THE UNIVERSITY OF MELBOURNE

Week 12
Security

COMP90007 Internet Technology

Prepared by: Chenyang Lu (Luke)

Tutor Feedback

https://apps.eng.unimelb.edu.au/casmas/index.php?r=qoct/feedback&subjCode=COMP90007

For students who attend Wednesday 10:00 am Tutorial, please choose Wed 5:15pm in dropdown list to leave feedback.

Your Tutor

Chenyang Lu (Luke)

- Email: chenyang.lu@unimelb.edu.au

- Workshop Slides: https://github.com/LuChenyang3842/Internet-technology-teaching-material

Day	Time	Location
Tue	18:15	Bouverie st –B114
Wed	10:00	Elec Engineering -122
Wed	17:15	Bouverie-sr 132

THE UNIVERSITY OF MELBOURNE

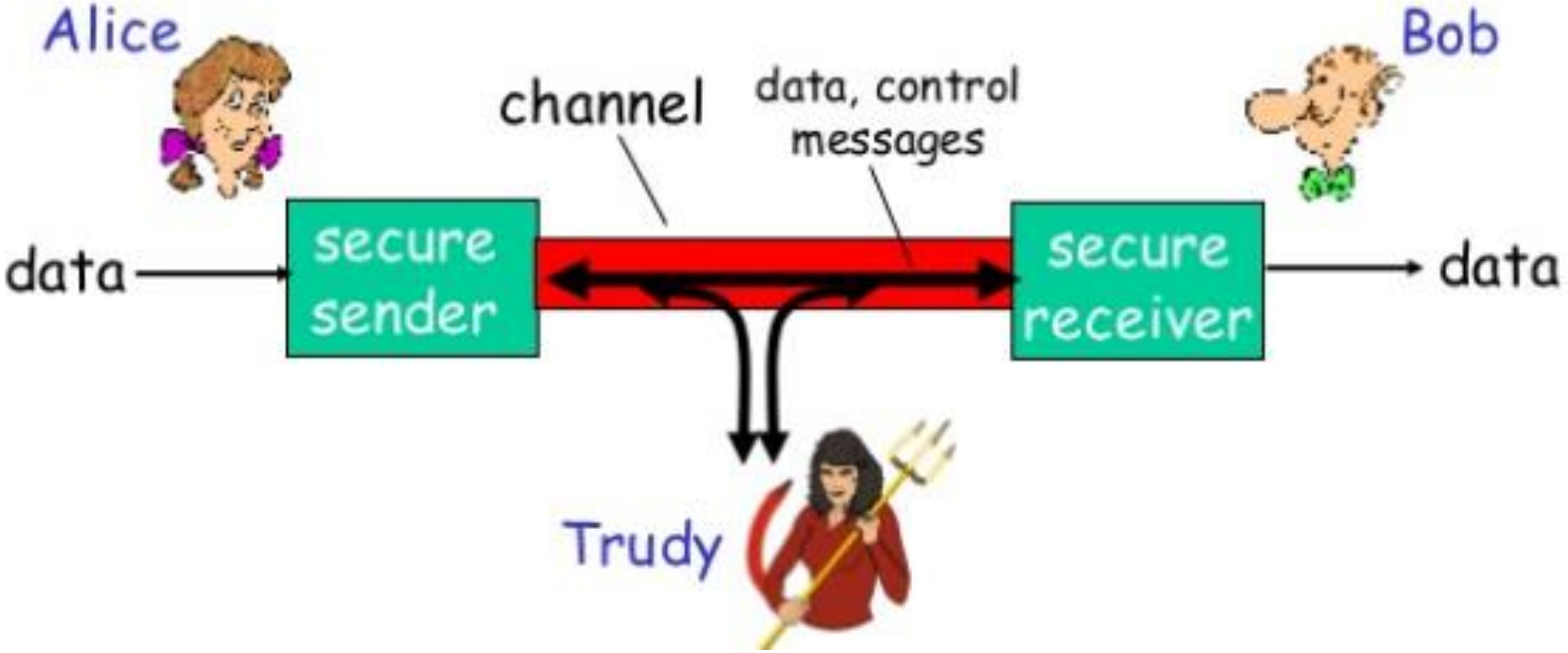
Security

Security

1. Encryption Model
2. Four key areas/aspects of network security
3. Two main categories
   - Symmetric key Algorithms
     - DES
     - AES
   - Asymmetric key Algorithms
     - RSA
4. Digital Signature (Message Digest)
5. Authentication
   1. Shared Keys (Diffie-Hellman key exchange)
   2. Key distribution
   3. Kerberos
   4. Public keys
6. Communication Security
   1. IPSec
   2. VPN
   3. FireWall
   4. Wireless security

