Week 1

Application Layer

message

Transport Layer

segment, process-to-process

Network Layer

datagram, host-to-host

Link layer

frame, transfer datagram between adjacent nodes

physical layer

bits

Network Edge :

clients and servers

Network core

interconnected routers

Transmission delay :

上传延迟

Propagation delay

传输延迟

d_{proc}, d_{queue} :

d_{proc} check bit errors

d_{queue} waiting time (> 1 will approaches infinite)

Ronting

Find path

determines source destination route taken by packets

Forwarding

move packets from router's input to appropriate router output

Week1 TUT

① 二进制 十进制 十六进制 转换

2 → 10:

$1010 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 = 10$

10 → 2:

2 | 67 1 ← remainder, last digit

2 | 33 1

2 | 16 0

2 | 8 0

2 | 4 0

2 | 2 0

1000011

2 | 1 | 1

2 → 16

1101 0101

② ①

0×2³ + 1×2² + 0×2¹ + 1×2⁰

= 5

1×2³ + 1×2² + 0×2¹ + 1×2⁰

= 13 → D

So it's D5

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	A
11	B
12	C
13	D
14	E
15	F

16 → 2

D 5

2 | 5 1

2 | 2 0

2 | 1 1

1 0 1

2 | 13 => 1 1

2 | 6 0

2 | 3 1

2 | 1 1

1101 0101

1 1 0 1

#对应的0不能 ignore

10 → 16

16 | 37 5 25

16 | 2 2

16 → 10

F 7 → $15\times16^1+7\times16^0=247$

② Probability

假设有 8 个数，那么 8 个当中有 2 个为 flipped 的可能是多少？其中 P(flip) = 0.1

$$P=\binom{8}{2}0.1^2\times(1-0.1)^6=$$

$$\binom{n}{k}=\frac{n!}{(n-k)!\,k!}$$

③ 单位转换

bits 小写

1 Bytes = 8 bits # Bytes 大写

1 Kb = 1000 bits # bit per second

1 Mbps = 10^6 bps

1 Gbps = 10^9 bps

Week 2 Link Layer

- Transfer frame between adjacent nodes
- Service Provided:
  - Error Detection (EDC)
  - Error Correction
- Implements in Each and every host NIC
  network interface card

1 EDC - error detection

- NOT 100% reliable

1.1 Single Bit Parity

在末尾增加新的 Parity bit，如果有 odd number 个1则为1；Even number 个则为0

1.2 Two Dimensional Bit Parity

allow detect error across 2 dimension

Bit Block	1	0	1	1	Row Parity
1	1	0	0	1	0
0	0	1	1	0	0
1	1	1	0	1	1
1	1	1	1	1	0
Column Parity	1	0	1	1	0

2 Cyclic Redundancy Check CRC

- G(x) = $x^4 + x^2 + 1$
  - Generator
  - Binary is 10101

- CRC bits:
  G(x)最高次项为1
  或比 Binary Expression 少1

- Error Pattern E(x)
  必须位数小于 Generator

因为如果 E(x) 大于等于 Generator
则可能会检测不到 Error 的情况

(1) Consider the following information bits 1010101 and CRC generator 1101. What are the coded bits to be transmitted? Show that the coded bits are divisible by 1101.

D
data

CRC generator: 1101

1101 | 1010101 000 → CRC code

1101
1111
1101
1001
1101
1000
1101
1010
1101
1110
1101
0011

进行XOR

Redundancy

R

Send 1010101011
Result

Remainder位数和 CRC code一致

CRC coded

(2) The system can be shown in the following figure.

CRC generato is 1101

Received bits – (minus) coded bits is called the error pattern (note that here "–" is equivalent to XOR). Obviously, all "0" error pattern means "no error".

Please show

(a) Error pattern 0000100 can be detected by the receiver.

E(x) < CRC generator

Not divisible by 1101

(b) Error pattern 0001101 cannot be detected by the receiver.

Divisible by 1101

E(x) > CRC generator

3 Multiple Access Protocol & Links

- Links : end to end , Broadcast

3.1 TDMA (Time division multiple access)

- Each node get fixed length slot (so no collision)
- disadvantage : idle node still need resource

3.2 FDMA (Frequency division multiple access)

- Usually every node get equal band width (can be different)
- disadvantage: Idle node still need resource

5.3 Slotted Aloha

Assume:

- all frames same size
- time divided into equal size slot (time to transmit 1 frame)
- only transmits at the begin of slot
- If 2 or more nodes transmit in the same slot, all nodes detect collision

Operation:

NO Collision: node can send new frame in next node's beginning

Collision: node retransmits frame in each subsequent slot with prob until success

Advantage:

Idle node will not use resource

Dis:

maybe one node can take all resources by keep transmitting

Max Efficiency: 求 $NP(1-P)^{N-1}$ 的导 得出 $P = 0.37$

At best: chanel used for useful transmission 37% of time

pure Aloha 只有 18%

Week 2 Tut

Slotted ALOHA:

P 一般表示 probability of success transmit for a node

- $P(\text{idle slot}) = \prod_{i=1}^{N}(1-p_i) = (1-p)^N$
  $P(X=0)$

N 表示所有 Node 的数量

X 表示 # of nodes attempt for transmission in one timeslot

- $P(\text{success slot}) = NP(1-p)^{N-1} = P_o(1-P_{other})^{N-1} + \cdots + P_N(1-P_{other})^{N-1}$
  ↳ 当 P_i 都相等

= Total Efficiency of chanel
= Normalized Throughput

- $P(\text{collision slot}) = 1 - P(\text{success slot}) - P(\text{idle slot})$

- $P(\text{a node success at a slot}) = P_i \prod_{k=1}^{N-1}(1-P_k) = P(1-P_{other})^{N-1}$
  ↳ $k \neq i$

$= P_s$ = Node i's Average Throughput

- $P(\text{a node success at } k^{th} \text{ node}) = \underbrace{(1-P_i)^{K-1}}_{\text{前 K 个都 Fail}} P_i$

成功传输平均需要 time slots 数是

$$E(D) = \frac{1}{P_s} = \frac{1}{P_i \prod_{k=1, k\neq i}^{N_1} (1-P_k)} = \frac{1}{p(1-p)^{N_1}}$$

$4p(1-p)^3$

Week 3 Link Layer

① CSMA

- listen before transmit { channel idle: send / channel busy: defer

	If channel sensed idle	If channel sensed busy
1-Persistent	the node immediately transmits its data	the node keeps sensing the channel and waits until it becomes idle
non-Persistent	the node transmits immediately	the node does not continuously sense the channel. Instead, it waits for a random period of time before sensing the channel again.
p-Persistent	The node transmits with a probability p. The node will redo the action with a probability (1−p) in the next time slot	the node will wait one slot and sense the chanel again.

② CSMA/CD (collision detection)

- 绝对 wired LANs 有效，因为 wireless LANs 会导致 received signal strength overwhelm by local transmission strength

- Operation: If a collision detect, the node will immediately send a 48 bits jamming signal to ensure all nodes aware of

the collision. 并且进入 Binary Exponential backoff

"发现 node"

- 从开始，随机时间窗口内选择一个时间点重传
- 如果再次尝试失败，则时间窗口大小会变成 $\{0, 1, 2, \cdots, 2^{m}-1\}$ 大，其中 m 是 collision 之前的次数

- Efficiency $= \dfrac{1}{1 + 5t_{prop}/t_{trans}}$

  - better than slotted Aloha
  - t_{prop}: max propagation delay between 2 nodes
  - t_{trans}: time to transmit max-size frame

- 计算 # of nodes in current network

$$\#\text{ of nodes} = \left[\frac{\text{Efficiency} \times \text{总的信道带宽 (bit/s)}}{\text{每个 node 每秒发送的数据 (bit)}}\right]$$

总的信道带宽 (bit/s) ← channel rate

N computers have been connected to a shared medium (a cable) of length 2 km. The maximum channel rate of the shared medium is 10 Mbps. Each computer generates 10 frames per second with each frame being 1000 bytes. The speed of wave propagation is 2×10^8 meters/second. What is the maximum number of nodes supported in the network if CSMA-CD is used on the shared medium?

$$t_{prop} = \frac{2 \times 1000}{2 \times 10^8} = 10 \text{ microsec}$$

$$t_{trans} = \frac{1000 \times 8}{10^7} = 800 \text{ microsec}$$

$$efficiency = \frac{1}{1 + 5\frac{10}{800}} = 0.94$$

$$\text{traffic of each node} = 10 \times 1000 \times 8 = 80000 \text{bit/sec}$$

$$\text{no. nodes} = \frac{9.4 \times 10^6}{80000} = 117$$

$t_{prop} = \frac{\text{Cable length (m)}}{\text{propagation speed (m/s)}}$ max

$t_{trans} = \frac{\text{1 Frame Size (bits)}}{\text{Channel rate (bps)}}$ max

③ Ethernet

- Only detect Error
- unreliable — Connectionless
-

④ IPv4

32 bits = 4×8 → 8个 binary number

129.56.78.123 → Binary is 0111 1011

129.56.78 | 123

⑤ MAC

48 bits = 2×4×6 → 总共6组

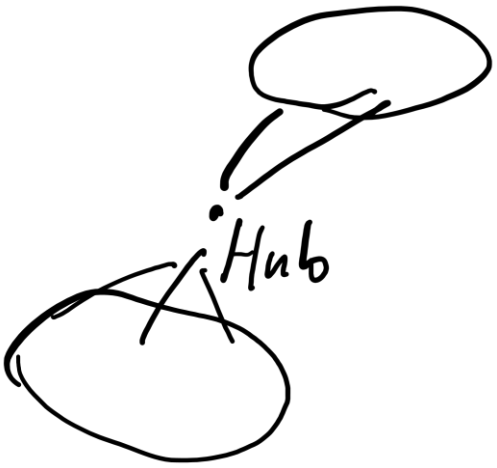
一组有2个 Hex

每个 Hex 占 4 bits

- 01:23:45:67:89:AB

. Broad cast is FF:FF:FF:FF:FF:FF

⑥ Hub VS Swith



| collision domain / subnet

no difference with line

multiple subnet

Week 4 Network Layer

① Read Forwarding Table — Longest Prefix Matching

Longest prefix matching

longest prefix matching
when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

选用能最大 match 的 interface

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

DA: 11001000 00010111 00011000 10101010 which interface?

