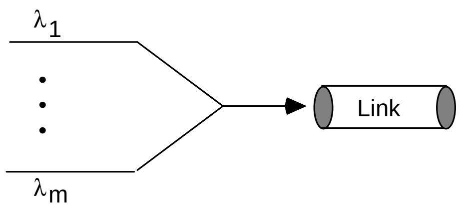
Lecture 12
CS/EE 5516 - Spring 1994

Ch 4. Multiaccess Communication

In Ch 2,3 we compared TDM, SM:



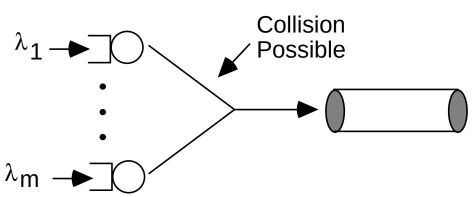
Q: When is TDM vs. SM preferable?

A: •λ_i small => SM (1 transmitter can use 100% of bandwidth)
want small delay => SM (wait m/2 on avg. for TDM, wait 0 for SM)
want large => TDM (fair access)

Q: Can SM be implemented in a LAN or satellite net?

A: [LAN is wireless net, coax/fiber cable, U/S TP.]

NO: No central controller!
Best you can do is sync stations to use slots.



<u>BG 4.2 Slotted Aloha</u>

-Theory Underlying Ethernet

<u>Model Assumptions</u>

m = # nodes

1. Equal Length Packets (Discrete Time:
   Sent on Slot Boundaries No xmit starts at 1.4)
   Xmit is one slot in duration

Implications:
1. Permits discrete time model
2. Prohibits carrier sensing (ethernet), which takes <1 slot time

2. Poisson Arrivals: Aggregate Rate λ
   Per node rate λ/m

3. Collision occurs (whenever my daughter interrupts my wife)
   or
   Perfect Reception (no info on #colliding xmitters, etc)

   1. Ignores noise errors
   2. Prohibits receiver that can capture 1 xmit in face of collisions
   3. Prohibits cellular radio; <u>all</u> nodes "see" collision

4. 0,1,e Immediate Feedback
   At end of slot <u>every</u> node learns if
   0 pkts succeeded(idle)
   1 pkt succeeded (idle)
   e: error (collision)

   => Time delayed feedback (satellites) complicate analysis, but no fundamental problem

5. Retransmission of collisions in later timeslot until succeeds (node called <u>backlogged)</u>

6A. No buffering.
Backlogged node ignores arriving traffic
=> delay w/GA is lowerbound

6B. • nodes:
There are • nodes, not m. Each arrival has its own node.
=> Delay w/ 6B is upper bound.

<u>BG 4.2.2 Slotted Aloha</u>

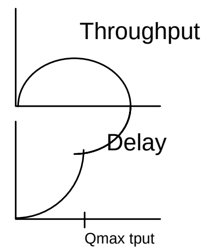
University of Hawaii

Q: What happens on collision? A:

- xmit on next slot?	No - Guarantees another collision
- wait random delay?	Yes, but longer delays possible

Q: Use no buffering assumption (6a). For any λ:
[what will mean delay be?
[what will max throughput be? Near 0? Near 1? Near 1/2?
(Given λ and q_r = prob retrans in current slot)

Q: How would you find answer?
Littles law?
M/M/1 queue? M/M/m ?
Markov chain?



A: Safe answer (w/ poisson arrivals): Markov chain.

Q: What is 1st step to creating chain?

A: Choose system <u> state</u> (ideally, 1 integer)

Q: What should state be?

A: # backlogs (n):
(Class: Why not #customers in system?
Why not #backlogs + #new arrivals at slot start?)

What happens if no backlog, 1 arrival at slot? => "1"
(See table on next page)
Do we need 2D? (#backlog + #arrivals?)

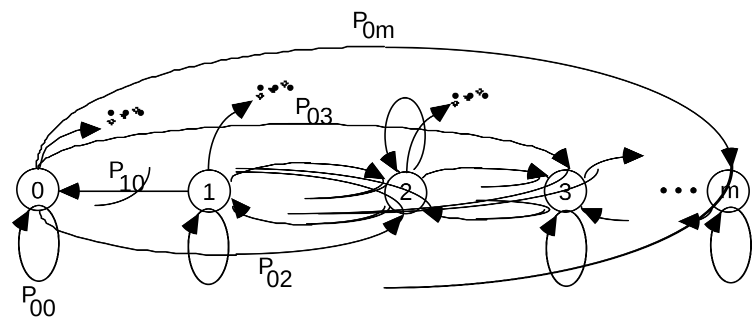
We can use 1 dimension: # in system = # backlogged = n

Consider transition from n to n + i. (one transition = one slot)
Q: What are legal values of i?