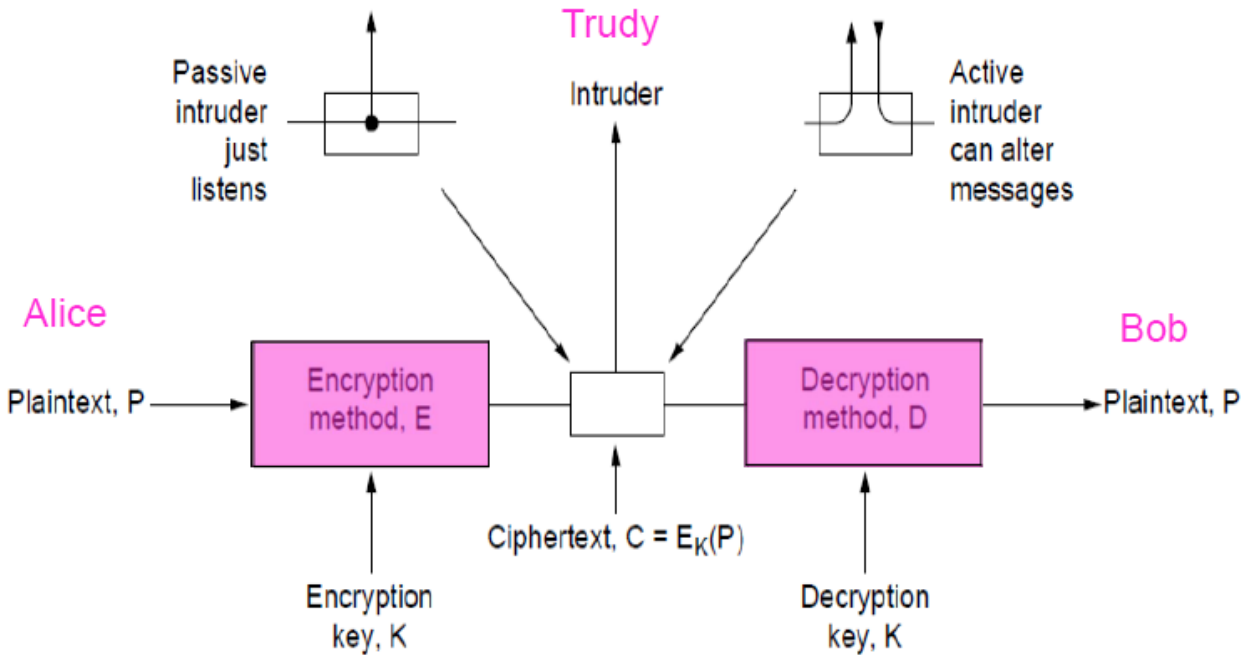
Security in computer science is all about a store:

- Bob, Alice (lovers!) want to communicate “Securely”
- Trudy (Intruder) may intercept, delete, add Message



Reference: https://www.slideshare.net/IsraelMarcus1/security-63063859

Encryption Model



Notation

- C = ciphertext,
- P = plaintext,
- E = encryption,
- D=decryption,
- K = key

- $C = E_K(P)$
- $P = D_K(C)$
- $D_K(E_K(P)) = P$

Modern Key-based Algorithm

- **Symmetric key algorithms**
  - Same key to Encrypt and Decrypt
  - Two Methods
    - Data Encryption Standard (DES)
    - Advanced Encryption Standard (AES)

- **Asymmetric key algorithms**
  - Different Key to encrypt and decrypt
  - Diffe-Hellman Key System:
    - **Key 1: public key**, usable by anyone **to encrypt** messages to the owner of the key, this key known to all
    - **Key 2: private key**, required **to decrypt** the message and known only by the owner of this key
    - Process:
      1. $C = E_{K1}(P)$
      2. $P = D_{K2}(C)$
      3. $DK2(E_{K1}(P)) = P$

RSA – An Asymmetric key algorithm

- Key Generation:
  1. Choose two large primes, p and q
  2. Compute n = p × q,
  3. Compute z = (p-1) × (q-1)
  4. Choose d to be relative prime to z, i. e. no common factor
  5. Find e such that (d × e) mod z = 1
- **Key used for Encryption (Public Key)** is (e,n) → Cipher = $Plain^e \ (mod\ n)$
- **Key used for Decryption (Private Key)** is (d,n) → Plain = $Cipher^d \ (mod\ n)$

Question 1

Given the RSA algorithm we studied last week,
if p = 3, q = 11 and if d = 3 and e = 7 instead of the version we saw in class, using the same character mapping we saw in class though, where A is 01 and B is 02, and C is 03 and so on, how would RSA work? Would it work at all? Show in detail what numbers would be computed and transmitted at both ends of a transmission if we want to send across a "D". Show where it fails if it does not work properly?

- Is that reasonable to choose d =3, e =7?
- If that's reasonable, can we show the detail process of encrypt "D" and decrypt "D"?

Question 1

Given the RSA algorithm we studied last week,
if p = 3, q = 11 and if d = 3 and e = 7 instead of the version we saw in class, using the same character mapping we saw in class though, where A is 01 and B is 02, and C is 03 and so on, how would RSA work? Would it work at all? Show in detail what numbers would be computed and transmitted at both ends of a transmission if we want to send across a "D". Show where it fails if it does not work properly?

