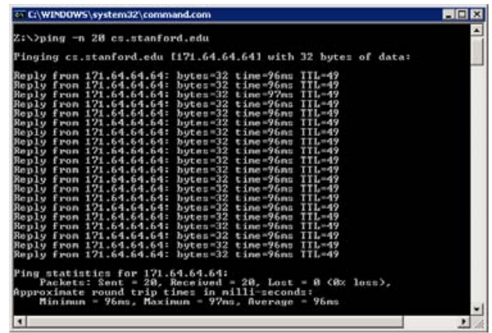
MA/CS-109: What took you so long?

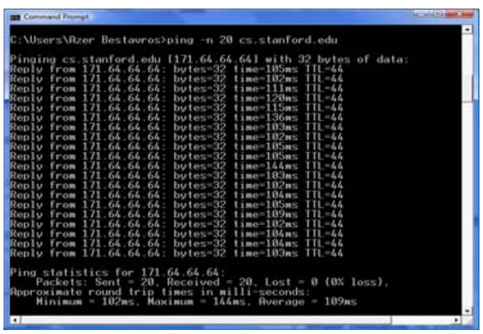
Azer Bestavros

Network delays (BU – Stanford)



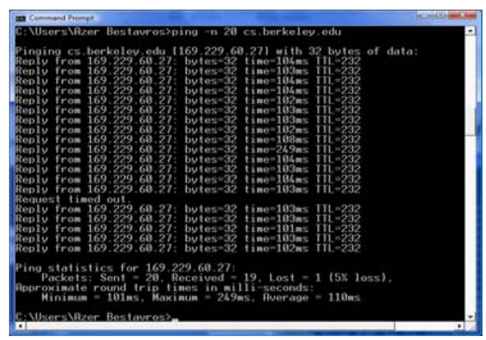
```
C:\WINDOWS\system32\command.com

Z:\>ping -n 20 cs.stanford.edu

Pinging cs.stanford.edu [171.64.64.64] with 32 bytes of data:

Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=97ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49
Reply from 171.64.64.64: bytes=32 time=96ms TTL=49

Ping statistics for 171.64.64.64:
    Packets: Sent = 20, Received = 20, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 96ms, Maximum = 97ms, Average = 96ms
```


Network delays (Home – Stanford)



```
Command Prompt

C:\Users\Azer Bestavros>ping -n 20 cs.stanford.edu

Pinging cs.stanford.edu [171.64.64.64] with 32 bytes of data:
Reply from 171.64.64.64: bytes=32 time=105ms TTL=44
Reply from 171.64.64.64: bytes=32 time=102ms TTL=44
Reply from 171.64.64.64: bytes=32 time=111ms TTL=44
Reply from 171.64.64.64: bytes=32 time=120ms TTL=44
Reply from 171.64.64.64: bytes=32 time=115ms TTL=44
Reply from 171.64.64.64: bytes=32 time=136ms TTL=44
Reply from 171.64.64.64: bytes=32 time=103ms TTL=44
Reply from 171.64.64.64: bytes=32 time=102ms TTL=44
Reply from 171.64.64.64: bytes=32 time=105ms TTL=44
Reply from 171.64.64.64: bytes=32 time=105ms TTL=44
Reply from 171.64.64.64: bytes=32 time=144ms TTL=44
Reply from 171.64.64.64: bytes=32 time=103ms TTL=44
Reply from 171.64.64.64: bytes=32 time=102ms TTL=44
Reply from 171.64.64.64: bytes=32 time=104ms TTL=44
Reply from 171.64.64.64: bytes=32 time=105ms TTL=44
Reply from 171.64.64.64: bytes=32 time=109ms TTL=44
Reply from 171.64.64.64: bytes=32 time=102ms TTL=44
Reply from 171.64.64.64: bytes=32 time=104ms TTL=44
Reply from 171.64.64.64: bytes=32 time=104ms TTL=44
Reply from 171.64.64.64: bytes=32 time=103ms TTL=44

Ping statistics for 171.64.64.64:
    Packets: Sent = 20, Received = 20, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 102ms, Maximum = 144ms, Average = 109ms
```

