EEEN20060 Communication Systems

Analysis of MAC Protocols

Brian Mulkeen

UCD School of Electrical, Electronic and Communications Engineering

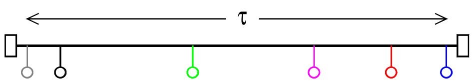
Scoil na hInnealtóireachta Leictrí, Leictreonaí agus Cumarsáide UCD

What do we want to know?

- Network Throughput
  - total rate of data transfer through system
  - bit/s or data block/s, from all users
  - or normalise: efficiency, block/frame time
- Average Transfer Delay
  - time from data block ready for transmission to final delivery at destination
  1. queuing time at sender
     - waiting behind other blocks that were ready earlier
  2. access delay, waiting for turn or permission
  3. contention delay, collisions & re-transmissions
  4. transmission time, actually sending frame
  5. propagation delay, sender to destination

2

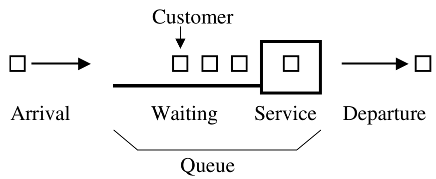
Scenario



- N devices share a channel
  - e.g. bus network, all in parallel on cable
- Blocks of data become ready at random times
  - often assume Poisson random process
  - queue to be transmitted
  - eventually sent as frame on channel
  - average time to transmit bits of frame is $\overline{T_F}$
- Propagation delay on channel
  - end-to-end (worst case) τ
  - average delay between two devices $\bar{\tau}$
- Ignore errors – short cable, separate problem

3

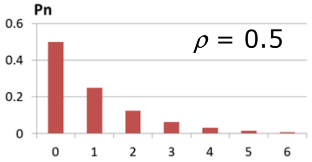
Queuing Theory

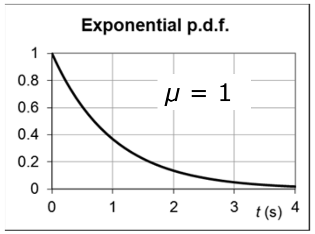


- In general, *customers*, here, data blocks
  - *arrive* into queue at random
    - here, Poisson random process
    - average rate of arrival λ (data block/s)
  - *wait* for some *service*, according to queue rule
    - here, service is transmission on channel
  - spend some time being served (service time)
    - here, frame transmission time – fixed or variable
    - average service rate is $\mu = \frac{1}{\overline{T_F}}$ (while busy)
  - then *depart* from the system

4

Queuing Theory

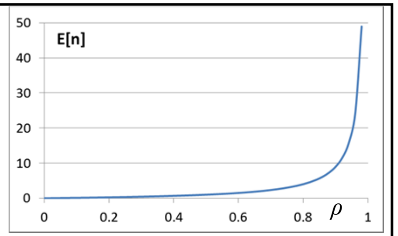




- For stable queue, need λ < μ
  - arrival rate < service rate
  - define ratio $\rho = \frac{\lambda}{\mu}$ as *link utilisation*, want < 1
- Consider random service times (frame size)
  - assume exponential prob. density function
    - as shown above – long frames less likely…
  - then prob. *n* customers in queue $P_n = \rho^n(1-\rho)$
    - assumes no limit on queue size…

5

Queue Size

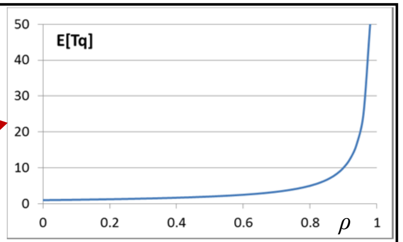


- Average number of customers in queue
  - know probability of *n* customers…
  - average: $\mathrm{E}[n] = \sum_{n=0}^{\infty} nP_n = \sum_{n=0}^{\infty} n\rho^n(1-\rho)$
  - get $\mathrm{E}[n] = \frac{\rho}{1-\rho}$ grows rapidly as $\rho \to 1$
- Assumptions (not always realistic):
    - arrivals are Poisson random process
    - service time (frame size) exponential p.d.f.
    - infinite queue (no limit on size)
  - but similar shape for other cases

6

Queuing Time

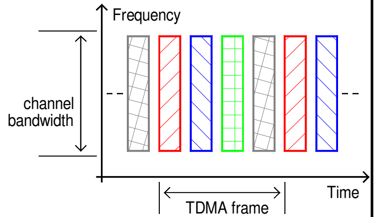
Normalised: time in units of average service time



- Average time in system (incl. service)
  - Little's formula: $\mathrm{E}[n] = \lambda \mathrm{E}[t_Q]$
  - so $\mathrm{E}[t_Q] = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu(1-\rho)}$
- Average time waiting: subtract service time
  - $\mathrm{E}[t_W] = \mathrm{E}[t_Q] - \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)}$
- Fixed service time, this becomes
  - $\mathrm{E}[t_W] = \frac{\rho}{2\mu(1-\rho)}$ only half as long

7

Example: Fixed TDMA



- Assume fixed-length link-layer frames
  - length T_F, chosen to fit in one time-slot
- Fixed and equal allocations to *N* devices
  - so *N* independent queues, arrival rates λ_i
  - service rate fixed, one block per cycle, $\mu = \frac{1}{T_C}$
  - so $\rho_i = \frac{\lambda_i}{\mu} = \lambda_i T_C$ and $\mathrm{E}[t_W] = \frac{\rho_i}{1-\rho_i}\frac{T_C}{2}$
- Cycle time is TDMA frame time, $T_C > NT_F$,
  - with overhead, let $T_C = (N+O)T_F$

8

Fixed TDMA...

