COMP90007 Internet Technology

Week7

Yiran (Scott) Ruan

Email: yrrua@unimelb.edu.au

Web Page: https://yiranruan.github.io

Question 1

- Six stations, A through F, communicate using the MACA protocol. Is it possible that two transmissions take place simultaneously? Explain your answer.

Two problems

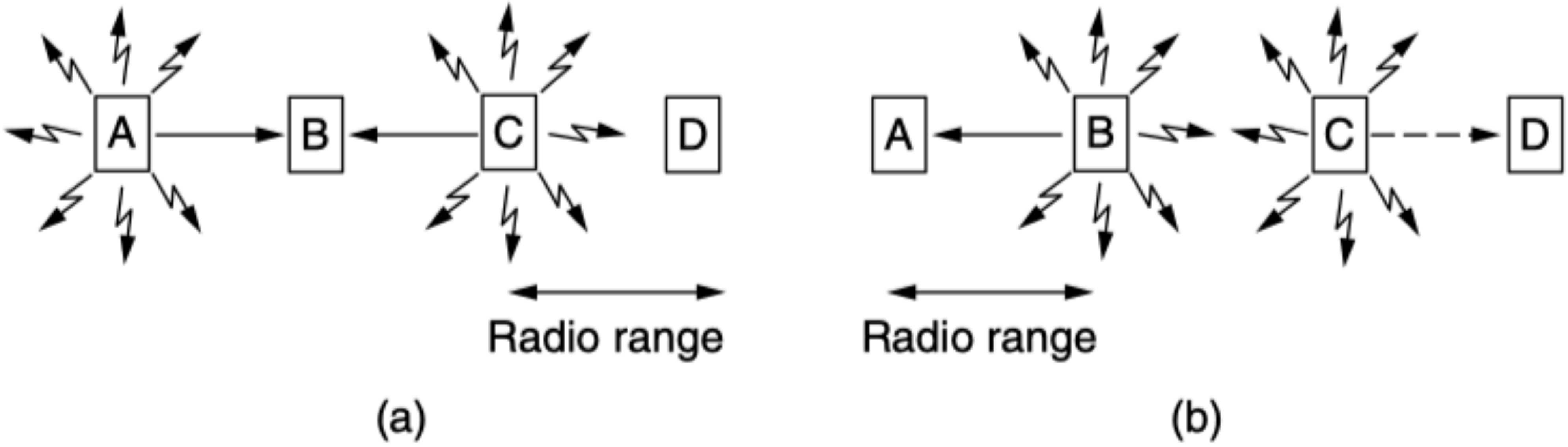

Figure 4-11. A wireless LAN. (a) *A* and *C* are hidden terminals when transmitting to *B*. (b) *B* and *C* are exposed terminals when transmitting to *A* and *D*.

Two problems

- The difficulty is that, before starting a transmission, a station really wants to know whether there is radio activity around the receiver.

MACA
(Multiple Access with Collision Avoidance)

- **Basic idea**: for the sender to stimulate the receiver into outputting a short frame, so stations nearby can detect this transmission and avoid transmitting for the duration of the upcoming (large) data frame. This technique is used instead of carrier sense.

MACA
(Multiple Access with Collision Avoidance)

