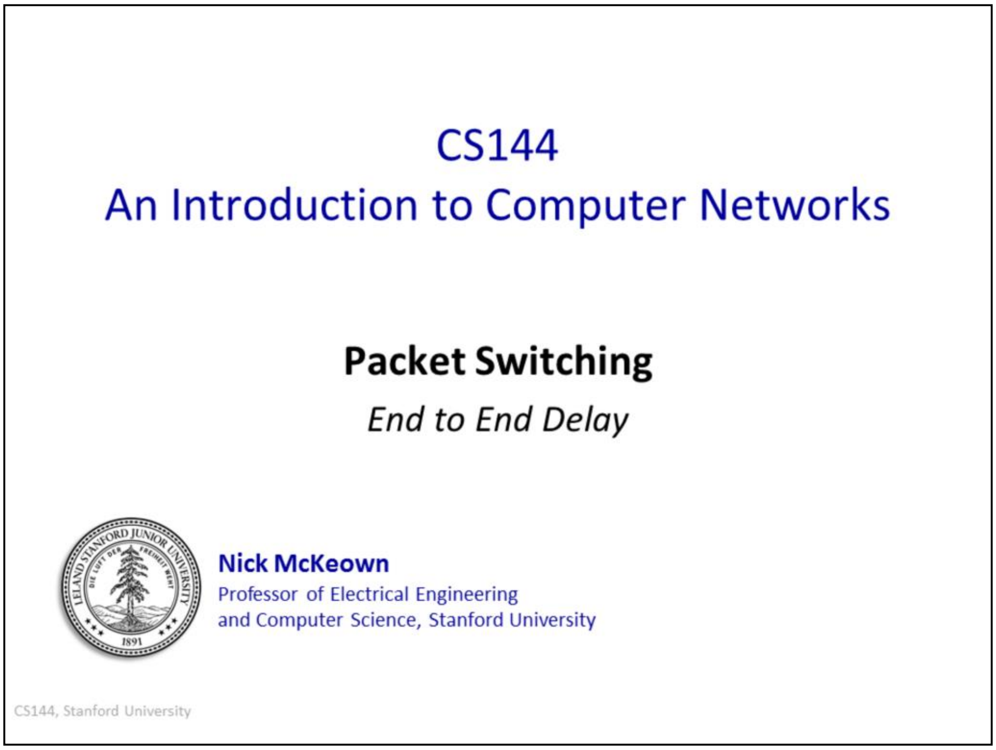
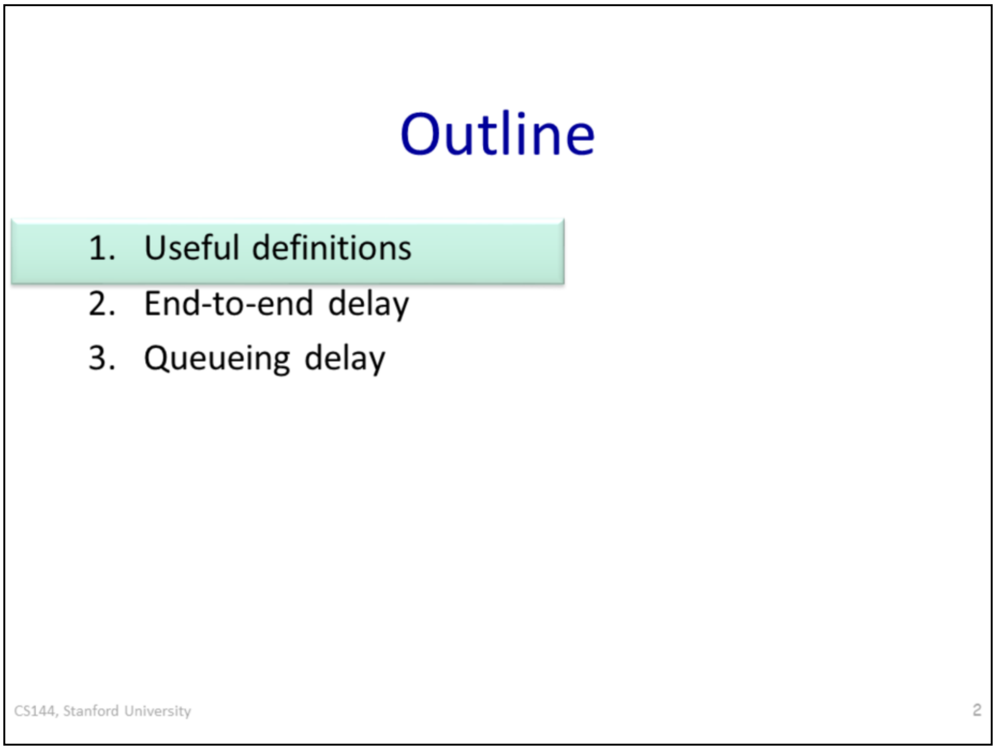
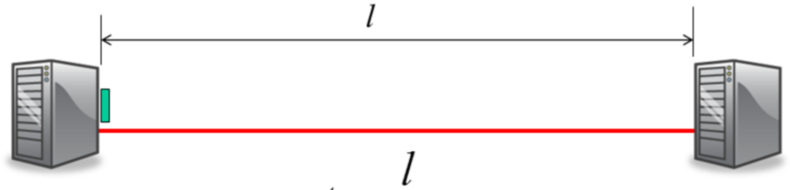


