Chapter 3 - Part 2

INTERNETWORKING

Eng. Haneen El-Masry

March, 2014
3.2 INTERNETWORKING

An internetwork is an arbitrary collection of networks interconnected to provide host to host packet delivery service.

The nodes that interconnect the networks are called routers.

The Internet is an internetwork.

An internetwork is a logical network built out of a collection of physical networks.

A physical network is either a directly connected or a switched network.

A physical network uses one technology.

The Internet Protocol (IP) is the key tool used today to build scalable, heterogeneous internetworks.

IP runs on all the nodes (both hosts and routers) in a collection of networks and allows these nodes and networks to function as a single logical internetwork.

**IP service model:** best-effort datagram (connectionless) delivery:

- Packets can get lost (IP service is unreliable).
- Packets can get delivered out of order.
- Packets can be delayed for a long time.

**Fragmentation and Reassembly**

Every network type has a maximum transmission unit (MTU): The largest IP datagram that it can carry in a frame.

Ethernet MTU is 1500 bytes, Wi-Fi MTU is 2312 bytes.

Source host and routers divide an IP datagram into fragments if datagram size > MTU.

All the fragments carry the same identifier.

Reassembly is done at destination host.

Destination host gives up reassembly if some fragments are lost.

IP does not attempt to recover from missing fragments.
Exercise 36:

At A:

Data Size= TCP MSG Data + TCP Header= 1024 + 20 =1044 B.

Malalaaah: 

Transport Layer هو عبارة عن حجم ال Packet التي رح تستلمها ال Network Layer من ال TCP Packet وهي هنا TCP Packet.

Datagram Size = Data + IP Header = 1044+20 =1064 B.

Datagram > MTU=1024 B

Then We need Fragmentation:

# Fragments= Γ Datagram/MTU 1 = Γ 1064/1024 1 = 2 fragments.

Each Fragment is a datagram, then each fragment consist of IP Header and Data.

F1: 


F2: 


لنجد حجم الداتا في كل fragments:

1- حجم ال MTU- IP Header للأكثرا يكون fragment.
2- حجم الداتا في ال fragments ما عدا الاخيرة يقبل القسمة على 8.
3- يبقى القسمة ع 8 أم لا؟؟
4- إذا بنخل حجم الداتا في كل (MTU – IP Header) .

أذا لا نجد العدد الأصغر من (MTU – IP Header) مباشرة والتي يقبل القسمة ع 8. 

الأخيرة = حجم الداتا الكلية – مجموع الداتا في ال fragments.

في السؤال:

MTU- IP Header= 1024 – 20 = 1004

لا تقبل القسمة ع 8 إذن رح نختار ال 1000 كحجم للداتا في F1 وبناء عليه رح يكون حجم الداتا في F2 1044-1000=44.
More Flag and Offset:

More flag هو بت واحد بيكون 1 إذا كان فيه بعد ال fragments .. fragment غيرها و 0 إذا لا. 

Offset من خلاله ينعرف رقم أول بايت في الداتا الموجودة في ال fragment بالنسبة للداتا الكلية وهو = مجموع الداتا الموجودة في كل ال fragments السابقة / 8

F1:
More = 1 ... Offset = 0/8 = 0

F2:
More = 0 ... Offset = 1000/8 = 125

At R:

Fragmentation عند Router وصله F1, F2 عمليات فragment F1 حجمها أكبر من 576 آذن رح نحتاج أن نعمل لها fragmentation بينما F2 حجمها أقل من 576 آذن رح نبعتها متا ما هيا.

Fragmentation of F1 at R:

#fragments = ⌈1020/576⌉ = 2

F1-I: 20 552
F1-II: 20 448

More flag and Offset:
F1-I:
More = 1 ,,,, Offset = 0

F1-II:
More = 1 ,,,, Offset = 552/8 = 69

لا هنا ال 1 ف في الانتهان لانهم ناتجن عن more بالأنس ف ال fragments 3 8
اذن رح يوصل ل بالشكل التالي :
Exercise 37:

Path MTU is the smallest MTU of any link on the current path (route) between two hosts.

Path MTU = 576 B.

