Spring 99

Assignment #2

COMPUTER NETWORKS

SAHAND KHOSHDEL – ST ID: 810196607

In the Name of God

Q1)

a) [N = 5; p = 0.1]

Throughput = E{packets transmitted successfully per slot}

= E{packets sent on the network per slot} × Pr{Successful transmission}

E{packets sent on the network per slot} = G

X : packets sent on the network ⇒ X ~ Bin (N,p)

$P_x^{(k)} = \binom{N}{k} p^k (1-p)^{N-k}$ ⇒ **E{X}** = N.p = 5 × 0.1 = **0.5**

Pr{Successful transmission}

= Pr{One node putting its packet on the network at the beginning of a slot and others not}

= $p(1-p)^4$ = (0.1)(0.9)^4 = **0.06561**

Throughput = 0.5 × 0.06561 = **0.0328**

b)

We know that the maximum value of the throughput (S) occurs when G (expected value of packets sent in a slot) becomes 1 but this was when the binomial distribution of the packets been sent tends to a poison distribution **that requires**

conditions that are not satisfied in this problem (N should be sufficiently large , p should be very small and Np should be a finite number) :

In This problem N=5, p=0.1; so we will deal with it as a general optimization problem :

Throughput = E{X}. Pr{Successful transmission} = Np . $p(1-p)^4$ = $\boldsymbol{Np^2(1-p)^4}$

For maximizing the Throughput with holding the node numbers(N) constant, we take the derivative of the equation above with respect to "p":

$$\frac{\partial(throughput)}{\partial p} = N[\,(2p)(1-p)^4 + p^2(4(p-1)^3) = 0$$

⇒ p = 0 , 1 , $\frac{1}{3}$ ⇒ $\boxed{p = \frac{1}{3}}$

Q2)

Throughput = Efficiency × $\boldsymbol{R_b}$

R_b: (bitrate of packets sent on the network(not necessarily successful)

$\boldsymbol{R_b} = 200\,\frac{packet}{sec} \times \frac{100\ bytes}{packet}$ = **20 KBps**

Efficiency = $\frac{S}{G}$ = ?

S = E{packets transmitted successfully} = Ge^{-G} (for sufficiently large population)

G = E{packets sent on the network} = $200\,\frac{packet}{sec} \times \frac{2.5\ msec}{slot} = 0.45\,\frac{packet}{slot}$

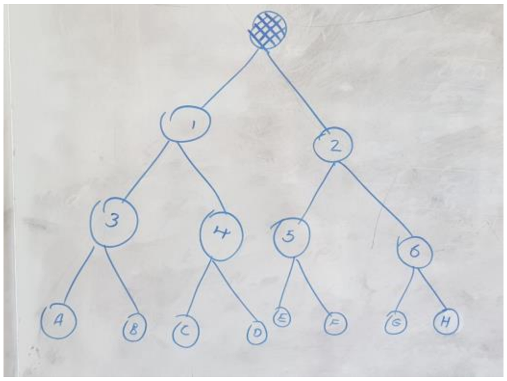
Efficiency = $\frac{S}{G} = e^{-G} = e^{-0.45} \approx$ **0.638**

⇒ **Throughput = 12.753 KBps**

Q3)

a) X: number of nodes that have packet to transmit

E{X} = ¼ (8) = 2



The whole procedure may consist more than 2 transmissions but as we are analyzing an average amount, we can simplify the procedure and assume there are always 2 transmissions needed to complete the procedure.