In the first video on packet switching I told you what packet switching is, and why the Internet uses it. Packet switching is going to feature prominently throughout this course --- many of the properties of the Internet follow from the choice of packet switching.



In this video I’m going to give you some useful definitions forpropagation delay and packetization delay, and we’re going to use these definitions to come up with an expression for the end-to-end delay of a packet. I’m also going to tell you about queueing delay and how it makes the end to end delay unpredictable. Finally, I’m going to explain why the unpredictable end to end delay means that applications like youtube and skype need to use a playback buffer.

Propagation Delay, t_l: The time it takes a single bit to travel over a link at propagation speed c.



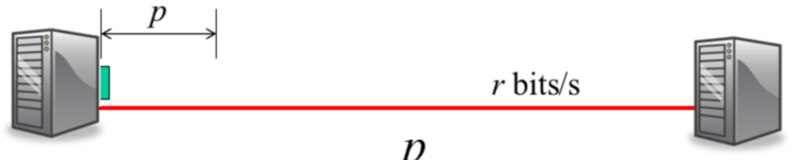
$$t_l = \frac{l}{c}$$

Example: A bit takes 5ms to travel 1,000km in an optical fiber with propagation speed 2 x 10^8 m/s.

We’ll start with the definintion of propagation delay. The propagation delay is the time it takes a single bit of information to travel over a link at propagation speed c. The propagation delay is simply determined by how long the link is, l in our case, and the speed the bit travels, c. We use the variable c because in most of the links we’re interested in the speed of propagation is close to the speed of light. For example, in a twisted pair of electrical cables a bit propagates at about 70% of the speed of light, and in an optical fiber the speed of propagation is a little bit slower. In most of our examples we’ll assume the bit propagates at 2 x 10^8 m/s, which is close enough.

Notice that the propagation delay doesn’t depend on the data-rate of the link. It doesn’t matter if the link is running at 1kb/s or at 10Gb/s – the propagation delay is just a function of the speed of propagation of each bit, and the length of the cable.

Packetization Delay, t_p: The time from when the first to the last bit of a packet is transmitted.