Short-range radio waves

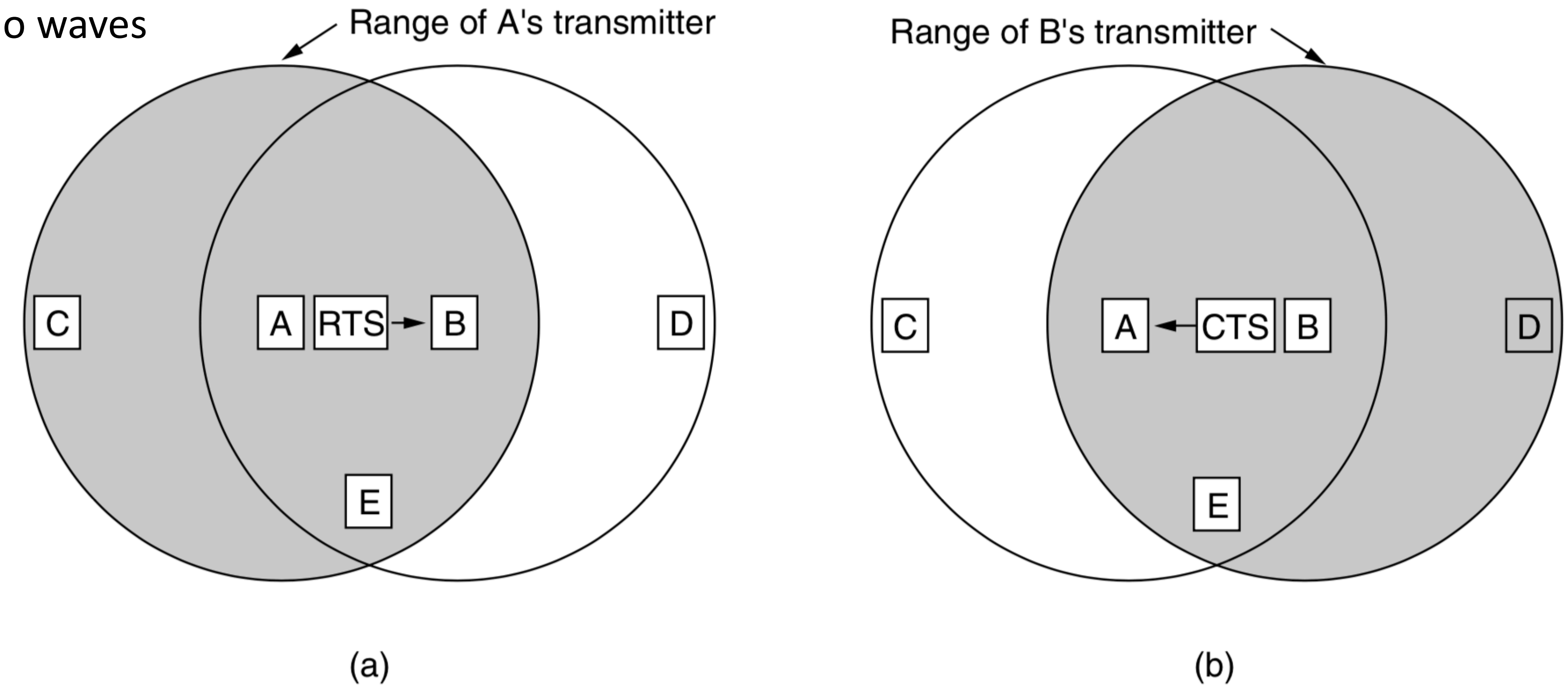

Figure 4-12. The MACA protocol. (a) *A* sending an RTS to *B*. (b) *B* responding with a CTS to *A*.

* Computer Network 279

MACA
(Multiple Access with Collision Avoidance)

RTS -> This short frame (30 bytes) contains the length of the data frame that will eventually follow.

CTS -> The CTS frame contains the data length (copied from the RTS frame).