At A:
# Fragments = \( \frac{\text{Datagram/MTU}}{\text{Path MTU}} \) = \( \frac{1064}{576} \) = 2 fragments.

<table>
<thead>
<tr>
<th>F1</th>
<th>20</th>
<th>552</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2</td>
<td>20</td>
<td>492</td>
</tr>
</tbody>
</table>

More flag and Offset:

F1:

More = 1 ,, offset = 0

F2:

More = 0 ,, offset = 552/8 = 69
**IP Addresses**

Each host/router interface is assigned a globally unique 32-bit IP address.

Dotted decimal notation: 129.186.3.6

IP addresses are hierarchical: network part + host part

All hosts attached to the same network have the same network part and different host part in their IP address.

IP addresses were originally divided into 3 classes:

![IP address classes diagram](image)

**Datagram Forwarding in IP**

Every datagram contains the IP address of the destination host.

If a node (host or router) is directly connected to destination network, then the node sends the datagram directly to destination host.

If a node is not directly connected to destination network, then the node sends the datagram to the next hop router.

The node finds the correct next hop router by consulting its **forwarding table**.

Each host has a **default router**: if the host is not directly connected to the destination network, then it sends the datagram to the default router.

Each router maintains a forwarding table

- Table contains \(<\text{NetworkNum}, \text{NextHop}>\) pairs.
- Table may contain a default entry specifying a default router that is used if none of the entries in the table matches destination host’s network number.
**Router Datagram Forwarding Algorithm**

- if (NetworkNum of destination = NetworkNum of one of my interfaces) then deliver packet to destination over that interface
- else if (NetworkNum of destination is in my forwarding table) then deliver packet to NextHop router
  - else deliver packet to default router

**Host Datagram Forwarding Algorithm**

- if (NetworkNum of destination = my NetworkNum) then deliver packet to destination directly
- else deliver packet to default router

Forwarding table lists only network numbers rather than all the hosts in the network.

**SubNetting**

Assigning one network number per physical network is inefficient.

**The Solution:**

Allocate a single network number to several physical networks, referred to as **Subnets**.

- All hosts on the same physical network have the same subnet number.
- All physical networks share the same network number.
- An IP address now has three parts: network + subnet + host.

IP address **bitwise AND** subnet mask = subnet number.

**Host Forwarding Algorithm**

- if (my subnet mask & destination IP address = my subnet number) Deliver packet to destination directly.
- else deliver packet to default router.
Router Forwarding Algorithm

D = destination IP address

for each forwarding table entry <SubnetNumber, SubnetMask, NextHop>

- if (SubnetMask & D = SubnetNumber)
  - if NextHop is an interface deliver packet directly to destination
  - else deliver packet to NextHop (a router)
- if no matches are found, then deliver packet to default router

Exercise 55:

<table>
<thead>
<tr>
<th>SubnetNumber</th>
<th>SubnetMask</th>
<th>NextHop</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.96.39.0</td>
<td>255.255.255.128</td>
<td>Interface 0</td>
</tr>
<tr>
<td>128.96.39.128</td>
<td>255.255.255.128</td>
<td>Interface 1</td>
</tr>
<tr>
<td>128.96.40.0</td>
<td>255.255.255.128</td>
<td>R2</td>
</tr>
<tr>
<td>192.168.153.0</td>
<td>255.255.255.192</td>
<td>R3</td>
</tr>
<tr>
<td>(default)</td>
<td></td>
<td>R4</td>
</tr>
</tbody>
</table>

ملاحظة:

عندما نواجه نتائج And وال SubnetworkAddress بين Nتابل Subnetting أول ما نلقي نظرة على Subnetting يتناقص وخصووصا إذا كان حاكي بالسؤال ان الراوتر يستخدم最长 Prefix Match للكل لذا Pool Matches if no matches are found, then deliver packet to default router

a) 128.96.39.10

128.96.39.10 & 255.255.255.128 = 128.96.39.0 >> Interface 0.

128.96.39.10 & 255.255.255.192= 128.96.39.0 >> No Matching

Then Use Interface 0 as the next hop.

b) 128.96.40.12

128.96.40.12 & 255.255.255.128 = 128.96.40.0 >> R2.

