February 1st, 2016

Exam of Packet switch architectures (2015/16)

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 a 3×3 input queued switch, with ports running at 8 Gbps. Assume that the internal timeslot corresponds to a 64 bytes packet. The following rates should be guaranteed:

$$R = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 4 \\ 0 & 4 & 2 \end{bmatrix} \quad \text{Gbps}$$

where R_{ij} is the rate from input i to output j.

1. Use the Paul algorithm to find a possible frame sequence F:
   - Show the corresponding Clos network
   - Show the Paul matrix evolution
   - Show the final F
2. Compute the average access delay (in nanoseconds) for each VOQ when adopting F
3. Compute the admissibility conditions for the traffic when adopting F

Problem B

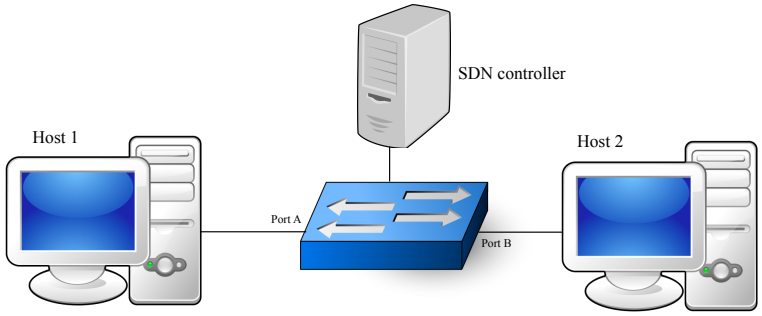
Consider an Internet traffic monitoring system based on a b-bit fingerprint $F(x)$ for flow x, adopting a bit string hash for the storage. The flow is identified by the couple $x = (IP_{\text{source}}, IP_{\text{destination}})$. Answer to the following question:

1. What is the total amount of memory (in kbytes) required for storage in the case $b = 32$ after $1,000$ flows have been inserted?
2. Describe a possible fingerprint function when $b = 32$
3. Prove the formula to compute the probability of false positive in function of b and of the number k of inserted flows
4. What would be the minimum value of b to guarantee a probability of false positive less than ϵ?
5. Is there any case in which the bit string array shows a null probability of false positive?
6. Show the pseudocode to insert a generic flow x into the monitoring system, after defining properly all the involved data structures.
7. Show the pseudocode to search a generic flow x
8. Show the pseudocode to delete a generic flow x
9. Does the considered monitoring system supports correctly the deletion operation? Why?

Problem C

Consider a SDN network based on Openflow as protocol adopted in the southbound interface.

1. Describe all the three possible kinds of Openflow messages, highlighting the purpose of each of them.
2. What is a flow table in Openflow?
3. Describe the internal architecture of an Openflow switch, highlighting all the main components and the kinds of adopted memories.
4. Consider the following scenario in which an Openflow switch is used to connect two hosts. Assume that Host 1 sends two Ethernet packets to Host 2, denoted as p_1 and p_2. Assume that the controller knows the network topology and adopts a reactive routing application, i.e. the flow tables are populated in realtime as soon as the traffic is received.



(a) Show the sequence of the packets that are exchanged in the data plane (between the hosts and the switch) and in the control plane (between the switch and the controller) by filling the following table:

Sequence number	Link	Packet

(b) Show the final flow table in the switch

Hints for the solution

Problem A

The Clos network is built with 3 I-stage modules 7×7, 7 II-stage modules 3×3 and 3 III-stage modules 7×7. We must connect R'_{ij} circuits from I-stage module i to III-stage module j:

$$R' = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 4 \\ 0 & 4 & 2 \end{bmatrix}$$

We do not report the Paul algorithm evolution. One possibility would be a frame with 7 matchings: $\mathcal{F} = [M_i]_{i=1}^{7}$, being:

$$M_1 = M_2 = M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad M_4 = M_5 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \qquad M_6 = M_7 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

The access delays are the following:

VOQ	Serving matchings	Average access delay (in timeslots)
11	M_1, M_2, M_3	(1+1+1+5+4+3+2)/7=17/7
12,23,31	M_4, M_5	(4+3+2+1+1+6+5)/7=22/7
13,21,32	M_6, M_7	(6+5+4+3+2+1+1)/7=22/7
21	M_1, M_2, M_3	(1+1+1+5+4+3+2)/7=17/7
31	M_1, M_2, M_3	(1+1+1+5+4+3+2)/7=17/7

The absolute delay can be computed by multiplying the last column for the duration T of each timeslot, being

$$T = \frac{64 \times 8 \text{ bit}}{8 \text{ Gbps}} = 64 \text{ ns}$$

When adopting the above frame $\mathcal{F}$, in terms of normalized arrival rates, the admissibility conditions will be

$$\Lambda' \leq R'/7$$

In terms of absolute arrival rates, the admissibility conditions will be:

$$\Lambda = R'/7 \times 8 \text{ Gbps} = \begin{bmatrix} 1.14 & 2.28 & 1.14 \\ 2.28 & 0 & 4.56 \\ 0 & 4.56 & 2.28 \end{bmatrix} \text{ Gbps}$$

Note that, by construction $R \leq \Lambda$.

Problem B

1. The bit string hash requires 2^b bits, independently from the stored elements. In the considered case, around 4 Gbit ($\approx$ 500 Mbytes) are required.
2. A possible hash function can be obtained by the EXOR between the source and destination IP address: $F(x) = IP_{\text{source}} \oplus IP_{\text{destination}}$.
3. The probability that a specific bit in the bit array equals to one when adding a flow is $p = 1/2^b$. Consider now a flow y which has not been inserted before. Now the probability that $F(y) \neq F(x)$ for any x already inserted is $(1-p)^k$. Thus, the probability of false positive is

$$\Pr(\text{false positive}) = 1 - \left(1 - \frac{1}{2^b}\right)^k \approx 1 - e^{-\frac{k}{2^b}}$$

4. By setting $\Pr(\text{false positive}) < \epsilon$, after some simple steps, we get

$$b > \log_2 \frac{k}{-\log_2(1-\epsilon)}$$

5. Only when no element has been inserted in the bit array, the probability of false positive is null.
6. Let BSA be the string array with 2^b bits, and $BSA[i] \in \{0, 1\}$ be the ith bit in the data structure. Let $F(x)$ be the fingerprint of flow x.

function INSERT(x)
$BSA[F(x)] = 1$
end function

7. **function** SEARCH(x)
if $BSA[F(x)] = 1$ **then**
return FOUND
else
return NOT_FOUND
end if
end function

8. **function** DELETE(x)
$BSA[F(x)] = 1$
end function

9. If the deletion operation occurs, then the bit string array can introduce false negatives and thus the deletion *must not be supported*.

Problem C

For the considered scenario, a possible sequence of packets is the following:

Sequence	Link	Packet
1	Host 1 $\rightarrow$ Switch	p_1
2	Switch $\rightarrow$ Controller	Packet-in(p_1, from port A)
3	Controller $\rightarrow$ Switch	Packet-out(p_1, to port B)
4	Controller $\rightarrow$ Switch	Flow-mod(send to port B all packets destined to MAC(Host 2))
5	Switch $\rightarrow$ Host 2	p_1
6	Host 1 $\rightarrow$ Switch	p_2
7	Switch $\rightarrow$ Host 2	p_2

The final flow table will be:

MAC-source	MAC-dest	IP-source	IP-dest	$\cdots$	Action
*	MAC(Host 2)	*	*	*	Send to port B