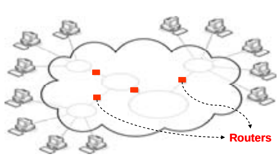

Network delays (Home – Berkeley)



```
Command Prompt
C:\Users\Azer Bestavros>ping -n 20 cs.berkeley.edu

Pinging cs.berkeley.edu [169.229.60.27] with 32 bytes of data:
Reply from 169.229.60.27: bytes=32 time=104ms TTL=232
Reply from 169.229.60.27: bytes=32 time=103ms TTL=232
Reply from 169.229.60.27: bytes=32 time=104ms TTL=232
Reply from 169.229.60.27: bytes=32 time=104ms TTL=232
Reply from 169.229.60.27: bytes=32 time=102ms TTL=232
Reply from 169.229.60.27: bytes=32 time=103ms TTL=232
Reply from 169.229.60.27: bytes=32 time=103ms TTL=232
Reply from 169.229.60.27: bytes=32 time=102ms TTL=232
Reply from 169.229.60.27: bytes=32 time=108ms TTL=232
Reply from 169.229.60.27: bytes=32 time=249ms TTL=232
Reply from 169.229.60.27: bytes=32 time=104ms TTL=232
Reply from 169.229.60.27: bytes=32 time=103ms TTL=232
Reply from 169.229.60.27: bytes=32 time=104ms TTL=232
Reply from 169.229.60.27: bytes=32 time=103ms TTL=232
Request timed out.
Reply from 169.229.60.27: bytes=32 time=103ms TTL=232
Reply from 169.229.60.27: bytes=32 time=103ms TTL=232
Reply from 169.229.60.27: bytes=32 time=101ms TTL=232
Reply from 169.229.60.27: bytes=32 time=103ms TTL=232
Reply from 169.229.60.27: bytes=32 time=102ms TTL=232

Ping statistics for 169.229.60.27:
    Packets: Sent = 20, Received = 19, Lost = 1 (5% loss),
Approximate round trip times in milli-seconds:
    Minimum = 101ms, Maximum = 249ms, Average = 110ms

C:\Users\Azer Bestavros>_
```

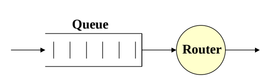

Understanding (network) delays

- Two components
  - Propagation delays
  - Other (router) delays



- Highway analogy
  - Propagation delays ~ time moving at posted speeds
  - Other delays ~ time stopped in traffic jams
- Propagation delay is fixed (for a fixed route) – you can check it with “ping”.

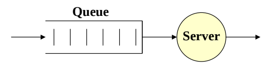
Queues: We can’t live without them!



- Queues are needed to manage the short-term mismatch between
  - the speed with which requests are made, and
  - the speed with which requests are served

1 Animation = 1K Words

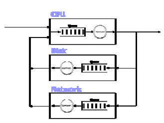
Queues are useful abstractions

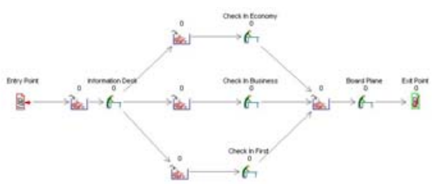


Not just for Internet routers, but also for
- Store check-out
- Car wash
- Airport check-in
- Fast-food lines
- Post office queue
- Toll booths
- …

Queues are useful abstractions

- Entire "systems" are interconnected queues!



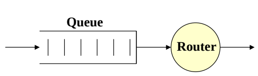


Queuing delay: Questions

- How does queuing delay relate to load?
  - Is the relationship "linear" – e.g., doubling the load results in doubling the delay?
- How does queuing delay relate to router speed?
  - When does it make sense to upgrade a router? or add a lane to the toll booths?

Modeling the queue at the router

- Think about the packets queued up at a router as a "population" that grows/shrinks as a result of
  - Packets arriving to the router (births), and
  - Packets leaving the router (deaths)



- Note:

  Births are the result of some external processes – i.e., not a function of the population!

