March 3rd, 2017

Exam of Switching technologies for data centers (2016/17)

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. The answers must be provided in correct English. Any notation must be defined.
Time available: 70 minutes.

Problem A

Consider a $N \times (N/2)$ rearrangeable switch, acting as concentrator, built with 2×2 basic modules, exploiting recursive factorization. Assume $N = 2^h$, for some positive integer $h \geq 2$.

1. Consider a first architecture with recursive factorization.
   (a) Compute the total number of basic modules, showing all the steps to solve the corresponding recursive equation.
   (b) Draw the complete network topology for a 8×4 switch and compute the total number of basic modules.
   (c) Can Paul and looping algorithms be used to configure the network? Why?
2. Consider a second architecture built by designing a symmetric $N \times N$ Benes network and then removing the useless modules starting from at the last stage.
   (a) Compute the total number of basic modules (starting from the formula for Benes networks).
   (b) Draw the complete network topology for a 8×4 switch and compute the total number of modules.
   (c) Can Paul and looping algorithms be used to configure the network? Why?
3. What is the best among the two architectures?

Problem B

Consider a traditional hash table with H buckets to store $<$key,value$>$ elements.

1. Define the concept of "hash function" and describe its properties.
2. Explain the two main relevant results regarding random policies for bins-and-balls models, describing all the involved assumptions.
3. Describe two different ways to implement hash tables that exploit the above two results.
4. For each of the two implementations:
   (a) Describe in pseudocode the insertion of an element; for simplicity, assume that the key does not appear already in the hash.
   (b) Describe in pseudocode the lookup of an element.
   (c) Evaluate the expected lookup time.
   (d) Evaluate the worst case lookup time.
   (e) Show an example of insertion of 12 elements when $H = 4$.

Problem C

Consider a 4×4 input queued switch with virtual output queues (VOQs), with each port running at 10 Gbps. Assume that the internal timeslot corresponds to a 64 bytes packet. The following rate matrix must be guaranteed:

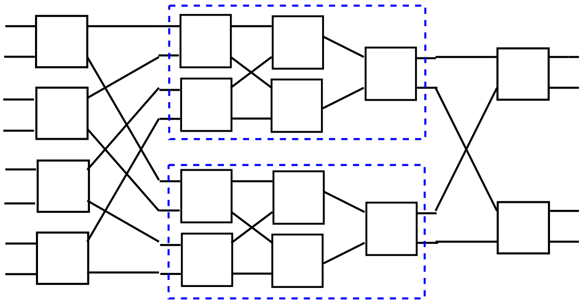
$$\hat{R} = \begin{bmatrix} 1 & 2 & 1 & 4 \\ 2 & 4 & 1 & 1 \\ 4 & 1 & 2 & 1 \\ 1 & 1 & 4 & 2 \end{bmatrix} \text{ Gbps}$$

1. Find the frame sequence F, using Paul algorithm.
2. What are the admissibility conditions for the traffic to achieve the maximum throughput?
3. What is the minimum worst-case access delay and the corresponding VOQs?
4. What is the maximum worst-case access delay and the corresponding VOQs?

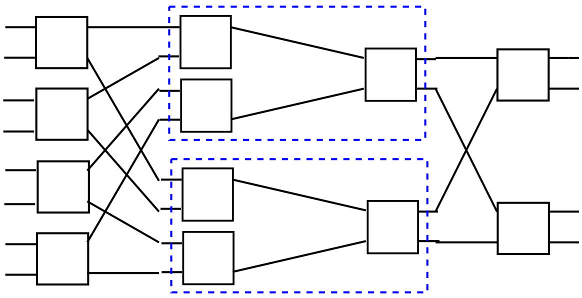
Hints for the solution

Problem A

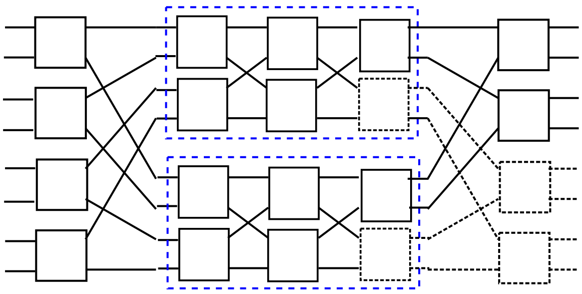
The problem is almost identical to ex. 33. In addition:
1.(b) Using recursive constructions, the 8×4 switch has complexity: $C(8,4) = 6C_2 + 2C(4,2) = (6+2\times5)C_2 = 16C_2$.
The corresponding network is:



Note that, using a simpler construction, $C(4,2) = 3C_2$ and thus $C(8,4) = 6C_2 + 2C(4,2) = (6 + 2 \times 3)C_2 = 12C_2$.
The corresponding network is:



2.(c) Using a 8×8 Benes network and removing 4 useless modules (shown as dotted), we obtain the following architecture, whose total complexity is $16C_2$. Thus, the complexity is equivalent to the first architecture.



Problem B

See the class notes. In addition:
Let $h(k)$ and $g(k)$ be two hash functions that map a key k into the interval $[1, H]$. Let $T = [T_i]_{i=1}^{H}$ be a table with H buckets, in which T_i is bucket i.

- For traditional hash tables:

```
function INSERT(k, v)                         ▷ k is the key and v is the value
    Add (k, v) in T_h(k)                      ▷ Add the element
end function
```

```
function LOOKUP(k)                            ▷ k is the key
    for each (k', v') ∈ T_h(k) do             ▷ Check bucket h(k)
        if k = k' then                        ▷ Key k found
            return v'
        end if
    end for
    return Not-found
end function
```

- For multiple-choice hash tables:

```
function INSERT(k, v)                         ▷ k is the key and v is the value
    if |T_h(k)| ≤ |T_g(k)| then               ▷ Find the smallest bucket
        Add (k, v) in T_h(k)                  ▷ Add the element
    else
        Add (k, v) in T_g(k)                  ▷ Add the element
    end if
end function
```

```
function LOOKUP(k)                            ▷ k is the key
    for each (k', v') ∈ T_h(k) do             ▷ Check first bucket h(k)
        if k = k' then                        ▷ Key k found
            return v'
        end if
    end for
    for each (k', v') ∈ T_g(k) do             ▷ Check second bucket g(k)
        if k = k' then                        ▷ Key k found
            return v'
        end if
    end for
    return Not-found
end function
```

Problem C

1. Using Paul algorithm, a possible frame of 8 timeslots $F = [M_i]_{i=1}^{8}$ is the following:

$$M_1 = M_2 = M_3 = M_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad M_5 = M_6 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad M_7 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad M_8 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

2. The traffic matrix Λ to be admissile and achieve the maximum throughput must satisfy:

$$\Lambda \leq \hat{R}/8 \times 10 = \begin{bmatrix} 1.25 & 2.50 & 1.25 & 5.00 \\ 2.50 & 5.00 & 1.25 & 1.25 \\ 5.00 & 1.25 & 2.50 & 1.25 \\ 1.25 & 1.25 & 5.00 & 2.50 \end{bmatrix} \text{ Gbps}$$

3. According to F, all the VOQs corresponding to M_1 experience a worst-case access delay equal to 5 timeslots, which is the minimum among all the VOQs.

4. According to F, all the VOQs corresponding to M_7 and M_8 experience a worst-case access delay equal to 8 timeslots, which is the maximum among all the VOQs.