Internet Protocols EBU5403 Live Lecture C3/C4 Tutorial and Office hour

Module organiser: Richard Clegg (r.clegg@qmul.ac.uk)

Michael Chai (michael.chai@qmul.ac.uk)

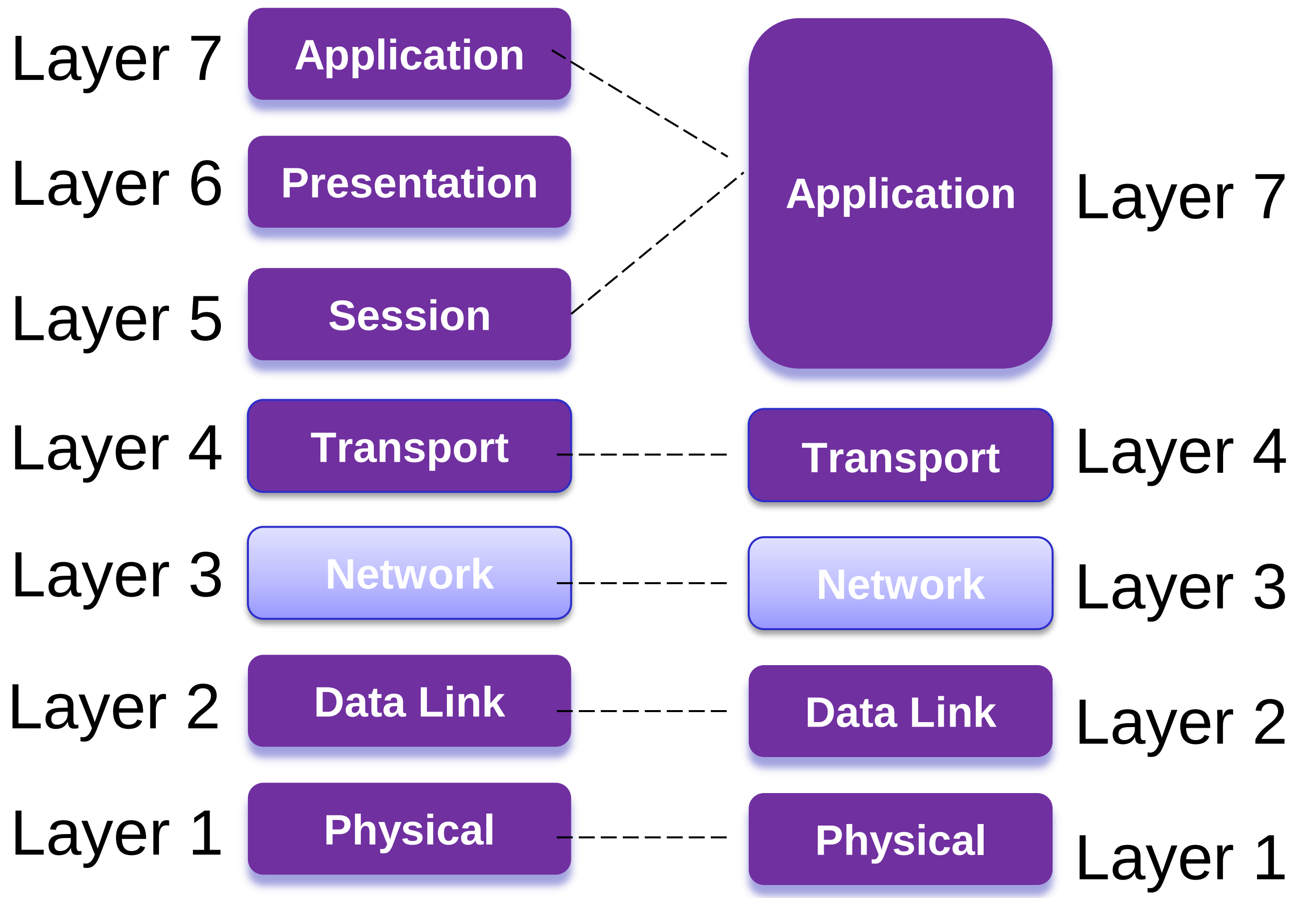
Cunhua Pan(c.pan@qmul.ac.uk)

<table>
<tr><th></th><th>Part 1</th><th>Part 2</th><th>Part 3</th><th>Part 4</th></tr>
<tr><td>Ecommerce + Telecoms 1</td><td colspan="2">Richard Clegg</td><td colspan="2" rowspan="2">Cunhua Pan</td></tr>
<tr><td>Telecoms 2</td><td colspan="2">Michael Chai</td></tr>
</table>

Structure of course

- Part A
  - Introduction to IP Networks
  - The Transport layer (part I)
- Part B
  - The Transport layer (part II)
  - The Network layer (part I)
  - Class test
- Part C
  - The Network layer (part II)
  - The Data link layer (part I)
  - Router lab tutorial (assessed lab work after this week)
- Part D
  - The Data link layer (part II)
  - Network management and security
  - Class test

