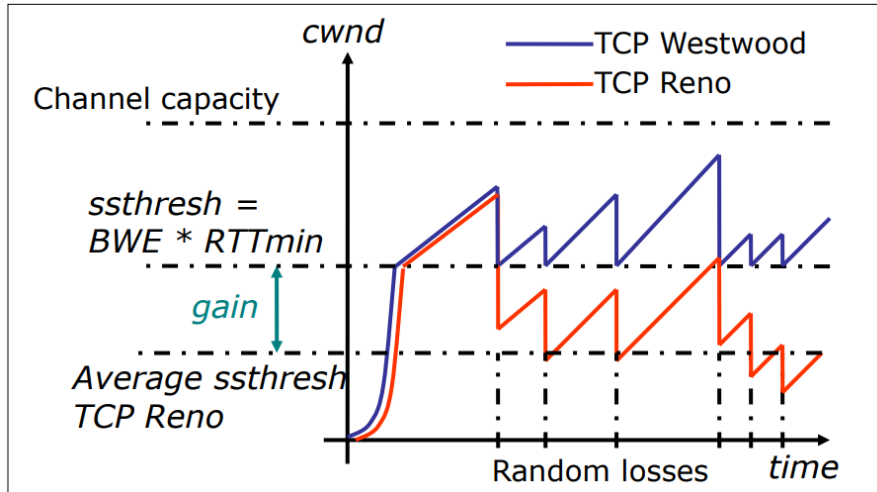


Figure 13: TCP Westwood vs TCP Reno

- Rate Estimation:
  - useful to enhance congestion control
  - computed by sampling/exponential filtering at sender
  - based on ACKs arrival times and amount of bandwidth delivered
    - ⇒ data ACKed are aggregated in interval  $T = RTT$
  - used by sender to set CWND and SSThresh
- Pros:
  - BWE allows to reach higher throughput
  - it changes only on sender side
- Cons:
  - wrong BWE over asymmetric links
  - No handling of high BER/disconnections
  - doubts about friendliness and fairness

#### 6.4.1.7 TCP Adaptive Selection

Characteristics:

- possibility to have different TCP variants concurrently → matching different characteristics of connections
- it can be applied in different ways depending on:
  - agent that performs TCP selection
  - use of cross-layer approach ⇒ not linked by the standard
  - changing TCP version on on-going connection ⇒ dynamic selection
- there are different modules that can be replaced in TCP-module-container
  - criteria:
    - TCP parameters → RTT, BWE ...
    - cross-layer informations
    - reliable channel estimation

#### 6.4.1.8 TCP Cubic

Characteristics:

- it is used by default in linux kernels
- optimized congestion control algorithm for high speed networks with high latency
- window → cubic function of time since last congestion event
- Algorithm:
  1. inflection point set CWND prior last congestion event

2. quickly initial growth
  3. slow down + stay stable around CWND value when congestion happens
  4. no loss happens → quickly growth again
- Differences with standard TCPs
    - TCP Cubic doesn't rely on receipt of ACKs to increase CWND
    - TCP Cubic's CWND depends only on the last congestion event
      - ⇒ less RTT-unfairness → window growth independent from RTT

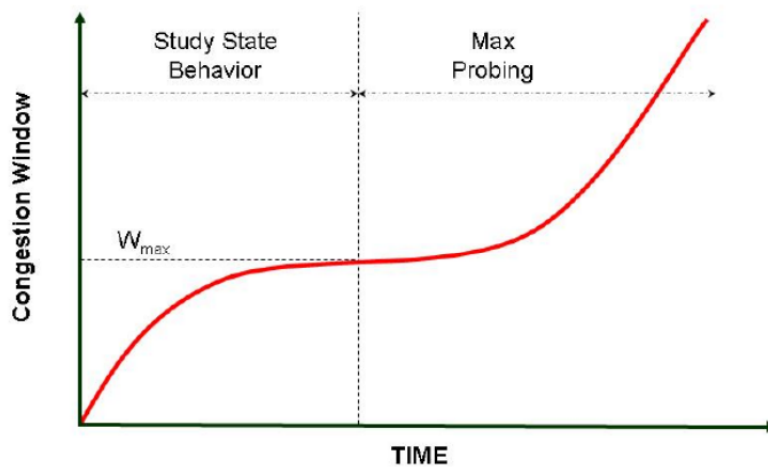


Figure 14: TCP Cubic: CWND growth

## 7 Vehicular Ad Hoc Networks (VANET)

### 7.1 IEEE 802.11p

- it provides connectivity to vehicles
  - beyond 4G → not so crucial
  - useful for safety/public apps
- parts taken from a and b versions
- it use a reserved sequency (5.9GHz)
  - 1000 m of transmission range
  - 26 Mbps → but it can be lower → sufficient for exchanging messages, ...
  - transmission up to 6 Mbps at 300m → nodes travelling at 200 km/h speed
- Problem:
  - connection with AP and vehicles
    - transmission range changes
    - transmission needs to be adapted
    - these can lead to sync problems among vehicles