$$t_p = \frac{p}{r}$$

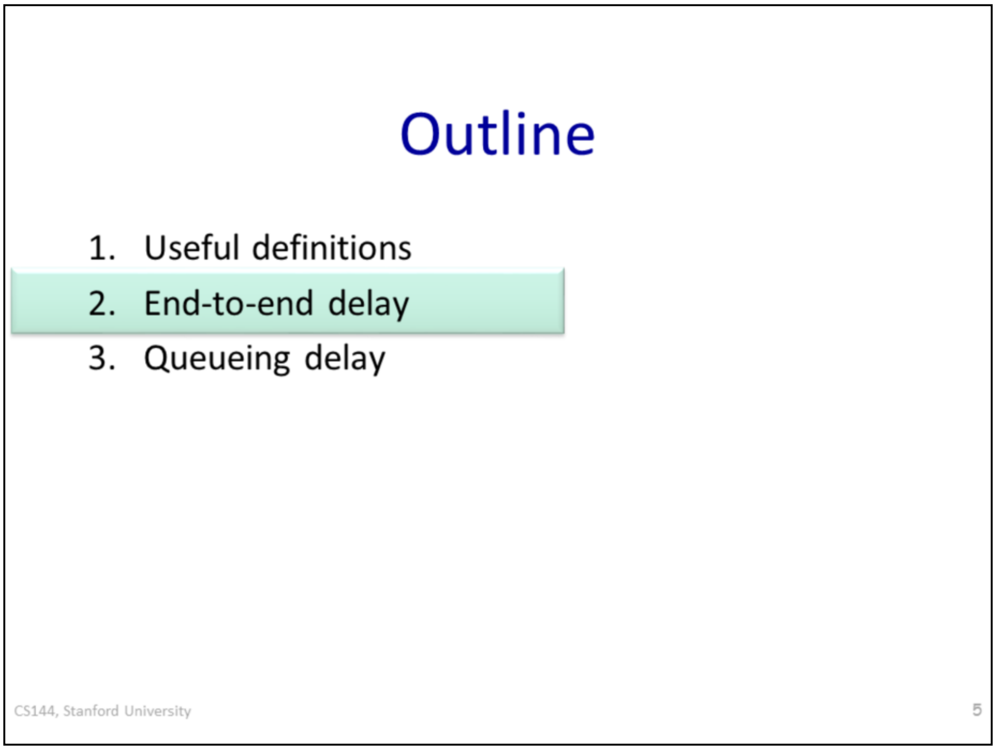
Example 1: A 64byte packet takes 5.12μs to be transmitted onto a 100Mb/s link.
Example 2: A 1kbit packet takes 1.024s to be transmitted onto a 1kb/s link.

CS144, Stanford University 4

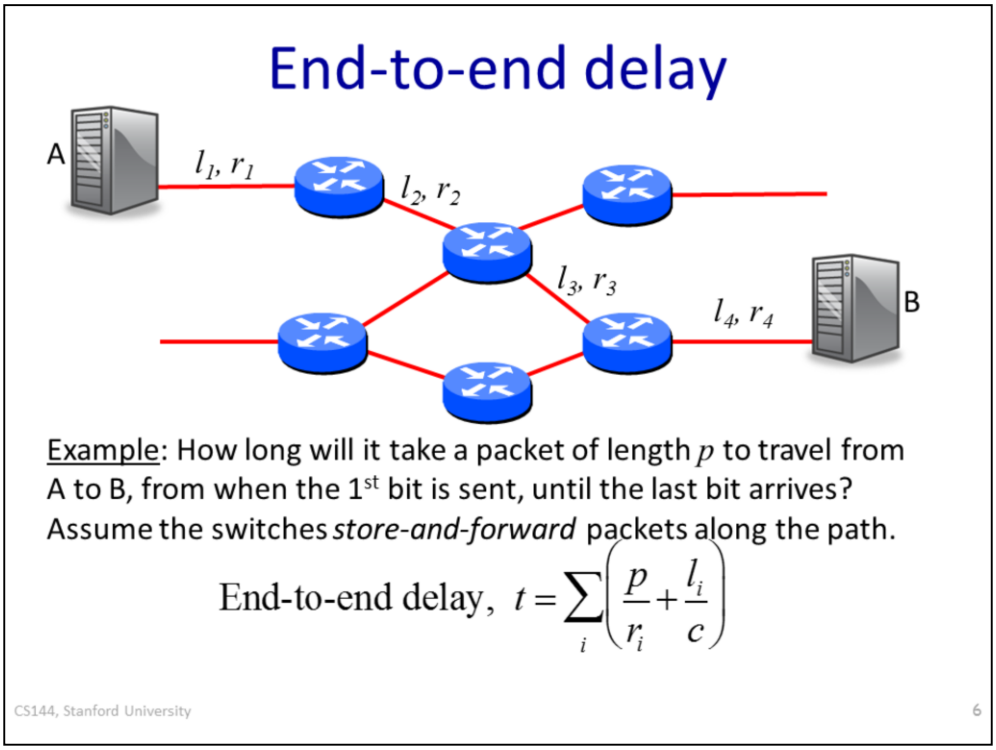
Another useful definition is the packetization delay. This is the time from when the first bit of a packet is put onto the link until the last bit is put onto the link. The packetization delay is determined by how fast we can put bits onto the link, or the data-rate, r. If a link runs at 1kb/s we can put 1,000 new bits onto the link every second. If it runs at 10Gb/s then we can put 10 billion every second.