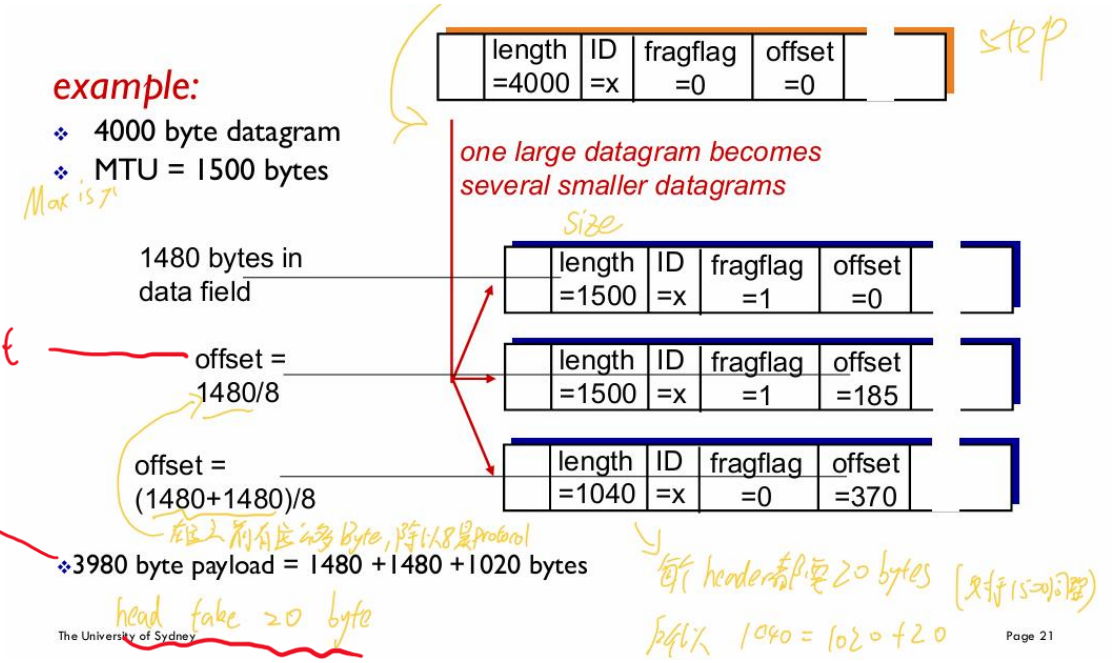
The University of Sydney Page 16

只比较网络部分，即子网掩码为1的部分

比如 192.168.72.123/24 只会比较 192.168.72

② Fragmentation

- 网络对于传输的最大数据包有限制，比如 Ethernet 的 MTU 是 1500 bytes



step

Size

offset

payload

(exclude head for every segment)

head take 20 byte

每个header都要20 bytes [对于IP协议]

所以 1040 = 1020 + 20

③ Subnet Mask

200.23.16.0/24 means subnet mask is 255.255.255.0

host portion

Since they take 24 bits in total

这部分也叫 Subnet portion

④ CIDR : Routing

2. Routing Table

A company has been granted a block of IP addresses starting with 150.12.16.0/24. The address space should be allocated to four subnets A, B, C and D. Subnet A needs 9 addresses, subnet B needs 18 addresses, subnet C needs 28 addresses, and subnet D needs 12 addresses. The IP addresses have been assigned in the following order A, B, C, and D (subnet A has the smallest IP addresses and subnet D has the largest IP addresses). What is the starting IP address of subnet C?

这是子网掩码

150.12.16. A 9-size need 4 bits
8bits
150.12.16.0000.0000
150.12.16.0000.1111
· begin with all 0
· end with all 1

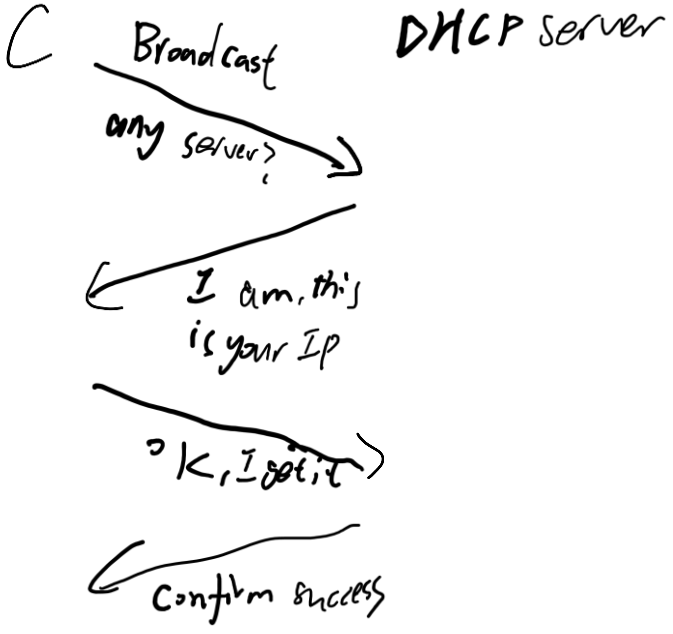
B 18 need 5 bits
150.12.16.0001.0000 ✗ 不是以0开头
150.12.16.0010.0000 ✓ 是以0开头
150.12.16.0011.1111

C 28 need ≤ bits
150.12.16.0100.0000 在4bits上加1,进2-base计算
150.12.16.0101.1111

K bits 可以承载 2^K-2 个 address
并因为排除 all 0 和 all 1

5 DHCP

dynamically get address from as server



⑤ NAT Router 1st type of Router

· 只影响 IP 和 Port number

Send side：在经过 Router 时

S 变成 router 的 global IP

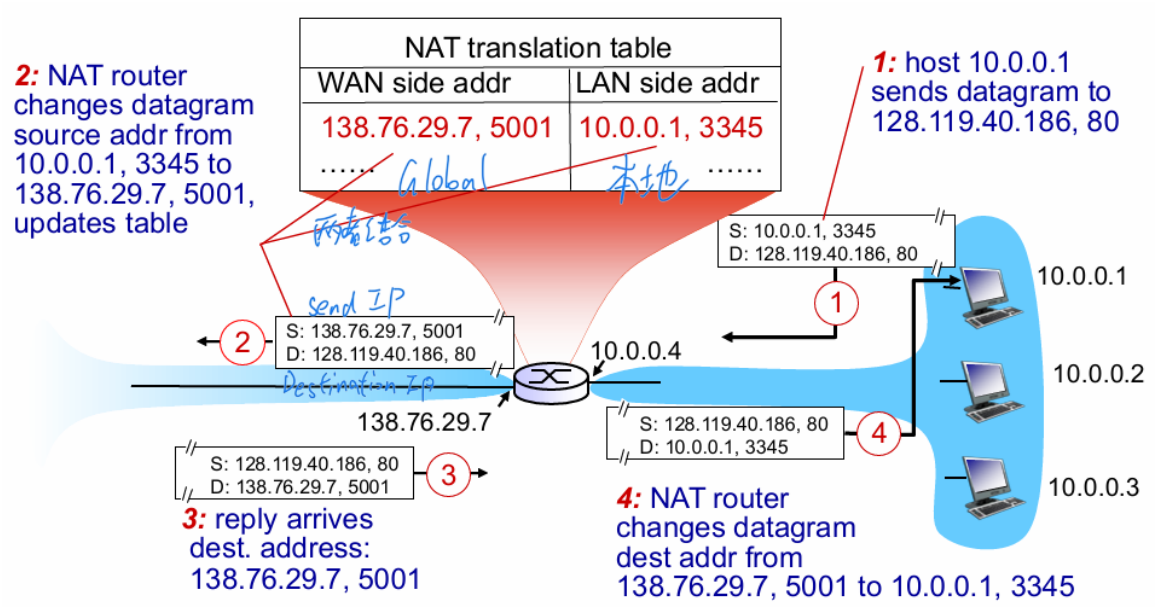
此及分配它的对应的 port Number

D 不变

Receive side：在经过 Receive Router 时

S 不变

D 变成 local IP 和 local port



7 ICMP

used by user & routers to communicate

Week 5

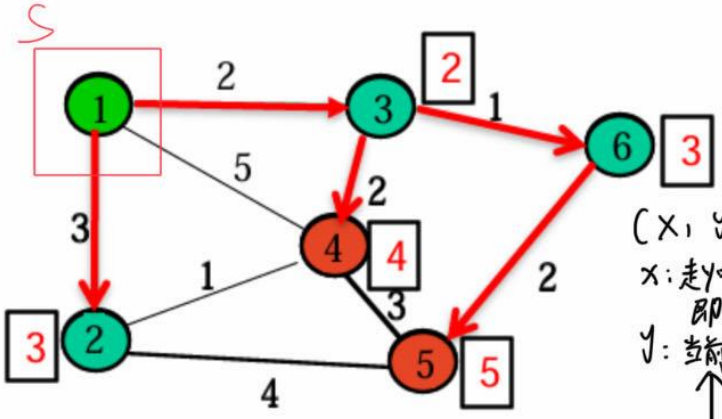
走点 右距

① Link state . LS . Dijkstra $O(n^2)$

Dijkstra's algorithm: Example

找到所 N_i 的 neighbours 并且记录它们到 starting node 的 最小 cost

找到 N_1 和 N_3 的所有 neighbour 并且记录它们到 starting node N_1 的 最小 cost



(x, y)