### 7.2 Vehicular Networks: System Model

Introducing the context

- Safe driving:
  - alert messages → delivered quickly to all cars following the one that is alerting (Abnormal Vehicles)
    - need to generate a chain reaction
  - there are some problems:
    - \* multiple transmission ⇒ when every node is broadcasting
    - \* possible congestion ⇒ alert unuseful → crashes still can happen

For these reason, a system model with these characteristics is introduced:

- high mobility of nodes
- variable transmission range
- a car cannot be sure to be the farthest car receiving that broadcast message
- some approaches:
  - MCDS (Minimum Connected Dominant Set):
    - \* minimum cardinality set of connected nodes ⇒ each other node in the network is connected to a node of the MCDS set
    - \* MCDS nodes have to broadcast the message

- \* it is optimal but non feasible solution → it needs creation of overlay structure ⇒ to cover the network
  - \* it is dangerous because of deterministic failure → what if a node in MCDS doesn't broadcast? → redundancy needed redundancy needed
  - \* implementation with  $n$  nodes ⇒  $O(n \cdot \log(n))$  control messages
- RD (Redundancy Avoidance):
- \* based on backoff mechanism → if there is a collision due congestion ⇒ it reduces frequency
  - \* if following vehicle has already broadcast ⇒ actual nodes don't
  - \* it isn't considered the number of hops in these schemas
- JS (Jamming Signal):
- \* it is Urban Multi-hop Broadcasting Protocol
  - \* it is used to determine next forwarder
  - \* vehicles which receive alert → it emits JS for an amount of time ⇒ proportional to the distance from sender
  - \* The last vehicle stopping the JS knows it is the last one → it forwards the alert message
  - \* JS phase delays the transmission of the message ⇒ not suitable for alert messages
- CW (Contention Window):
- \* vehicles set CW inversely proportional to distance from sender
  - \* this implies:
    - no control traffic
    - something unrealistic ⇒ transmission rate not known → it needs to be determined in order to find the last one

### 7.3 Fast Broadcasting

Characteristics:

- fast broadcast is a solution designed to have alert messages covering the area of interest in as less time as possible (as few hops as possible)
- there are two types:
  - probabilistic:
    - \* Pro → reliable
    - \* Cons → End-to-End delay
  - deterministic:
    - \* Pro → End-to-End delay

- \* Cons → reliable

- how it works:

→ there are two phases:

- \* estimation phase → vehicles exchange hello messages to collect info  
→ in order to estimate their own transmission range.
- \* broadcasting phase → transmission range estimation used to forward asap alert message to destination

→ here there is a detailed description of the two phases

1. estimation phase:

- \* continuously run
- \* time is splitted into rounds
- \* one hello messages randomly sent every time round
- \* hello contains
  - sender's position
  - maximum frontward distance from which another vehicle has been heard transmitting an hello message
- \* About messages:
  - two types:
    - ★ hello → information to estimate transmission range
    - ★ alert → sender's transmission range
  - variables:
    - ★ CMBR (Current Maximum Backward Range)  
→ computed by hearing hello from the back ⇒ someone could hear
    - ★ CMFR (Current Maximum Frontward Range)  
→ computed by hearing hello from the front ⇒ I could hear

2. broadcast phase:

- \* alert is generated by an Abnormal Vehicle
- \* alert is sent in broadcast to warn following vehicles
- \* alert includes estimated transmission range for that hop
- \* node receiving alert waits an amount of time  
→ proportional to the node's position with respect to the estimated maximum transmission range  
(→ near ⇒ more time)
- \* CW calculated as follow with R = range and D = distance:

$$CW = \lfloor \left( \frac{R_{max} - D}{R_{max}} \cdot (CW_{max} - CW_{min}) \right) + CW_{min} \rfloor$$

- if another car (farther from source) already forwarded  
⇒ the other doesn't forward
  - wrong estimation ⇒ possibly huge delay
  - dynamic transmission rate ⇒ CW can be lower for estimation
- ROFF (RObust Fast Forwarding):
  - it is a multi-hop deterministic delay based
  - estimation → every vehicle sends hello every round  
→ neighbourhood discovery