# 1.

a)

Best-effort: there are no guarantees that packets will be delivered, time of delivery may vary, packets can be lost, delayed, reordered, duplicated, corrupted, etc. This is easy to run over any type of link layer. The Internet offers this because

b)

Please do not throw salami pizza away

Layers:

1. Physical Layer
   1. Defines characteristics of the medium through which data (raw bits) is transferred; encodes raw signals into bits
2. Data Link Layer
   1. Structures and frames physical layer bit stream, detects errors, controls media access
3. Network Layer
   1. Creates a network given hosts, names and addresses them, abstracts protocols with IP
4. Transport Layer
   1. Delivers structured data with TCP/UDP, may provide reliability, ordering, framing, congestion control
5. Session Layer
   1. Manages multiple transport layer connections, coordinates audio, video flows
6. Presentation Layer
   1. Manages presentation, representation and conversion of data (character sets, markup, format, content negotiation)
7. Application Layer
   1. Implements user application protocols, Domain Name Service

c)

Original state (time 0):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0/\* | 4/B | 5/C | 2/D | /- |
| B | 4/A | 0/\* | 2/C | /- | /- |
| C | 5/A | 2/B | 0/\* | /- | 1/E |
| D | 2/A | /- | /- | 0/\* | 1/E |
| E | /- | /- | 1/C | 1/D | 0/\* |

Time 1:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0/\* | 4/B | 5/C | 2/D | 3/D |
| B | 4/A | 0/\* | 2/C | 6/A | 3/C |
| C | 5/A | 2/B | 0/\* | 2/E | 1/E |
| D | 2/A | 6/A | 2/E | 0/\* | 1/E |
| E | 3/D | 3/C | 1/C | 1/D | 0/\* |

Time 2:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0/\* | 4/B | 4/D | 2/D | 3/D |
| B | 4/A | 0/\* | 2/C | 4/C | 3/C |
| C | 4/E | 2/B | 0/\* | 2/E | 1/E |
| D | 2/A | 4/E | 2/E | 0/\* | 1/E |
| E | 3/D | 3/C | 1/C | 1/D | 0/\* |

d)

|  |  |
| --- | --- |
| Telephone network | The Internet |
| + Guaranteed capacity | - Available capacity varies |
| - One communication can block others (multiple connections impossible) | + Connectivity guaranteed (multiple connections usual) |
| - Entire message can be lost | - Messages split into multiple packets, packets can be lost |

e)

Server:

* Create socket in IPv4 family
* Bind to some host with port number
* Start listening to incoming connection requests
* Accept connection
* Send/receive requests/messages
* Close client socket

Client:

* Create socket
* Connect to some host with port number
* Send/receive requests/messages
* Close socket

f)

The important problem in symmetric cryptography is the question of how to safely distribute the secret key between communicating hosts. Public key cryptography solves it by encrypting the random, ephemeral session key (to be used with symmetric cryptography) with a public key and requiring the other host to have a private key to decrypt it and keep using symmetric cryptography.

## 2.

a)

i) LRU:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stream | C | D | A | B | D | A | C | B | C | D | A |
| Slot 1 | C (0) | C (0) | C (0) | B (3) | B (3) | B (3) | C (6) | C (6) | C (8) | C (8) | C (8) |
| Slot 2 |  | D (1) | D (1) | D (1) | D (4) | D (4) | D (4) | B (7) | B (7) | B (7) | A (10) |
| Slot 3 |  |  | A (2) | A (2) | A (2) | A (5) | A (5) | A (5) | A (5) | D (9) | D (9) |
| Cache miss | \* | \* | \* | \* |  |  | \* | \* |  | \* | \* |

ii) LFU:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stream | C | D | A | B | D | A | C | B | C | D | A |
| Slot 1 | C (1) | C (1) | C (1) | B (1) | B (1) | B (1) | C (1) | B (1) | C (1) | C (1) | C (1) |
| Slot 2 |  | D (1) | D (1) | D (1) | D (2) | D (2) | D (2) | D (2) | D (2) | D (3) | D (3) |
| Slot 3 |  |  | A (1) | A (1) | A (1) | A (2) | A (2) | A (2) | A (2) | A (2) | A (3) |
| Cache miss | \* | \* | \* | \* |  |  | \* | \* | \* |  |  |

b)