Answer:

1. p = 3, q =11
2. n = p × q =3 × 11 =33
3. z = (p-1) × (q-1) = 20
4. Choose d to be relative prime to z (no common factor),
   - Here we choose d = 3, there's no common factor between z and d (20 and 3), Correct choice
5. Find e such that (d × e) mod z = 1 .
   - Here we choose e =7. (7 ×3) mod 20 =1. Correct choice
6. **Key used to do Encryption** is (e,n) → (7,33)
7. **Key used to do Decryption** is (d,n) →(3,33)

The plain Text is D represented by 04
Encryption : Cipher = $Plain^e \ (mod\ n) = 4^7 \ (mod\ 33) = 16$
Decryption : Plain = $Cipher^d (mod\ n) = 16^3 \ (mod\ 33) = 4$ Correct, successfully convert back to original plain text

Detail explanation of Question1:

It works! Please refer to Week 11 Lecture 2 slides 4 and 6.

p = 3, q = 11 means z is (3 - 1) x (11 - 1) = 20

d is chosen to be 3 which has no common factors with z which is good. e is 7 which means (d x e) is 3 x 7 = 21. Thus 21 mod 20 is 1 which is another good choice!

n is p x q = 33 .
For encryption the pair to use 7,33 which is the public key ,and for decryption 3, 33 is used which is the private key! To send "D", first we see that it has numerical value is 4 as per this question's suggestion. And 4 ^ 7 = 16384.

16384 mod 33 = 16 is found next (Ok to use a calculator here but not necessary if you see 4 ^ 7 is 2 ^ 14

16 is sent in transmission and then we take 16 ^ 3 = 4096 upon receipt, and then 4096 mod 33 to get 4 which concludes decryption, 4 is "D" in our coding. Eureka!

Question 2

Using the RSA algorithm we saw in class, please design a simple algorithm to sign documents where the sender cannot refute the fact that a document was signed by herself later on. Which property of RSA your algorithm relies on?

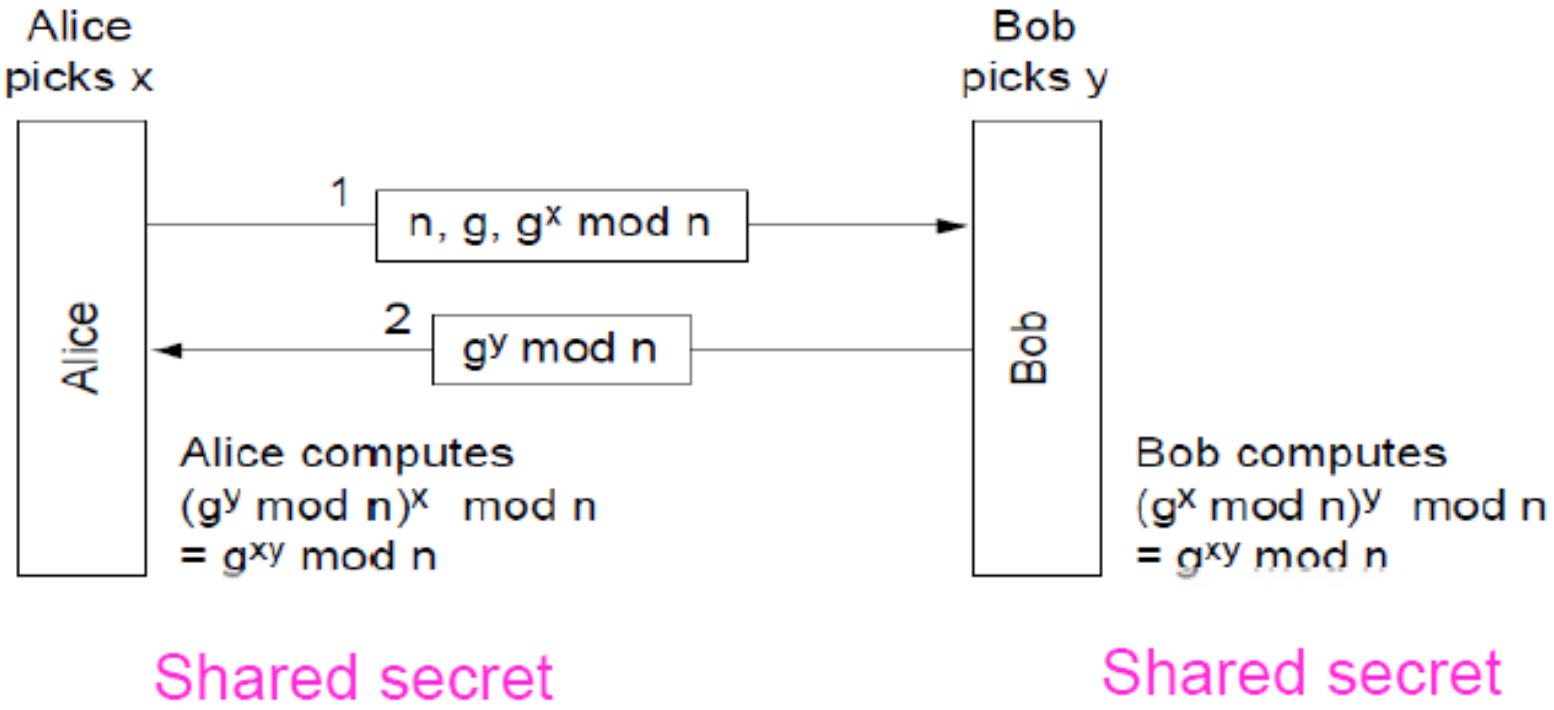
Ans. The algorithm is as follows:

- Someone sends a document for signature to person A.
- Our person A signs it by using her private key PrK to, pretty much uses private key to lock the document.
- Then this message is sent to whoever needs it. We can add the plain text message to this communication as well or the original document can be accessible from a webpage etc.
- Receiver uses the public key of the sender A, say PuK, to open the message and if the text matches the original plaintext of the document then sender A should be the one who signed this document as there is no other person who can lock the document with her private key as only she knows that key; which only our public key is capable of countering the effect of...
- We rely on the property that E(D(P)) = P in RSA as well as D(E(P)) = P using these key pairs.