x: 走y中描述的路线, x是Current Node的上一个Node

即 $S \to \cdots \to N_x \to N_{current\ Node}$

y: 当前列所代表的Node (简称Current Node) 到 Starting Node 的距离

$N_1 \to N_4$ cost = 5

$N_1 \to N_3 \to N_4$ cost = 4 ✓

Iteration	Tree (我们从 start Node 开始)	N_2	N_3	N_4	N_5	N_6
Initial	{1}	(1,3)	(1,2)	(1,5)	(-1,∞)	(-1,∞)
1	{1,3} (2nd closest node is N_3)	(1,3)		(3,4)	(-1,∞)	(3,3)
2	{1,2,3} (3rd closest node is N_2)			(3,4)	(2,7)	(3,3)
3	{1,2,3,6}			(3,4)	(6,5)	
4	{1,2,3,4,6}				(6,5)	
5	{1,2,3,4,5,6}	(1,3)	(1,2)	(3,4)	(6,5)	(3,3)

$N_6 \to N_3 \to N_1$ $N_5 \to N_6 \to N_3 \to N_1$

② Distance Vector

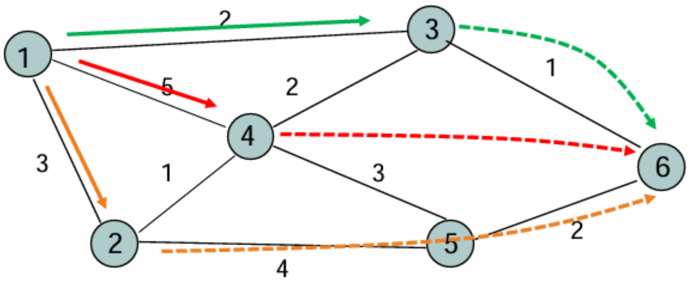
- only know neighbour

Destination node 6

$y = 6$

$D1 = \min\{3+D2,\ 2+D3,\ 5+D4\}$ $x=1$, $y=6$, $n=\{2,3,4\}$
$D2 = \min\{3+D1,\ 1+D4,\ 4+D5\}$ $x=2$, $y=6$, $n=\{1,4,5\}$
$D3 = \min\{2+D1,\ 2+D4,\ 1\}$
$D4 = \min\{5+D1,\ 1+D2,\ 2+D3,\ 3+D5\}$
$D5 = \min\{4+D2,\ 3+D4,\ 2\}$

How to solve: Use an iterative procedure!
Use this Bellman-ford equation every round, until convergence.



初始化 iteration，所于所有的初始化应一行都这么写

(N_x, y):
N_x: 走了的路线，N_x 表示 Current Node 下一个 node
Current Node → N_x → …… → Destination Node
y: 当前列所表示的 node（称为 Current Node）到 Destination Node 的最短距离

Iteration	Node 1	Node 2	Node 3	Node 4	Node 5
Initial	(-1, ∞)	(-1, ∞)	(-1, ∞)	(-1, ∞)	(-1, ∞)
1	(-1, ∞)	(-1, ∞)	(6,1)	(-1, ∞)	(6,2)
2	(3,3)	(5,6)	(6, 1)	(3,3)	(6,2)
3	(3,3)	(4,4)	(6, 1)	(3,3)	(6,2)

iter2 只能用 iter1 的信息

iter3 只能用 iter2 的信息

End condition: End when next iteration has same answer to previous one

（这里缺失了这一步）

之所以会重新回到 N_3 是因为对于 $Node_4$ 和 $Node_1$ 来说，它们离 Destination $Node_6$ 的最短路径需要经过 $Node_3$

可以使用 "split horizon with reverse poisoning" 解决

→ $Node_3$ → $Node\ 6$ cost 1 ✓

⇐ { Node 3 → Node 4 → Node 3 → Node 6 cost 5
Node 3 → Node 1 → Node 3 → Node 6 cost 5 }

$D_3 = \min\{2+D_1,\ 2+D_4,\ 1+D_6\}$

计算 Routing table at a node

Consider the following network in which distance vector routing is used. Each node in this network sends its routing table using a vector of size 6 where each entity of the vector represents the cost to the corresponding node, i.e. (Cost-to-Node1, Cost-to-Node2, Cost-to-Node3, Cost-to-Node4, Cost-to-Node5, Cost-to-Node6).

The following cost vectors have just arrived at router 1 from its neighbours:

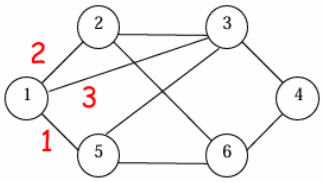
from 2: (2,0,2,7,4,1);
from 3: (3,2,0,3,1,4);
from 5: (1,3,1,4,0,3).

The link costs between node 1 and nodes 2, 3, and 5 are 2, 3, and 1, respectively. What is the routing table at node 1?

① 找 neighbour

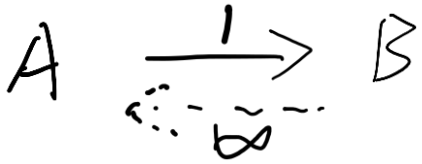
from 1 : (0,2,2,5,1,3) ← Result

	Cost	Next Node
For Node 2: min {2+0, 3+2, 1+3} = 2	2	2
For Node 3: min {2+2, 3+0, 1+1} = 2	2	5
For Node 4: min {2+7, 3+3, 1+4} = 5	5	5
For Node 5: min {2+4, 3+1, 1+0} = 1	1	5
For Node 6: min {2+1, 3+4, 1+3} = 3	3	2



① 去哪 都要经过它的 neighbours

Reverse Poison



Split Horizon

Week 6 network layer

- IGP intra-AS local

  所有不带BGP的都属于IGP

- BGP inter-AS Global

  Hop 数决定，

① AS Path choosing

Find best inter-As route

(based on hop and 运营商)

Find best intra-AS Route

1. local Prefer (这条的)
2. shortest Path

If there is a Tie: (cause by same # of hope for inter-AS routes)

Hot potato

② ARP. address resolution Protocol

use it to ask MAC address

③ Normal Router 2nd type of Router

经过 Router 时

· Source

IP 不变

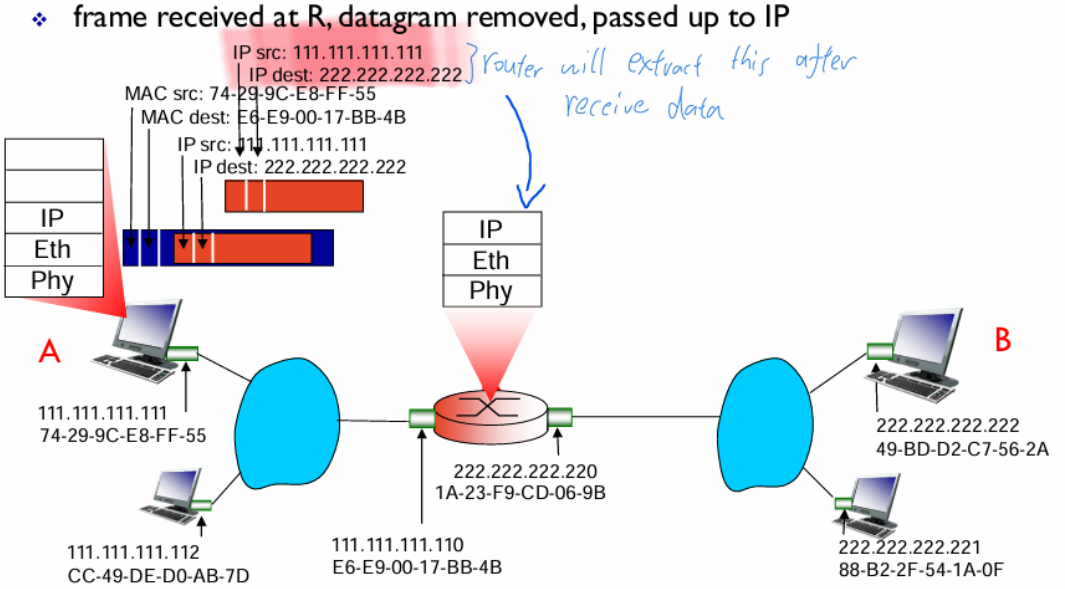
MAC 变成当前的

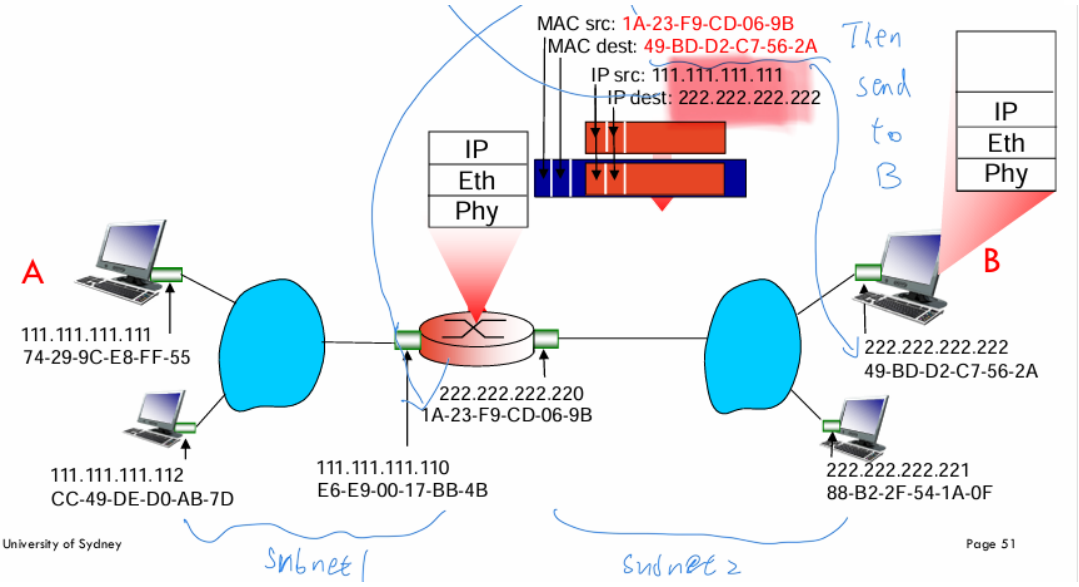
Destination

IP 不变

MAC 变成下一个

- frame sent from A to R
- frame received at R, datagram removed, passed up to IP