Network Layer

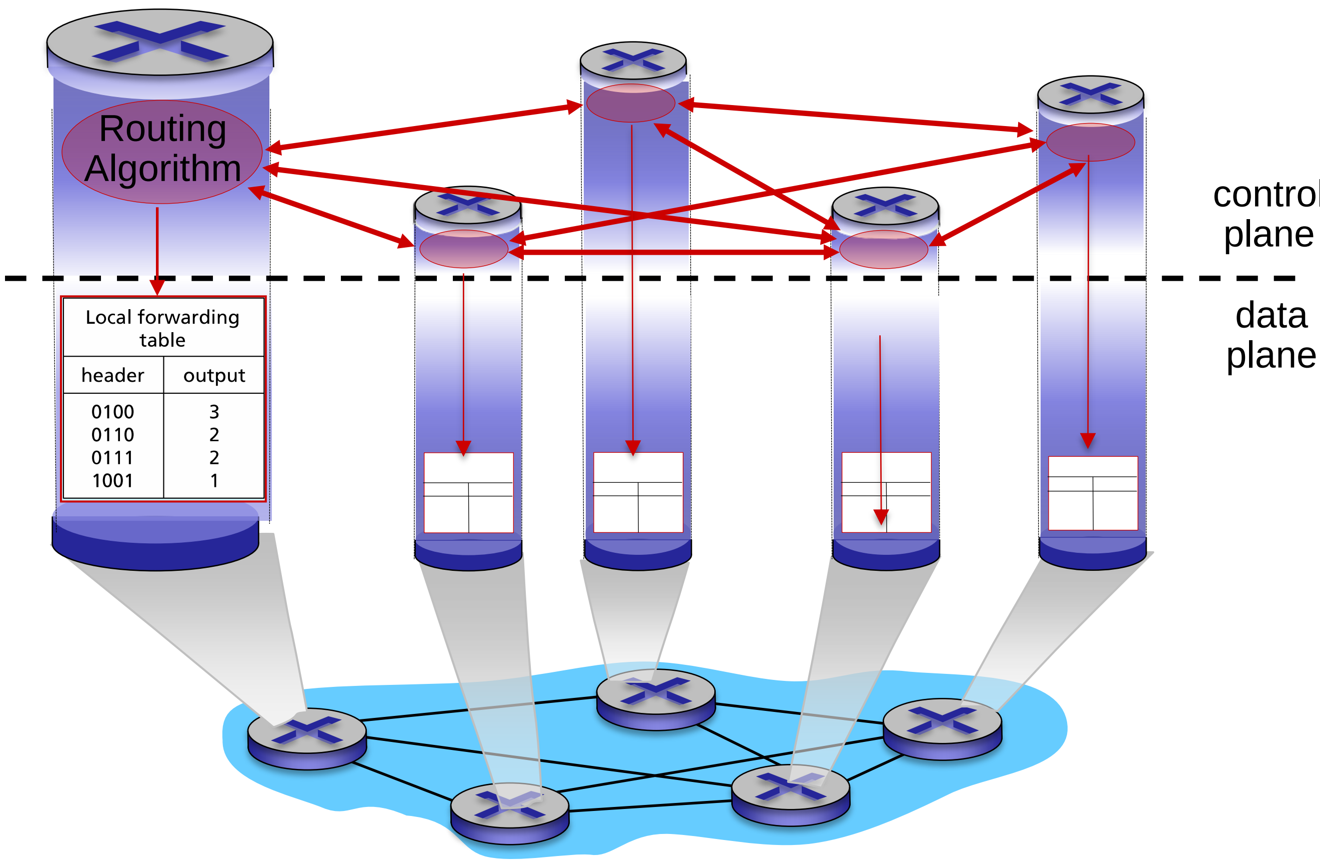


Reminder of lecture contents

- Lecture C3
  - SDN control plane
  - ICMP: The Internet Control Message Protocol
  - Lab demonstration
- Lecture C4
  - Introduction, services
  - Error detection, correction
- Tutorial

