***Homework 2. InternerWorking, Pierre FLEITZ***   
  
**1) ARP :**

1. *State of the five ARP caches after the IPv4 unicast datagram has been delivered to host H2 :*

|  |  |  |
| --- | --- | --- |
| H1 | | |
| IP | MAC | Port |
| A | a | North |

|  |  |  |
| --- | --- | --- |
| H3 | | |
| IP | MAC | Port |
| B | b | North |

|  |  |  |
| --- | --- | --- |
| R1 | | |
| IP | MAC | Port |
| E | e | West |
| D | d | East |

|  |  |  |
| --- | --- | --- |
| H2 | | |
| IP | MAC | Port |
| A | a | South |

|  |  |  |
| --- | --- | --- |
| H4 | | |
| IP | ARP | Port |
| E | e | South |

1. *State of the bridges B1’s MAC address table after the IPv4 unicast datagram has been delivered to host H2*

|  |  |
| --- | --- |
| B1 MAC Address table | |
| MAC | Port |
| d | South |
| a | East |

**2) UDP and Fragmentation :**

1. 7400 / 1480 = 5. Then we gonna have 5 fragments.
2. *Each fragment has its own header with most of the fields repeated, but some changed :*

|  |  |  |  |
| --- | --- | --- | --- |
| Fragment | MF | Offset | Length |
| Number 1 | 1 | 0 | 1480 bytes (bytes 0 to 1479) |
| Number 2 | 1 | 185 | 1480 bytes (bytes 1480 to 2959) |
| Number 3 | 1 | 370 | 1480 bytes (bytes 2960 to 4439) |
| Number 4 | 1 | 555 | 1480 bytes (bytes 4440 to 5919) |
| Number 5 | 0 | 740 | 1480 bytes (bytes 5920 to 7399) |

**3) Routing :**

1. *Initial routing state of D :*

|  |  |  |
| --- | --- | --- |
| Destination | Metric | Next-Hop |
| 208.218.2.0/24 | 1 | - |
| 208.218.4.0/24 | 1 | - |

1. *Routing state of D after it has received the initial distance-vector from E :*

|  |  |  |
| --- | --- | --- |
| Destination | Metric | Next-Hop |
| 208.218.2.0/24 | 1 | - |
| 208.218.4.0/24 | 1 | - |
| 208.218.5.0/24 | 2 | 208.218.4.2 |

1. *RIP messages :*

**Reminder :** Type of RIP msg : (see p.294, Figure 11.11 and example Figure 11.13 p296 from TCP/IP book, I will not represent the two ”spaces” between Network and Distance)

|  |  |  |
| --- | --- | --- |
| Command | Version | / |
| Family | | / |
| Network | | |
| Distance | | |

|  |  |
| --- | --- |
| Information excracted | |
| Dest | Cost |
| 208.218.2.0 | 1 |
| 208.218.4.0 | 2 |
| 208.218.5.0 | 2 |
| 208.218.1.0 | 1 |

From A to B, so 208.218.1.1 to 208.218.1.2, interface used East :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Msg1 | 2 | 1 | | / | |
| 2 | | | / | |
| 208.218.2.0/24 | | | | |
| 1 | | | | |
| Msg2 | 2 | 1 | | / | |
| 2 | | | / | |
| 208.218.4.0/24 | | | | |
| 2 | | | | |
| Msg3 | 2 | 1 | | / | |
| 2 | | | / | |
| 208.218.5.0/24 | | | | |
| 2 | | | | |
| Msg4 | 2 | | 1 | | / |
| 2 | | | | / |
| 208.218.1.0/24 | | | | |
| 1 | | | | |

From A to D, so 208.218.2.1 to 208.218.2.2, interface used South :

|  |  |  |  |
| --- | --- | --- | --- |
| Msg1 | 2 | 1 | / |
| 2 | | / |
| 208.218.1.0/24 | | |
| 1 | | |
| Msg2 | 2 | 1 | / |
| 2 | | / |
| 208.218.2.0/24 | | |
| 1 | | |
| Msg3 | 2 | 1 | / |
| 2 | | / |
| 208.218.4.0/24 | | |
| 2 | | |
| Msg4 | 2 | 1 | / |
| 2 | | / |
| 208.218.5.0/24 | | |
| 2 | | |

|  |  |
| --- | --- |
| Information extracted | |
| Dest | Cost |
| 208.218.1.0 | 2 |
| 208.218.2.0 | 1 |
| 208.218.4.0 | 2 |
| 208.218.5.0 | 2 |

From A to E, so 208.218.2.1 to 208.218.2.3, interface used South :

|  |  |  |  |
| --- | --- | --- | --- |
| Msg1 | 2 | 1 | / |
| 2 | | / |
| 208.218.1.0/24 | | |
| 1 | | |
| Msg2 | 2 | 1 | / |
| 2 | | / |
| 208.218.2.0/24 | | |
| 1 | | |
| Msg3 | 2 | 1 | / | The message is prepared with the combinaison of split horizon and poison reverse, in minde. A obtened information about these 2 networks by router E. Here A sends an update to E so it replaces the actual value of the hop count for the 2 networks with 16 (infinity) to prevent any confusion for E |
| 2 | | / |
| 208.218.4.0/24 | | |
| 16 | | |
| Msg4 | 2 | 1 | / |
| 2 | | / |
| 208.218.5.0/24 | | |
| 16 | | |

|  |  |
| --- | --- |
| Information extracted | |
| Dest | Cost |
| 208.218.1.0 | 1 |
| 208.218.2.0 | 1 |
| 208.218.4.0 | 16 |
| 208.218.5.0 | 16 |