The data-rate of a link is determined by how close together we can pack the bits. If, for example, a link runs at 1Gb/s, then we can put one bit onto the link every nano second. We'll see in a later video about Physical Links what limits the data-rate of a link.

Notice that the packetization delay is only a function of the length of a packet and the data-rate. It makes no difference how long the link is, or how fast bits propagate along it.



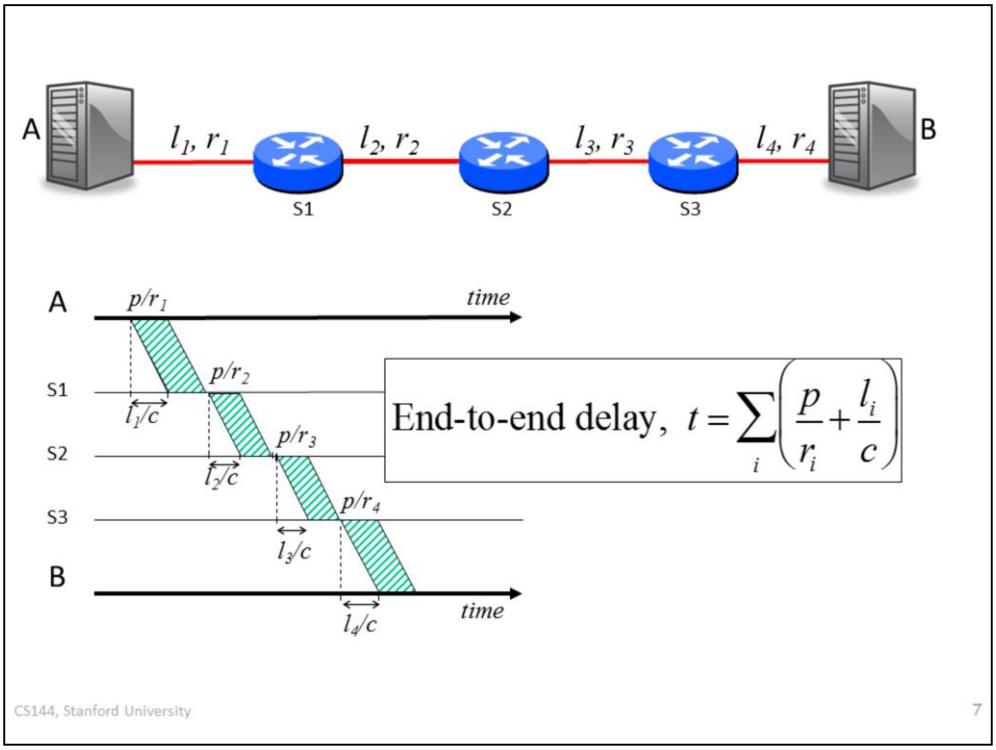
Next we're going to see how we can use the two different types of delay to determine the end-to-end delay – which is the time it takes a packet to go across the network from the source to the destination.



The end-to-end delay is the time from when we send the first bit onto the first link, until the last bit of the packet arrives at the destination. We can calculate the end-to-end delay by adding up the propagation and packetization delays on every link along the path.

In the example shown here, the packet is going to traverse 4 links. First, we calculate the time it takes the first bit to reach the first switch, then we calculate the time for the last bit to reach the first switch. We repeat the process for each link along the path.

Here is expression for the overall end-to-end delay.