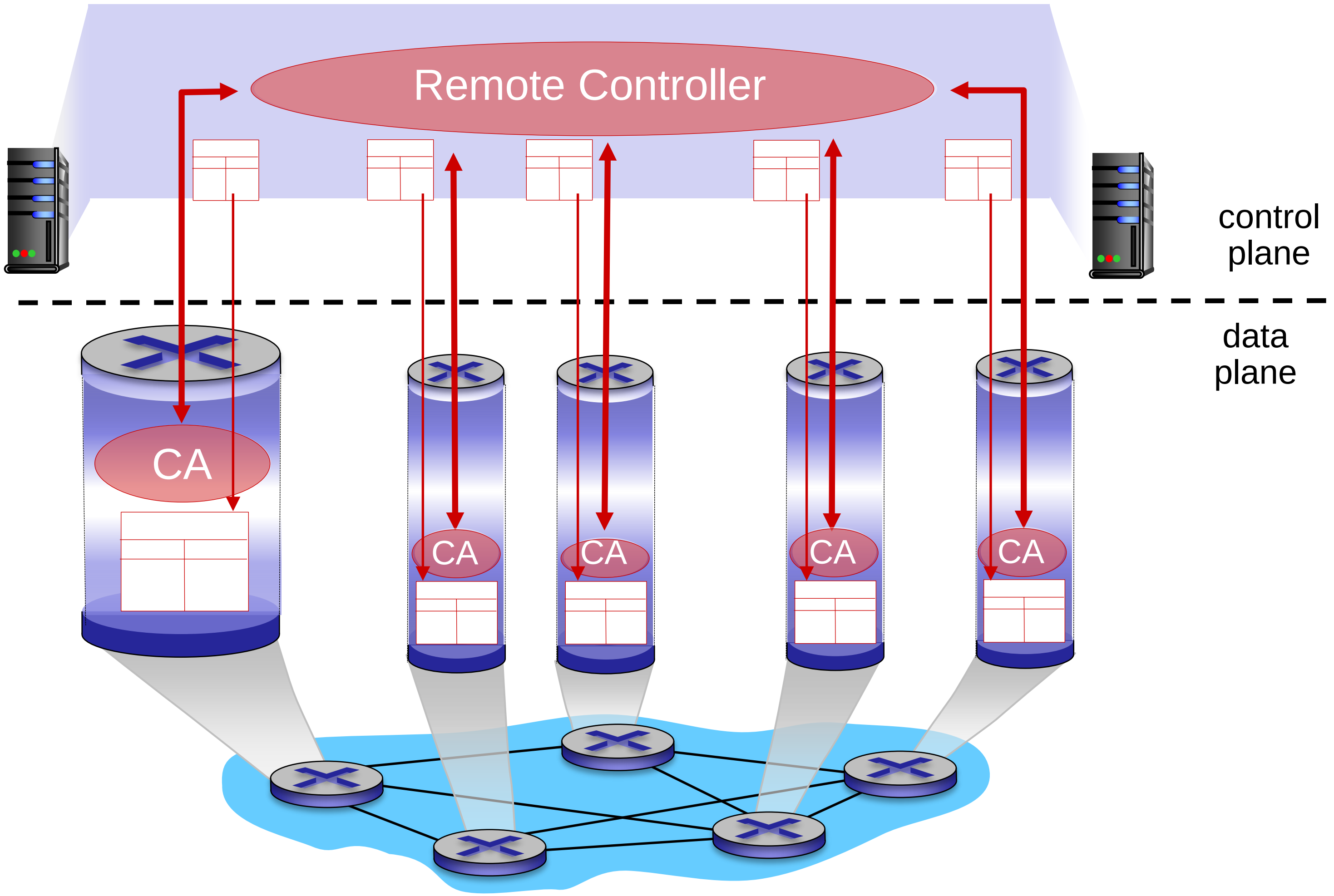
Recall: per-router control plane

Individual routing algorithm components *in each and every router* interact with each other in control plane to compute forwarding tables



Recall: logically centralized control plane

A distinct (typically remote) controller interacts with local control agents (CAs) in routers to compute forwarding tables



Software defined networking (SDN)

Question: *Why* a *logically centralized* control plane?

Software defined networking (SDN)

Question: Why a logically centralized control plane?

- easier network management: avoid router misconfigurations, greater flexibility of traffic flows
- table-based forwarding (recall OpenFlow API) allows "programming" routers
  - centralized "programming" easier: compute tables centrally and distribute
  - distributed "programming: more difficult: compute tables as result of distributed algorithm (protocol) implemented in each and every router
- open (non-proprietary) implementation of control plane

ICMP: internet control message protocol

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Traceroute example

Tracing route to www.bupt.edu.cn
[124.127.207.2]
(from QMUL)

In QMUL

1	2 ms	2 ms	2 ms	161.23.60.2
2	3 ms	3 ms	3 ms	172.23.22.17
3	2 ms	2 ms	13 ms	172.23.48.194
4	2 ms	2 ms	2 ms	172.23.56.1
5	3 ms	3 ms	3 ms	172.23.8.14
6	3 ms	2 ms	2 ms	172.23.8.18
7	2 ms	2 ms	2 ms	172.23.56.10
8	3 ms	2 ms	2 ms	172.23.52.17
9	2 ms	2 ms	4 ms	172.23.16.162

10	3 ms	3 ms	3 ms	146.97.143.217
11	3 ms	2 ms	3 ms	146.97.35.233
12	4 ms	4 ms	3 ms	146.97.33.1
13	3 ms	3 ms	3 ms	146.97.35.206