Note: This is the concept of digital signature (Signature of message Digest)!

Question 3

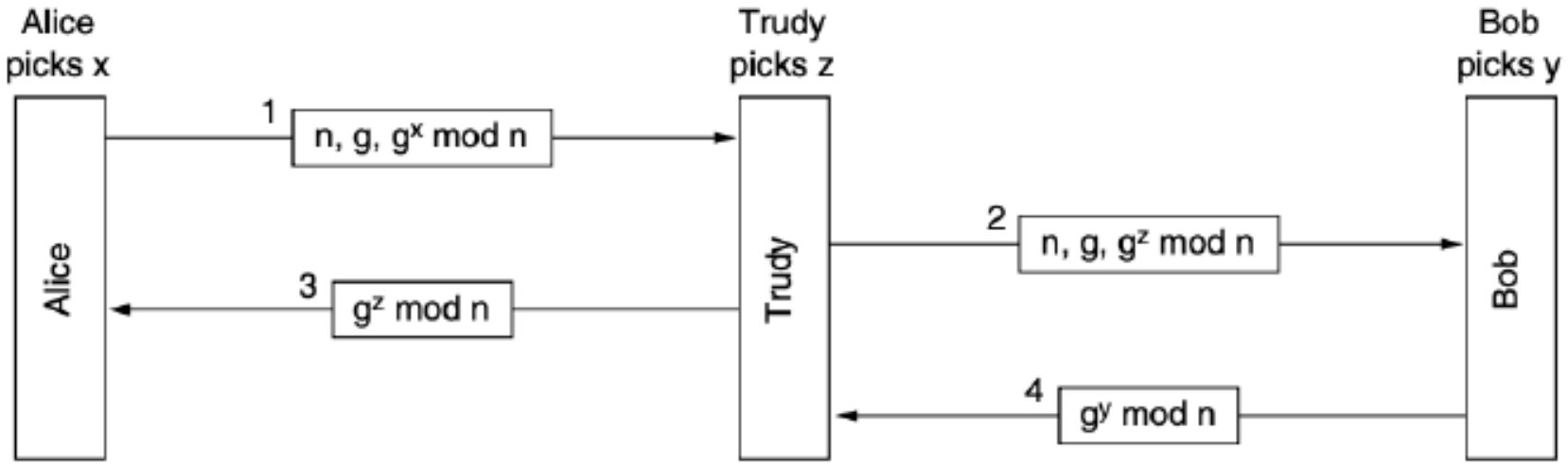
Given the Diffie-Hellman key challenge in the lectures please develop the full flow chart for the man-in-the-middle (MITM) attack, with step numbers and messages sent, show details about how this attack would work.



Question 3

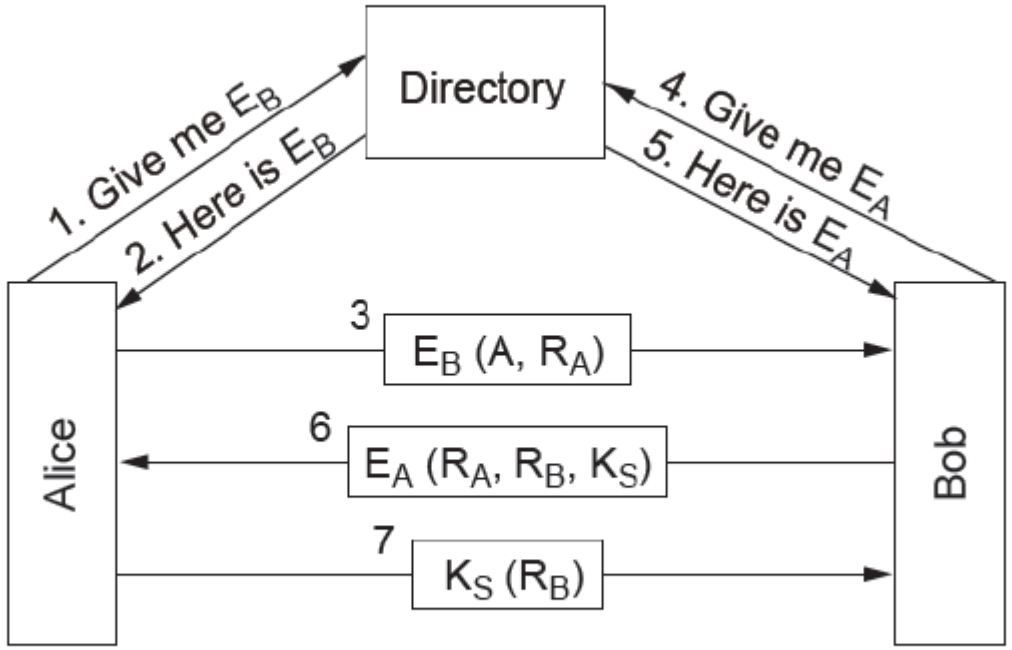
Given the Diffie-Hellman key challenge in the lectures please develop the full flow chart for the man-in-the-middle (MITM) attack, with step numbers and messages sent, show details about how this attack would work.