128.96.40.12 & 255.255.255.192= 128.96.40.0 >> No Matching

Then Use R2 as the next hop.
c) **128.96.40.151**

128.96.40.151 & 255.255.255.128 = 128.96.40.128 >> No Matching

128.96.40.151 & 255.255.255.192 = 128.96.40.128 >> No Matching

Then Use R4 as the next hop.

d) **192.4.153.17**

192.4.153.17 & 255.255.255.128 = 192.4.153.0 >> No matching

192.4.153.17 & 255.255.255.192 = 192.4.153.0 >> Use R3 as the next hop.

e) **192.4.153.90**

192.4.153.90 & 255.255.255.128 = 192.4.153.0 >> No matching

192.4.153.90 & 255.255.255.192 = 192.4.153.64 >> No Matching.

>> Use R4 as the next hop.

**Exercise 56:***

Apply each subnet mask and, if the corresponding subnet number matches the SubnetNumber column, then use the entry in Next-Hop.

(a) Applying the subnet mask 255.255.254.0, we get 128.96.170.0. Use interface 0 as the next hop.

(b) Applying subnet mask 255.255.254.0, we get 128.96.166.0. (Next hop is Router 2.) Applying subnet mask 255.255.252.0, we get 128.96.164.0. (Next hop is Router 3.) However, 255.255.254.0 is a longer prefix, so use Router 2 as the next hop.

(c) None of the subnet number entries match, so use default Router R4.

(d) Applying subnet mask 255.255.254.0, we get 128.96.168.0. Use interface 1 as the next hop.

(e) Applying subnet mask 255.255.252.0, we get 128.96.164.0. Use Router 3 as the next hop.
**Classless InterDomain Routing (CIDR)**

Suppose we assign the class C network numbers from 192.4.16 through 192.4.31.

Observe that the top 20 bits of all the addresses in this range are the same:

(11000000 00000100 0001)

We have effectively created a 20-bit network number.

To represent a network number, or prefix, we place a / X after the prefix where X is the prefix length in bits.

For example, the 20-bit prefix for all the networks 192.4.16 through 192.4.31 is represented as 192.4.16/20

We have a single network number that can be used in forwarding tables!

An ISP can advertise a single route to all the customers if the customers share a common prefix (Network Number).

**IP Forwarding**

Each forwarding table entry is a <NetNumber/MaskLength, NextHop> pair

When a packet comes in:

Router scans the forwarding table entry by entry, looking for a match. A match occurs when destination IP address & Mask = NetNumber.

If multiple entries match, the entry with the longest match wins.
Exercise 72:

<table>
<thead>
<tr>
<th>Net Mask</th>
<th>Net/MaskLength</th>
<th>NextHop</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF.F0.0.0</td>
<td>C4.50.0.0/12</td>
<td>A</td>
</tr>
<tr>
<td>FF.FF.0.0</td>
<td>C4.5E.10.0/20</td>
<td>B</td>
</tr>
<tr>
<td>FF.F0.0.0</td>
<td>C4.60.0.0/12</td>
<td>C</td>
</tr>
<tr>
<td>FF.FC.0.0</td>
<td>C4.68.0.0/14</td>
<td>D</td>
</tr>
<tr>
<td>80.0.0.0</td>
<td>80.0.0.0/1</td>
<td>E</td>
</tr>
<tr>
<td>C0.0.0.0</td>
<td>40.0.0.0/2</td>
<td>F</td>
</tr>
<tr>
<td>C0.0.0.0</td>
<td>00.0.0.0/2</td>
<td>G</td>
</tr>
</tbody>
</table>

### a) C4.5E.13.87

C4.5E.13.87 & FF.F0.0.0 = C4.50.0.0 >> Next Hop: A

C4.5E.13.87 & FF.FF.0.0 = C4.5E.10.0 >> Next Hop: B

B is the longest match then the next hop is B.

### b) C4.5E.22.09

C4.5E.22.09 & FF.F0.0.0 = C4.50.0.0 >> Next Hop: A

No other matches, then the next hop is A.

