**contents**

**i) fundamentals in wireless transmissions {20%}**

**Shannon & Nyquist formula**

**ii) technologies for different scenarios/applications {wifi}**

**comparisons of network technologies 20-30%**

**--> Wireless technologies: Wifi, Manet**

**Wifi**

Wi-Fi is a [wireless networking](https://www.cisco.com/c/en/us/solutions/small-business/resource-center/networking/wireless-network.html?ccid=cc001530) technology that allows devices such as computers (laptops and desktops), mobile devices (smart phones and wearables), and other equipment (printers and video cameras) to interface with the Internet. It allows these devices--and many more--to exchange information with one another, creating a network.

Internet connectivity occurs through a wireless router. When you access Wi-Fi, you are connecting to a wireless router that allows your Wi-Fi-compatible devices to interface with the Internet.

**Wireless LANs**

♣ Infrared (IrDA) or radio links (Wavelan)

♣ Advantages

– very flexible within the reception area

– Ad-hoc networks possible

– (almost) no wiring difficulties

♣ Disadvantages

– low bandwidth compared to wired networks

– many proprietary solutions

• Bluetooth, HiperLAN and IEEE 802.11

**Wireless LANs vs. Wired LANs**

♣ Destination address does not equal destination location

♣ The media impact the design

– wireless LANs intended to cover reasonable geographic distances must be built from basic coverage blocks

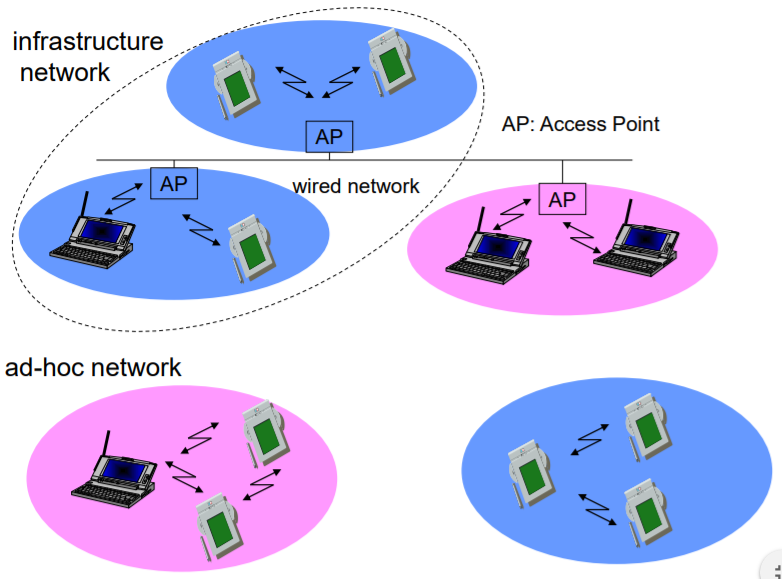
♣ Impact of handling mobile (and portable) stations

– Propagation effects

– Mobility management

– Power management

**Infrastructure vs. Ad hoc WLANs**



**How are infrastructure and ad hoc networks different?**

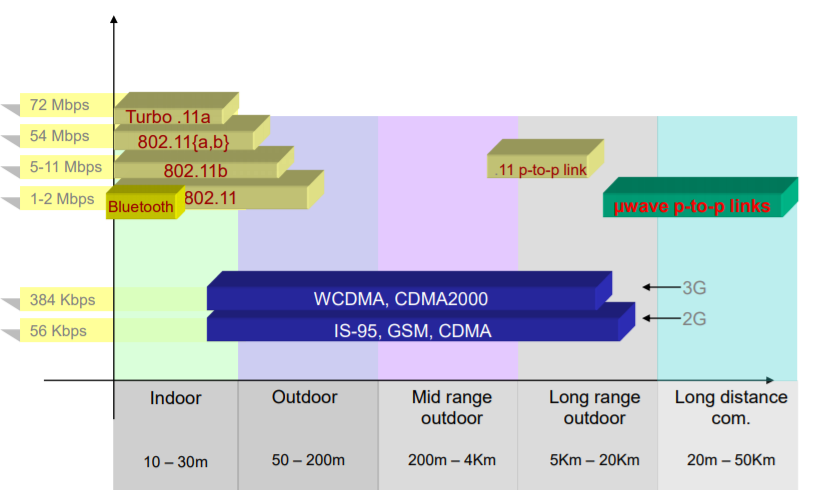
Wireless networks can operate in one of two modes: infrastructure or ad hoc.

In *infrastructure* mode, all devices on a wireless network communicate with each other through an access point (wireless router).

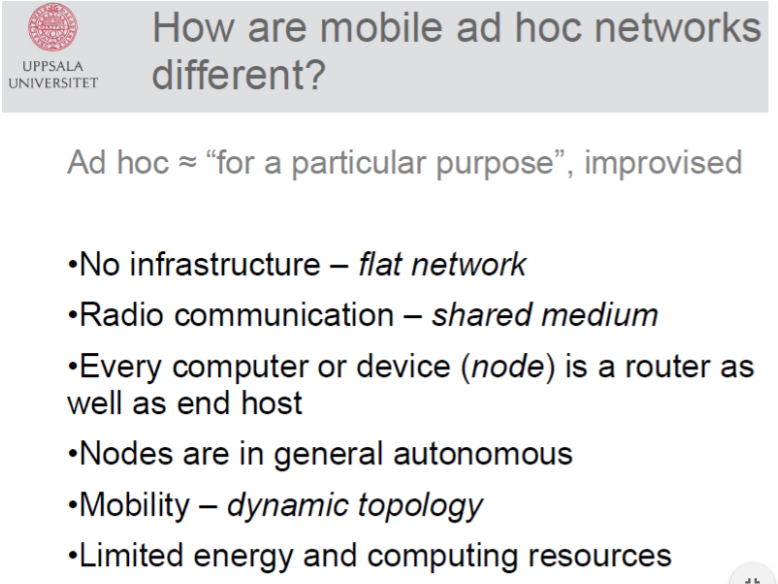
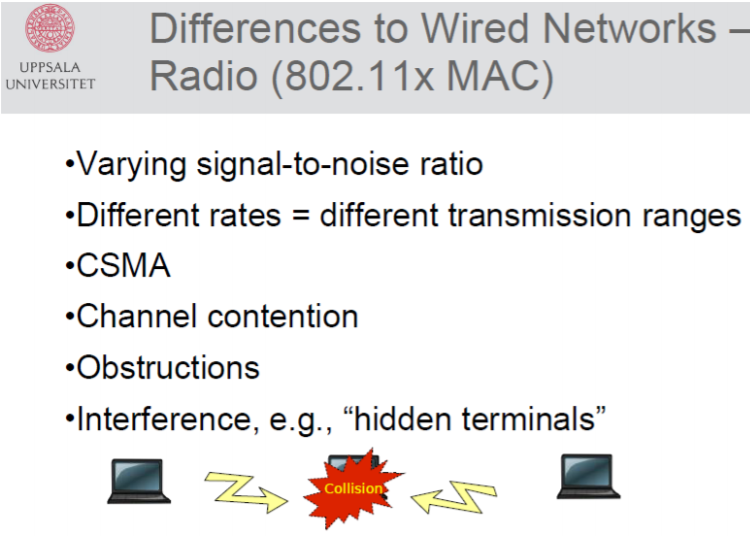
In *ad hoc* mode, a computer with a wireless network adapter communicates directly with a printer equipped with a wireless print server.

|  | **Infrastructure** | **Ad hoc** |
| --- | --- | --- |
| ***Characteristics*** | | |
| Communication | Through an access point | Directly between devices |
| Security | More security options | WEP or no security |
| Range | Determined by the range and number of access points | Restricted to the range of individual devices on the network |
| Speed | Usually faster | Usually slower |
| ***Requirements for all devices*** | | |
| Unique IP address for each device | Yes | Yes |
| Mode set to | Infrastructure mode | Ad hoc mode |
| Same SSID | Yes, including the access point | Yes |
| Same channel | Yes, including the access point | Yes |

**Wireless Technology Landscape (Cảnh quan công nghệ không dây)**

**** ****

**Manet**

**** ****

A **wireless ad hoc network**[[1]](https://en.wikipedia.org/wiki/Wireless_ad_hoc_network#cite_note-1) (**WANET**) or **Mobile ad hoc network** (**MANET**) is a decentralized type of [wireless network](https://en.wikipedia.org/wiki/Wireless_network).[[2]](https://en.wikipedia.org/wiki/Wireless_ad_hoc_network#cite_note-auto-2)[[3]](https://en.wikipedia.org/wiki/Wireless_ad_hoc_network#cite_note-S._Manoj_2004-3)[[4]](https://en.wikipedia.org/wiki/Wireless_ad_hoc_network#cite_note-4)[[5]](https://en.wikipedia.org/wiki/Wireless_ad_hoc_network#cite_note-5)[[6]](https://en.wikipedia.org/wiki/Wireless_ad_hoc_network#cite_note-6) The network is [ad hoc](https://en.wikipedia.org/wiki/Ad_hoc) because it does not rely on a pre-existing infrastructure, such as [routers](https://en.wikipedia.org/wiki/Router_(computing)) in wired networks or [access points](https://en.wikipedia.org/wiki/Wireless_access_point) in managed (infrastructure) wireless networks.[[7]](https://en.wikipedia.org/wiki/Wireless_ad_hoc_network#cite_note-7) Instead, each [node](https://en.wikipedia.org/wiki/Node_(networking)) participates in routing by [forwarding](https://en.wikipedia.org/wiki/Packet_forwarding) data for other nodes, so the determination of which nodes forward data is made dynamically on the basis of network connectivity and the [routing algorithm](https://en.wikipedia.org/wiki/Routing_algorithm) in use.