- Access delay – waiting for turn
  - average = half cycle time
  - but no contention delay
- Average transfer delay: $T_i = \frac{\rho_i}{1-\rho_i}\frac{T_C}{2} + \frac{T_C}{2} + T_F + \bar{\tau}$
- Total throughput: $\lambda = \sum_{i=1}^{N} \lambda_i$
  - assuming all queues stable, $\lambda_i < \mu$
  - max throughput needs frame in every time-slot
- Note:
  - *if* equal traffic from all, $\rho_i = \frac{\lambda}{N} T_C = \lambda T_F \frac{N+O}{N}$
  - *if* could combine all data into one queue...
  - service rate (on same channel) $\frac{1}{T_F}$, so $\rho = \lambda T_F$
  - average transfer delay: $T = \frac{\rho}{1-\rho}\frac{T_F}{2} + T_F + \bar{\tau}$

9

Example: Polling, independent devices

- Devices generate traffic as in TDMA
  - queue, wait for poll before transmitting
- Cycle time now variable
  - depends on who has data, and how much...
- Cycle time has fixed component, *latency*, *L*
  - minimum cycle time, time to poll all devices
  - centralised polling: $L \approx N(T_{poll} + T_{response} + 2\bar{\tau})$
    - for each device, send poll, received after prop. delay
    - device sends response, received after prop. delay
    - times depend on bits in messages and channel bit rate
  - token passing: $L \approx N(T_{token} + \bar{\tau})$
    - for each device, time to transmit token, prop. delay
- Access delay = half average cycle time

10

Polling...

- Waiting time in queue?
  - difficult to calculate in general, often simulate...
- Example:
  - device may send one frame when polled
  - so service rate is one frame per cycle
- How many devices will send frame in cycle?
  - device will send if queue not empty
  - prob. queue empty = $P_0 = 1 - \rho$
  - prob. of device i sending = ρ_i
  - expected no. frames in cycle = $\sum_{i-1}^{N} \rho_i$
- Average cycle time: $\bar{t_c} = L + \overline{T_F} \sum_{i=1}^{N} \rho_i$
  - hence average service rate...

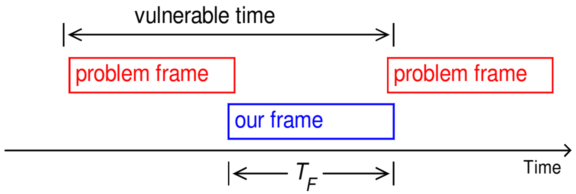
11

Example: Polling, master driven

- Devices generate data when polled
  - e.g. sensors in fire detection system
    - report status when asked, assume ask regularly
    - never want to transmit independently
  - assume equal priority, all polled at same rate
- Throughput depends on polling rate
  - design question is: how often can we poll?
- Cycle time: $T_C = N(T_{poll} + T_{response} + 2\bar{\tau})$
  - for each device, send poll, including data
  - device sends response, with data for master
    - times depend on bits in messages and channel bit rate
  - no queue at devices, no real access delay
    - nothing random in this system
  - so throughput is *N* blocks per cycle...

12

Example: Aloha



- Devices transmit frame when ready
  - collision if two (or more) frames overlap
- Assume data blocks fixed length (to simplify)
  - giving frame time T_F
- Vulnerable time $2T_F$
  - collision with our frame if any other device becomes ready in this interval…

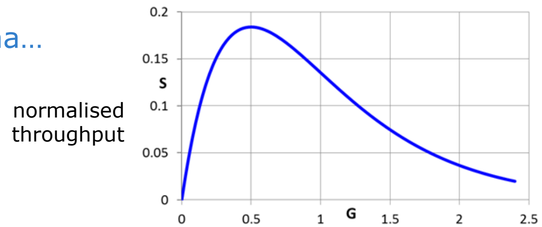
13

Aloha…

- After collision, back-off, re-transmit…
- To simplify analysis, assume:
  - data blocks arrive for transmission (across all devices) as Poisson random process
  - average arrival rate (new data blocks) λ
  - back-off algorithm makes total transmission attempts a Poisson random process also
  - average rate of attempts $\lambda' > \lambda$
- Probability of success
  - P_S = prob. no transmit attempts in time $2T_F$
  - for Poisson random process, $P_S = e^{-\lambda' 2T_F}$
  - or P_S = fraction of attempts that succeed: $P_S = \frac{\lambda}{\lambda'}$
  - equating these: $\lambda = \lambda' P_S = \lambda' e^{-\lambda' 2T_F}$

14

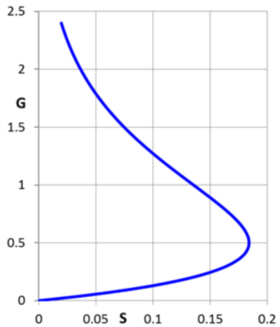
Aloha…



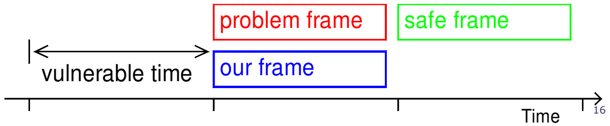
- Usually normalise to frame time
  - normalised throughput $S = \lambda T_F$ block/frame time
  - normalised attempt rate $G = \lambda' T_F$ attempt/fr. time
- Then $S = Ge^{-2G}$
  - differentiate: $\frac{dS}{dG} = e^{-2G}(1-2G)$ zero for max…
  - max throughput at G = 0.5 attempt/frame time
  - and $S_{max} = \frac{1}{2e} \approx 0.184$ block/frame time

15

Aloha…



- Alternative view
  - desired throughput on X-axis
  - see two ways to get it
  - one with many collisions and long delays…
- Back-off algorithm vital!
  - want to stay on lower part of curve…
- Slotted Aloha
  - vulnerable time T_F
  - analysis similar…



16