A: -1 • i • m-n for n > 0
2 • i • m for n = 0

List exhaustive cases based on i:

i	#arrivals	#backlogged that retransmit	result
-1	0	1	1
0	0	0	0
0	0	2,3,...,n	e
0	1	0	1
1	1	1,2,...,n	e
2,3,...,m-n	i	anything	e



$Q_a(i,n)$ = Prob[i unbacklogged nodes xmit in a slot]

$= \binom{m-n}{i} (1-q_a)^{m-n-1} * q_a{}^i$

$Q_r(i,n)$ = Prob [i backlogged nodes xmit]

$= \binom{n}{i} (1-q_r)^{n-i} q_r{}^i$

Returning to table, $P_{n,n+i}$ (-1 • i • m-n) can be calculated in terms of $Q_a(i,n)$ and $Q_r(i,n)$.

Now we can derive transitive probabilities:

i	#arrivals	#backlogged that retransmit	result	Probability
-1	0	1	1	$P_{n,n-1} = Q_a(0,n)Q_r(1,n)$
0	0	0	0	
0	0	2,3,...,n	e	$P_{n,n} = Q_a(0,n)[1-Q_r(1,n)]+Q_a(1,n)Q_r(0,n)$
0	1	0	1	
1	1	1,2,...,n	e	$P_{n,n+1} = Q_a(i,n)[1-Q_r(0,n)]$
2,3,..., m-n	i	anything	e	$P_{n,n+i} = Q_a(i,n)$

Solve for pn, then N_{bar} = Expect # backlogged packets then use Little's law: T = $\lambda/2$ to get mean delay.

(N_{bar} near 0 means almost no delay to xmit)

[Recall: will be lower bound on delay, because we assume unbuffered system]

Something is missing in analysis:

Dynamics of system

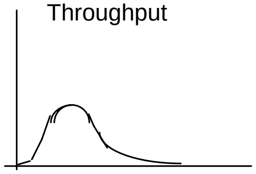
small q_r => large delay
large q_r => n grows and thus delay grows (more collisions)

The problem is that "optimal" q_r is function of n, but n is a result of q_r.

=> we want to explore this **feedback** with our model.

n small => make q_r large
n large => make q_r small

Let's calculate the departure rate, so we can graph:



Departure rate = E [successful xmits in one slot]

= Pr [one node successfully transmits]

= P_{succ}

P_{succ} = (see table, for "1" result)

= P[0 arrive, 1 trans] + P[1 arrive, 0 retrans]

= $Q_a(0,n)\ Q_r(1,n) + Q_a(1,n)Q_r(0,n)$

If we substitute our expressions for Qa and Qr, and use the approximation **$(1-x)^y = e^{-xy}$** for small x, then:

$$P_{succ} = G(n)e^{-G(n)}$$

G(n) = E[# attempted xmits in a slot when system in state n]

= $(m-n)q_a + nq_r$

= arrival rate + retransmission rate

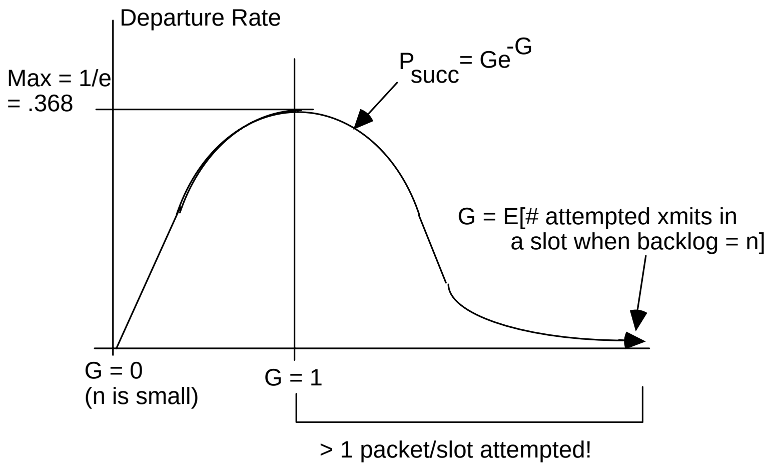
This is the same as Pr[1 arrival] in $\tau = 1$ interval in Poisson process with parameter $G(n)\tau$: ($\tau = 1$ interval is one slot)

Pr[n departures/arrivals] $= e^{-G(n)\tau} * (G(n)\tau)^n / n!$, with $n = 1$, $\tau = 1$

$= G(n)\ e^{-G(n)}$

<u>Approximate</u> # departures in slot as Poisson process with parameter G(n), which varies with system state.