UK JANET (Joint Academic Network)
London

CHINANET (first hop is UK end of Connection)

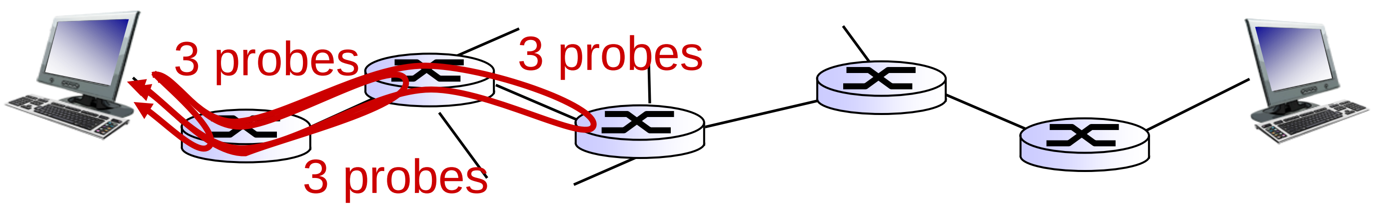
14	6 ms	5 ms	4 ms	202.97.52.97
15	189 ms	186 ms	189 ms	202.97.52.25
16	177 ms	177 ms	195 ms	202.97.53.245
17	178 ms	177 ms	175 ms	202.97.53.109
18	*	*	*	Request timed out.
19	175 ms	175 ms	232 ms	106.120.254.18
20	193 ms	199 ms	200 ms	124.127.161.242
21	200 ms	201 ms	201 ms	124.127.207.2

China Networks Internet eXchange Beijing.

Hop 18 does not respond to ICMP packets but allows them to pass on hence the * * *.

Traceroute and ICMP

- source sends series of UDP segments to destination
  - first set has TTL =1
  - second set has TTL=2, etc.
  - unlikely port number
- when datagram in *n*th set arrives to nth router:
  - router discards datagram and sends source ICMP message (type 11, code 0)
  - ICMP message include name of router & IP address



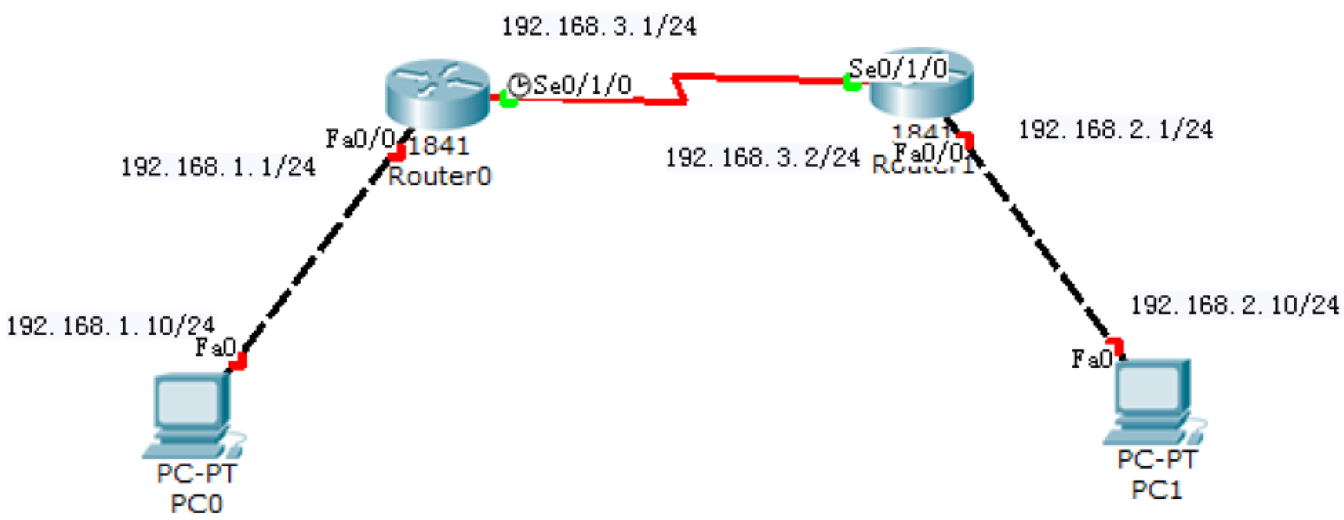
- when ICMP message arrives, source records RTTs

stopping criteria:

- UDP segment eventually arrives at destination host
- destination returns ICMP "port unreachable" message (type 3, code 3)
- source stops

Lab Demonstration

Exercise: Simple RIP network



Link layer services

- *framing, link access:*
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, destination
    - different from IP address!
- *reliable delivery between adjacent nodes*
  - we learned how to do this already (transport layer)!
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates
    - *Q:* why both link-level and end-end reliability?