Week 7 Network Layer

① Traditional Routing

- 由 Router 決定發送路徑
- Always forward datagram to the same Next hope
- Have no global view
- Cannot make decision based on traffic

② SDN:

- 在 control plan 有 Global view
- 可跟據 traffic 動态調整 route

②.① OPen flow (在 Data plan)

- 發送 info 給 control plan, 以获取

forwarding table

- 具体怎么用看A2 Q1

查完 Action：

Forward to ...

Drop

Week 7 Transport layer

① Multiplexing at sender

Demultiplexing at receiver

② UDP

- Unreliable, Connectionless
- Unorder delivery

2.1 UDP Checksum

把头拆成多个 16 bits, 并对它们进行加法

example: add two 16-bit integers

	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
carryout	1															
wraparound		1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
sum		1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum		0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

add again

第1次加后发现进位了

Note: when adding numbers, a carryout from the most significant bit needs to be added to the result

③ RDT (Reliable data transfer)

RDT 1.0：假设 network layer is perfect

no loss

no error

RDT 2.0：使用 ACKs 和 NAKs 解决处理 Error detecting 的问题

会 stuck 于 Acks NAKs lost

RDT 2.1：考虑到了 ACKs 和 NAKs 损坏的可能

- Add sequence number to each packets

RDT 2.2：取消 NAKs，采用 ACKs

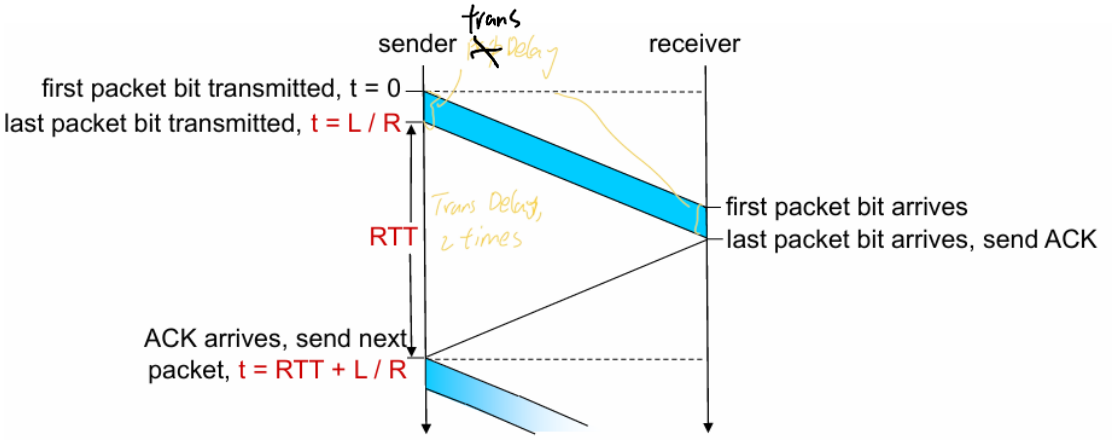
- # of seq being Ack

RDT 3.0：解决 NAK/Ack lost 的问题（导致 stuck）

- 会在 Timeout 之后重传

RDT3.0 效率方程

$$Efficiency = \frac{T_{trans}}{RTT + T_{trans}}$$



$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

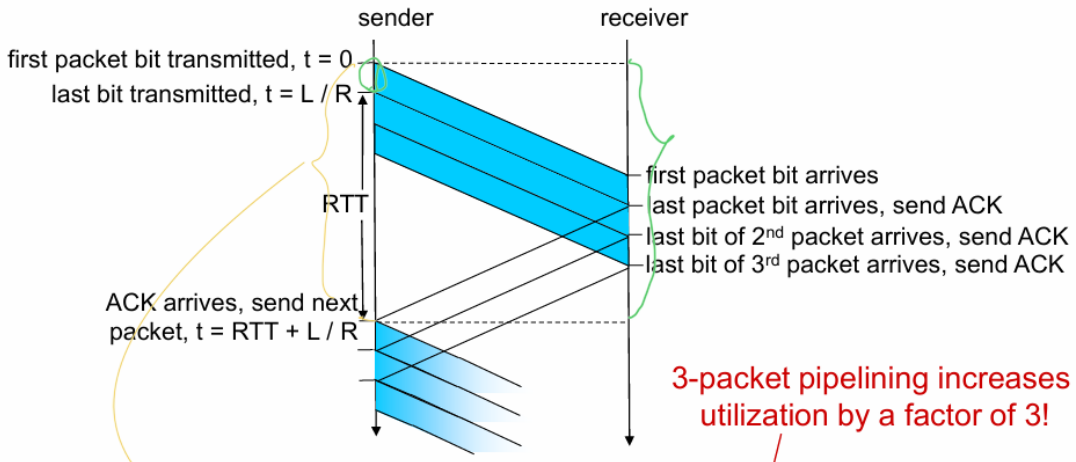
实际利用率

↳ 2次 T_{prop}
1次 T_{trans}

4 Pipelined Protocal

- 解决 RDT3.0 low efficiency 的问题
- allow 一次传多个

Pipelining: increased utilization



3-packet pipelining increases utilization by a factor of 3!

$$U_{sender} = \frac{3L/R}{RTT + L/R} = \frac{.0024}{30.008} = 0.00081$$

为什么乘3

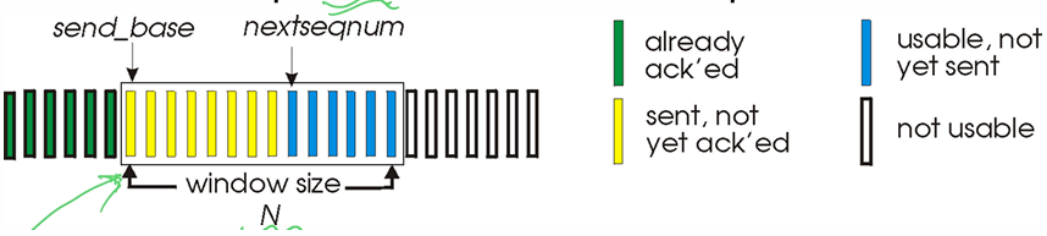
3个pkt比存先高效3倍

蓝色部分 sender在工作，白色的时候没有

4.1 Go back N GBN

- Receiver only send cumulatie ack
- ACK_9 represent ACK_{0-9}

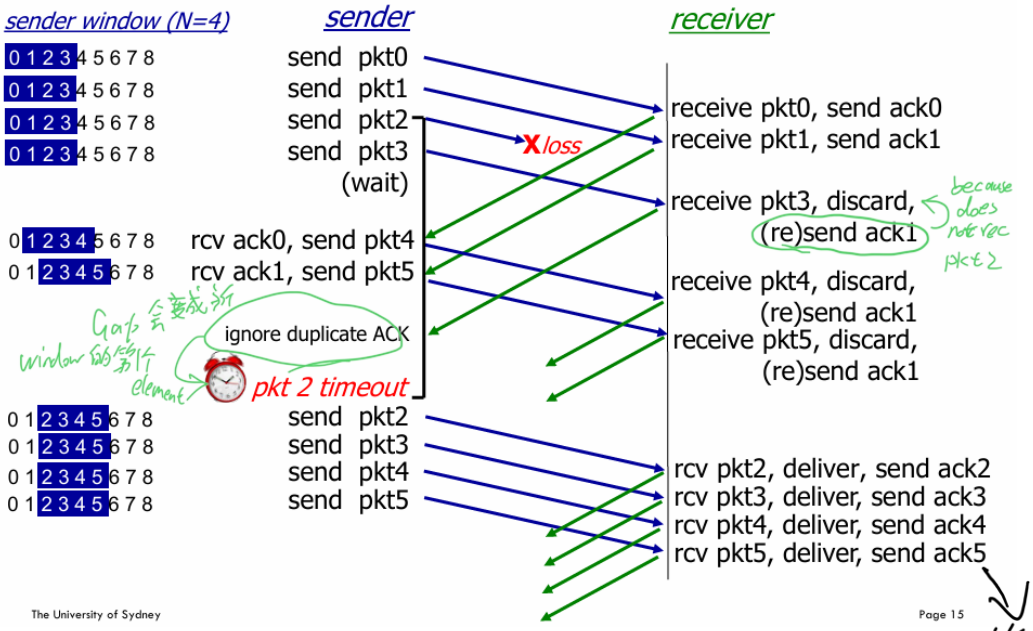
– "window" of up to N, consecutive unacked pkts allowed