Simplifying assumptions

- There is only one congested router for a given route
- There is enough space for as many packets as needed
- An average of B packets arrive every second (birth rate)
- When there are packets in the queue, an average of D packets depart every second (death rate)
- Time advances in very tiny intervals T
- One of three things may happen in a given interval T
  - A birth – a packet is added to the queue
  - A death – a packet is removed from the queue
  - Neither – nothing changes

Router Utilization (U)

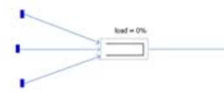
- B is a measure of the traffic intensity (demand)
- D is a measure of the router service capacity
- U = B / D is a measure of how busy the router is

- If U > 1
  - Router has no chance to keep up with demand
  - We know what will happen -- not interesting

- We will assume that U < 1 (i.e., B < D)

Modeling queue population (q)

- At any point in time, the queue is either:
  - empty q=0
  - has 1 packet q=1
  - has 2 packets q=2
  - has 3 packets q=3
  - has 4 packets q=4
  - … …
  - has i-1 packets q=i-1
  - has i packets q=i
  - has i+1 packets q=i+1
  - … …



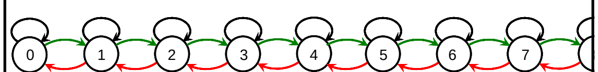
How does the population change?

- Consider the very small interval of time T

- As per our assumptions, T is so small that only one of 3 things can happen:
  - A birth
  - A death
  - Neither

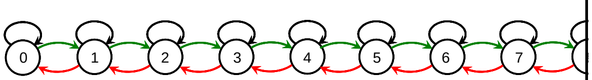
- A birth will add 1 to the population; a death will subtract 1 from the population; otherwise, the population does not change

Evolution of Queue Population



- We have seen this before!
- It's a random walk on a graph
  - Nodes are the different possible queue sizes
  - Edges denote birth, death, or neither events

Evolution of Queue Population

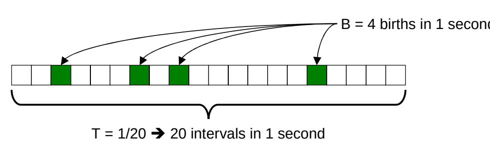


- We can find the relative frequency of the different nodes, if we can figure out the chances of following the various edges

Chances of a birth in any interval

- In one second, we expect B births
- Consider very small intervals T

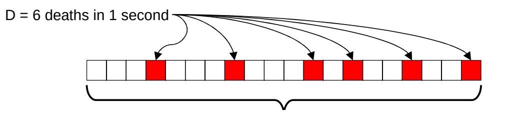
- Example: B = 4 (births/sec) and T = 1/20 = 0.05 sec



- Chances of a birth in an interval is B/(1/T) = **B*T** = 0.2

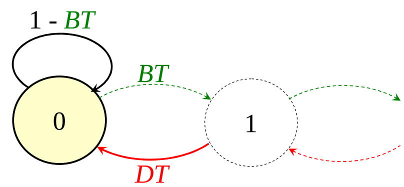
Chances of a death in an interval

- In one "busy" second, we expect D deaths
- Consider very small intervals T
- Example: D = 6 (deaths/sec) and T = 1/20 = 0.05 sec



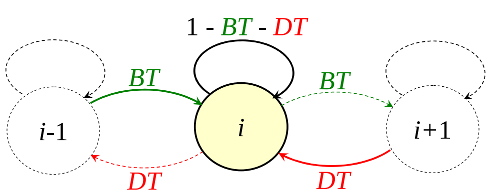
- Probability of a death in an interval is D/(1/T) = **D*T** = 0.3

Evolution of queue population size



$$\Pr(0) = DT\,\Pr(1) + (1 - BT)\Pr(0)$$
$$\Pr(1) = \frac{B}{D}\Pr(0)$$
$$\Pr(1) = U\,\Pr(0)$$

Evolution of queue population size



$$\Pr(i) = BT\,\Pr(i-1) + DT\,\Pr(i+1) + (1 - BT - DT)\Pr(i)$$
$$\Pr(i+1) = \left(1 + \frac{B}{D}\right)\Pr(i) - \frac{B}{D}\Pr(i-1)$$
$$\Pr(i+1) = (1+U)\Pr(i) - U\,\Pr(i-1)$$

Relationship between graph nodes

$$\Pr(1) = U\,\Pr(0)$$
$$\Pr(2) = (1+U)\Pr(1) - U\,\Pr(0) = U^2\,\Pr(0)$$
$$\Pr(3) = (1+U)\Pr(2) - U\,\Pr(1) = U^3\,\Pr(0)$$
$$\ldots$$