Man-in-the-middle (MITM) attack



Question 4

Leveraging the authentication protocol using Public-Key cryptography, we send across two additional numbers, R_A and R_B. Why are these needed? Why not Alice sends only her name to Bob but needs a R_A as well?



- A,B: Identity of Alice and Bob
- R_i's are the challenges (a big random number), where I identifies the challenger
- K_S: Session Key

Question 4

Leveraging the authentication protocol using Public-Key cryptography, we send across two additional numbers, R_A and R_B. Why are these needed? Why not Alice sends only her name to Bob but needs a R_A as well?

Ans. Without R_A, Bob can still send back an acknowledgement but Alice cannot be sure that whether the responding person is Bob or not. The R_A is needed to prove that Bob opened the initial message with his private key, saw R_A, and in the response message sends it to Alice to prove this. Same is true for the role of RB.

Question 5

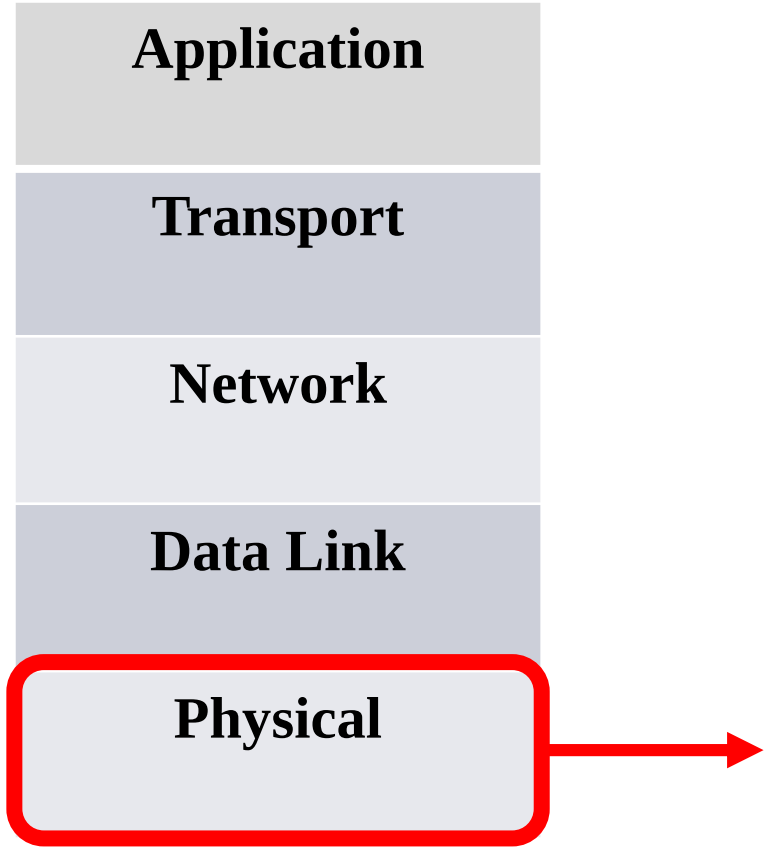
Please list, summarize the four key areas/aspects of network security.

Ans. The four key areas/aspects are:
Secrecy: keeping information hidden from a general audience, i.e., except the intended party
Authentication: Ensuring the user you are giving the content to has the valid id/credentials
Non-repudiation: Proving that the content belongs to/send by a named sender
Integrity control: Ensuring the content is not tampered with, e.g., during transport