### c) C3.41.80.02

C3.41.80.02 & 80.0.0.0.0 = 80.0.0.0 >> Next Hop: E

No other matches, then the next hop is E.

### d) 5E.43.91.12

5E.43.91.12 & C0.0.0.0 = 40.0.0.0 >> Next Hop: F

No other matches, then the next hop is F.
Exercise 73:

(a) F  (b) B  (c) E  (d) A  (e) D  (f) C

3.3 ROUTING

Routing is the process by which forwarding tables are built.

There are two main classes of routing protocols:

1- Distance vector:

Each node constructs a one-dimensional array (a vector) containing the “distances” (costs) to all other nodes and distributes that vector to its immediate neighbors.

2- Link state:

When a router is initialized, it determines the link cost on each of its network interfaces. The router then advertises this set of link costs to all other routers in the internet topology, not just neighboring routers.
Exercise 46:

(a) | Information Stored at Node | Distance to Reach Node |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>∞</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>∞</td>
</tr>
<tr>
<td>F</td>
<td>∞</td>
</tr>
</tbody>
</table>

(b) | Information Stored at Node | Distance to Reach Node |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>∞</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
</tr>
<tr>
<td>F</td>
<td>9</td>
</tr>
</tbody>
</table>

(c) | Information Stored at Node | Distance to Reach Node |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>6</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>6</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
</tr>
<tr>
<td>F</td>
<td>9</td>
</tr>
</tbody>
</table>

ملاحظة: 
في جدول (b) بتعتمد عالمعلومات الموجودة في جدول (c) عالمعلومات اللي في جدول (a)
Exercise 47:

(a) | Information Stored at Node | Distance to Reach Node |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>∞</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>∞</td>
</tr>
<tr>
<td>E</td>
<td>∞</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

(b) | Information Stored at Node | Distance to Reach Node |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>∞</td>
<td>4</td>
</tr>
</tbody>
</table>

(c) | Information Stored at Node | Distance to Reach Node |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>
Exercise 52:

The cost=1 links show A connects to B and D; F connects to C and E. F reaches B through C at cost 2, so B and C must connect. F reaches D through E at cost 2, so D and E must connect. A reaches E at cost 2 through B, so B and E must connect. These give:

As this network is consistent with the tables, it is the unique minimal solution.

Exercise 53:

The following is an example network topology.

Exercise 62:

<table>
<thead>
<tr>
<th>Step</th>
<th>confirmed</th>
<th>tentative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(A,0,-)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(A,0,-)</td>
<td>(D,2,D) (B,5,B)</td>
</tr>
<tr>
<td>3</td>
<td>(A,0,-) (D,2,D)</td>
<td>(B,4,D) (E,7,D)</td>
</tr>
<tr>
<td>4</td>
<td>(A,0,-) (D,2,D) (B,4,D)</td>
<td>(E,6,D) (C,8,D)</td>
</tr>
<tr>
<td>5</td>
<td>(A,0,-) (D,2,D) (B,4,D) (E,6,D)</td>
<td>(C,7,D)</td>
</tr>
<tr>
<td>6</td>
<td>(A,0,-) (D,2,D) (B,4,D) (E,6,D) (C,7,D)</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 63:

<table>
<thead>
<tr>
<th>Step</th>
<th>Confirmed</th>
<th>Tentative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(A,0,-)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(A,0,-)</td>
<td>(B,1,B) (D,5,D)</td>
</tr>
<tr>
<td>3</td>
<td>(A,0,-) (B,1,B)</td>
<td>(D,4,B) (C,7,B)</td>
</tr>
<tr>
<td>4</td>
<td>(A,0,-) (B,1,B) (D,4,B)</td>
<td>(C,5,B) (E,7,B)</td>
</tr>
<tr>
<td>5</td>
<td>(A,0,-) (B,1,B) (D,4,B) (C,5,B)</td>
<td>(E,6,B)</td>
</tr>
<tr>
<td>6</td>
<td>(A,0,-) (B,1,B) (D,4,B) (C,5,B) (E,6,B)</td>
<td></td>
</tr>
</tbody>
</table>