***MANET characteristics: Why Ad Hoc Networks ?***

◼ Setting up of fixed access points and backbone infrastructure is not always viable

❖ Infrastructure may not be present in a disaster area or war zone

❖ Infrastructure may not be practical for shortrange radios; Bluetooth (range ~ 10m)

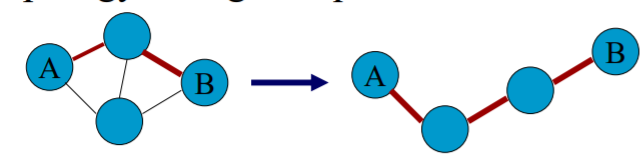
◼ Ad hoc networks:

❖ Do not need backbone infrastructure support

❖ Are easy to deploy

❖ Useful when infrastructure is absent, destroyed or impractical

◼ Host movement frequent

◼ Topology change frequent 

◼ No infrastructure. Multi-hop wireless links.

◼ Data must be routed via intermediate nodes.

***Mobile Ad-hoc Network***

◼ Self-configuring network of mobile routers (and associated hosts) connected by wireless links

◼ This union forms a random topology

◼ Routers move randomly free

◼ Topology changes rapidly and unpredictably

◼ Standalone fashion or connected to the larger Internet

◼ Suitable for emergency situations like natural or human-induced disasters, military conflicts, emergency medical situations, etc.

**IEFT MANET Working Group**

◼ goal: ❖ to standardize an interdomain unicast routing protocol which provides one or more modes of operation, each mode specialized for efficient operation in a given mobile networking “context”, where a context is a predefined set of network characteristics.

◼ a dozen candidate routing protocols have been proposed

**--> Wireless communication challange: hidden terminal, exposed terminal problems**

**Wireless Media**

♣ Physical layers used in wireless networks

– have neither absolute nor readily observable boundaries outside which stations are unable to receive frames – are unprotected from outside signals

– communicate over a medium significantly less reliable than the cable of a wired network

– have dynamic topologies

– lack full connectivity and therefore the assumption normally made that every station can hear every other station in a LAN is invalid (i.e., STAs may be “hidden” from each other)

– have time varying and asymmetric propagation properties

**Limitations of the mobile environment**

• Limitations of the Wireless Network

• limited communication bandwidth

• frequent disconnections

• heterogeneity of fragmented networks

• Limitations Imposed by Mobility

• route breakages

• lack of mobility awareness by system/applications

• Limitations of the Mobile Device

• short battery lifetime

• limited capacities

**Wireless v/s Wired networks**

♣ Regulations of frequencies

– Limited availability, coordination is required

– useful frequencies are almost all occupied

♣ Bandwidth and delays – Low transmission rates

• few Kbps to some Mbps

– Higher delays • several hundred milliseconds

– Higher loss rates • susceptible to interference, e.g., engines, lightning

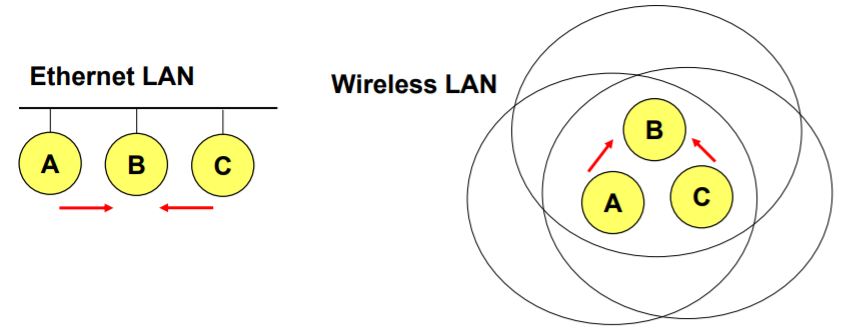
♣ Always shared medium

– Lower security, simpler active attacking

– radio interface accessible for everyone

– Fake base stations can attract calls from mobile phones

– secure access mechanisms important

**Difference Between Wired and Wireless **

♣ If both A and C sense the channel to be idle at the same time, they send at the same time.

♣ Collision can be detected at sender in Ethernet.

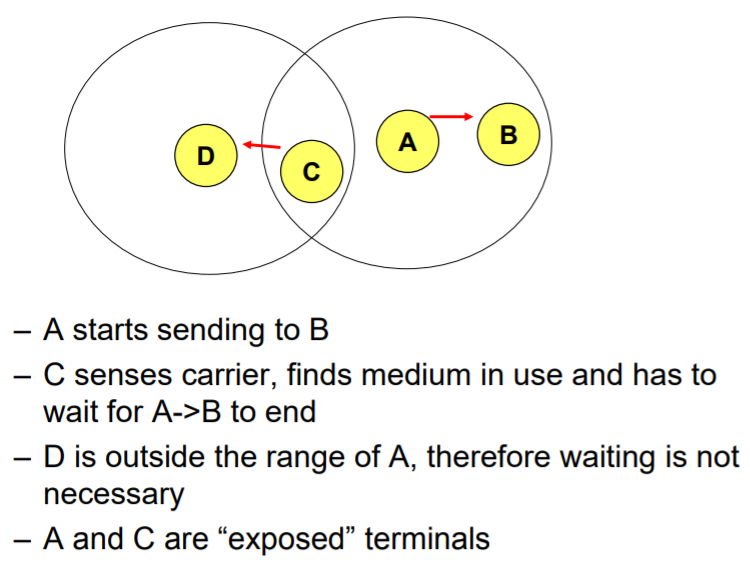
♣ Half-duplex radios in wireless cannot detect collision at sender.

**Hidden Terminal Problem**

In [wireless networking](https://en.wikipedia.org/wiki/Wireless_networking), the **hidden node problem** or **hidden terminal problem** occurs when a [node](https://en.wikipedia.org/wiki/Node_(networking)) can communicate with a [wireless access point](https://en.wikipedia.org/wiki/Wireless_access_point) (AP), but cannot directly communicate with other nodes that are communicating with that AP.[[1]](https://en.wikipedia.org/wiki/Hidden_node_problem#cite_note-1) This leads to difficulties in [medium access control](https://en.wikipedia.org/wiki/Medium_access_control) sublayer since multiple nodes can send data packets to the AP simultaneously, which creates interference at the AP resulting in no packet getting through.

Although some loss of packets is normal in wireless networking, and the higher layers will resend them, if one of the nodes is transferring a lot of large packets over a long period, the other node may get very little [goodput](https://en.wikipedia.org/wiki/Goodput" \o "Goodput).

Practical protocol solutions exist to the hidden node problem. For example, Request To Send/Clear To Send (RTS/CTS) mechanisms where nodes send short packets to request permission of the access point to send longer data packets. Because responses from the AP are seen by all the nodes, the nodes can synchronize their transmissions to not interfere. However, the mechanism introduces latency, and the overhead can often be greater than the cost, particularly for short data packets.



Trái Hidden Phải Expose

**Exposed Terminal Problem**

In [wireless networks](https://en.wikipedia.org/wiki/Wireless_networks), the **exposed node problem** occurs when a node is prevented from sending packets to other nodes because of [co-channel interference](https://en.wikipedia.org/wiki/Co-channel_interference) with a neighboring transmitter. Consider an example of four nodes labeled R1, S1, S2, and R2, where the two receivers (R1, R2) are out of range of each other, yet the two transmitters (S1, S2) in the middle are in range of each other. Here, if a transmission between S1 and R1 is taking place, node S2 is prevented from transmitting to R2 as it concludes after [carrier sense](https://en.wikipedia.org/wiki/Carrier_sense) that it will interfere with the transmission by its neighbor S1. However note that R2 could still receive the transmission of S2 without interference because it is out of range of S1.[[1]](https://en.wikipedia.org/wiki/Exposed_node_problem#cite_note-macaw-1)

[IEEE 802.11 RTS/CTS](https://en.wikipedia.org/wiki/IEEE_802.11_RTS/CTS) mechanism helps to solve this problem only if the nodes are synchronized and packet sizes and data rates are the same for both the transmitting nodes. When a node hears an RTS from a neighboring node, but does not generate the corresponding CTS, that node can deduce that it is an exposed node and is permitted to transmit to other neighboring nodes.[[1]](https://en.wikipedia.org/wiki/Exposed_node_problem#cite_note-macaw-1)

If the nodes are not synchronised (or if the packet sizes are different or the data rates are different) the problem may occur that the sender will not hear the CTS or the ACK during the transmission of data of the second sender.