Link layer services (more)

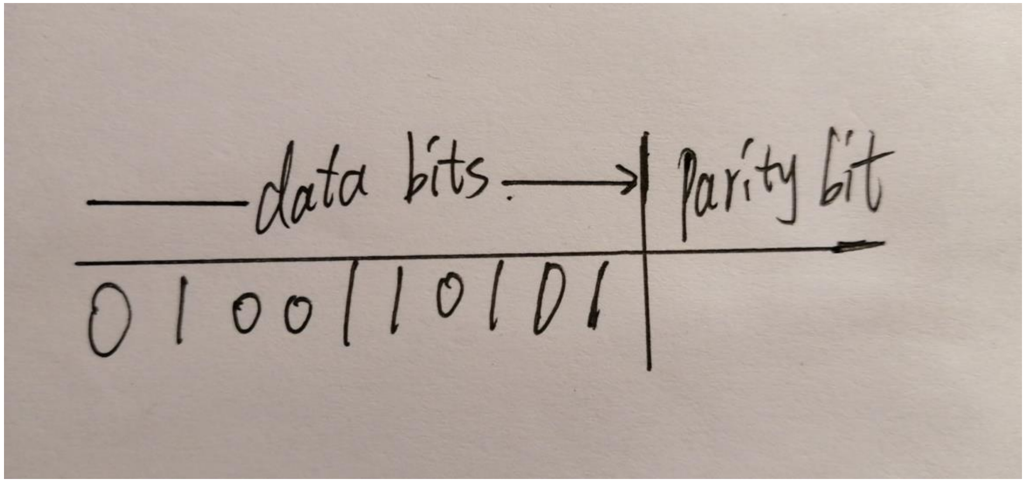
- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error detection:*
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- *error correction:*
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- *half-duplex and full-duplex*
  - with half duplex, nodes at both ends of link can transmit, but not at same time

Link layer services (more)

- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error detection:*
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- *error correction:*
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- *half-duplex and full-duplex*
  - with half duplex, nodes at both ends of link can transmit, but not at same time

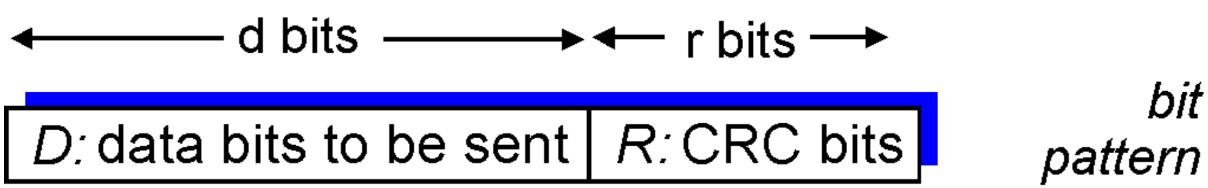
Parity checking

Question: What are the parity bits when using even parity checking and odd parity checking?



Cyclic redundancy check

- more powerful error-detection coding
- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi, ATM)