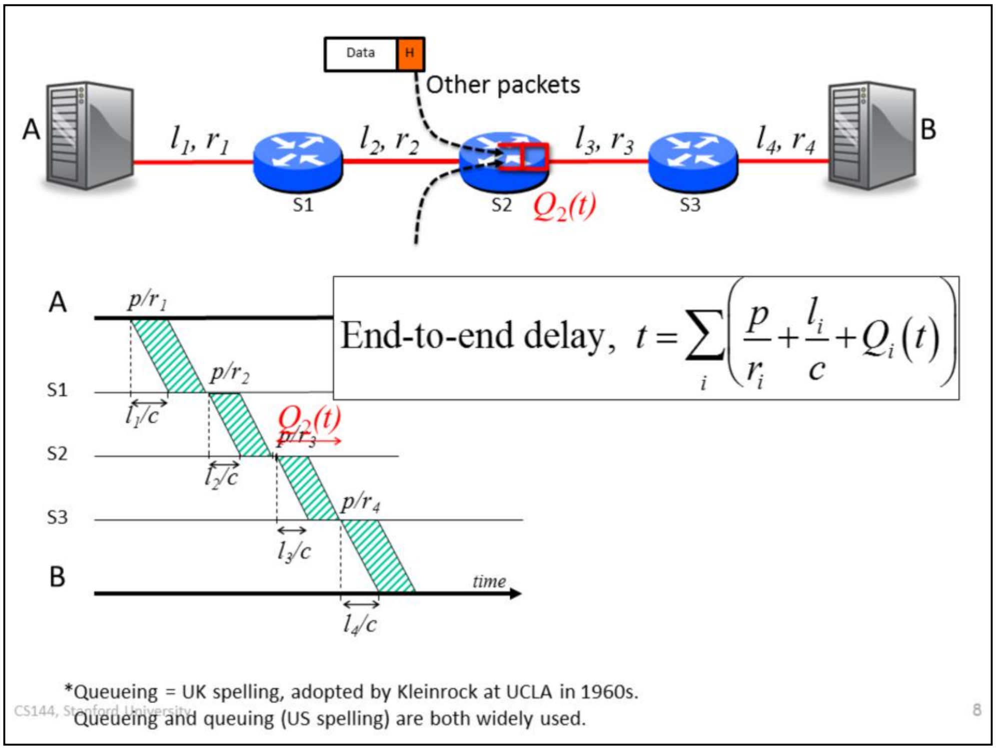
Now let's take a closer look at what's going on in detail. I've stretched out the links and switches into a single path and removed the rest of the network.

This is a timeline showing how the bits and packets travel along the links. The first bit takes l1/c to go from A to S1. The last bit of the packet is sent p/r1 bits after the first, and so the whole packet has arrived at S1 by l1/c + p/r1.

Internet routers are what we call "store and forward" packet switches. This means they wait until the whole packet arrives until they look up the address and decided where to send it next. They could instead start forwarding the packet after they've seen the header and not wait for the whole packet to arrive – ssomething called "cut through" switching. But Internet routers generally don't do this. In a later video and in some of the exercises we'll see some examples of cut through packet switches.

Getting back to our example, our router stores and then forwards the packet. On the link from S1 to S2 it takes the first l2/c to reach S2, and the last bit arrives p/r2 seconds later. This continues until the packet reaches B.

Our expression for the end-to-end delay adds up all the components of delay along the path.



OK, so I've not told you the whole story yet. The thing about packet switching is that your packets share the links with packets from other users. <click> When several packets show up at the same time wanting to go on the link from S2 to S3, then some of the packets have to wait in the router's queue, or packet buffer. We say the link from S2 to S3 is congested because there are lots of packets queued waiting to travel along it. The packet buffer helps prevent us from having to drop packets – the bigger the buffer, the less likely we are to have to drop a packet waiting to travel across the link. Packet buffers are fundamental to packet switching. If we didn't have packet buffers, then we'd lose a packet every time two packets showed up at the same time wanting to travel across the same link.