The exposed node problem is not an issue in [cellular networks](https://en.wikipedia.org/wiki/Cellular_network) as the power and distance between cells is controlled to avoid it.

**Effect of mobility on protocol stack**

♣ Application – new applications and adaptations

♣ Transport – congestion and flow control

♣ Network – addressing and routing

♣ Link – media access and handoff

♣ Physical – transmission errors and interference

**--> MAC Layer: DCF w/ CSMA/CA, DCF w/ RTS/CTS, PCF**

**MAC layer**

The **M**edia **A**ccess **C**ontrol **L**ayer is one of two sublayers that make up the Data Link Layer of the [OSI](https://www.webopedia.com/definitions/osi/) model. The MAC layer is responsible for moving data [packets](https://www.webopedia.com/definitions/packet/) to and from one [Network Interface Card](https://www.webopedia.com/definitions/network-interface-card-nic/) (NIC) to another across a shared [channel](https://www.webopedia.com/definitions/channel/).

See a [breakdown of the seven OSI layers](http://webopedia.internet.com/reference/osi-layers/) in the [Quick Reference](http://webopedia.internet.com/quick_ref/) section of Webopedia.

The MAC sublayer uses MAC [protocols](https://www.webopedia.com/definitions/protocol/) to ensure that signals sent from different stations across the same channel don’t [collide](https://www.webopedia.com/definitions/collision/).

Different protocols are used for different shared networks, such as [Ethernets](https://www.webopedia.com/definitions/ethernet/), [Token Rings](https://www.webopedia.com/definitions/token-ring-network/), [Token Buses](https://www.webopedia.com/definitions/token-bus-network/), and [WANs](https://www.webopedia.com/definitions/wide-area-network-wan/).

Also see “[The OSI Reference Model – Understanding Layers](https://www.webopedia.com/insights/understanding-layers/)” in Webopedia’s “[Did You Know…](https://www.webopedia.com/reference/)” section.

♣ Traffic services

– Asynchronous Data Service (mandatory) – DCF

– Time-Bounded Service (optional) - PCF

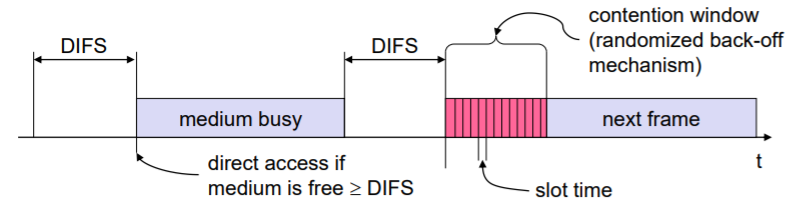
♣ Access methods

– DCF CSMA/CA (mandatory) • collision avoidance via randomized back-off mechanism • ACK packet for acknowledgements (not for broadcasts)

– DCF w/ RTS/CTS (optional) • avoids hidden/exposed terminal problem, provides reliability

– PCF (optional) • access point polls terminals according to a list

**CSMA/CA**

*DCF Inter-Frame Space*

– station which has data to send starts sensing the medium (Carrier Sense based on CCA, Clear Channel Assessment)

– if the medium is free for the duration of an Inter-Frame Space (IFS), the station can start sending (IFS depends on service type)

– if the medium is busy, the station has to wait for a free IFS plus an additional random back-off time (multiple of slot-time)

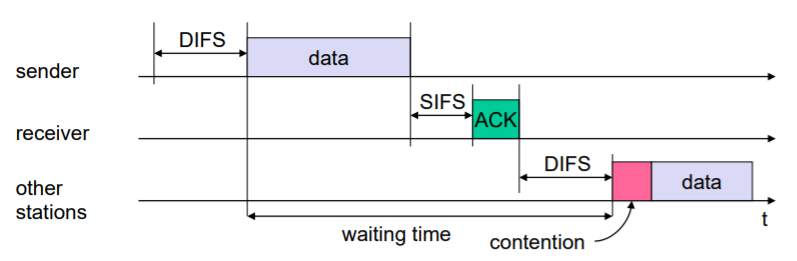
– if another station occupies the medium during the back-off time of the station, the back-off timer stops (fairness)

**DCF – basic access**

♣ If medium is free for DIFS time, station sends data

♣ receivers acknowledge at once (after waiting for SIFS) if the packet was received correctly (CRC)

♣ automatic retransmission of data packets in case of transmission errors

****

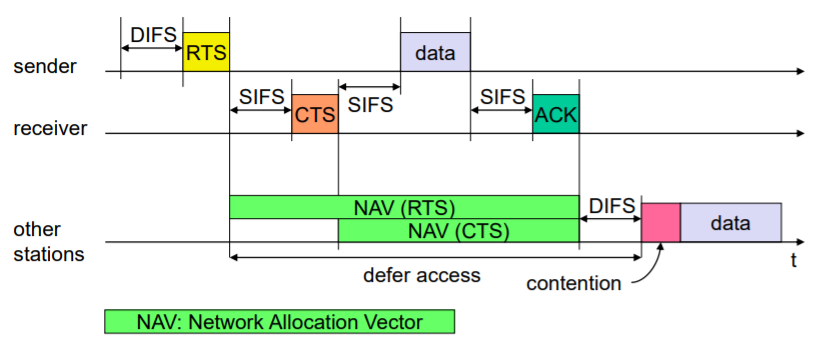
**RTS/CTS**

♣ If medium is free for DIFS, station can send RTS with reservation parameter (reservation determines amount of time the data packet needs the medium)

♣ acknowledgement via CTS after SIFS by receiver (if ready to receive)

♣ sender can now send data at once, acknowledgement via ACK

♣ other stations store medium reservations distributed via RTS and CTS

****

**PCF**

Point coordination function (PCF) is a media access control (MAC) technique used in IEEE 802.11 based WLANs, including Wi-Fi. It resides in a point coordinator also known as access point (AP), to coordinate the communication within the network. The AP waits for PIFS duration rather than DIFS duration to grasp the channel. PIFS is less than DIFS duration and hence the point coordinator always has the priority to access the channel

**Coexistence of PCF and DCF (sự tồn tại)**

♣ A Point Coordinator (PC) resides in the Access Point and controls frame transfers during a Contention Free Period (CFP)

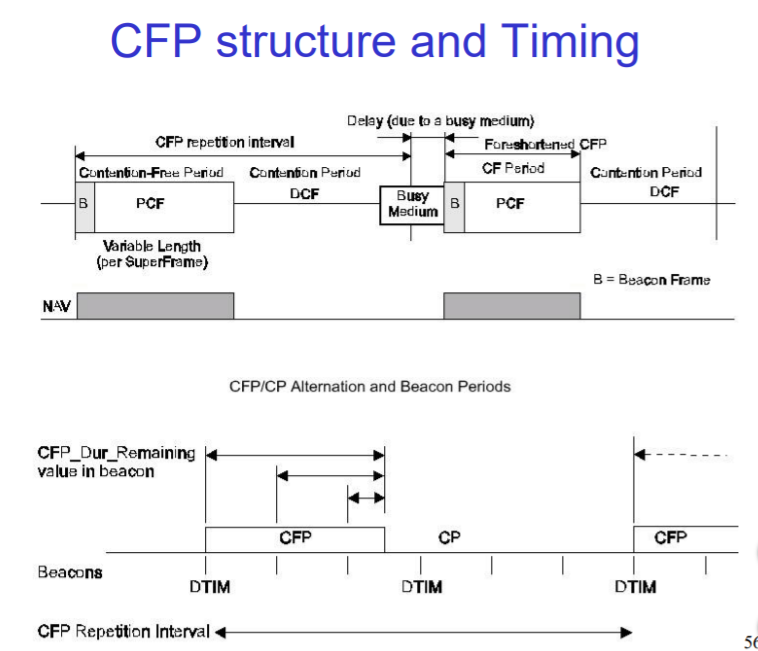
♣ A CF-Poll frame is used by the PC to invite a station to send data. Stations are polled from a list maintained by the PC ♣ The CFP alternates with a Contention Period (CP) in which data transfers happen as per the rules of DCF

♣ This CP must be large enough to send at least one maximum-sized packet including RTS/CTS/ACK

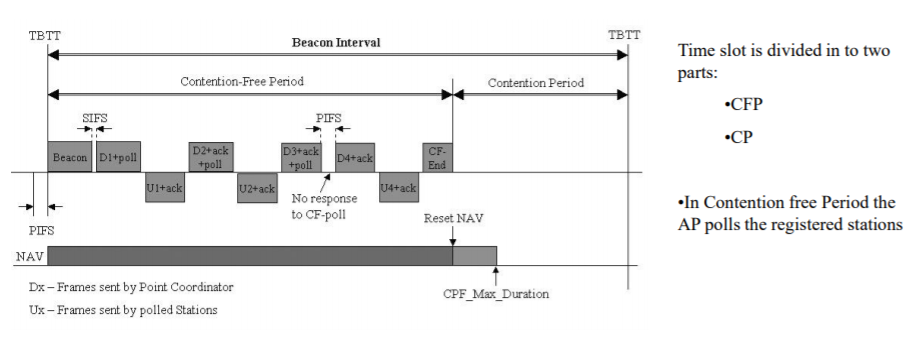
♣ CFPs are generated at the CFP repetition rate

♣ The PC sends Beacons at regular intervals and at the start of each CFP

♣ The CF-End frame signals the end of the CFP



**Point Coordination Function**



• PC waits for PIFS.