Y : number of time slots each 2 packets need to be transmitted

3 modes can possibly occur :

Mode #1:

Both packets are for nodes which have the same parent (A&B , C&D ,etc)

In this mode both we will have 3 collisions(main node,both reach 1 ,both reach 3)

- There won't be any problem for when we have C&D to transmit as searching for the status of the node 3 which results 'idle' prevents us searching node 4's status as we now It'll be 'collision':

So we need 5 slots to transmit both packets (Y = 5)

Mode #2:

Packets have the same grandparent (ex: A,C)

In this mode both we will have 2 collisions(main node, both reach 1)

So we need 4 slots to transmit both packets

Mode #3:

Packets have the same grand-grandparent (ex: A,E)

In this mode both we will have 1 collision(Only on the main node)

So we need 3 slots to transmit both packets

- All the possible modes are equally probable

E{Y} = $(\frac{1}{3} \times 3) + (\frac{1}{3} \times 4) + (\frac{1}{3} \times 5)$ = **4** ⇒

Expected number of time slots to finish transmission : 4

b)

By considering the previous assumption which there are 2 nodes attempting to transmit their packets:

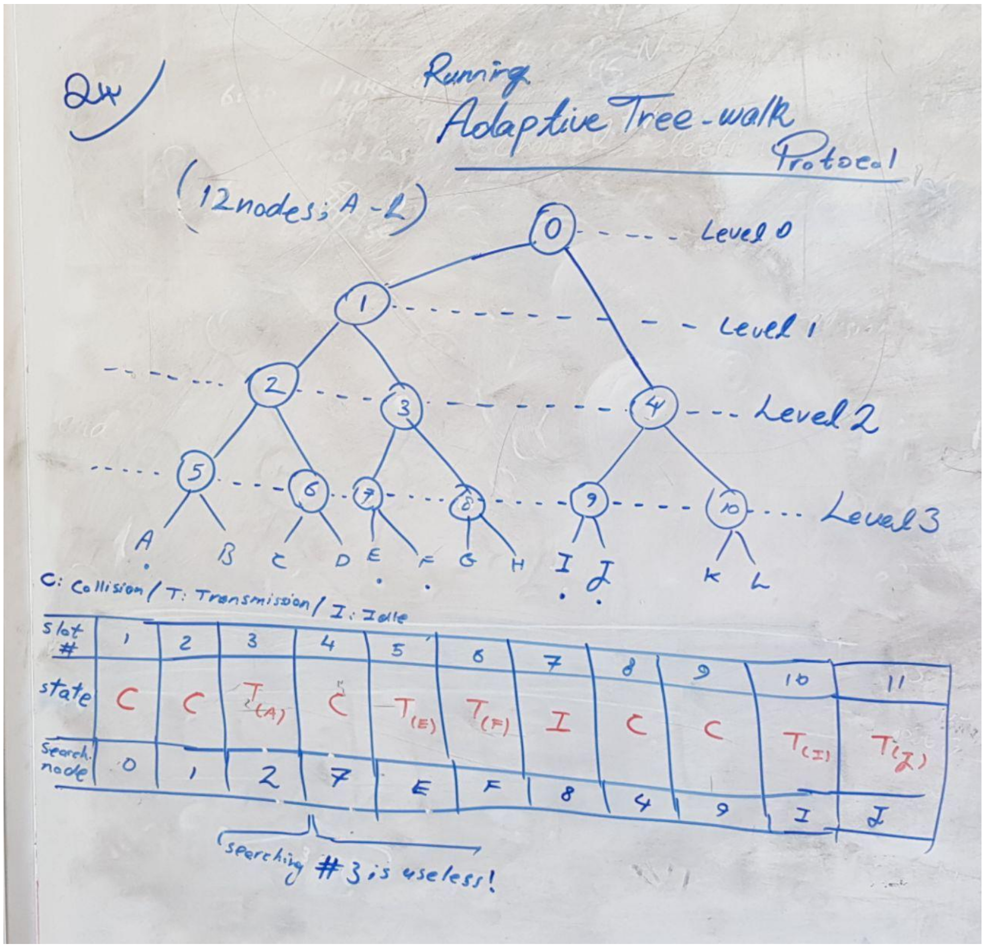
Z: E{slots per packet(node)} = It's equally likely that the node will be the first or second node to transmit in each mode so :