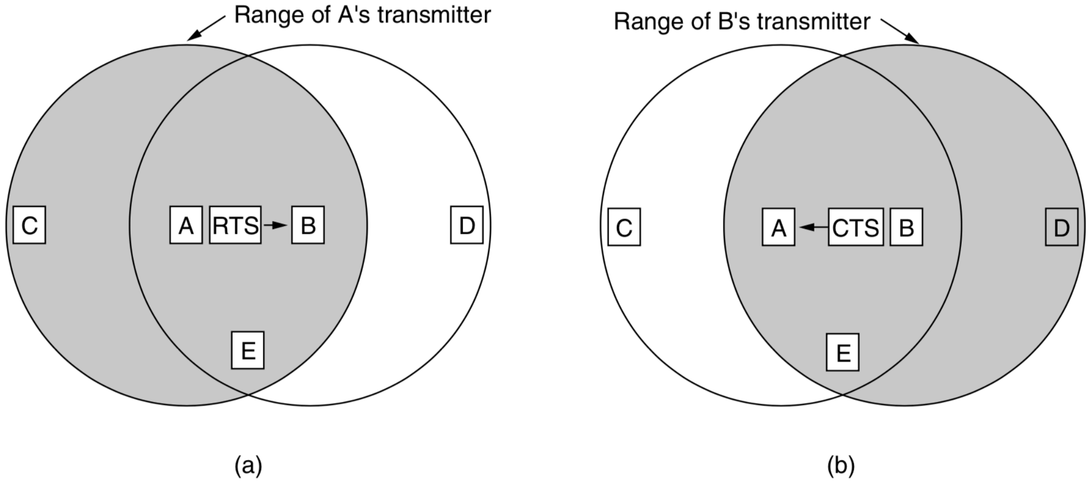

Figure 4-12. The MACA protocol. (a) *A* sending an RTS to *B*. (b) *B* responding with a CTS to *A*.

* Computer Network 279

Solution 1

- Yes. Imagine that they are in a straight line and that each station can reach only its nearest neighbours. Then A can send to B while E is sending to F.

Question 2

- If there are n independent paths between two nodes in a network, and the probability that an individual path is working is p, what is the probability of these two nodes being connected? Assume path failures are independent.
- Hint: first try to calculate what is the probability that all paths have failed

Solution 2

Answer:

Pr(nodes connected)

= 1 − Pr(no connection)

= 1 − Pr(all paths failed)

= 1 − Pr(individual path failure)^n (assuming independent events)

= 1 − [1 − Pr(individual path working)]^n

= $1 - (1 - p)^n$

Question 3

- A network on the Internet has a subnet mask of `255.255.240.0`. What is the maximum number of hosts that it can handle?

IP Address

- 32 bits
- It is important to note that an IP address does not actually refer to a host. It really refers to a **network interface.**
    - if a host is on two networks, it must have two IP addresses.
    - in practice, most hosts are on one network and thus have one IP address.
    - 127.0.0.1

IP address

- IP address: Two portion
    - Network
        - the **same value** for all hosts on a single network
        - Corresponds to a contiguous block of IP address space (Prefix)
    - Host

Prefix

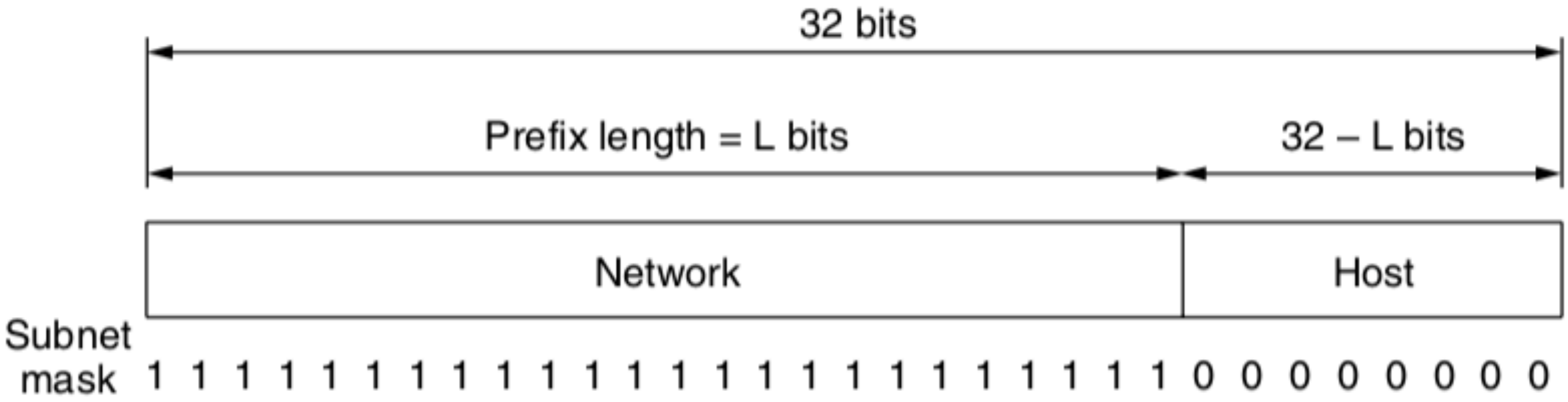

Figure 5-48. An IP prefix and a subnet mask.