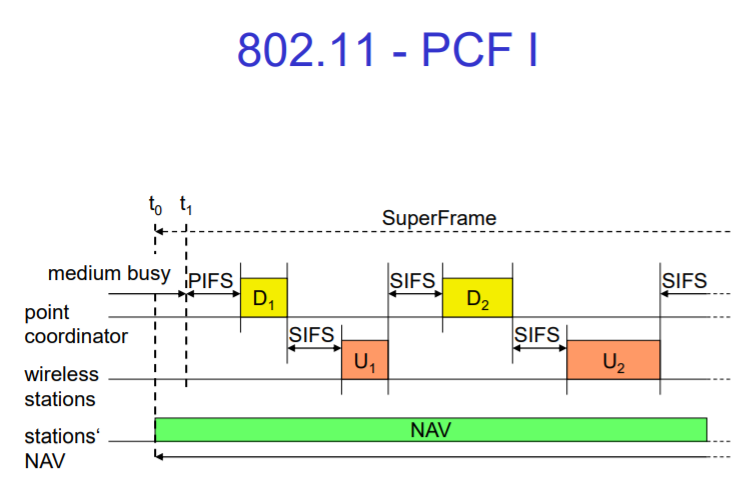
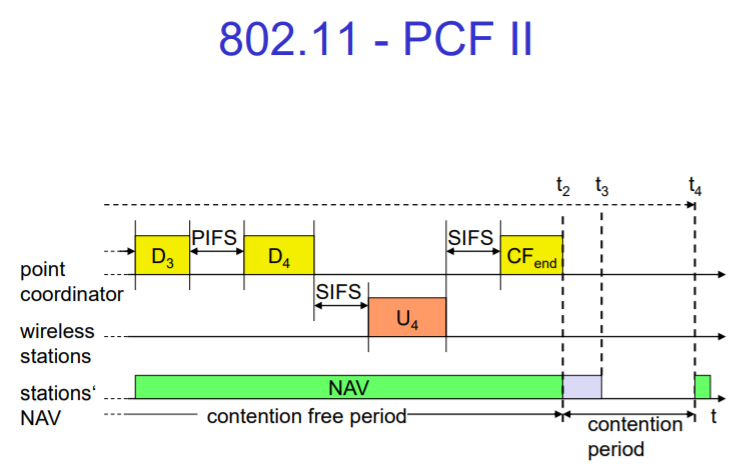
• Sends a Beacon Frame to start the polling sequence

•All other STAs set their NAVs to the value of CPF\_Max\_Duration

•After SIFS, PC Sends Data + Polled STA

•STA1,Sends the Data + Ack

•PC Data + Polled STA + Ack

Trái PCF I – Phải PCF II

**Throughput – DCF vs. PCF**

♣ Overheads to throughput and delay in DCF mode come from losses due to collisions and backoff

♣ These increase when number of nodes in the network increases

♣ RTS/CTS frames cost bandwidth but large data packets (>RTS threshold) suffer fewer collisions

♣ RTC/CTS threshold must depend on number of nodes

♣ Overhead in PCF modes comes from wasted polls

♣ Polling mechanisms have large influence on throughput

♣ Throughput in PCF mode shows up to 20% variation with other configuration parameters – CFP repetition rate

♣ Saturation throughput of DCF less than PCF in all studies presented here (‘heavy load’ conditions)

**--> \*Manet applications in real-world**

**MANET Applications**

◼ ad hoc conferencing

◼ home networking

◼ emergency services

◼ personal area network (PAN)

◼ ubiquitous computing

❖ “computers are all around us, constantly performing mundane tasks to make our lives a litter easier”

❖ “Ubiquitous intelligent internetworking devices that detect their environment, interact with each other, and respond to changing environmental condition will create a future that is as challenging to imagine as a science fiction scenario.”

**MANET applications – military**

- unknown terrain

- Limit the range of communication – directional antennas

- destroyed infrastructure

**- disaster relief**

**-** earthquakes, tsunamis, hurricanes

- wiped out infrastructure

- search & rescue

**- economic & commercial**

- community mesh networks

- access extensions

- personal area networks (PANs)

- Ad hoc gamming (on subway, café, etc)

**--> MANET Routing: Reactive vs Proactive routing, AODV vs OLSR**

**MANET vs. Traditional Routing**

● Every node is potentially a router in a MANET, while most nodes in traditional wired networks do not route packets ■ Nodes transmit and receive their own packets and, also, forward packets for other nodes

● Topologies are dynamic in MANETs due to mobile nodes, but are relatively static in traditional networks

● Routing in MANETs must consider both Layer 3 and Layer 2 information, while traditional protocols rely on Layer 3 information only ■ Link layer information can indicate connectivity and interference

● MANET topologies tend to have many more redundant links than traditional networks

● A MANET “router” typically has a single interface, while a traditional router has an interface for each network to which it connects ■ Routed packet sent forward when transmitted, but also sent to previous transmitter

● Channel properties, including capacity and error rates, are relatively static in traditional networks, but may vary in MANETs

● Interference is an issue in MANETs, but not in traditional networks ■ For example, a forwarded packet from B-to-C competes with new packets sent from A-to-B

● Channels can be asymmetric with some Layer 2 technologies ■ Note that the IEEE 802.11 MAC assumes symmetric channels

● Power efficiency is an issue in MANETs, while it is normally not an issue in traditional networks

● MANETs may have gateways to fixed network, but are typically “stub networks,” while traditional networks can be stub networks or transit networks

● There is limited physical security in a MANET compared to a traditional network ■ Increased possibility of eavesdropping, spoofing, and denial-of-security attacks

● Traditional routing protocols for wired networks do not work well in most MANETs ■ MANETs are too dynamic ■ Wireless links present problems of interference, limited capacity, etc.

**MANET Routing**

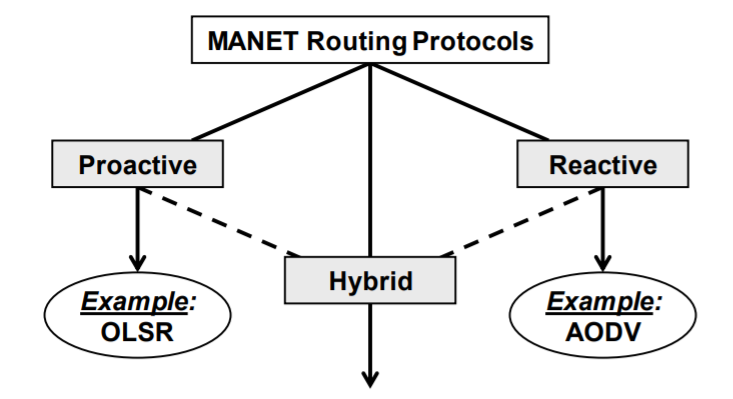
● Nodes must determine how to forward packets ■ Source routing: Routing decision is made at the sender ■ Hop-by-hop routing: Routing decision is made at each intermediate node

● Difficult to achieve good performance ■ Routes change over time due to node mobility ■ Best to avoid long delays when first sending packets ■ Best to reduce overhead of route discovery and maintenance ■ Want to involve as many nodes as possible – to find better paths and reduce likelihood of partitions

**MANET Routing Approaches**

● Decision time ■ Proactive or table-driven – maintain routing tables ■ Reactive or on-demand – determine routing on an as-needed basis

● Network structure ■ Hierarchical – impose a hierarchy on a collection of nodes and reflect this hierarchy in the routing algorithm ○ May use a proactive protocol for routing within a cluster or zone ○ May use a reactive protocol for routing between distinguished “cluster heads” ■ Non-hierarchical – make decisions among all nodes

****

**Common Features (các tính năng chung)**

● MANET routing protocols must… ■ Discover a path from source to destination ■ Maintain that path (e.g., if an intermediate node moves and breaks the path) ■ Define mechanisms to exchange routing information

● Reactive protocols ■ Discover a path when a packet needs to be transmitted and no known path exists ■ Attempt to alter the path when a routing failure occurs

● Proactive protocols ■ Find paths, in advance, for all source-pair destinations ■ Periodically exchange routing information to maintain paths

**OLSR**

The **Optimized Link State Routing Protocol** (**OLSR**)[[1]](https://en.wikipedia.org/wiki/Optimized_Link_State_Routing_Protocol#cite_note-1) is an [IP](https://en.wikipedia.org/wiki/Internet_Protocol) routing protocol optimized for [mobile ad hoc networks](https://en.wikipedia.org/wiki/Mobile_ad_hoc_network), which can also be used on other [wireless ad hoc networks](https://en.wikipedia.org/wiki/Wireless_ad_hoc_network). OLSR is a proactive [link-state routing protocol](https://en.wikipedia.org/wiki/Link-state_routing_protocol), which uses *hello* and *topology control* (TC) messages to discover and then disseminate link state information throughout the mobile ad hoc network. Individual nodes use this topology information to compute next hop destinations for all nodes in the network using shortest hop forwarding paths.