THE UNIVERSITY OF MELBOURNE

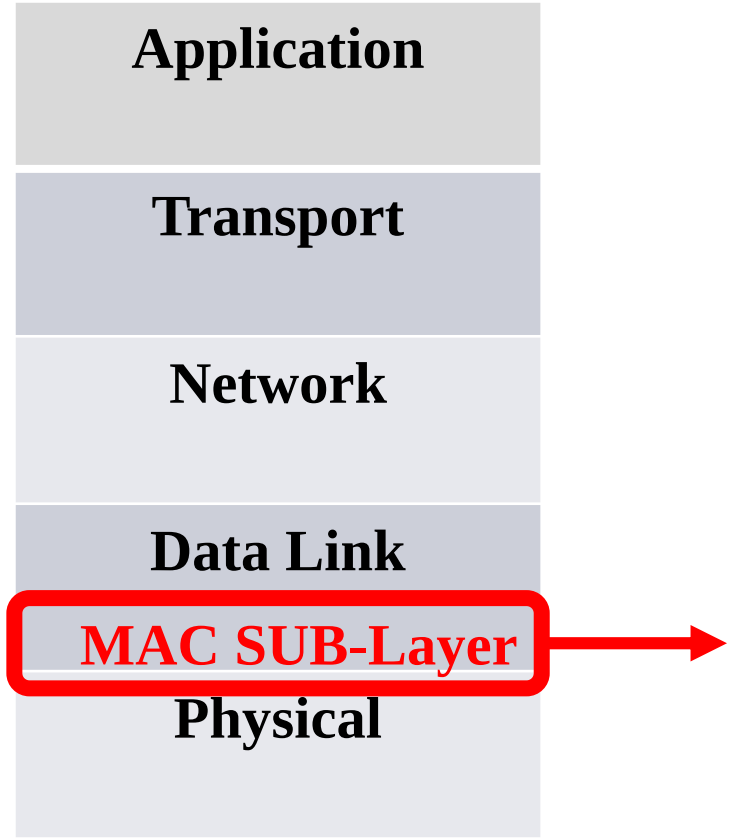
Review

Physical Layer



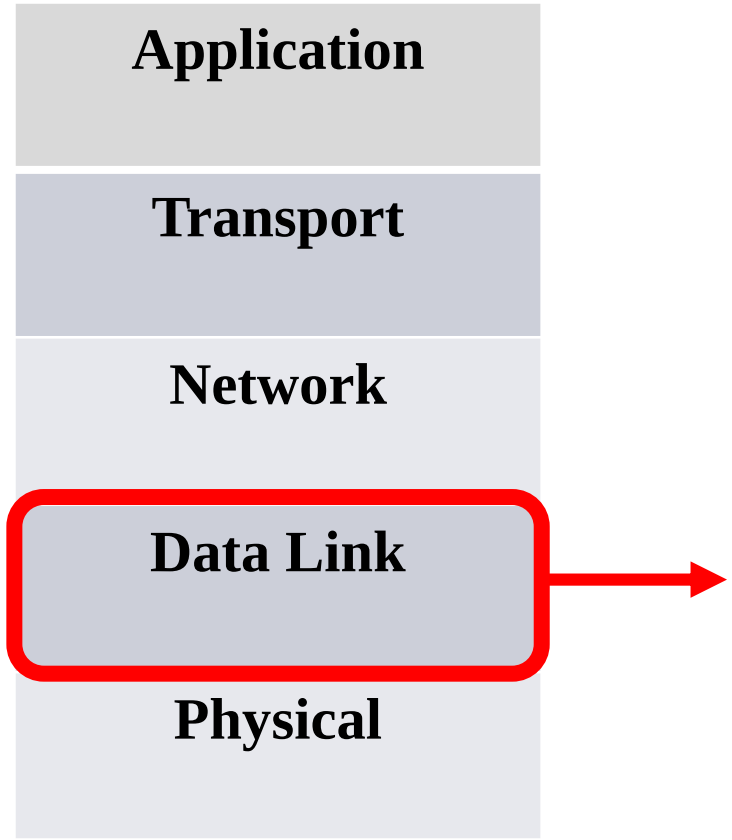
1. Latency = Transmission Delay + Propagation Delay
2. Topology (Linear, Ring, Full Mesh, Simplex, Half Duplex, Full Duplex)
3. Sampling
4. Max Data Rate

Mac-sub Layer



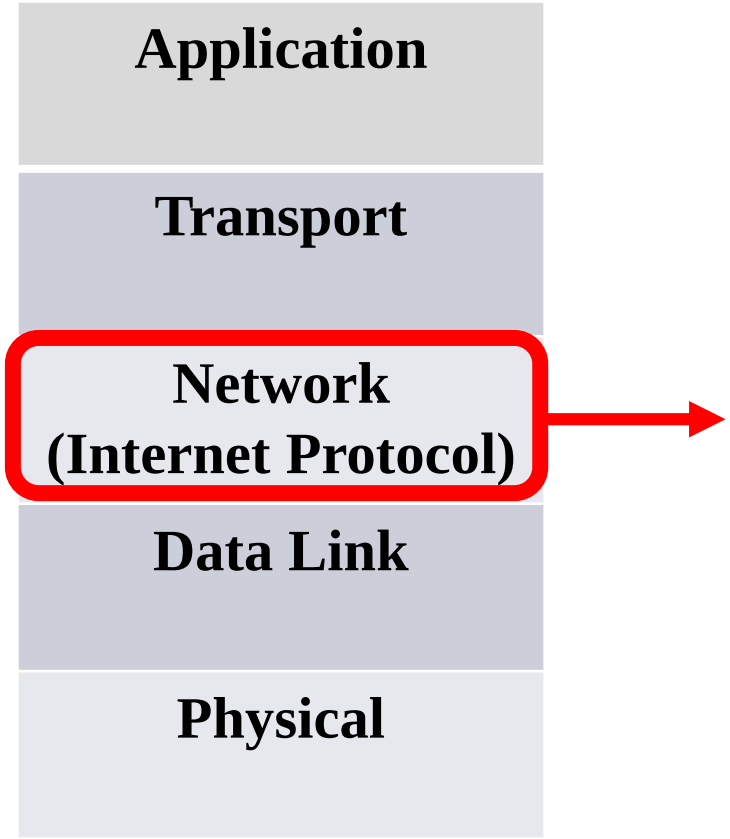
1. Contention
   - ALOHA
   - Carrier Sense Multiple Access (CSMA)
2. Collision Free
   - CSMA/CD – Binary Countdown
   - CSMA/CD - bit map
3. Limited Contention
   - CSMA/CD - Adaptive Tree Walk Protocol
4. MACA/MACAW (for Wireless LANs)