$D * 2^r \text{ XOR } R$ *mathematical formula*

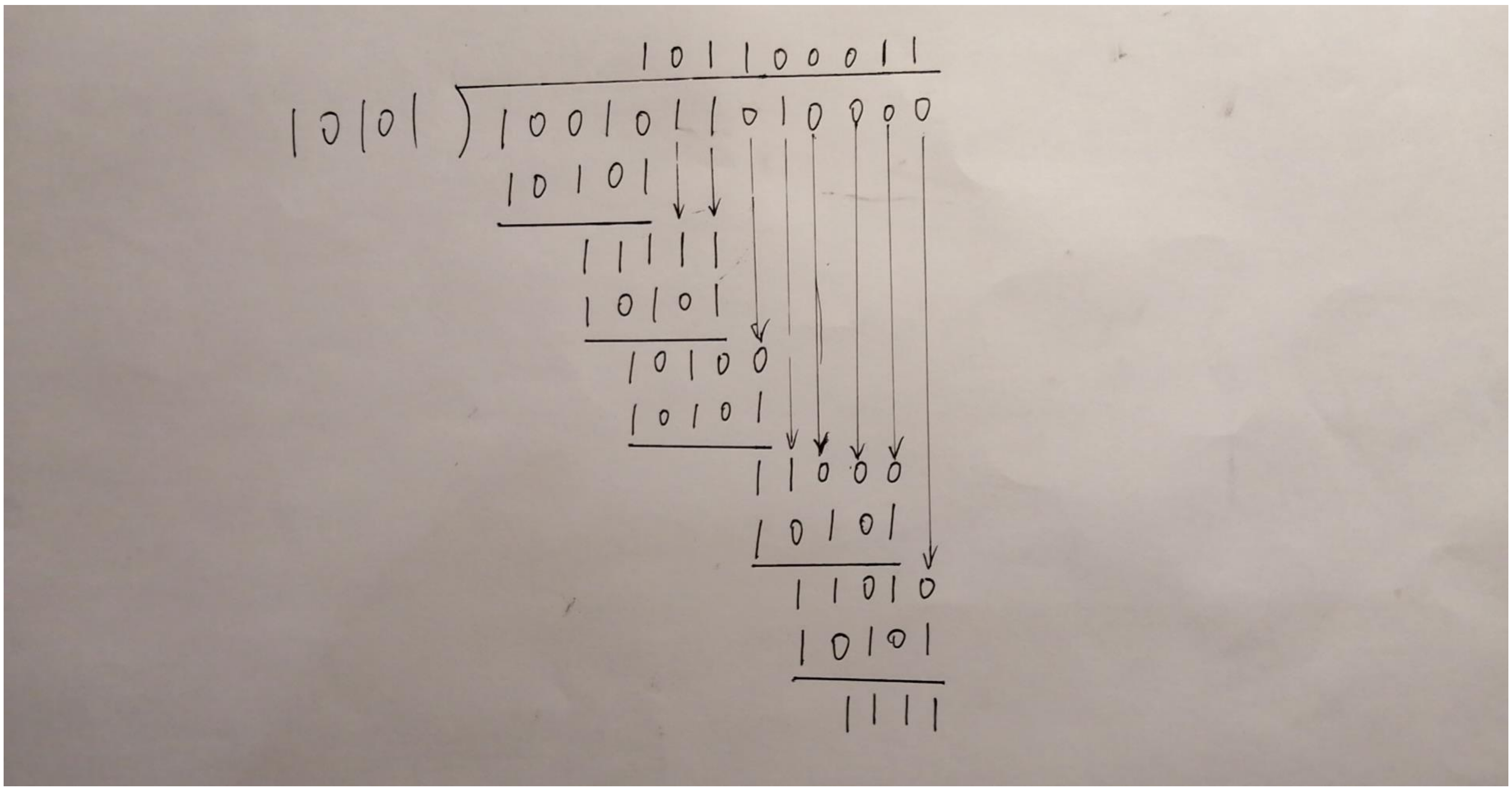
CRC Question

Using CRC, find CRC bit, R.

- Data bits, D=100101101
- Generator bits, G=10101
- CRC bits, R=?
- Frame=?

