September 23th, 2016

Exam of Packet switch architectures (2015/16)

Exam of Switch and router architectures (2014/15)

Rules for the exam. It is **forbidden** to use notes, books or calculators. Use only draft paper provided by the professor. When needed, use approximations.
Time available: 70 minutes.

Problem A

Consider an input queued switch of size $N \times M$ supporting both unicast and multicast traffic. Assume that each input is equipped with just 2 queues: one queue for broadcast packets and one queue for multicast (but not broadcast) traffic, including unicast traffic. Consider a scheduling algorithm that allows fanout-splitting and serves broadcast traffic at highest priority. As a reminder, the fanout set of a multicast packet is the set of its destination ports.
Describe in pseudo-code the scheduling algorithm, using the following notation. At each timeslot, let `B[i]` be the size of the queue for broadcast packets at input i. Let `M[i]` be the size of the queue for multicast/unicast packets at input i. Assume that function `destInMCQueue(j,i)` returns `true` iff output `j` belongs to the fanout set of the packet at the head of the multicast queue `M[i]`. Let `X` be the matrix describing the switching configuration chosen in the current timeslot, based on the state of the queues. More precisely, `X[i][j]` is a Boolean variable, which assumes the value `true` iff the crosspoint from input `i` to output `j` is active, i.e. a packet must be sent from input i to output j in the current timeslot.

Problem B

Consider a flow-monitor tool that stores all the active flows using a table implemented with a simple vector array. A fingerprint scheme is adopted to map any flow to an integer in the interval $[0, 1023]$; the fingerprint is designed to identify flows independently from their direction, i.e. the two flows of a bidirectional communication are identified by the same value. Let a flow f be defined in terms of a 4-uple with the notation reported in the following table:

Source IP address	$x_1.x_2.x_3.x_4$
Destination IP address	$y_1.y_2.y_3.y_4$
Layer-4 source port	p_x
Layer-4 destination port	p_y

1. Design a fingerprint scheme $F(f)$ that satisfies all the design constraints above and able to deal also with flows without any layer-4 port.

2. Consider the sequence of operations in the above table. For each operation, show the content of table T and the probability of false positive after the operation. Assume the table initially empty: $T = \{\}$.

Time	Operation	Table T	Prob. false positive
1	insert(1.2.3.4, 5.6.7.8, 1, 2)		
2	insert(2.3.4.5, 6.7.8.9, 3, 4)		
3	insert(2.3.4.5, 9.8.7.6, 5, 2)		
4	insert(3.4.0.3, 5.7.9.5, 2, 1)		
5	delete(1.2.3.4, 5.6.7.8, 1, 2)		
6	delete(2.3.4.5, 9.8.7.6, 5, 2)		
7	delete(3.4.0.3, 5.7.9.5, 2, 1)		
8	delete(2.3.4.5, 6.7.8.9, 3, 4)		

3. In the above sequence of operations, add an operation at some specific time $t \in [0, 10]$ for which a false positive event occurs. Note that the false positive can occur for other values of t. What is the largest time interval during which the false positive would occur?

4. Show the pseudo-code to insert, delete and search an element in a table exploiting fingerprinting.

Problem C

Draw an 8×8 Banyan network, having Baseline layout; identify the nodes and the edges of the network. Connect: 2/3, 3/5, 4/0, 5/2, 6/1. Is it possible? Why? Connect: 2/0, 3/2, 4/4, 5/6, 6/7. Is it possible? Why?

Hints for the solution

Problem A

It is a variant of ex. 107 but now the scheduler allows fanout splitting.

```
// initialize the data structures
for j=1...M      // for each output port
    output_reserved[j]=false
    for i=1...N // for each input port
        X[i][j]=false
// first, try to serve broadcast packets
for i=1...N      // for each input port
    if (B[i]>0)
       // found broadcast packet with all available outputs
       for j=1...N
           X[i][j]=true
           output_reserved[j]=true // (not needed)
       return // ends since switching configuration is maximal
// second, try to serve multicast packets
for i=1...N
    if (M[i]>0) // check if the mc queue is non-empty
       for j=1...M // for each output port
           // check is any output in the fanout set is available
           if (destInMCQueue(j,i)==true AND output_reserved[j]==false)
                 output_reserved[j]=true
                 X[i][j]=true
```

Problem B

1. The fingerprint can be defined as follows:

$$F(f) = (x_1 + x_2 + x_3 + x_4 + y_1 + y_2 + y_3 + y_4 + p_x + p_y) \mod 1024$$

   where $p_x = p_y = 65,536$ whenever layer-4 port is not defined.

2. Thus the evolution of the hash table is the following:

Time	Operation	Table T	Prob. false positive
1	insert(1.2.3.4, 5.6.7.8, 1, 2)	$\{39\}$	$1/1024$
2	insert(2.3.4.5, 6.7.8.9, 3, 4)	$\{39, 51\}$	$2/1024$
3	insert(2.3.4.5, 9.8.7.6, 5, 2)	$\{39, 51, 51\}$	$2/1024$
4	insert(3.4.0.3, 5.7.9.5, 2, 1)	$\{39, 39, 51, 51\}$	$2/1024$
5	delete(1.2.3.4, 5.6.7.8, 1, 2)	$\{39, 51, 51\}$	$2/1024$
6	delete(2.3.4.5, 9.8.7.6, 5, 2)	$\{39, 51\}$	$2/1024$
7	delete(3.4.0.3, 5.7.9.5, 2, 1)	$\{51\}$	$1/1024$
8	delete(2.3.4.5, 6.7.8.9, 3, 4)	$\{\}$	0

3. E.g., Search(2.3.0.9, 10.7.0.1, 10, 9) during any time $t \in (2, 8)$ would lead to a false positive.

4. See the solution of point 5 of ex.114.

Problem C

The exercise is almost identical to ex.32.