**OLSR Concepts**

● Proactive (table-driven) routing protocol ■ A route is available immediately when needed

● Based on the link-state algorithm ■ Traditionally, all nodes flood neighbor information in a linkstate protocol, but not in OLSR

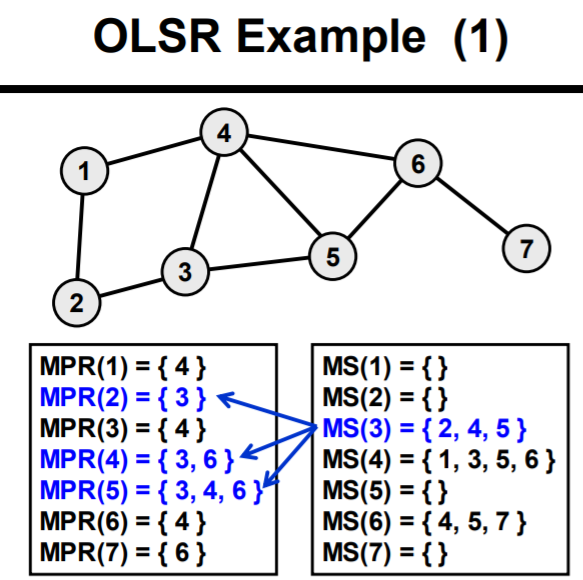
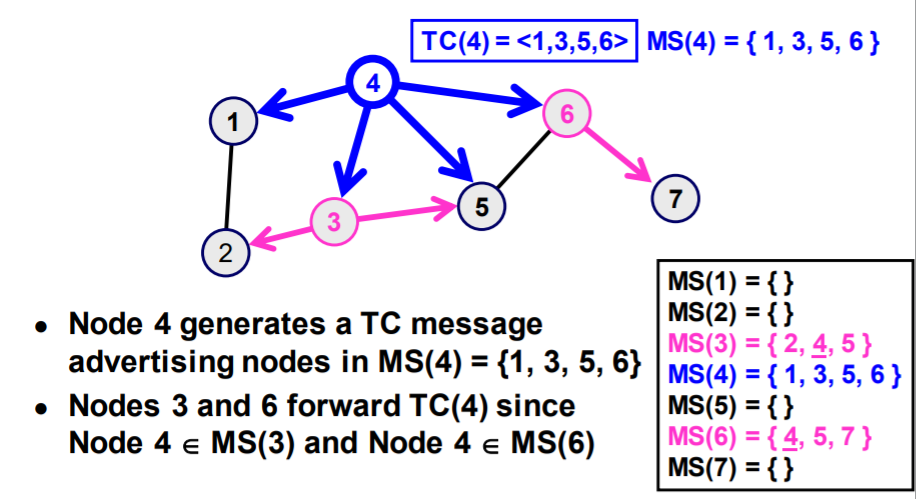
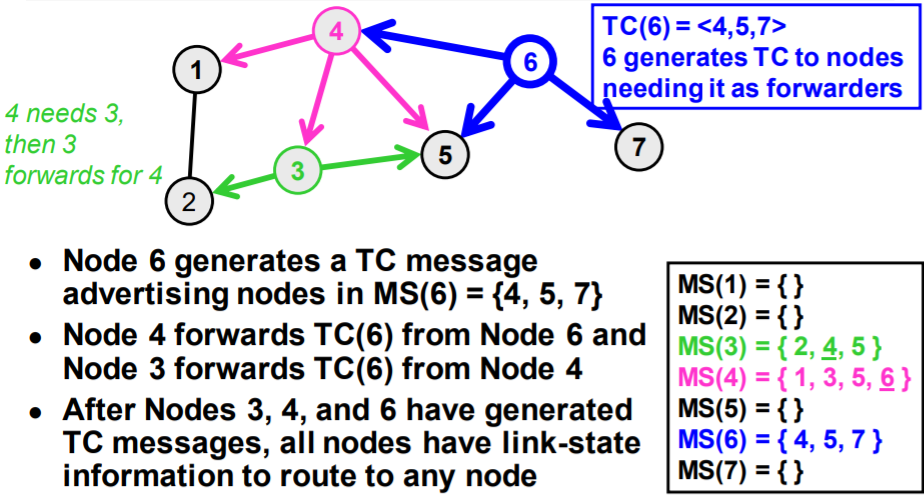
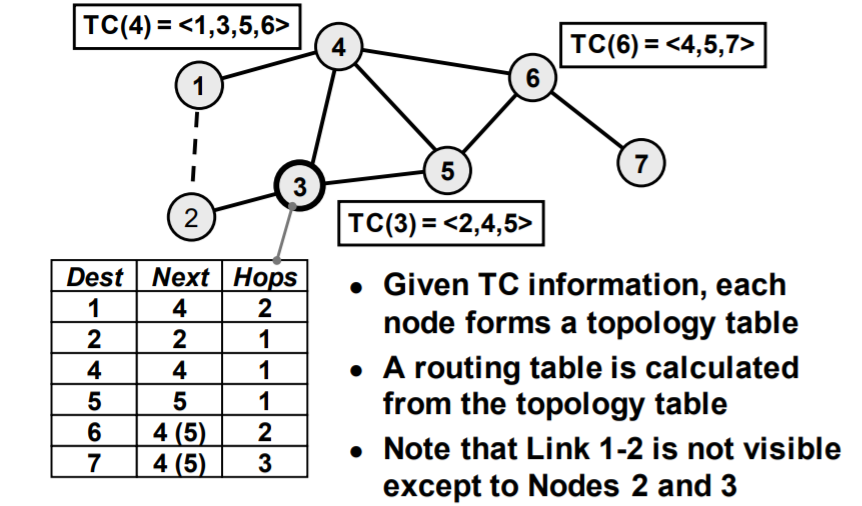
● Nodes advertise information only about links with neighbors who are in its multipoint relay selector set ■ Reduces size of control packets

● Reduces flooding by using only multipoint relay nodes to send information in the network ■ Reduces number of control packets by reducing duplicate transmissions

● Does not require reliable transfer, since updates are sent periodically

● Does not need in-order delivery, since sequence numbers are used to prevent out-of-date information from being misinterpreted

● Uses hop-by-hop routing ■ Routes are based on dynamic table entries maintained at intermediate nodes Introduction to Wireless Networks: IP Routing and M

**AODV**

**Ad hoc On-Demand Distance Vector** (**AODV**) **Routing** is a [routing protocol](https://en.wikipedia.org/wiki/Routing_protocol) for [mobile ad hoc networks](https://en.wikipedia.org/wiki/Mobile_ad_hoc_network) (MANETs) and other [wireless ad hoc networks](https://en.wikipedia.org/wiki/Wireless_ad_hoc_network). It was jointly developed on July 2003 in [Nokia Research Center](https://en.wikipedia.org/wiki/Nokia_Research_Center), [University of California, Santa Barbara](https://en.wikipedia.org/wiki/University_of_California,_Santa_Barbara) and [University of Cincinnati](https://en.wikipedia.org/wiki/University_of_Cincinnati) by C. Perkins, [E. Belding-Royer](https://en.wikipedia.org/wiki/Elizabeth_Belding) and [S. Das](https://en.wikipedia.org/wiki/Samir_Das).[[1]](https://en.wikipedia.org/wiki/Ad_hoc_On-Demand_Distance_Vector_Routing#cite_note-1)

AODV is the routing protocol used in [ZigBee](https://en.wikipedia.org/wiki/ZigBee) – a low power, low data rate [wireless ad hoc network](https://en.wikipedia.org/wiki/Wireless_ad_hoc_network). There are various implementations of AODV such as MAD-HOC, Kernel-AODV, AODV-UU, AODV-UCSB and AODV-UIUC.[[2]](https://en.wikipedia.org/wiki/Ad_hoc_On-Demand_Distance_Vector_Routing#cite_note-2)

**AODV Concepts**

● Pure on-demand routing protocol ■ A node does not perform route discovery or maintenance until it needs a route to another node or it offers its services as an intermediate node ■ Nodes that are not on active paths do not maintain routing information and do not participate in routing table exchanges

● Uses a broadcast route discovery mechanism

● Uses hop-by-hop routing ■ Routes are based on dynamic table entries maintained at intermediate nodes ■ Similar to Dynamic Source Routing (DSR), but DSR uses source routing

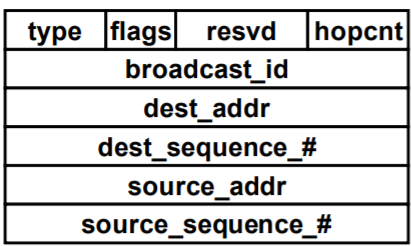
● Local HELLO messages are used to determine local connectivity ■ Can reduce response time to routing requests ■ Can trigger updates when necessary