Now we can look at Throughput: Tput = Departure rate • .368



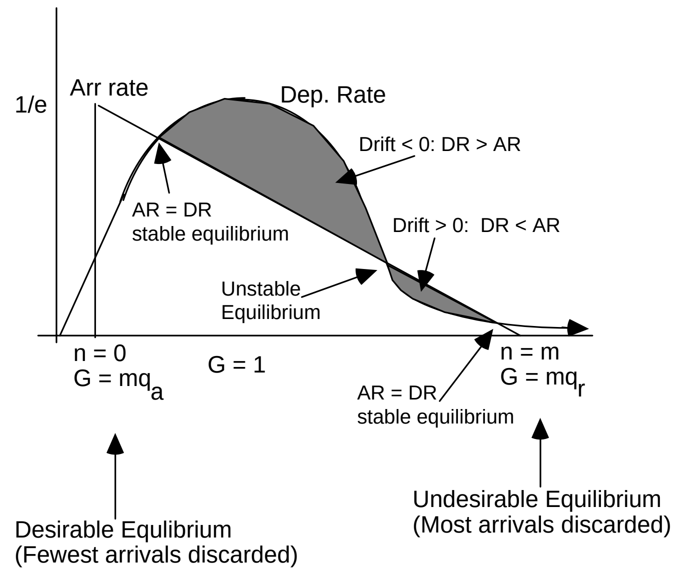
Same behavior we discussed in Internet: # collisions grows, driving down departure rate.
=> Arrivals are lost.

Now we have a graph of departure rate vs. G(n). But how is arrival rate related to departure rate?

Let's superimpose a graph of arrival rate:

E[arrival rate] = $(m-n)q_a$

Note that x axis is n, as well as G(n)



We will define DRIFT as the shaded region:

Drift in state n (D_n) = expected change in backlog over one slot time.

$\boxed{D_n = E[\text{arrival rate}] - E[\text{departure rate}]}$

Using the expression from above:

$D_n = (m-n)q_a - P_{succ}$

P_{succ} = Pr[successfull xmit at a node]

$D_n > 1$: AR > DR -> n will increase
$D_n < 1$: DR > AR -> n will decrease

Increase q_r: If scale for n is fixed, causes horizontal contraction of G(n) scale.
=> Unstable equilibrium occurs at smaller n!

Decrease q_r: Unstable equilibrium occurs at larger n (but delay also increases) & lots of arrivals discarded. As you decrease q_r, eventually have only 1 (desirable) stable point.

<u>Summary of 4.2, BG</u>

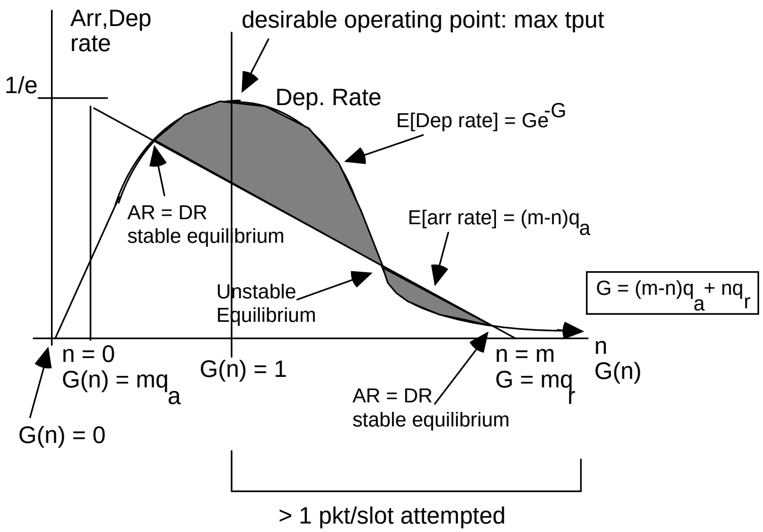
- Want delay/tput as function of offered load
- 6 assumptions
- Consider no buffering
- Model as discrete time Markov chain:
  State = # backlogged nodes
- Model gives E[#backlogged nodes] = N_{bar}, Little's law for T(delay)
- Steady state chain solution does not reveal "equilibrium" point.
- Also, model is based on fixed q_r. In reality, q_r may vary.
  small n => increase q_r
  large n => decrease q_r

  How can we capture this feedback dynamic relationship?