Z: $(\frac{1}{6} \times 2) + (\frac{1}{6} \times 3) + (\frac{1}{6} \times 3) + (\frac{1}{6} \times 4) + (\frac{1}{6} \times 4) + (\frac{1}{6} \times 5) = \frac{1}{6}(21)$ = **3.5**

Q4)

Running the adaptive tree-walk protocol:

- We can design the binary tree in the most efficient way in order to reduce the number of time slots needed to 9 ; but as we don't know which nodes are transmitiing at the beginning of the procedure we should design it in the most general way for supporting any scenario



- As we don't know whether G or H have packets that may have caused collision on time slot #3 (on node #1as sell as other nodes) we should check them (result: they are idle)
- **Total time slots** needed for transmission with a standard binary tree is **11**

Q5)

a)

X: number of nodes that have a packet to transmit in each procedure

- X~Bin(100,0.2) ⟹E{X} = Np = 20

  We can assume 20 nodes want to transmit packets in each procedure:

- 100 reservation bits will always be a part of transmission but the length od data packets depends on the amount of nodes which have packets to transmit.

Efficiency = $\frac{data\ packet\ length}{total\ \ lemgth} = \frac{20\times5}{(20\times5)+100} = 50\ \%$

b)

Considering 20 nodes waiting for transmission as the average:

N_x = Node number x, waiting for transmission

Waiting time (in bits) (for x transmissions) = 5x+100

Waiting time (in bits) (for a single transmission of node #x) = $\boldsymbol{\frac{5x+100}{x} = 5 + \frac{100}{x}}$

Average bits for a single transmission $= \frac{\sum_{x=1}^{20}(5+100/x)}{20} = \frac{100+100\sum_{x=1}^{20}\frac{1}{x}}{20} = \ = 5 + 5(\sum_{x=1}^{20}\frac{1}{x}) =$ **22.98(bits)**

Average waiting time for a single transmission: (Assuming all bits are transferred with rate R_b:

$$\boldsymbol{t_{avg(delay)} = \frac{22.98}{R_b}}$$

Q6)

a)

Bit map protocol:

Packet transmission slots = 20 bytes

Efficiency = $\frac{data\ length}{total\ \ lemgth}$ = $\frac{20\times1}{512+20\times1}$ = **3.76%**

Bit-map protocols efficiency decreases by decreasing number of nodes that are ready for transmission (non-efficient when the network has low traffic)

Adaptive Tree-walk Protocol:

In a binary tree with 512 ending nodes there will be 9 levels (generations) from the root to the ending nodes so before the packet can get transmitted 9 collisions will occur.

The packet of the single node will be sent on the 10'th time slot.

Both time slots where collisions occur and time slot where data is transmitted will last for T_p

(T_p : Packet time slot duration)

Efficiency = $\frac{Data\ transmission\ time}{Total\ time}$ $=$ $\frac{\#Data\ Transmission\ time\ slots}{\#Total\ time\ slots}$ = $\frac{1}{9+1}$ = **10%**

- **Despite the large amount of nodes, the adaptive Tree-walk protocol seems to perform better in a low-traffic network**

b) Bit-map protocol:

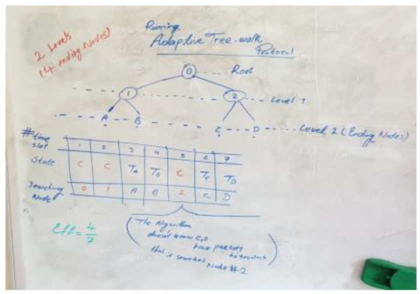
Efficiency = $\frac{data\ length}{total\ \ lemgth}$ = $\frac{20\times512}{512+20\times512}$ =$\frac{20}{21}$ = **95.24%**

The efficiency increased due to the increase in the data length. So Bit-map protocol has a great efficiency for high-traffic networks and this efficiency increases in larger networks, Although it is in a trade-off with a high delay time that's needed for data transmission.