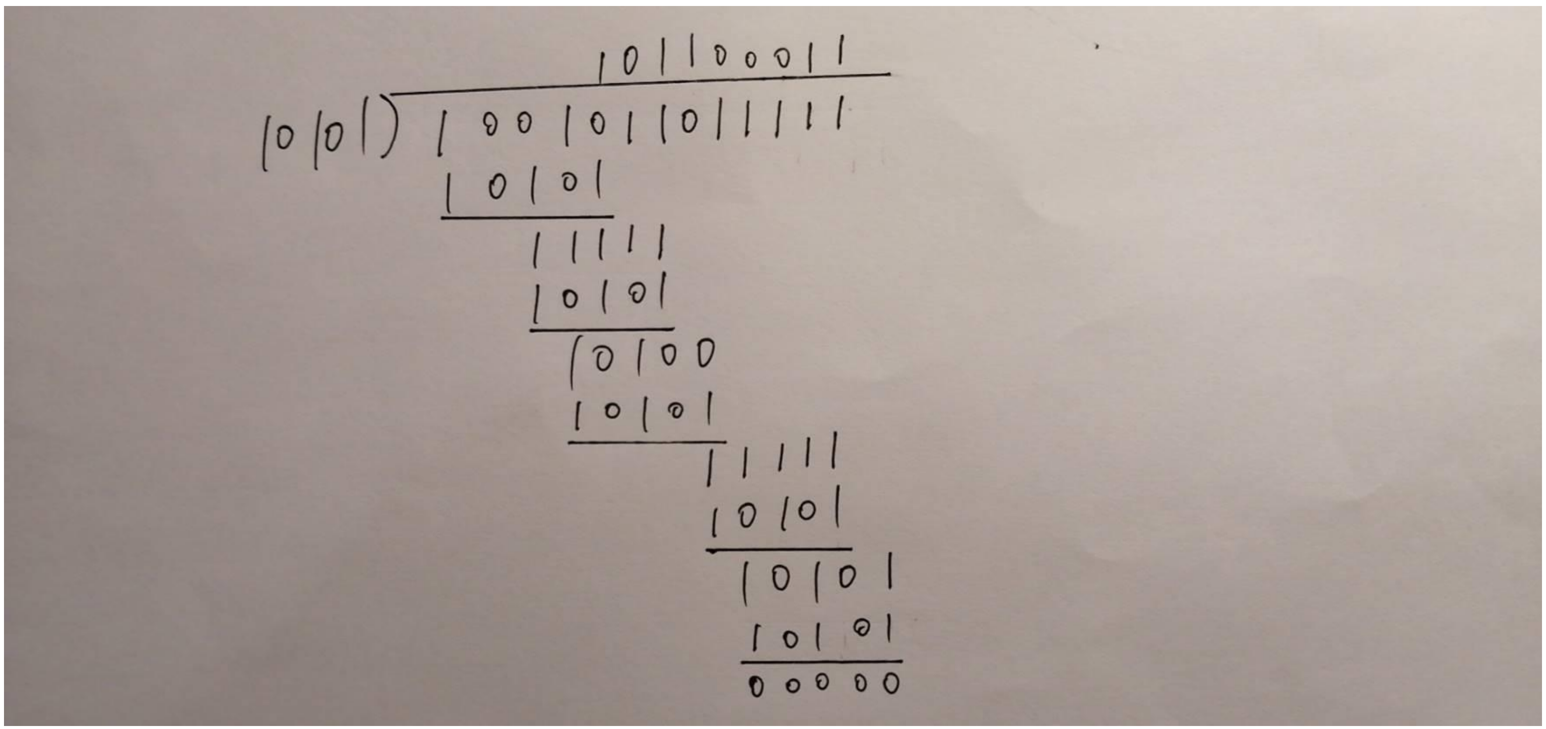
CRC Question

- Solutions



CRC Question

Frame: 1001011011111; G=10101



What have we learned

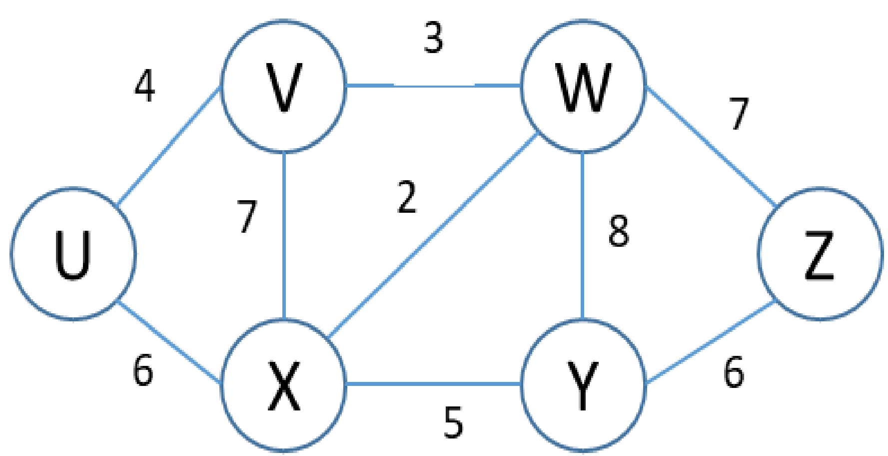
- Network layer (Final part)
  - SDN control plane
  - ICMP: The Internet Control Message Protocol
  - Lab demonstration
- Link layer (Part I)
  - Introduction, services
  - Error detection, correction

Tutorial/Office Hour

- Please feel free to ask questions if you do not understand.
- Do not be shy.

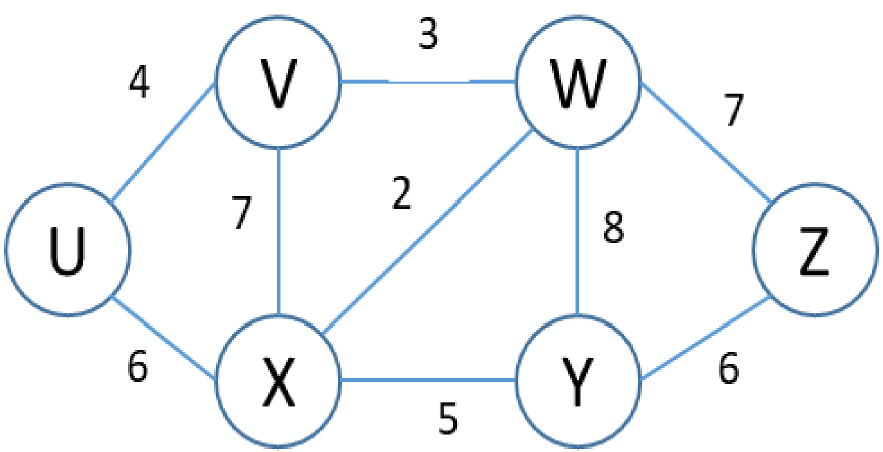
2017/2018 B Q3b

- With reference to the topology shown in Figure 1, use Dijkstra's algorithm to generate the Shortest Path Tree originating from router Z and use it to construct the corresponding routing table. It is important to clearly show how you arrived at your answer.