FCFS:

* Scheduling order/execution times:
  + Time: 0; Ready queue: P1; next process: P1
    - P1 executes: 0 – 5
  + Time: 5; Ready queue: P2, P3, P4; next process: P2
    - P2 executes 5 – 7
  + Time: 7; Ready queue: P3, P4, P5; next process: P3
    - P3 executes 7-10
  + Time: 10; Ready queue: P4, P5; next process: P4
    - P4 executes 10-15
  + Time: 15; Ready queue: P5; next process: P5
    - P5 executes 15-16
* Waiting times:
  + P1: 0-0 = 0; P2: 5-1 = 4; P3: 7-1 = 6; P4: 10-3 = 7; P5: 15-6 = 9
  + Avg: 5.2
* Turnaround times:
  + P1: 5-0 = 5; P2: 7-1 = 6; P3: 10-1 = 9; P4: 15-3 = 12; P5: 16-6 = 10
  + Avg: 8.4

SJF:

* Scheduling order/execution times:
  + Time: 0; Ready queue: P1; next process: P1
    - P1 executes: 0 – 5
  + Time: 5; Ready queue: P2, P3, P4; next process: P2
    - P2 executes 5 – 7
  + Time: 7; Ready queue: P3, P4, P5; next process: P5
    - P5 executes 7-8
  + Time: 8; Ready queue: P4, P3; next process: P3
    - P3 executes 8-11
  + Time: 11; Ready queue: P4; next process: P4
    - P4 executes 11-16
* Waiting times:
  + P1: 0-0 = 0; P2: 5-1 = 4; P3: 8-1 = 7; P4: 11-3 = 8; P5: 7-6 = 1
  + Avg: 4
* Turnaround times:
  + P1: 5-0 = 5; P2: 7-1 = 6; P3: 11-1 = 10; P4: 16-3 = 13; P5: 8-6 = 2
  + Avg: 7.2

Non-pre-emptive priority:

* Scheduling order/execution times:
  + Time: 0; Ready queue: P1; next process: P1
    - P1 executes: 0 – 5
  + Time: 5; Ready queue: P2, P3, P4; next process: P3
    - P3 executes 5-8
  + Time: 8; Ready queue: P2, P4, P5; next process: P5
    - P5 executes 8-9
  + Time: 9; Ready queue: P4, P2; next process: P4
    - P4 executes 9-14
  + Time: 14; Ready queue: P2; next process: P2
    - P2 executes 14-16
* Waiting times:
  + P1: 0-0 = 0; P2: 14-1 = 13; P3: 5-1 = 4; P4: 9-3 = 6; P5: 8-6 = 2
  + Avg: 5
* Turnaround times:
  + P1: 5-0 = 5; P2: 16-1 = 15; P3: 8-1 = 7; P4: 14-3 = 11; P5: 9-6 = 3
  + Avg: 8.2

Pre-emptive RR (quantum 3):

* Scheduling order/execution times:
  + Time: 0; Ready queue: P1 (5); next process: P1
    - P1 executes: 0 – 3
  + Time: 3; Ready queue: P2 (2), P3 (3), P4 (5), P1 (2); next process: P2
    - P2 executes 3-5
  + Time: 5; Ready queue: P3 (3), P4 (5), P1 (2); next process: P3
    - P3 executes 5-8
  + Time: 8; Ready queue: P4 (5), P1 (2), P5 (1); next process: P4
    - P4 executes 8-11
  + Time: 11; Ready queue: P1 (2), P5 (1), P4 (5); next process: P1
    - P1 executes 11-13
  + Time: 13; Ready queue: P5 (1), P4 (5); next process: P5
    - P5 executes 13-14
  + Time: 14; Ready queue: P4 (2); next process: P4
    - P4 executes 14-16
* Waiting times:
  + P1: (0-0)+(11-3) = 8; P2: 3-1 = 2; P3: 5-1 = 4; P4: (8-3)+(14-11) = 8; P5: 13-6 = 7
  + Avg: 5.8
* Turnaround times:
  + P1: 13-0 = 13; P2: 5-1 = 4; P3: 8-1 = 7; P4: 16-3 = 13; P5: 14-6 = 8
  + Avg: 9