Adaptive Tree-walk Protocol:

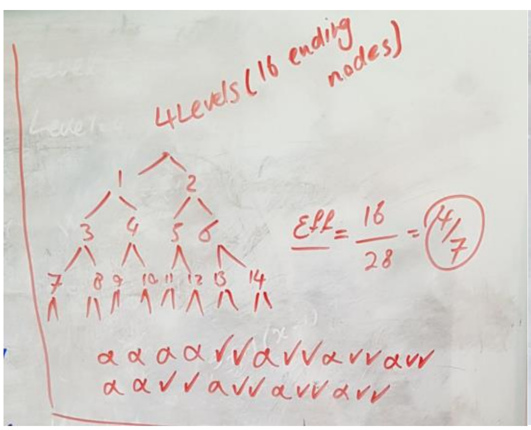
In order to simplify the problem, first we consider a smaller binary tree with all ending nodes having packets to transmit.

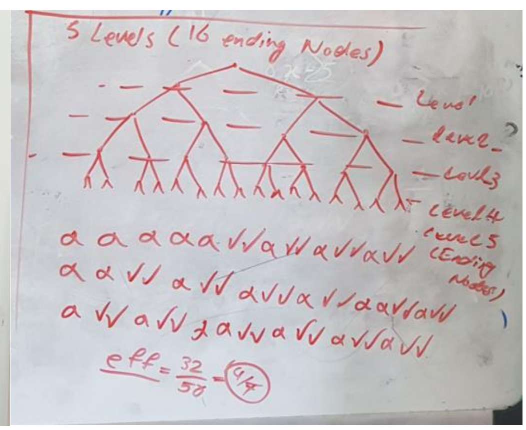
2-level binary tree with 4 ending Nodes:(Efficiency = 4/7)



4-level binary tree with 16 ending Nodes:

5-level binary tree with 32 ending Nodes:





Surprisingly we see the **Efficiency** doesn't change by increasing the Levels if **"All the Ending nodes want to transmit Data"** and remains at the value: $\frac{4}{7}$ = **57.14%**

- So when the network has a high traffic **Bit-map protocol** will be **more Efficient**

Q7)

FDMA: M frequency channels, N nodes: (M<N)

Channel selection ⇒ Uniform ⇒ Pr (Channel #k | transmission) = $\frac{1}{M}$

Pr{collision} = Pr{All channels are currently in use & we have allocation requests }

- We should notice that collisions only ocurs when two ore more nodes allocating a common channel interfere with each other (use the same frequency band) that causes the node's signals becoming non-separable.
- But collision doesn't affect those nodes who are using other frequency bands

So:

Pr{k nodes transmitting successfully } =

Pr{k channels are in use and others (N-k) are idle} + Pr{k channels are in use and (M-k) channels are invalid due to collision} = p1 + p2

⇒ **p1** = $\binom{N}{k}\boldsymbol{p^{k}(1-p)^{N-k}}$ / p2 = ?

(Explaining p2 : k channels are allocated to k nodes and some (= k') of the other nodes are causing collision on other channels in all possible ways (At least 2(M-k) nodes are required to cause collision among all (M-k) remaining channels)

- We assume the Nodes population is very larger than the number of channels (frequency slots) as they are in a cellular mobile network for example)

So:

- N>2M; by using this assumption all the channels can experience collision at th same time so the probability of k nodes transmitting data successfully while others experience collision (p2) will be a non-zero value:

  N>2M $\Rightarrow$ **N>2(M-k)** (so collision may be possible in the remaining M-k channels)

$\Rightarrow$ p2 = Pr{k nodes using k channels exclusively} . Pr{other nodes being distributed in a way that each of the (M-k) remaining channels host at least 2 nodes} = p3.p4