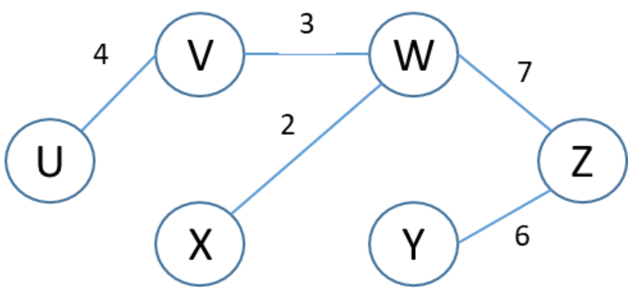


2017/2018 B Q3b

Step	N'	D(y),p(y)	D(w),p(w)	D(v),p(v)	D(x),p(x)	D(u),p(u)
0	Z	6,z	7,z	∞	∞	∞
1	ZY		7,z	∞	11,y	∞
2	ZYW			10,w	9,w	∞
3	ZYWX			10,w		15,x
4	ZYWXV					14,v
5	ZYWXVU					



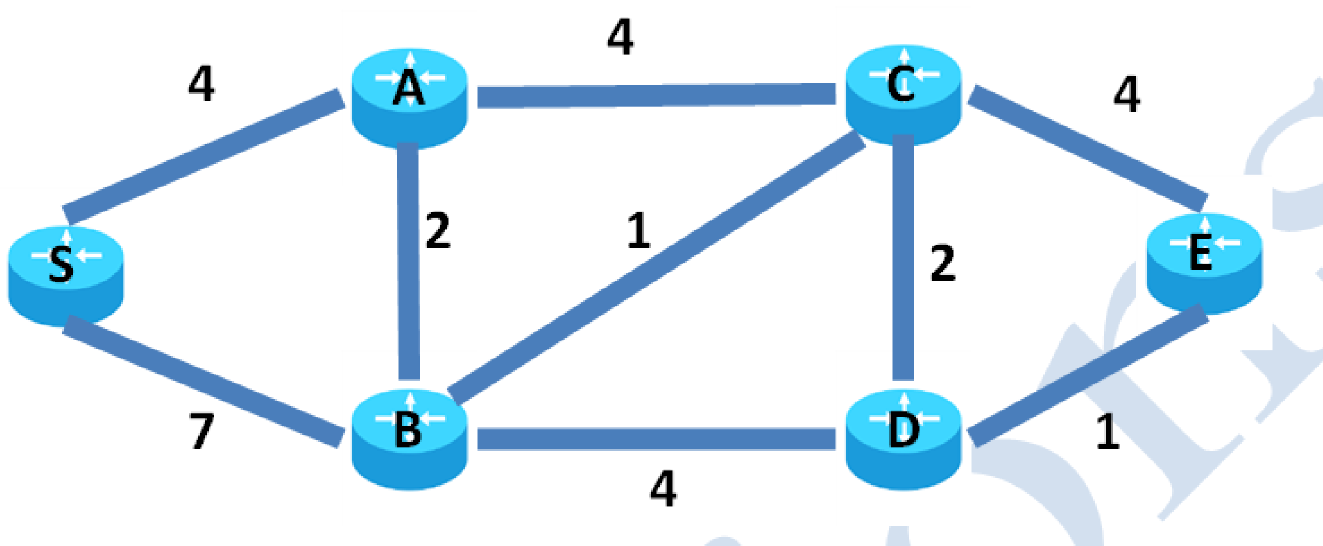
2017/2018 B Q3b



destination	link
y	(z,y)
w	(z,w)
x	(z,w)
v	(z,w)
u	(z,w)

2018/2019 A Q3a

a) Consider the network below. Routes are to be found using Dijkstra's algorithm beginning at S and finishing at E. Complete the table given. N' is the set of nodes where the distance is known, D(x) is the currently estimated distance to node x and p(x) is the node prior to x on the current iteration – the first line of the table is filled for you.



2018/2019 A Q3a

N' "known" nodes	D(A) p(A)	D(B) p(B)	D(C) p(C)	D(D) p(D)	D(E) p(E)
S	4,S	7, S	--	--	--
S,A		6,A	8,A	--	--
S,A,B			7,B	10,B	--
S,A,B,C				9,C	11,C
S,A,B,C,D					10,D
S,A,B,C,D,E					

2018/2019 A Q3b

b) Internet Protocol version 6: **[6 marks]**

i) Name 4 fields present (not including flags) in the IPv4 header that have no equivalent in IPv6.

(4 marks)

ii) Briefly explain tunnelling within the context of IPv4 to IPv6 transition.

2018/2019 A Q3b

1) Diffserv Codepoint (DSCP), ECN, Fragment offset, header checksum, header length, protocol, (half mark for TTL or identification)
2) Tunelling allows IPv6 packets to move over IPv4 network. IPv6 packet is copied as the payload in the IPv4 packet