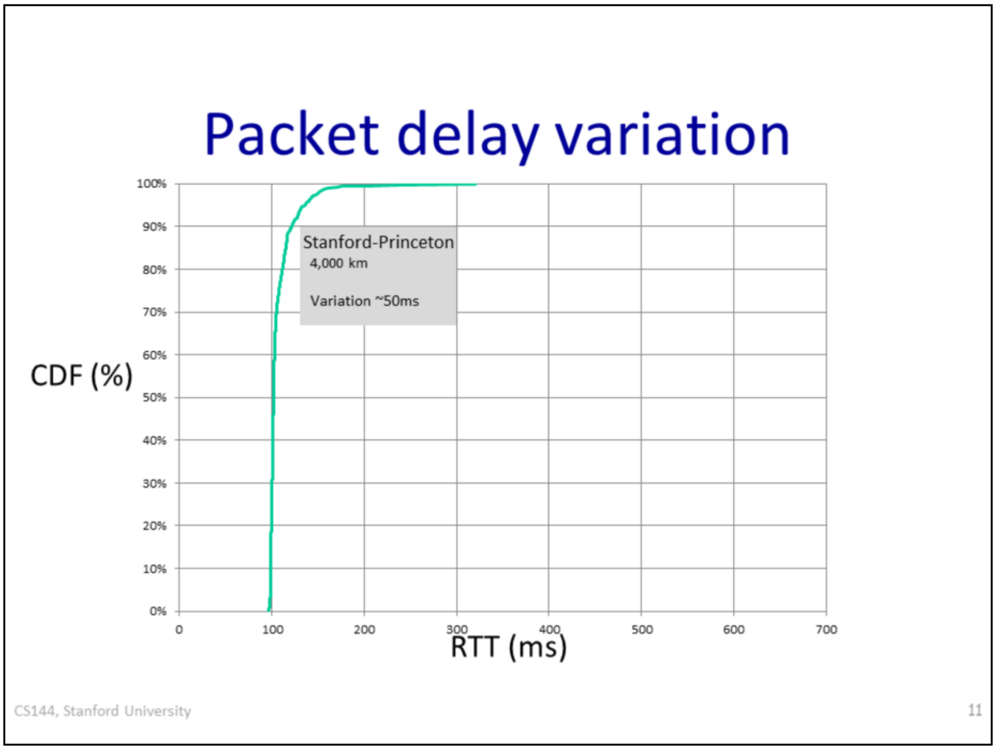
The packet buffer changes our expression for the end to end delay. If our packet arrives and the queue has packets in it, then the packet has to wait for the other packets ahead of it in the buffer. By default, the packet buffer in an Internet router is FCFS, which means the packets are transmitted in the order they arrived. So if there are three packets ahead of us, we have to wait for three packetization delays before its our turn to depart.

Here is our improved expression for the end to end delay, taking into consideration the queueing delay at each packet switch along the way. It's really important to remember that the queueing delay is unpredictable – it depends on the traffic sent by other users in the network. As far as we are concerned, the queueing delay is a random variable. It's the only random variable in our expression for end-to-end delay – everything else is deterministic.

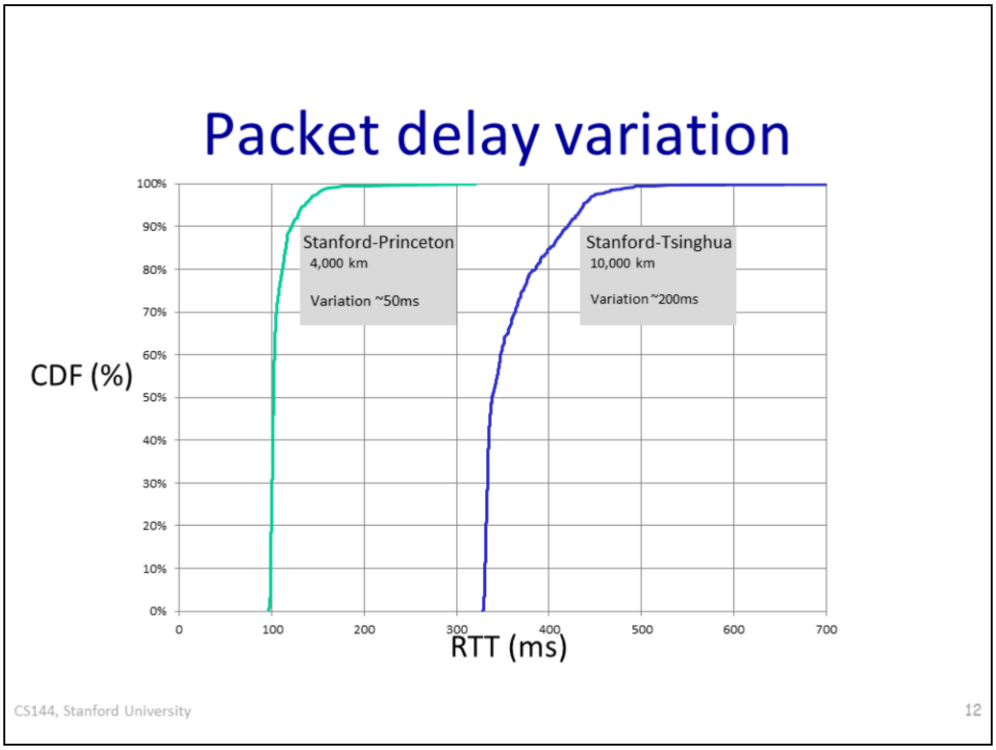
In case you don't believe me that the end-to-end delay is unpredictable, let's measure it. I'm going to use a very widely used tool called "ping" to measure the end-to-end delay between my computer and other computers in the Internet. Ping actually measures the RTT, which is the end-to-end delay in both directions. You'll find the ping command on your computer and you can use it to repeat the measurements I'm about to do on your own computer.