$\Rightarrow$ **p3 =** $\boldsymbol{\binom{N}{k} p^k}$

p4 $=\sum_{k'} \Pr\{\text{k' nodes being idle}\} \times \Pr\{\text{N} - \text{k} - \text{k' causing collision on M} - \text{k remaining channels}\}$

p4 = $\sum_{k'}$ p5 $\times$ p6

p5 = $\boldsymbol{\binom{N-k}{k'} (1-p)^{k'}}$

(p6 =?) $\Rightarrow$ $(c_1 + c_2 + \cdots + c_{(M-k)})$ = N-k-k' (M-k channels allocated to remaining nodes)

$\forall c_i \geq 2 \Rightarrow (c'_1 + c'_2 + \cdots + c'_{(M-k)})$ = N-k-k'-2(M-k) ; $\forall c'_i \geq 0$

The number of possible answers for this "Diophantine equation" equals: $\binom{N-k-k'-2(M-k)+(M-k)}{(M-k)}$

$\Rightarrow$ **p6 =** $\boldsymbol{(p)^{N-k-k'} . \binom{N-k-k'-2(M-k)+(M-k)}{(M-k)} = (p)^{N-k-k'} . \binom{N-M-k'}{(M-k)}}$

p4 = $\sum_{k'} \binom{N-k}{k'} (1-p)^{k'} \times (p)^{N-k-k'} \binom{N-M-k'}{(M-k)}$

$\Rightarrow$ **p4 =** $\boldsymbol{\sum_{k'=0}^{N-k-2(M-k)} [\binom{N-k}{k'} (1-p)^{k'} \times (p)^{N-k-k'} \binom{N-M-k'}{(M-k)}]}$

(The number of idle nodes can vary between 0 and $[N-k-2(M-k)]$ = N – 2M +k)

p2 = p3.p4

$\Rightarrow$ **p2 =** $\binom{N}{k} p^k \sum_{k'=0}^{N-2M+k} [\binom{N-k}{k'} (1-p)^{k'} \times (p)^{N-k-k'} \binom{N-M-k'}{(M-k)}]$

Pr(exactly k Nodes transmitting successfully on k Channels) = p1 + p2

$= \binom{N}{k} p^k (1-p)^{N-k} + \binom{N}{k} p^k \sum_{k'=0}^{N-2M+k} [\binom{N-k}{k'} (1-p)^{k'} \times (p)^{N-k-k'} \binom{N-M-k'}{(M-k)}]$

a) Throughput Calculation:

X: number of nodes transmitting successfully

Throughput = E{X} = $\sum_k (P_X^{(k)}.\, X)$ =

$\sum_{k=0}^{M} k\,.\,[\binom{N}{k} p^k (1-p)^{N-k} + \binom{N}{k} p^k \sum_{k'=0}^{N-2M+k} [\binom{N-k}{k'} (1-p)^{k'} \times (p)^{N-k-k'} \binom{N-M-k'}{(M-k)}]]$

b) For calculating the value of **p** which **maximizes Throughput:**

$\frac{\partial(throughput)}{\partial p} = 0 \Rightarrow$ **Since Summation is a linear operator :**

$$\sum_{k=0}^{M} k\,.\binom{N}{k} \frac{\partial}{\partial p} p^k (1-p)^{N-k} + \sum_{k=0}^{M} k\,.\sum_{k'=0}^{N-2M+k} \binom{N}{k} \binom{N-M-k'}{(M-k)} \frac{\partial}{\partial p} p^{N-k'} (1-p)^{k'} = 0$$

$$\frac{\partial}{\partial p} p^k (1-p)^{N-k} = (k)\, p^{k-1} (1-p)^{N-k} - (N-k)\, p^k (1-p)^{N-k-1}$$

$$\frac{\partial}{\partial p} p^{N-k'} (1-p)^{k'} = (N-k')\, p^{N-k'-1} (1-p)^{k'} - k'\, p^{N-k'} (1-p)^{k'-1}$$

$\Rightarrow$ Optimum p

c) Max Throughput = Throughput(p = p_{opt})

d) If N approaches infinity the first term will approach zero as (1-p) is a value between 0 and 1.