● Sequence numbers are assigned to routes and routing table entries ■ Used to supersede stale cached routing entries

● Every node maintains two counters ■ Node sequence number ■ Broadcast ID

**AODV Route Request**

● Initiated when a node wants to communicate with another node, but does not have a route to that node

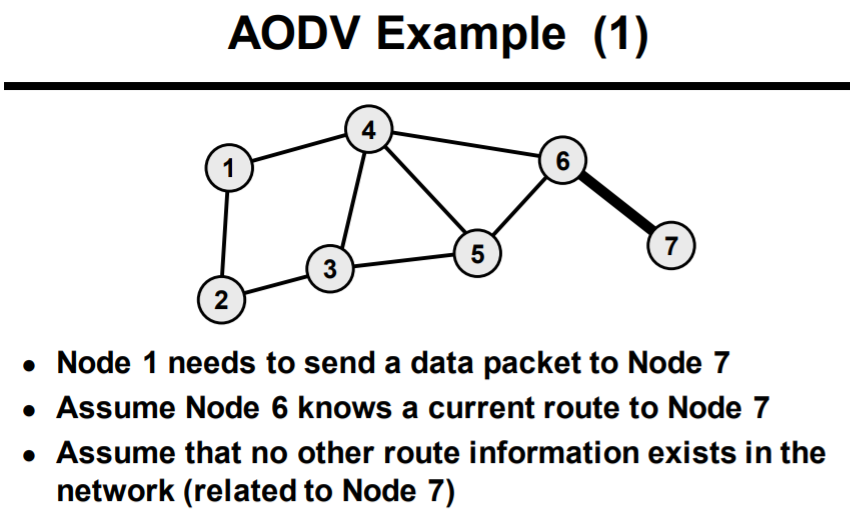
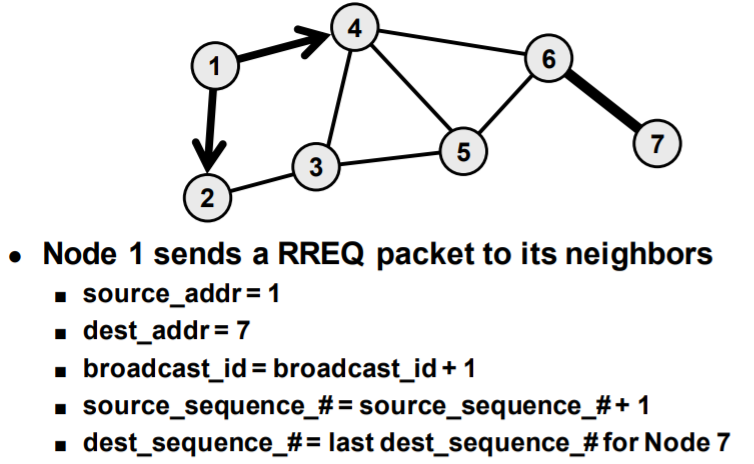
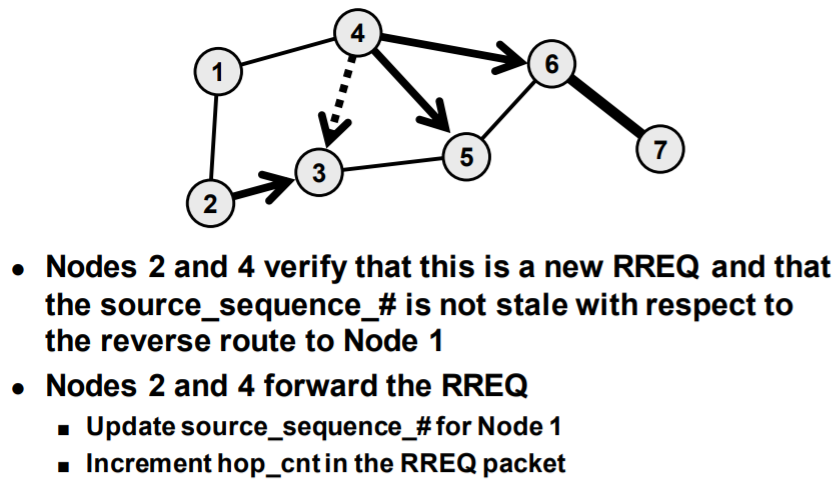
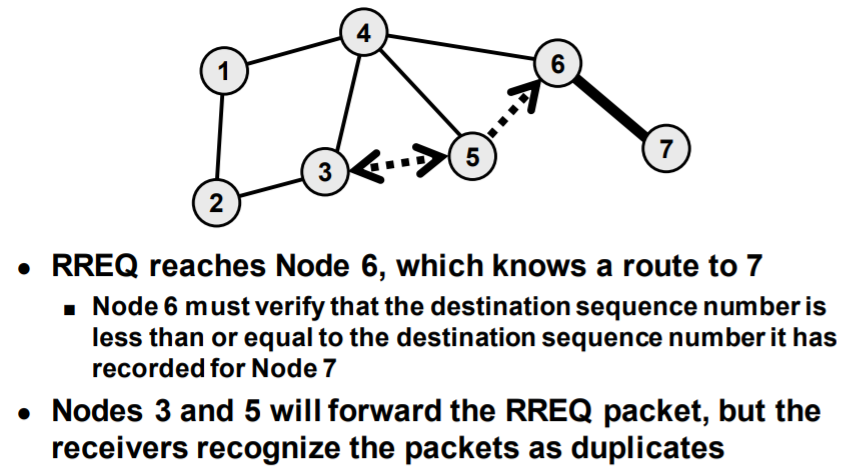
● Source node broadcasts a route request (RREQ) packet to its neighbors 

● Sequence numbers ■ Source sequence indicates “freshness” of reverse route to the source ■ Destination sequence number indicates freshness of route to the destination

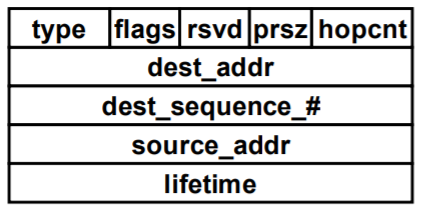
● Every neighbor receives the RREQ and either … ■ Returns a route reply (RREP) packet, or ■ Forwards the RREQ to its neighbors

● (source\_addr, broadcast\_id) uniquely identifies the RREQ ■ broadcast\_id is incremented for every RREQ packet sent ■ Receivers can identify and discard duplicate RREQ packets

● If a node cannot respond to the RREQ ■ The node increments the hop count ■ The node saves information to implement a reverse path set up (AODV assumes symmetrical links) ○ Neighbor that sent this RREQ packet ○ Destination IP address ○ Source IP address ○ Broadcast ID ○ Source node’s sequence number ○ Expiration time for reverse path entry (to enable garbage collection)