1. *Routing states of D and E after they havec received the distance-vector from A :*

|  |  |  |
| --- | --- | --- |
| Routing state of E | | |
| Destination | Metric | Next-Hop |
| 208.218.1.0/24 | 2 | 208.218.2.1 |
| 208.218.2.0/24 | 1 | - |
| 208.218.4.0/24 | 1 | - |
| 208.218.5.0/24 | 1 | - |

**4) ICMP :**

|  |  |  |
| --- | --- | --- |
| Routing state of D | | |
| Destination | Metric | Next-Hop |
| 208.218.1.0/24 | 2 | 208.218.2.1 |
| 208.218.2.0/24 | 1 | - |
| 208.218.4.0/24 | 1 | - |
| 208.218.5.0/24 | 2 | 208.218.4.2 |

1. The source host, here H1, sends ICMP echo request messages (type : 8, code : 0) ; the destination, here H2, will respond with an ICMP echo reply message.
2. *Traceroute from H1 to H3 :*

|  |  |
| --- | --- |
| A to H1 | |
| Type | Time exceeded (type: 11, code: 0) |
| @ Dest | 90.59.1.2 |

|  |  |
| --- | --- |
| H1 to A | |
| Type | Echo Request |
| TTL | 1 |
| @ Dest | 90.59.5.2 |

|  |  |
| --- | --- |
| B to H1 | |
| Type | Time exceeded (type: 11, code: 0) |
| @ Dest | 90.59.1.2 |

|  |  |
| --- | --- |
| H1 to B | |
| Type | Echo Request |
| TTL | 2 |
| @ Dest | 90.59.5.2 |

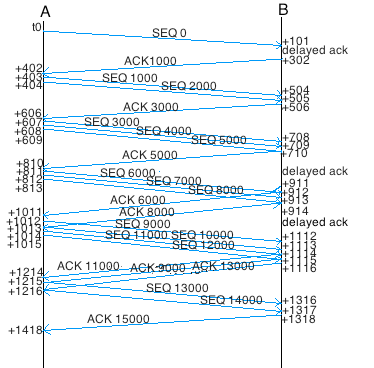
|  |  |
| --- | --- |
| H3 to H1 | |
| Type | Destination unreachable (type: 3, code: 3) |
| @ Dest | 90.59.1.2 |

|  |  |
| --- | --- |
| H1 to C | |
| Type | Echo Request |
| TTL | 3 |
| @ Dest | 90.59.5.2 |

|  |  |
| --- | --- |
| C to H1 | |
| Type | Time exceeded (type: 11, code: 0) |
| @ Dest | 90.59.1.2 |

|  |  |
| --- | --- |
| H1 to H3 | |
| Type | Echo Request |
| TTL | 4 |
| @ Dest | 90.59.5.2 |

***5) TCP :***

1. MSS = Maximum Segment Size. The MSS used by TCP is :   
   MSS = MTU – 20 ( IP header ) - 20 ( TCP header) = 1040 – 20 – 20 = 1000 bytes
2. To calcul the bandwidth delay product we need to do : RTTxBW. So here we have : Bandwidth-delay product = ((100\*2)\*10^3)\*(4\*10^6)=800KB. Therefore the received window is not big enough ! If we want A to be able to fully utilize the channel, The received window should be at least 800KB.
3.   
   Sorry for the quality of the schema, kind of hard to do something clean with a chronogramme like this… Hope you will understand what I tried to do.  
     
     
     
     
     
     
     
     
     
   When we are after connection establishment we have : RTO = max ( 2,5+G, 3 ) = 3s, at T0. SRTT = RTTVAR = 0.

*Values of SRTT, RTTVAR and RTO for the 4 first segement sent by A :*

* Segment 1 : SRTT = 0s, RTTVAR = 0s, RTO = 3s.
* Segment 2 :

We can see in the precedent figure that the ACK for the first segment is received at t0+402. Then we have RTT = 402 ms (delayed ack..). Therefore SRTT = 0,402 s and RTTVAR = 0,402/2 = 0,201 s. (Because this is the first measurement made on this connection !).  
And RTO = SRTT + max(G,4\*RTTVAR) = 0,402 + 0,804 = 1,206.

So Segment 2 : SRTT = 0,402s , RTTVAR = 0,201s, RTO = 1,206s

* Segment 3 : (Same thing) SRTT = 0,402s , RTTVAR = 0,201s, RTO = 1,206s.
* Segment 4 :

We can see in the precedent figure that the ACK for the fourth segment is received at t0+606 and the segment was sent at to+404. Then we have RTT = 202 ms.

SRTT’ = 7/8 \* SRTT + 1/8 \* RTT = 0,377 s

RTTVAR = 0,75\*0,201 + 0,25\*|0,402-0,202| = 0,2008 s.  
And RTO = SRTT + max(G,4\*RTTVAR) = 0,377 + 0,803 = 1,18s.

So Segment 4 : SRTT = 0,377s , RTTVAR = 0,2008s , RTO = 1,18s.  
  
 d) At t0+1418ms.