Prefix

- Prefixes are written by giving the lowest IP address in the block and the size of the block.
- The size is determined by the number of bits in the network portion; the remaining bits in the host portion can vary.
- the size must be a power of two

Prefix

- For example, 128.208.42.151
- CONVERT to Binary format

10000000 11010000 00101010 10010111

- If the prefix contains **2^8** addresses, so leaves 19 bits for the network portion
- lowest IP address: 10000000 11010000 00100000 00000000
- Size: 19
- it is written as 128.208.32.0/19

Subnet mask

- The length of the prefix corresponds to a binary mask of 1s in the network portion.

11111111 11111111 11110000 00000000

- It can be ANDed with the IP address to extract only the network portion.

11111111 11111111 11110000 00000000

AND 10000000 11010000 00101010 10010111

10000000 11010000 00100000 00000000

Question 3

- A network on the Internet has a subnet mask of `255.255.240.0`. What is the maximum number of hosts that it can handle?

Solution 3

- Convert IP address to binary representation
- 255.255.240.0
- -> 11111111 11111111 11110000 0000000
- -> 11111111111111111111 00000000000
- -> 20 1s : the length of prefix is 20
- -> the number of bits leaved for host is

  32 – 20 = 12 bits
- -> $2^{12} = 4096$ host addresses

Question 4

- IPv6 uses 16 bytes addresses. If a block of 1 million addresses is allocated every picosecond, how long will the addresses last?

Solution 4

With 16 bytes there are 2^{128} IP addresses. If we allocate them at a rate of $10^6 / 10^{-12} = 10^{18}$ addresses per second. Therefore it will take 3.4×10^{20} seconds to run out of IP addresses, which is about 10^{13} years.

This number is 1000 times the age of the universe. Of course, the address space is not flat, so they are not allocated linearly, but this calculation shows that even with an allocation scheme that has an efficiency of 1/1000 (0.1 percent), one will never run out.

Question 5

- A router an entry in its table that can be represented with mask as `135.46.56.0/21`. What is the maximum number of hosts that this network can represent?

Solution 5

- `135.46.56.0/21`
- 21 bits means network has 21 bits reserved, and remaining 11 bits are for hosts.
- Hence maximum number of hosts is 2^11 = 2048

Question 6

- What are the benefits and disadvantages of Transparent fragmentation in Network Layer?

Path MTU

- Hosts usually prefer to transmit large packets because this reduces packet overheads such as bandwidth wasted on header bytes.
- An obvious internetworking problem appears when a large packet wants to travel through a network whose **maximum packet size** is too small.
- **Path MTU** (**Path Maximum Transmission Unit**).
  - source does not usually know the path a packet will take through the network to a destination, so it certainly does not know how small packets must be to get there. This packet size is called the **Path MTU** (**Path Maximum Transmission Unit**).

fragmentation

- Allow routers to break up packets into **fragments**, sending each fragment as a separate network layer packet
- Issue: converting a large object into small fragments is considerably easier than the reverse process.