从断点重发

GBN in action

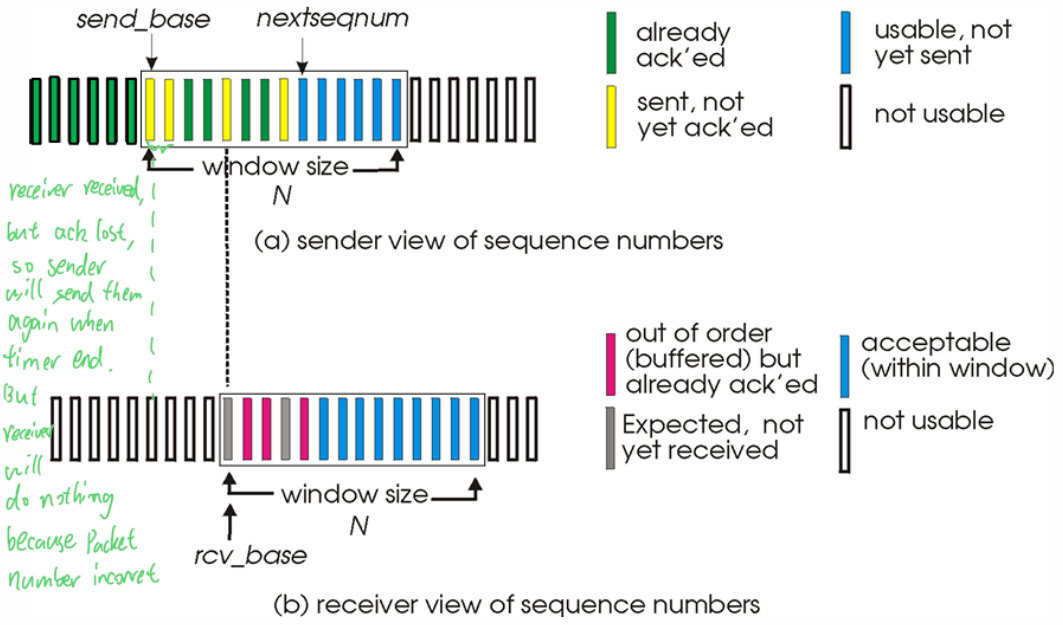
不保留 unorder pkt



带表 ack 5前的都OK,

4.2 Selective repeat SR

Selective repeat: sender, receiver windows

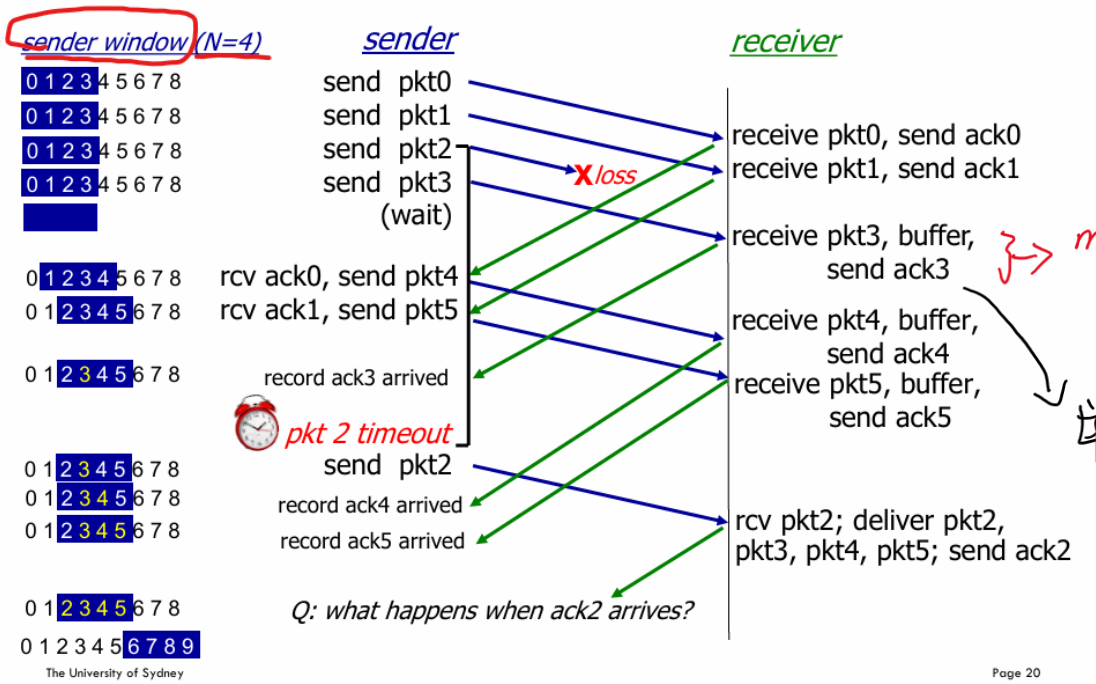


receiver received, but ack lost, so sender will send them again when timer end. But receiver will do nothing because Packet number incorrect