Data Link Layer



1. Framing methods
   - Character Count
   - Flag Bytes with Byte stuffing
   - Flag with Bit stuffing
2. Error Control
   - Error Bounds (Hamming Distance)
   - Detecting (parity, checksum, CRC)
   - Correcting (Hamming code)
   - Re-transmission
3. Flow Control
   - Feedback Based Flow Control
     - Stop and wait
     - Sliding window
   - ~~Rate based flow control~~

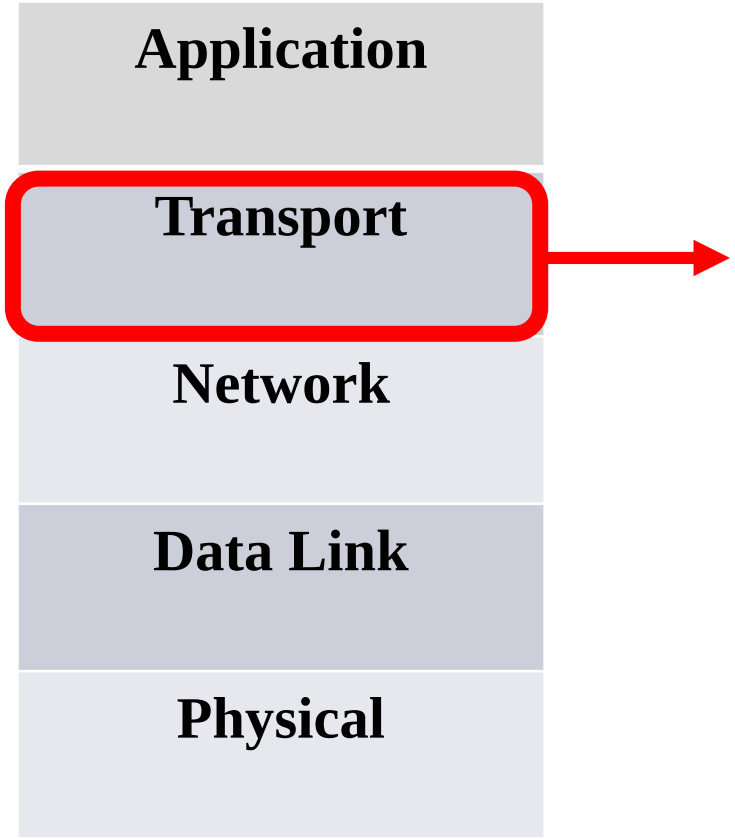
Network Layer



1. Routing Method
   - Virtual-Circuit subnet (Connection Oriented)
   - Datagram subnet (Connectionless)
2. IPV4
   - Datagram
   - IP address and subnetting (important!)
3. Fragmentation
4. Routing algorithm (Manage Routing Table)

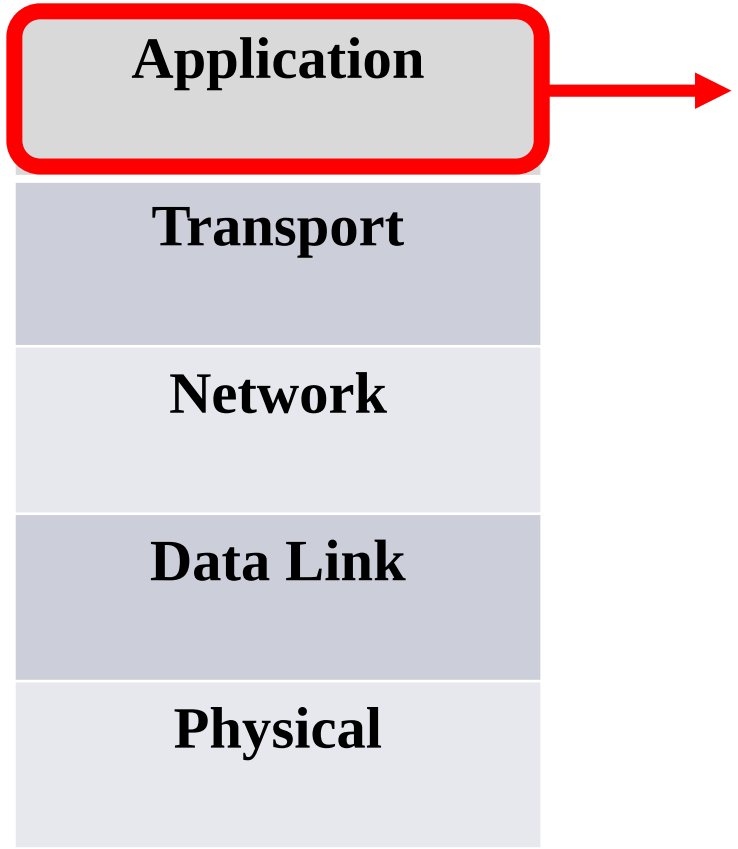
★
- Non-adaptive
  - Shortest Path routing
  - Flooding
- Adaptive
  - Distance Vector Routing
  - Links state routing
- Hierarchical Routing
- Broadcasting Routing
- Multicasting Routing