**** ****  

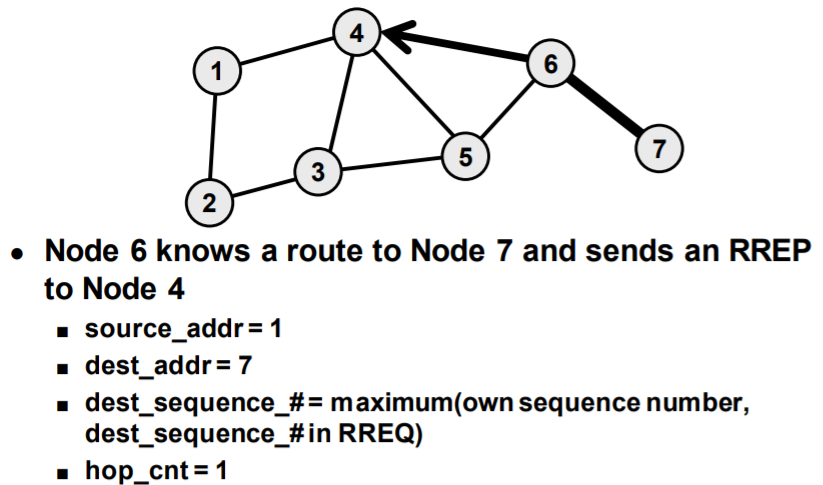
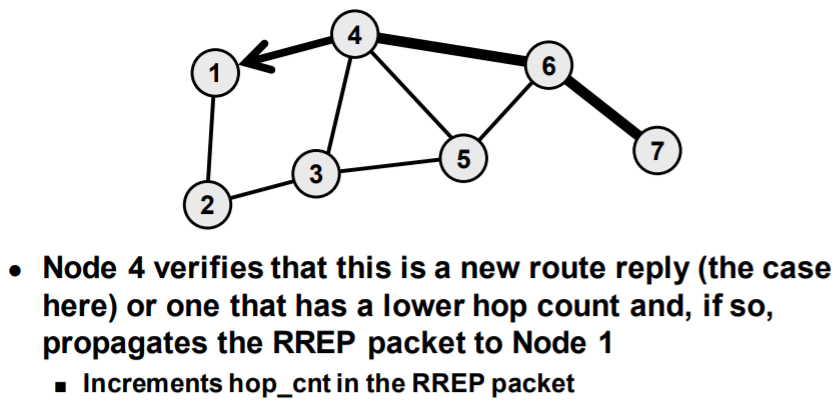
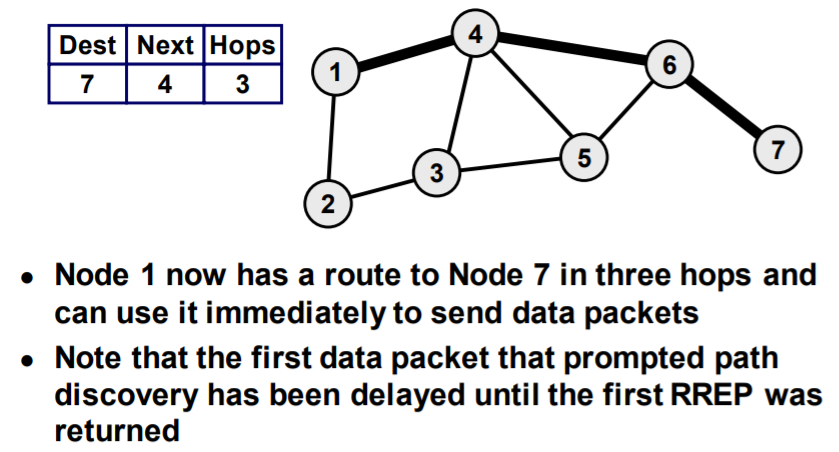
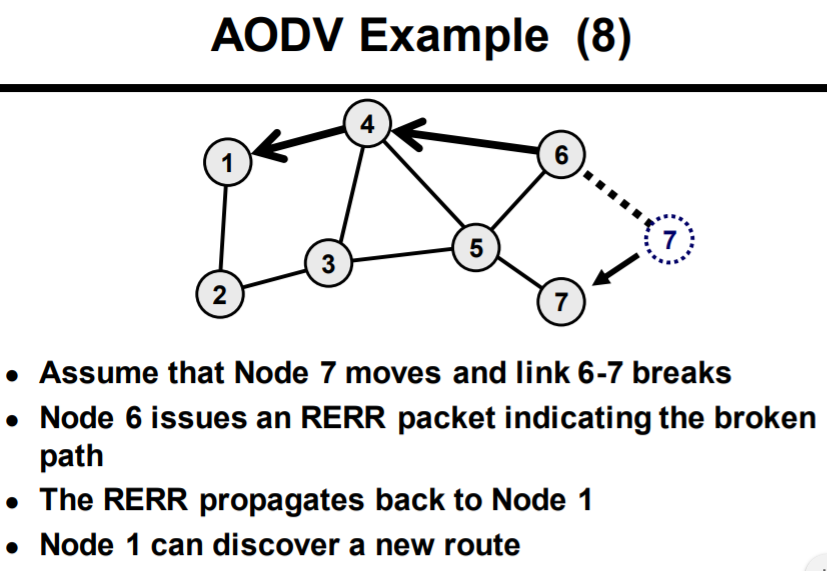
AODV Route Reply

● If a node receives an RREQ packet and it has a current route to the target destination, then it unicasts a route reply packet (RREP) to the neighbor that sent the RREQ packet 

● Intermediate nodes propagate the first RREP for the source towards the source using cached reverse route entries

● Other RREP packets are discarded unless… ■ dest\_sequence\_# number is higher than the previous, or ■ destination\_sequence\_# is the same, but hop\_cnt is smaller (i.e., there’s a better path)

● RREP eventually makes it to the source, which can use the neighbor sending the RREP as its next hop for sending to the destination

● Cached reverse routes will timeout in nodes not seeing a RREP packet  
****   

**AODV Route Maintenance**

● Route changes can be detected by… ■ Failure of periodic HELLO packets ■ Failure or disconnect indication from the link level ■ Failure of transmission of a packet to the next hop (can detect by listening for the retransmission if it is not the final destination)

● The upstream (toward the source) node detecting a failure propagates an route error (RERR) packet with a new destination sequence number and a hop count of infinity (unreachable)

● The source (or another node on the path) can rebuild a path by sending a RREQ packet

**iii)flows of operations [network kernel, drivers, packet transmission,packet reception/forwarding] 30-40%**

**--> Netfilter packet flow description**

**Netfilter** is a [framework](https://en.wikipedia.org/wiki/Software_framework) provided by the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) that allows various [networking](https://en.wikipedia.org/wiki/Computer_network)-related operations to be implemented in the form of customized handlers. Netfilter offers various functions and operations for [packet filtering](https://en.wikipedia.org/wiki/Packet_filter), [network address translation](https://en.wikipedia.org/wiki/Network_address_translation), and [port translation](https://en.wikipedia.org/wiki/Port_translation), which provide the functionality required for directing packets through a network and [prohibiting](https://en.wikipedia.org/wiki/Firewall_(computing)) packets from reaching sensitive locations within a network.

Netfilter represents a set of [hooks](https://en.wikipedia.org/wiki/Hooking) inside the Linux kernel, allowing specific [kernel modules](https://en.wikipedia.org/wiki/Kernel_module) to register [callback](https://en.wikipedia.org/wiki/Callback_(computer_programming)) functions with the kernel's networking stack. Those functions, usually applied to the traffic in the form of filtering and modification rules, are called for every packet that traverses the respective hook within the networking stack

**--> List of actions in Netfilter and their meaning: Accept, Recject, Drop**

ALLOW (aka ACCEPT)

Pemit a packet to traverse the firewall. This would be the behaviour if the firewall was not present.

REJECT

Prohibit a packet from passing. Send an ICMP destination-unreachable back to the source host [unless the icmp would not normally be permitted, eg. if it is to/from the broadcast address].

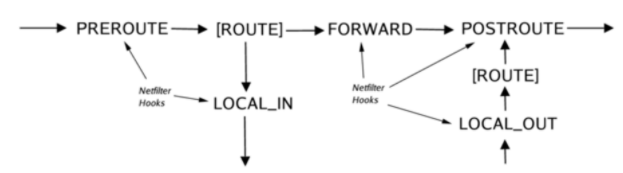
DROP (aka DENY, BLACKHOLE)

Prohibit a packet from passing. Send no response.

Netfilter is an internal Linux kernel program that provides flexibility for different network operations to accommodate different processes. Netfilter has options such as: packet filtering, network address translation, and port translation.

Netfilter's methods will provide guidance for packets traveling through the network as well as the work of preventing packets within vulnerable range in a computer network.

Netfilter represents a set of hooks in a computer network (in programming hooks include a series of techniques used to change or enhance the behavior of an operating system, applications, or software components. other by blocking functions such as calls, messages or events between software components) so it allows defining the modules to register the callback function to the kernel's network stack (the callback function is the function of the user called in another function or procedure during the operation, a function written to invoke the callback function, its argument must be the type of pointer that allows to declare the callback function).

The following is the basic pattern of a series of hooks: 

The left part is where the Packets package goes: through simple correct checksum (such as IP checksum) they pass through the netfilter framework hook NF\_IP\_PRE\_ROUTING [1]. Then they enter the transitional code, they will be decided whether they will be attached to another interface or will be processed too. The transition code can also take care to delete packets.

If it is attached to the original interface, the netfilter framework will be called to the NF\_IP\_LOCAL\_IN [2] hook, before passing to be processed.

If it is attached to another interface, the netfilter framework will call the NF\_IP\_FORWARD hook [3].

The packet then passes the last netfilter hook, the NF\_IP\_POST\_ROUTING [4] hook, before being put on the chain again.