重发 miss 的

Selective repeat in action

存储 unorder Pkt



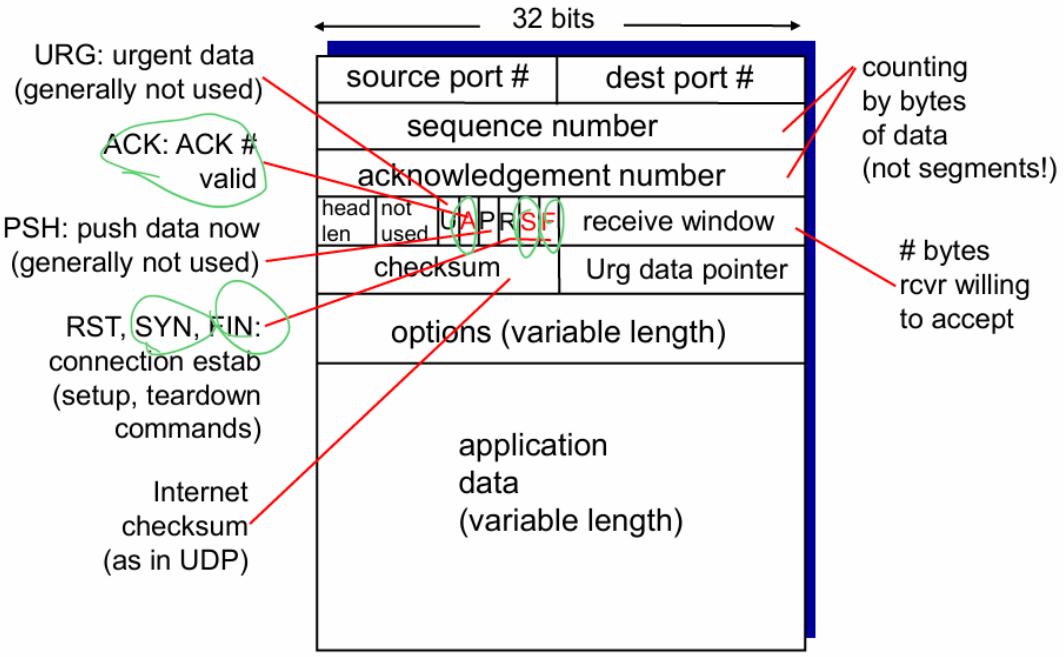
make sure order is correct

这里是和TCP不一样的，TCP这里会返回 ack1

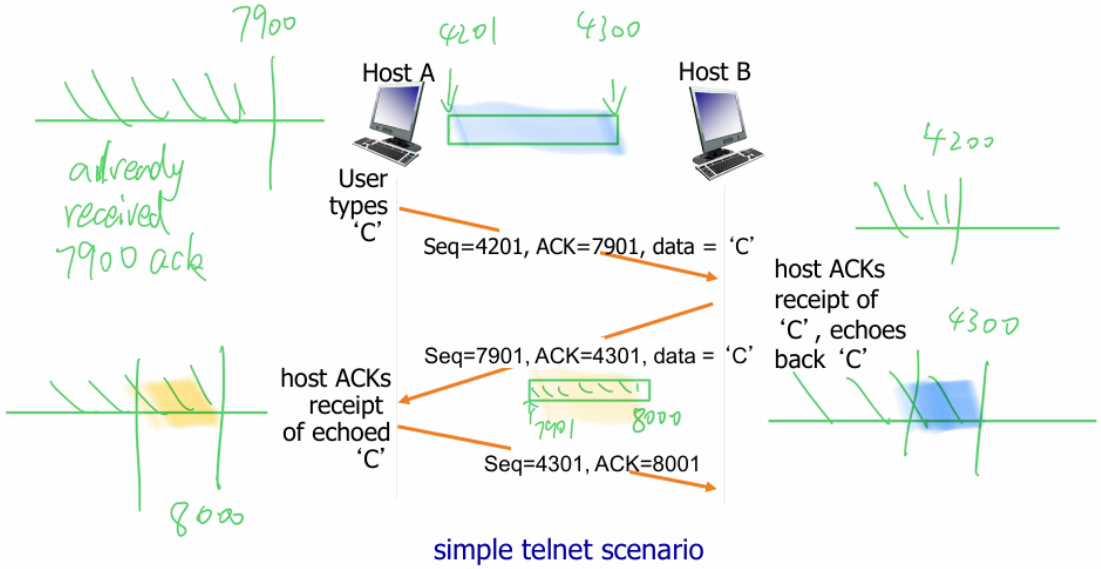
15 TCP

- In-order, Reliable
- 结合了 GBN 和 SR

TCP segment structure



TCP seq. numbers, ACKs



simple telnet scenario

注意 sender 和 Receiver 的 Seq 和 ACK 分别计算

(5.1) Set TimeOut period

波动越大的RTT，要越大的 Safety Margin

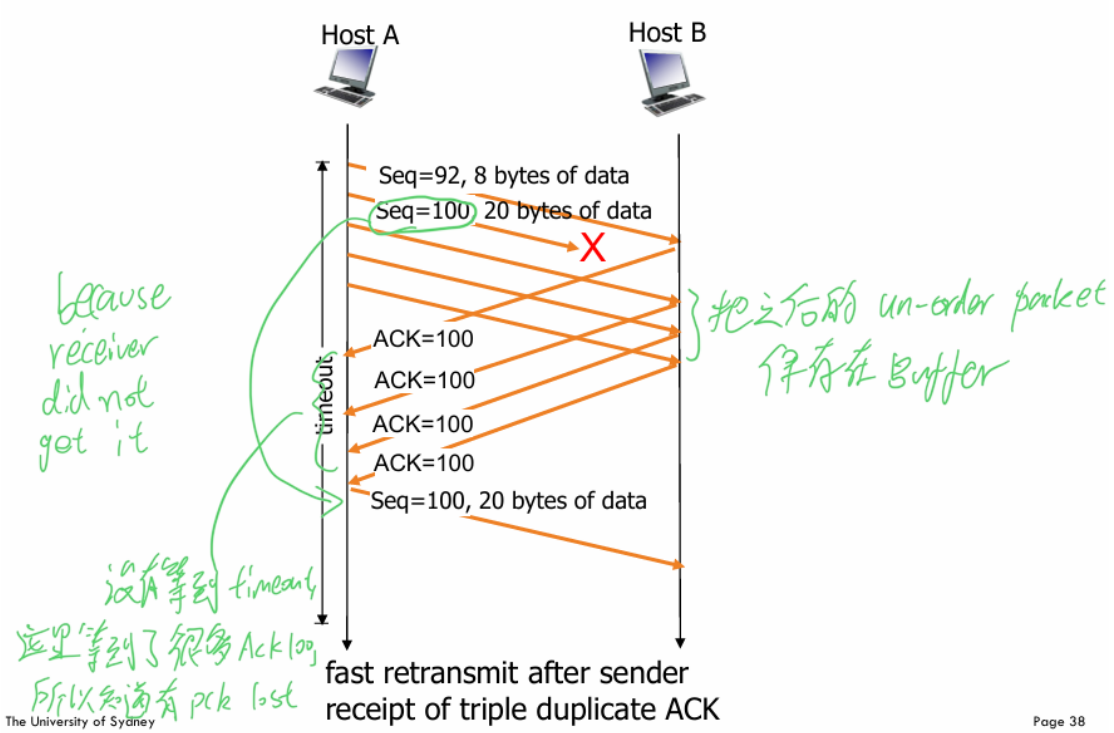
$$\text{EstimatedRTT}_i = (1-\alpha)\,\text{EstimatedRTT}_{i-1} + \alpha\,\text{SampleRTT}_i$$

★ $$\text{DevRTT}_i = (1-\beta)\,\text{DevRTT}_{i-1} + \beta\,|\text{SampleRTT}_i - \text{EstimatedRTT}_i|$$

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 \times \text{DevRTT}$$

(5.2) Retransmit

- Time Out
- Fast Retransmit — 3 duplicates



5.3 Flow Control rwnd

- receiver advertise "free buffer" spare by including rwnd in TCP header
- guarantee receiver buffer will not overflow
  - Focus on

5.4 Congestion Control

cwnd

- Focus on reduce traffic on network

5.4.1 TCP Tahoe (old)

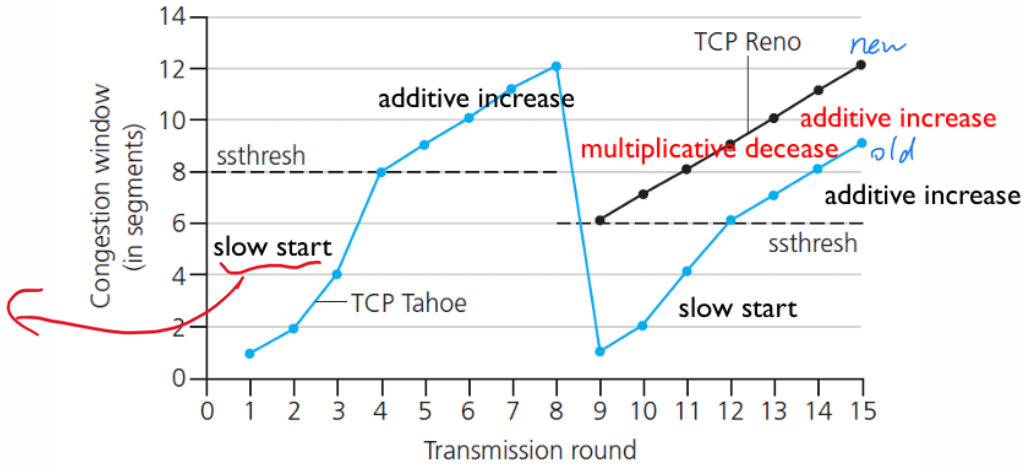
- set cwnd to 1

5.4.2 TCP RENO (new)

- Fast recovery
  - time out → set to 1
  - 3 duplicate → Cut in half

TCP: switching from slow start to CA

Congestion Avoidance 는 slow start 후



before ssthresh

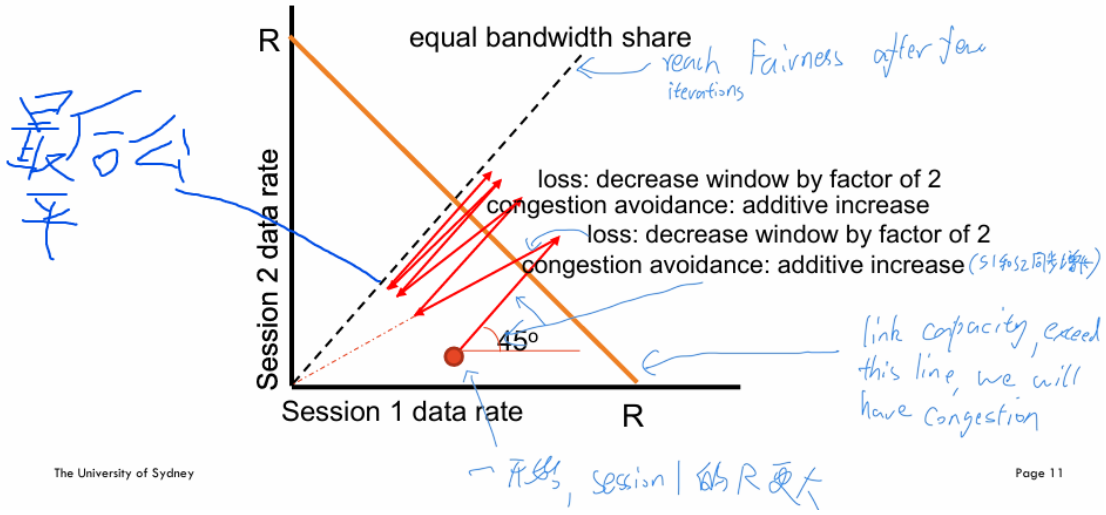
{ 1 ≤ 0 (5.4) + (5.3) } ⇒ Sender window size = max{rwnd, cwnd}

3.5 TCP Fairness

- 对每个TCP公平，但如果一个App开多个TCP Connection就不公平

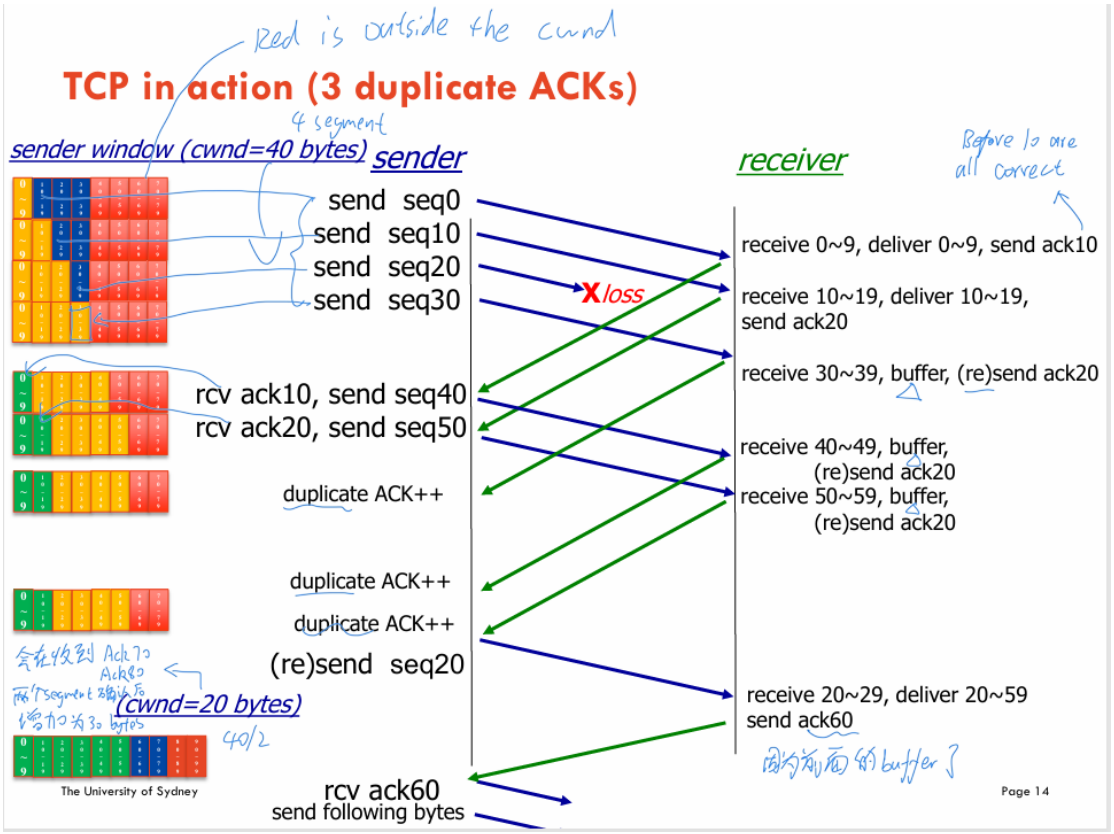
two competing sessions:

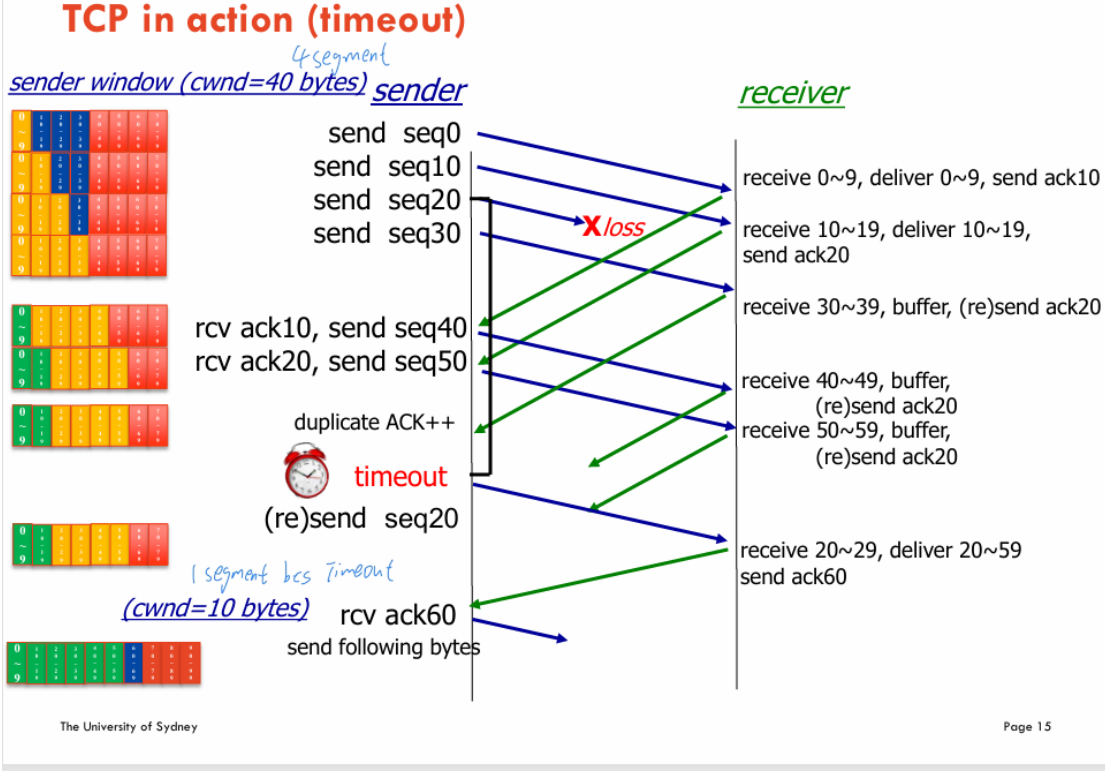
- additive increase gives slope of 1, as throughout increases
- multiplicative decrease decreases throughput proportionally



最后公平

一开始，session1的R更大


Red is outside the cwnd
TCP in action (3 duplicate ACKs)
4 segment
sender window (cwnd=40 bytes)
sender
receiver
Before 1o are all correct
send seq0
send seq10
send seq20
send seq30
X loss
receive 0~9, deliver 0~9, send ack10
receive 10~19, deliver 10~19, send ack20
receive 30~39, buffer, (re)send ack20
rcv ack10, send seq40
rcv ack20, send seq50
receive 40~49, buffer, (re)send ack20
receive 50~59, buffer, (re)send ack20
duplicate ACK++
duplicate ACK++
duplicate ACK++
(re)send seq20
(cwnd=20 bytes)
receive 20~29, deliver 20~59
send ack60
rcv ack60
send following bytes
The University of Sydney
Page 14


TCP in action (timeout)
4segment
sender window (cwnd=40 bytes)
sender
receiver
send seq0
send seq10
send seq20
send seq30
X loss
receive 0~9, deliver 0~9, send ack10
receive 10~19, deliver 10~19, send ack20
receive 30~39, buffer, (re)send ack20
rcv ack10, send seq40
rcv ack20, send seq50
receive 40~49, buffer, (re)send ack20
receive 50~59, buffer, (re)send ack20
duplicate ACK++
timeout
(re)send seq20
1 segment bcs Timeout
(cwnd=10 bytes)
receive 20~29, deliver 20~59
send ack60
rcv ack60
send following bytes
The University of Sydney
Page 15

Week 9 Application Layer

email	TCP	SMTP	25
remote terminate access	TCP	Telnet	23
web	TCP	HTTP, HTTPS	80, 443
file transfer	TCP	FTP	20, 21
Streaming	UDP or TCP	HTTP, RTP	
Internet phone	UDP or TCP	SIP, RTP	

① HTTP

Non-Persistent: get one object each time

Persistent: get multiple

一般不计算最后一个 return 的时间

Cookie 存在 client web server

- Msg:

200	ok
301	move permently
400	Bad
404	not Found

② SMTP

- Email: user agents, mail server, email protocol
- SMTP is ~~,~~ stateful

②.1 POP3

- stateless
- download

(2.2) IMAP (new)

- stateful
- keep msg on server

Week 10 Application Layer

① DNS 用 UDP

- host name → IP address
- load distribution avoid congestion
- There are 13 Root DNS

Root
|
TLD .com .Au
|
Authoritive amazon.com
|
local (find by DHCP)

- DNS Resource Record RR

1. Type A

hostname → IPv4

2. Type NS

Find corresponding TLD Suchas .com .Au

3. Type CNAME (Canonical)

设定 别名 alias

4. Type MX

.

② P2P

(2.1) P2P transfer VS client-server

client-server:

$$D_{c-s} \geq \max\left\{\frac{NF}{U_s}, \frac{F}{d_{min}}\right\}$$

P2P

$$D_{p2p} \geq \max\left\{ \underbrace{\frac{F}{U_s}}, \underbrace{\frac{F}{d_{min}}}, \underbrace{\frac{NF}{U_s + \sum U_i}} \right\}$$

- Tit-for-Tat
  - the more one share, the faster it can download
  - re-evaluate top 4 every 10 sec
    - ↳ optimically unchoke
      find potential better neighbour

Week 11

4 components of network security

- Confidentiality: only sender receiver can understand msg
- **Authendication**: sender and receiver want to identify each other
- Msg integrity: msg does not change
- Availability: available for user

① Cryptography

①.① Symmetric key

- $K_S(K_S(m)) = m = K_B(K_A(m))$
- 使用 DES 主流加密度
- 问题在于怎么 share K_S，选择要RSA

1.2 RSA

- both Encry, Decry take very long time
- Pubic key and Private key generate by one person
- $K_B^+(K_B^-(m)) = K_B^-(K_B^+(m))$
- 通过 RSA 加密 Symmetric key, 选择双方对称 symmetric key
- Secure because factor big number is hard