- Drift = D_n = E[Arr rate] - E[depart. rate]

  $= (m-n)q_a = G(n)e^{-G(n)}$

  $(m-n)q_a$ = poisson arrival with rate λ
  $G(n)e^{-G(n)}$ = Approx. depart. process as Poisson w/parm G(n)
  => this is Pr[1 depart] w/τ = 1 using Poisson distribution (assumes λ small (« 36.8%)
  G is "attempt rate"



Now consider infinite node assumption:

attempted departure rate (including collisions) =

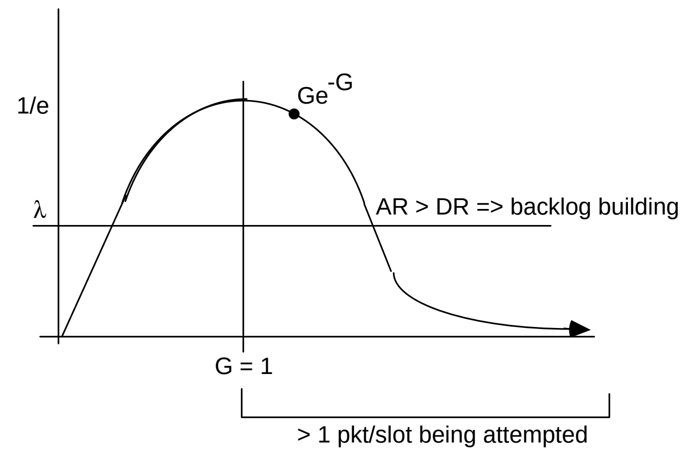
$G(n) = \lambda + nq_r$

<u>Arrival rate curve is horizontal line (constant λ)</u>

Markovian chain has no state distribution

Expected backlog increases w/o bound!

=> Expected delay is • for any λ.



Practical considerations:

1) if not slotted (var length msgs), max tput drops in half

2) We'll see, later, way to get close to 50% tput.

3) In practice, if λ << 1/e and q_r "moderate" (not to small)
System may run for long time w/o entering unstable state (remain there for long time)

Then restart system to recover.

However, better solutions exist:

BG 4.2.3 Stabilized Slotted Aloha => change q_r dynamically

P_{succ} has max at G(n) = 1.

Problem: Nodes don't know what n is; must estimate from feedback.

Example: increase q_r in idle slot
decrease q_r in e slot

Definition:

multiaccess system is stable if, given λ, E[delay/pkt] is finite.

Definition: Max stable tput: least upper bound of arrival rates for which system is stable

Ordinary SA is unstable: Max stable TPUT = 0 (with 6b (• buffer
Stabilized SA is stable: Max stable TPUT = 1/e assumtion)

p286-287

Ethernet differs from above: limited feedback

Each node has feedback only when it xmits. Violates earlier assumption. (Other differences - carrier sense)

Binary Exponential Backoff

- xmit arrival immediately

- If xmit unsuccessful i times, then probability of retransmit in successive slots is $1/2^i$

Max stable TPUT -> 0 as m->•, however!

Open Problem: Does any strategy exist that is stable, with this limited feedback?

Q: Do you think that Ethernets really are limited to 36.8% x 10Mbit/sec?

A: No! We saw that in graphs from [AC] paper!

Q: What's going on then? How was Ethernet invented?

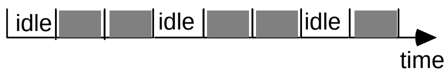
A: Metcalfe was reading Proceedings of Fall Joint Computer Conference 1970:
The Aloha System - another alt. for computer comm, to sleep.

Redid analysis in 1973, not sure if he ever finished at Harvard; left for PARC.

Published Ethernet paper in 1976 with Boggs

BG 4.4 Carrier Sensing

Recall that slotted ALOHA has idle & busy slots of equal length.