Two solutions

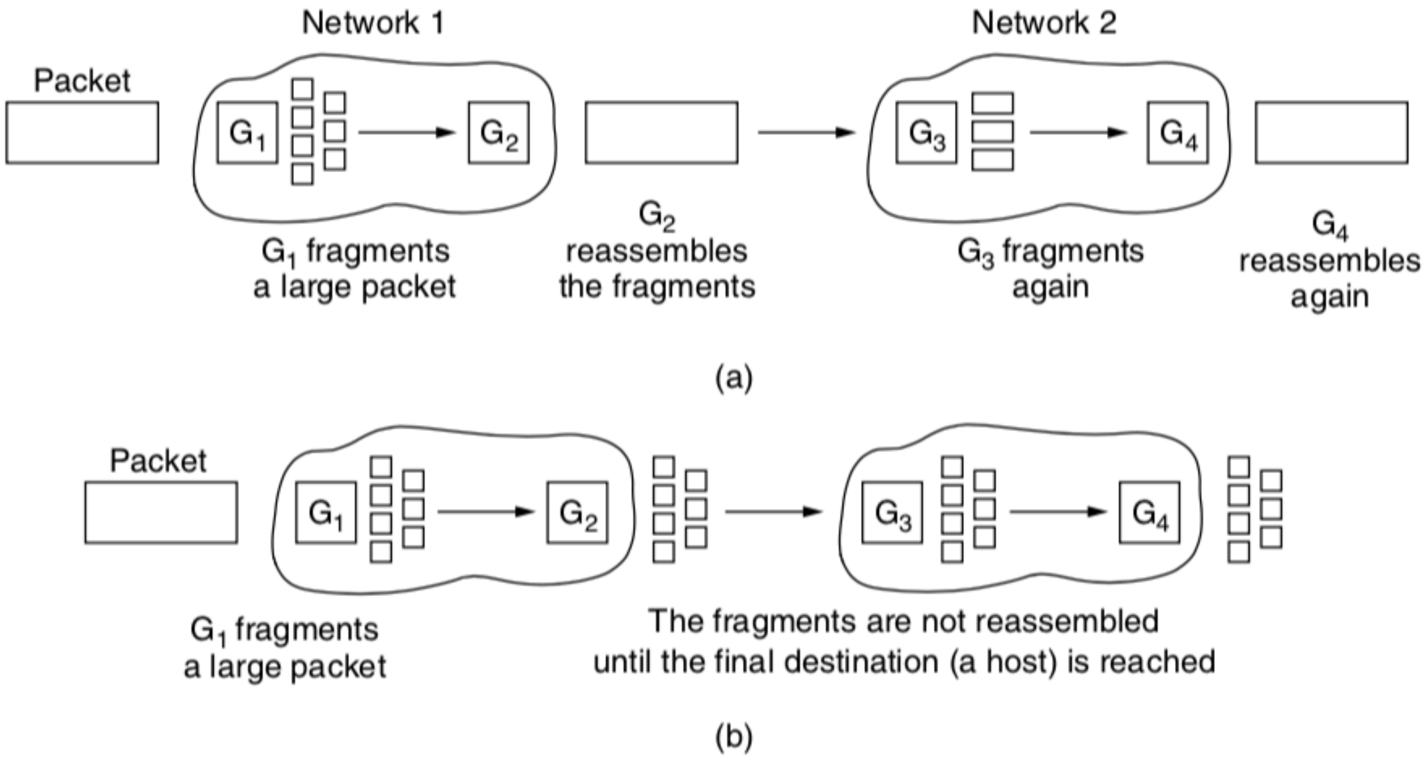
Network 1
Network 2
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
G_2 reassembles the fragments
G_3 fragments again
G_4 reassembles again
(a)
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached
(b)

Figure 5-42. (a) Transparent fragmentation. (b) Nontransparent fragmentation.

Transparent

- Transparent -> No awareness of fragmentation
- In this approach, when an oversized packet arrives at *G1*, the router breaks it up into fragments. Each fragment is addressed to the same exit router, *G2*, where the pieces are recombined. In this way, passage through the small-packet network is made transparent.
- Subsequent networks are not even aware that fragmentation has occurred.

Transparent

- The exit router must know when it has received all the pieces, so either a count field or an ‘‘end of packet’’ bit must be provided.
- Because all packets must exit via the same router so that they can be reassembled, the routes are constrained
- The amount of work that the router may have to do
    - buffer the fragments
    - decide when to throw them away if not all of the fragments arrive
    - be repeatedly fragmented and reassembled

Untransparent

- refrain from recombining fragments at any intermediate routers. Once a packet has been fragmented, each fragment is treated as though it were an original packet. The routers pass the fragments and reassembly is performed only at the destination host.
  - Less work on router

Solution 6

- Good design paradigm and encapsulation of fragmentation within each network. Transparent fragmentation is straightforward to implement and use but has problems. For one thing, the exit router must know when it has received all the pieces, so either a count field or an ‘‘end of packet’’ bit must be provided. Also, because all packets must exit via the same router so that they can be reassembled, the routes are constrained. By not allowing some fragments to follow one route to the ultimate destination and other fragments a disjoint route, some performance may be lost. More significant is the amount of work that the router may have to do. It may need to buffer the fragments as they arrive, and decide when to throw them away if not all of the fragments arrive. Some of this work may be wasteful, too, as the packet may pass through a series of small packet networks and need to be repeatedly fragmented and reassembled.