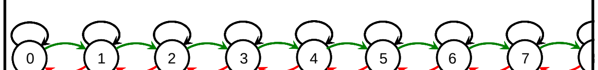
$$\Pr(i) = U^i\,\Pr(0)$$

But it must all add up to 100%

$$\Pr(0) + \Pr(1) + \Pr(2) + \ldots = 1$$
$$\Pr(0) + U\,\Pr(0) + U^2\,\Pr(0) + U^3\,\Pr(0) + \ldots = 1$$
$$\Pr(0)\left[1 + U + U^2 + U^3 + \ldots\right] = 1$$
$$\Pr(0)\left[\frac{1}{1-U}\right] = 1$$

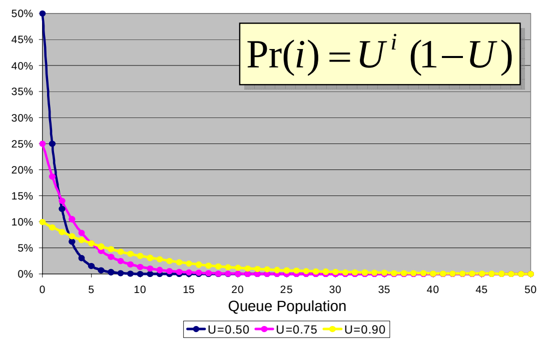
$$\Pr(0) = 1 - U$$

Chance of different queue lengths



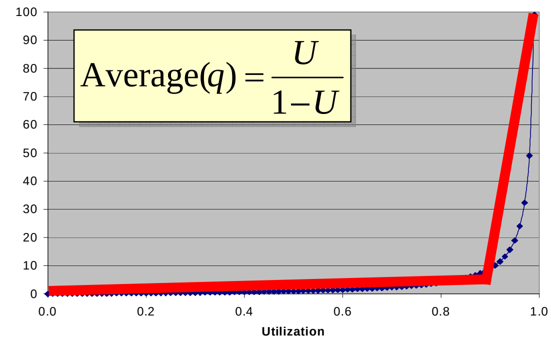
$$\Pr(i) = U^i\,(1-U)$$
$$\text{where } U = \frac{B}{D}$$

Chances of queue population = *i*



$$\Pr(i) = U^i\,(1-U)$$

Mean population (size) of the queue

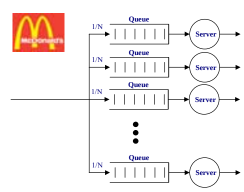


$$\text{Average}(q) = \frac{U}{1-U}$$

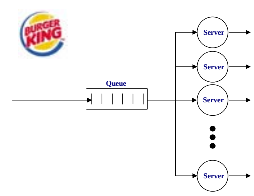
Basic observations from model

- Queues build up slowly with demand, when utilization is low
- Queues build up very fast with demand, when utilization is high
- Model explains why we often perceive lines to be either non existent or very long (network is either quite fast or very slow)
- If you want to ensure that lines will be short, then make sure utilization stays below ~ 80-85%
- Pushing a system to its capacity will backfire…

One Queue or N Queues?



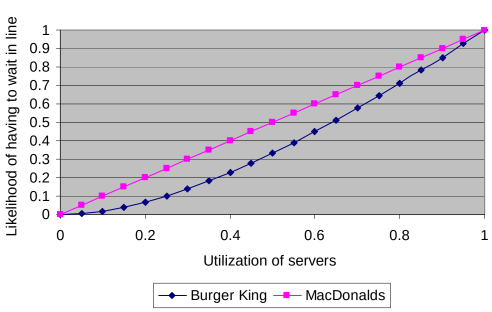
As in Supermarkets, Toll-boths, …



As in Airport check-in, Bank tellers…

How are these two designs different?
What are the tradeoffs?

One queue or two queues?



Using similar exercises we can…

- Study the behavior of queues that are served by multiple servers as opposed to a single one
- Establish that airport queues (single line served by multiple agents) are better than supermarket queues (one line per agent)
- Predict the loss rate of packets if queues have limited storage capacities
- Analyze collections of queues that are interconnected
- …