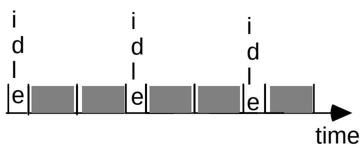


Why waste a full slot for IDLE?

If there (max prop delay + detection delay) << 1

(where max prop delay + detection delay has ß as an upper bound)

Then we can terminate idle slots after time ß < 1



The process of detecting idle slots is like detecting (sensing) an (imaginary) carrier.

Hence the name: CSMA (carrier sense, multiple access)

BG 4.4.1 CSMA Slotted Aloha

Two differences from <u>ordinary</u> slotted aloha:

1) Terminate idle periods after time ß

2) If packet arrives at a node while xmit is in progress, the packet is regarded as backlogged. Several variations of when to transmit this packet:

A) Non-Persistent CSMA:

i) xmit packet arriving in idle slot in next slot as usual
ii) xmit packet arriving in busy slot w/prob q_r <u>after each subsequent idle slot</u>

(better utilization than 1-persistent, due to lower prob. of collision for arrivals; <u>but</u> higher delay than 1-persistent)

B) P-Persistent CSMA:

i) (same rule)
ii) use probability p, not q_r, for 1st try. Use q_r for subsequent tries.

<u>1-Persistent CSMA is special</u> case of P-Persistent. Only difference from ordinary slotted aloha is carrier sense.

(better performance than ordinary slotted aloha due to CS mechanism).

BG 4.1 CSMA slotted Aloha

- Modelling

two slot durations: { ß = .01 - .10
{ 1

Idle slot always terminate after ß time units and a new slot begins.

Model of BG4.2 can be modified. (No longer transitions at discrete time)

Modify drift time to derive attempted departure:

rate(BG fig 4.16): (assumption 6b (• nodes))

$$D = ge^{-g}/(ß + 1 - e^{-g})$$

w/max of $1/(1 + •(2ß))$

and $g(n) = \lambda ß + q_r n$

Important result:

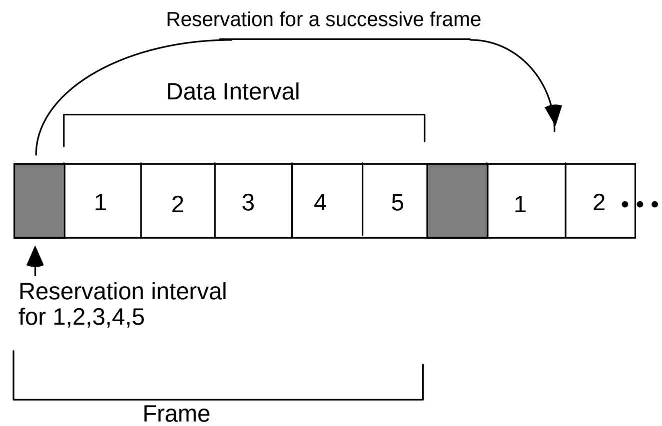
$\lim_{ß \to 0} D = 1$ => near 100% tput

See figure BG 4.16, Tanenbaum3.5

BG 4.5 Multiaccess Reservations

General idea:

if ß large (as in a satellite), let stations send tiny packets in contention mode or TDM to reserve future slots.
=> higher overall efficiency.



Using reservations w/SA yields 36% max utilization for reservation interval,
88% for entire BW
(assumes reserve slot is 5% of data slot)

[BG 4.5.2] Ethernet = CSMA/CD

Collision Detect added to CSMA at start of slot:
one reservation slot.
S > 1/(1 + 6.2ß)   (vs. 1/(1+ 1.4•ß for no CD w/non persistent)
(for ß = 1/100, CSMA/CD = 94, CSMA = 87)

[p BG 320] IEEE 802 standards

IEEE 802.3 (Ethernet) is:
- CSMA/CD

- binary exp. backoff
- <u>unslotted</u> persistent

[Stallings 3rd ed, p 425] see figure 11.23

<u>BG 4.5.3 Token Rings</u>

- IBM - Zurich lab developed during Ethernet standardization process
- Better delay characteristics under heavy load than CSMA/CD
- A conservative protocol from a conservative company!

(See fig. 1 [Hawe,Kempf,Kirby - DigTech Journal, Sept 86])

(See fig 3.20 [Tanenbaum])

(See fig 11-15 [Stallings])

(See fig 3-26, 3-27 [Tanenbaum])

(See fig 3-21 [Tanenbaum])