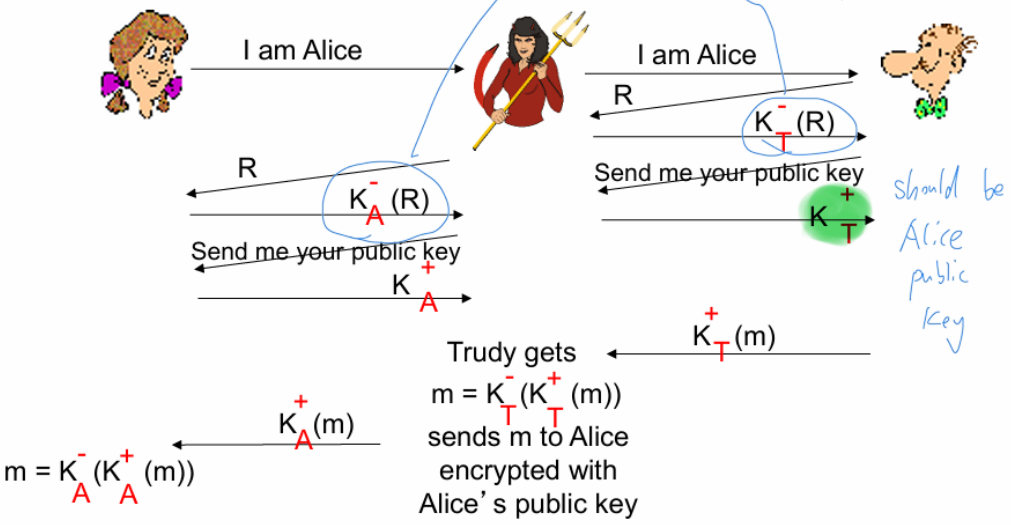
2 Authendication

V4: 需要线下交换 Symmetric key

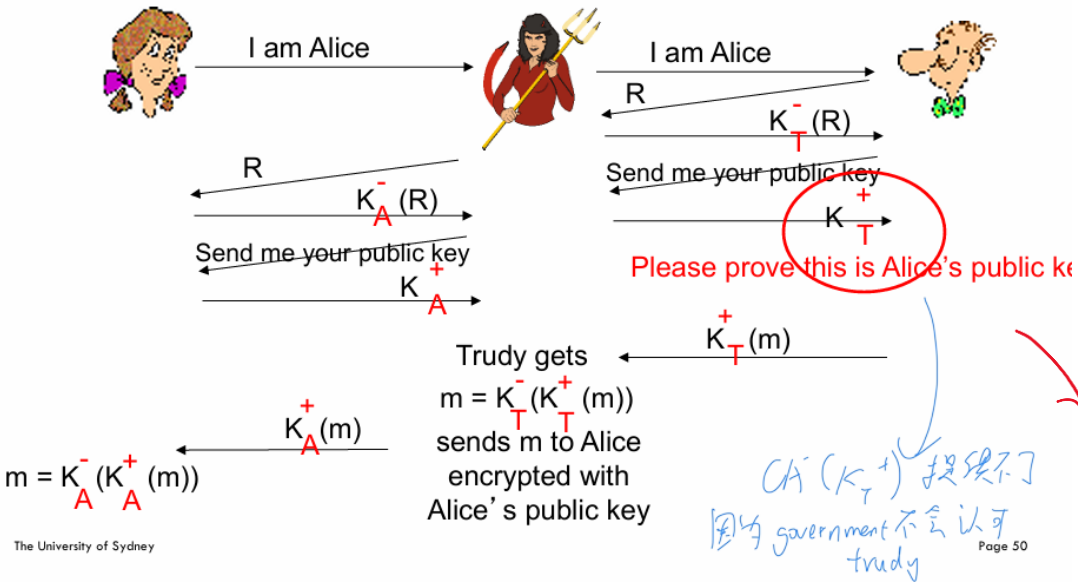
V5:

ap5.0: security hole

man (or woman) in the middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)

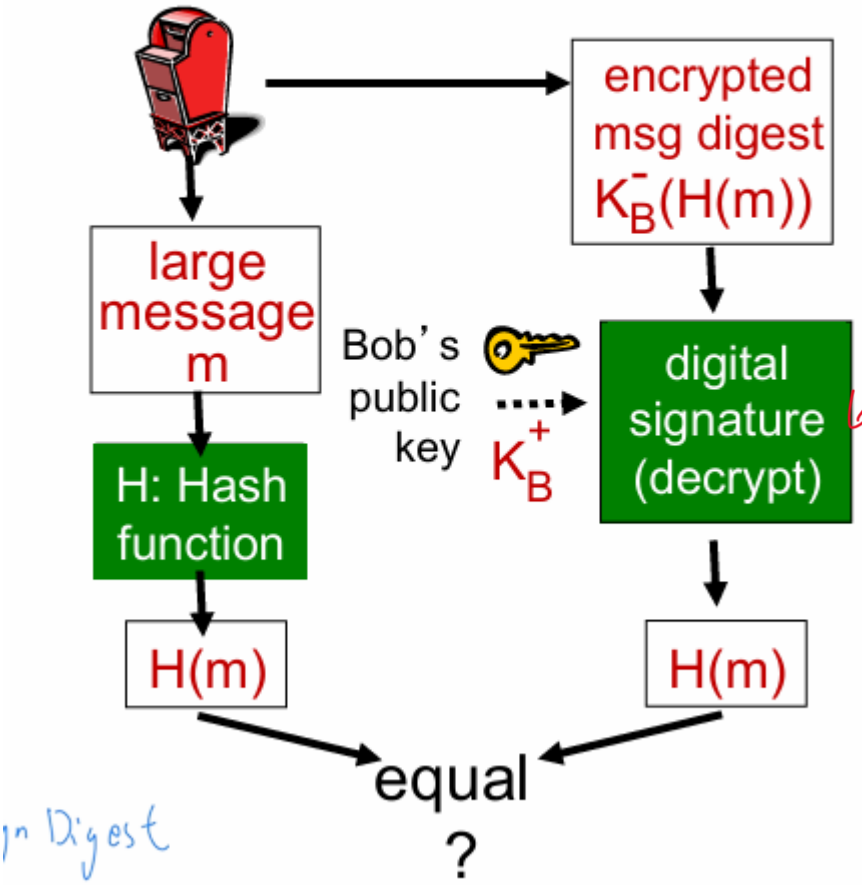


man (or woman) in the middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)



③ Digital Signature: Digest

Alice verifies signature, integrity of digitally signed message:



用RSA 只sign digest 以hash为input的时间

in Digest

Week 12

① Secure email (PGP)

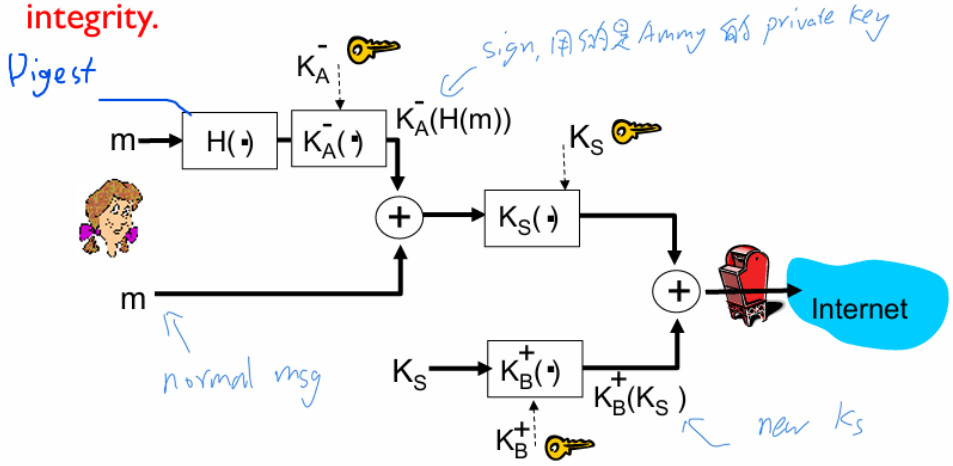
- want to send confidential email

Not best way: very slow

Both Alice and Bob's public key need to be certificated by Government CA

Secure e-mail (continued)

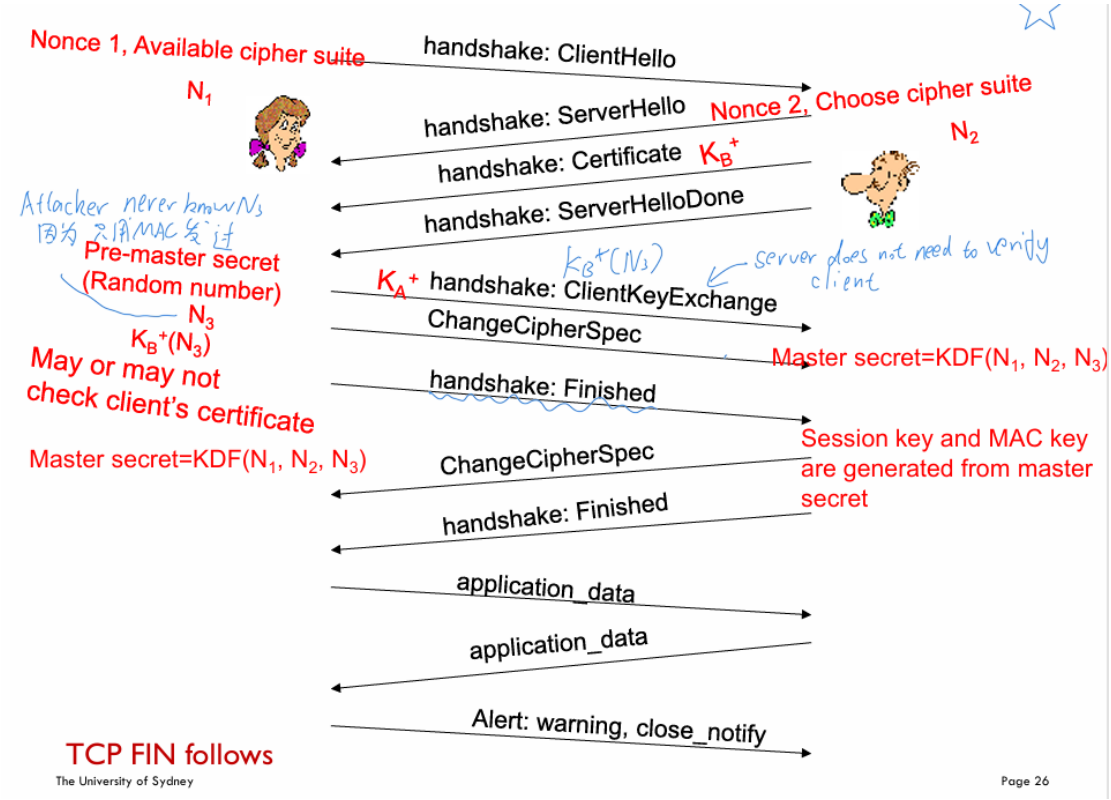
Alice wants to provide secrecy, sender authentication, message integrity.



Alice uses three keys: her private key, Bob's public key, newly created symmetric key

② SSL:

client connect to host



· 一开始由 client 提供选择方式，由 server 从中选一个

· 在 Master Secret 建立后，再对上面消息、确认，免得 Trudy 选了个 weak 传输方式

③ IPsec Protocols — 用于 VPN，保护企业内网

AH: authentication + integrity ; Not confidentiality

ESP: all 3

↳ 更常用

(4) IDS

- deep packet inspection
- examine correlation