Midterm Solution

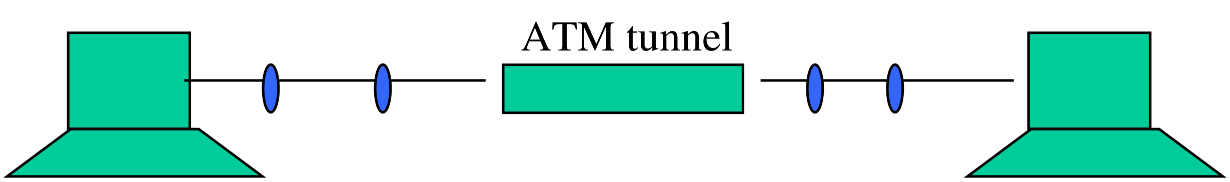
1. Short Answer Questions

(a) Switched Ethernet:

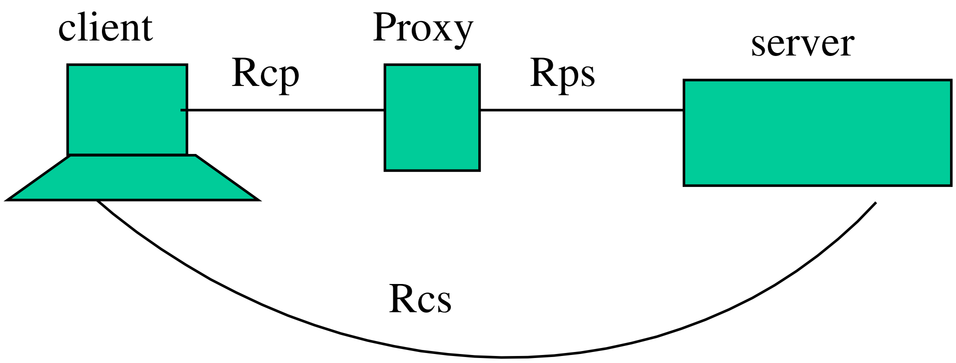
Collision possible if a station transmits at the same time switch sends it a packet.

(b)Tunneling & fragmentation

A tunnel is a logical link & transparent to the rest of the network.
Note: If this were simply a fragmentation issue without tunneling, then one could argue both ways(reassemble @destination or intermediate routers)



(c)Web Proxy Caching:



hit response time = Rcp

miss response time = Rcp + Rps

response time without proxy = Rcs

Typically,Rcp <Rcs

Rcp+Rps > Rcs

Due to cookies, hit ratio h is small, so the average response time = h.Rcp +(1-h)(Rcp +Rcs)

With a small h, proxy is beneficial, if

h*Rcp+(1-h)(Rcp+Rcs)< Rcs

not beneficial, if

h*Rcp +(1-h)(Rcp+Rcs)> Rcs

(d)Intra-domain Routing v/s inter-domain
-Scalability: hierarchical routing scales better
-Policy: Allows policy-based routing

2. CSMA/CD(K.R.CH5. Prob17)
(a)Channel alternates between productive & non-productive states:
(1) find # of slots in unproductive state(x)
Let Y be # of slots until a success, so Y is a Geo R.V.
Probability of a success in m^{th} slot:
$P(Y=m)=(1-ß)^{m-1}ß$
where ß = probability of success
$ß=N*P*(1-P)^{N-1}$
(2)find the # of wasted slots = $X=E(Y)-1=(1- ß) / ß$
(E(Y)=1/ ß for Geo R.V)

$$E = efficiency = \frac{n}{n+x} = \frac{n}{n + \frac{1-(1-1/N)^{N-1}}{(1-1/N)^{N-1}}}$$

(b) maximize E: dE/dP=0

more simply:

maximize E=minimize x=maximize ß:

d ß/dP=0

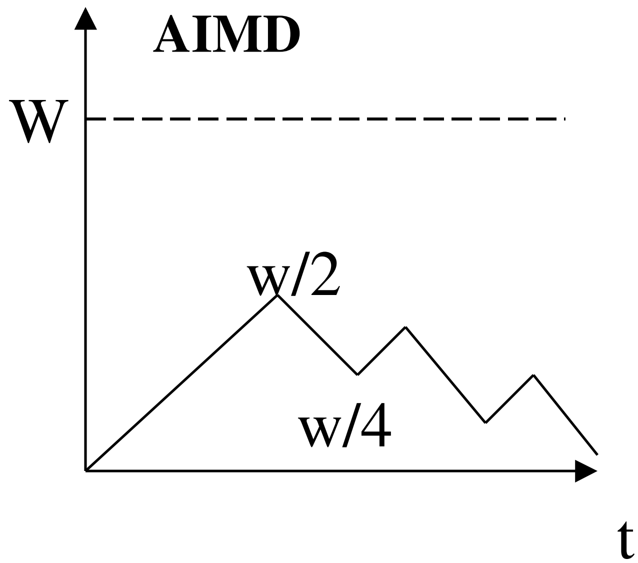
so P=1/N

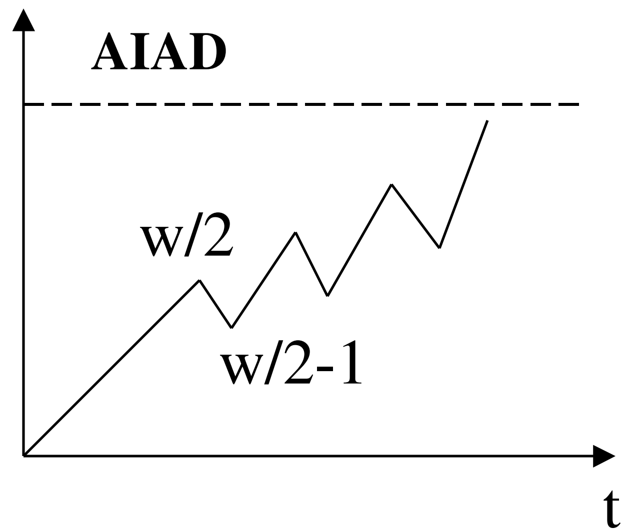
c)substitute p=1/N

$$E = \frac{n}{n + \frac{1-(1-1/N)^{N-1}}{(1-1/N)^{N-1}}}$$

$$\lim_{n\to\infty} E = \frac{n}{n + \frac{1-\frac{1}{e}}{\frac{1}{e}}} = \frac{n}{n+e-1}$$

3.Congestion Control





initially:

1

inc by 1

w/2

reduced by 1/2

w/4

w/4+w/8

initially:

1

inc by 1

w/2

reducedby1

w/2-1

w/2-1+w/8

1 ⟵ steady state ⟶ w-2

This leads to unfair share of the channel for 2 TCP sessions

4. Reliable Multicast

Sender does not know the # or identity of the receivers, so we can't use ACKs(won't know how many ACKs to receive). So a NACK-based protocol design is suitable.
Assumption: each receiver receives some but not all packets.

Receiver:

- •if receive an in-order packet with no corruption, hand to application all packets so far
- •if receive out-of-order packet, buffer & send NACKs for missing packet
- •if corrupted/already received packet, discard

Sender:

- •Assume infinite buffer size
- •Just keep sending packets & buffer them
- •On a NACK, retransmit packet

Essentially, similar to sample exam problem #4
&K.R. Ch3 #11(with minor modification)

(b) State of router:

•buffer packets & answer NACKs

•keep state of NACKs that have been sent to source & do not send a NACK from another receiver if one has been sent already