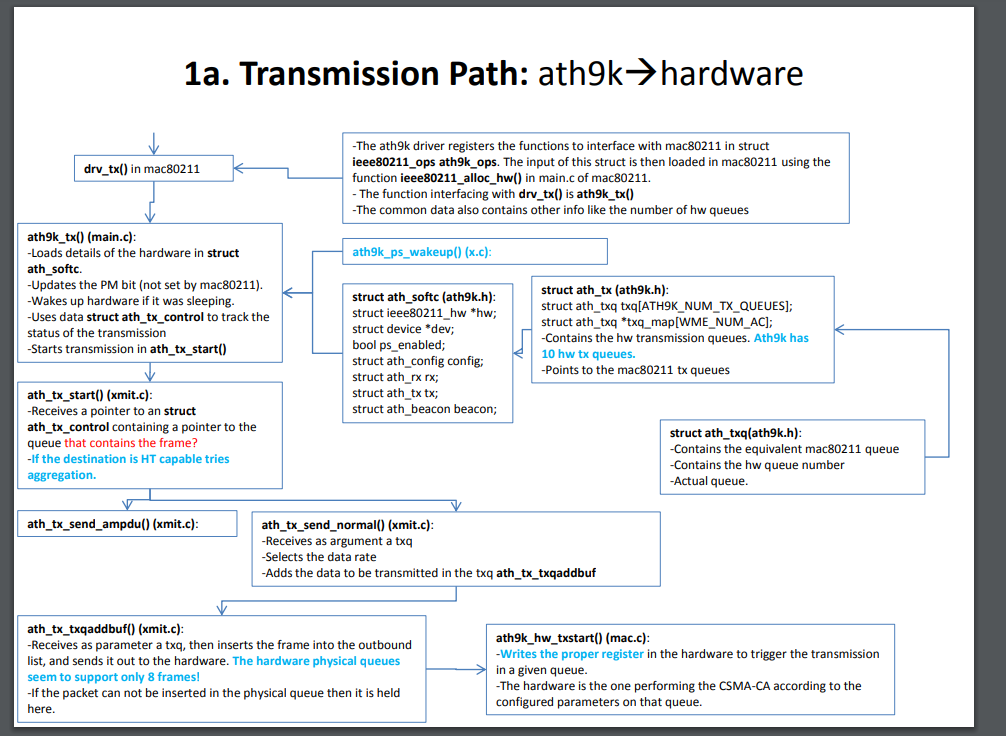
Hook NF\_IP\_LOCAL\_OUT [5] will be called for packets generated locally. Here you can see the route after the hook is called: in fact the forwarding code will be called first (to configure the IP address and some IP options).

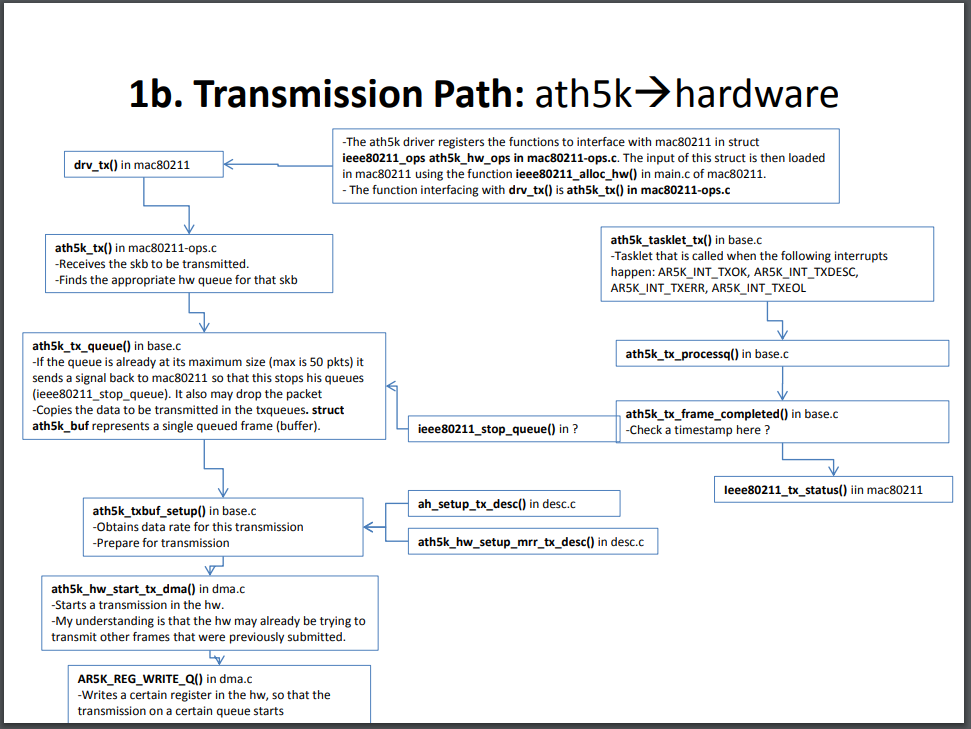
**--> List of hooks and their definition**

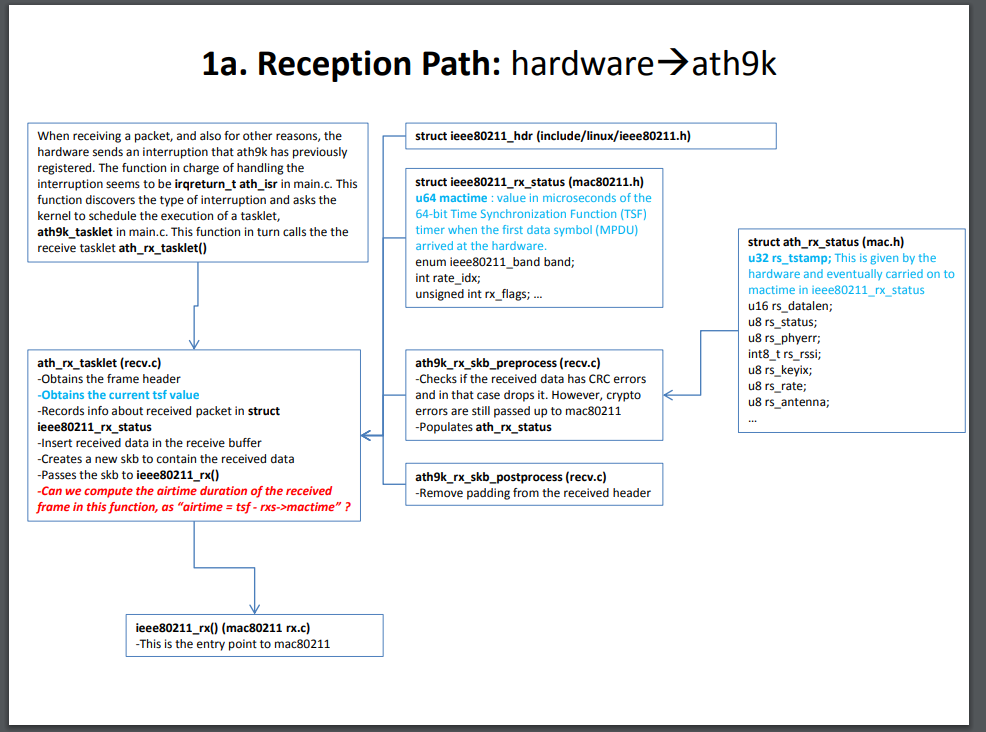
|  |  |
| --- | --- |
| **Hook** | **Được gọi lúc** |
| NF\_INET\_PRE\_ROUTING | Before routing decisions, are activated as soon as the packet enters the network |
| NF\_INET\_LOCAL\_IN | After routing decisions, if packets go to this host |
| NF\_INET\_FORWARD | If packets are passed to another interface |
| NF\_INET\_LOCAL\_OUT | Before packets locally were sent |
| NF\_INET\_POST\_ROUTING | Before packets can be sent or forwarded after routing. |

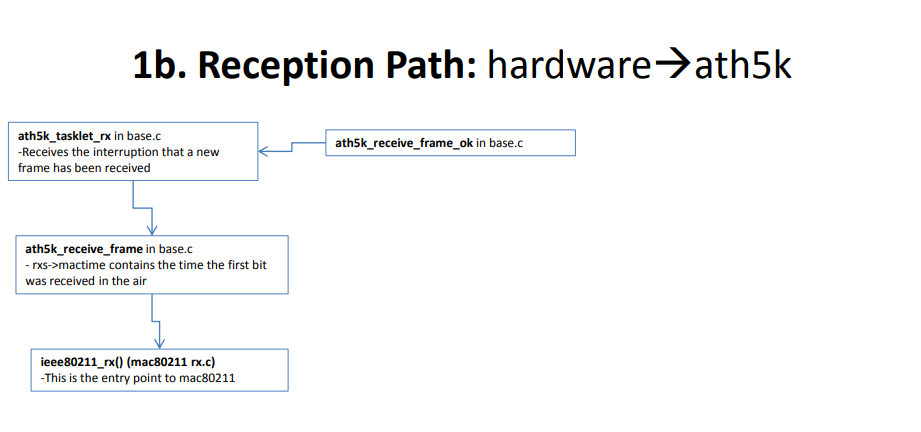
**--> Linux TCP/IP Stack packet flow description**

**--> Linux kernel ath5k, ath9k driver**









**iv) alternative questions {projects}20%**

**--> Your project 1: ideas, devices type, network architecture**

**Ideas:** With the development of computer vison, the use of cameras to recognize objects has become more and more popular. In this project, I have built a fire detection application based on Jetson Nano device.

**Device types:** Jetson nano is a small computer with Jetpack operating system with AI modules built in. Especially in computer vision, the Jetson nano is an indispensable device.

**Architecture**

In the fire detection application we build on the Yolov4 model.

Model yolov4 works based on the following steps: training input data, configuring parameters (number of layers to be detected, image size, scan frequency, ...), run the model. In the image to find the fire, yolo will divide the image into a grid of 7x7 squares. The model will then locate the flame to be detected based on its similarity to the trained data. The detection of a flame is based on the following principle: at each square that yolo has divided, it will determine if there is any flame in it. If it does, it sends the x, y, height, and length parameters of the image and then a bounding box will be drawn around the flame to be detected. the output will be a frame containing the images surrounded by the bounding box.