Transport Layer

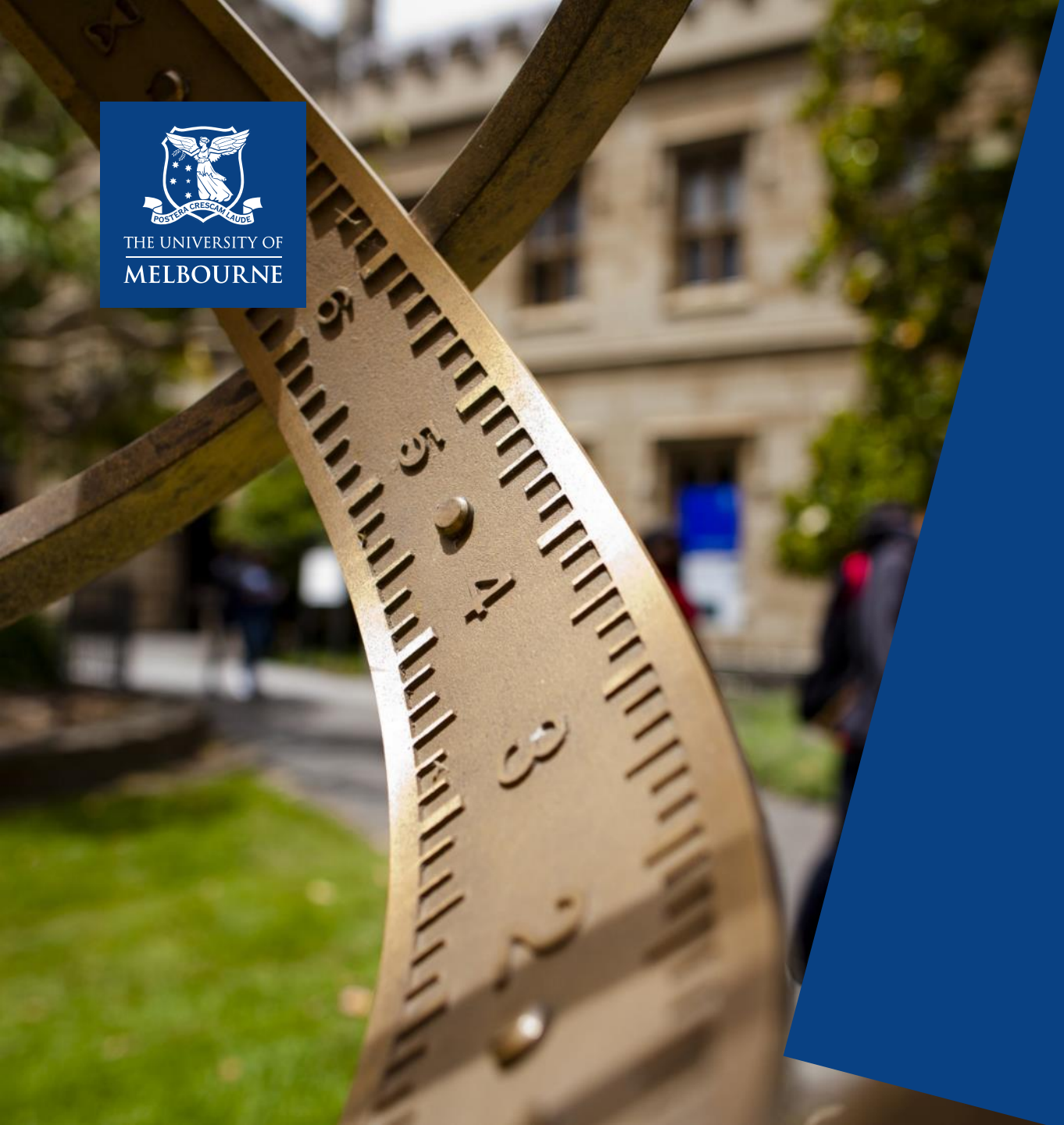


1. Transport Layer Encapsulation
2. Transmission Control Protocol (TCP)
   - Connection-establishment
     - Three-way handshake
   - Connection-release
     - Asymmetric
     - Symmetric
   - TCP segment header
   - TCP based socket
   - Error-control
   - Flow-control
   - Congestion-control
     - ❑ Slow start, Additive increase
     - ❑ Tahoe, Reno
3. User Datagram Protocol (UDP)
4. Techniques for achieving Quality of Service (QoS)

Application Layer



1. DNS (Domain Name server)
   - Query process
2. WWW components
3. HTTP (Hyper transfer protocol)
   - Non-persistent/ Persistent
   - HTTP Request Message format
   - HTTP Request method
   - HTTP Error code
4. Cookie/Session
5. Web Cache
6. Multi-media data transmission
7. Email



Final words

Final words

- Tutorial only covers part of the content, please also review lecture slides.
- Make sure you do all the tutorial questions again before final exam.
- Make sure you do all the past exam paper!

THE UNIVERSITY OF MELBOURNE

All the best!