Insert the “ping” video here.

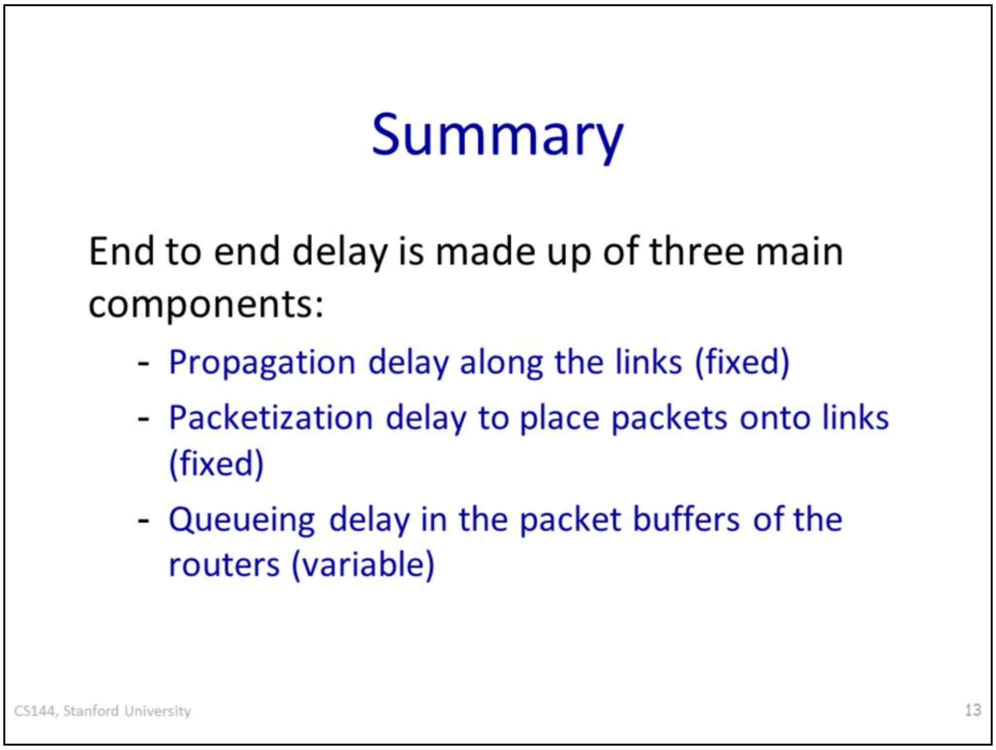
<end>



I used ping to measure a few hundred RTT values from my computer at Stanford to Princeton. The graph shows the CDF, or cumulative distribution function of the RTTs I measured. The overall variation is about 50ms, with 90% of the samples between 100 and 120ms.



I repeated the experiment from Stanford to Tsinghua university in China, about 10,000 km away. As I would expect, the RTT values are much larger because the propagation delay is much higher. But notice that the RTT samples have much greater variance – they range from about 320ms to over 500ms. This variance comes from my ping packets encountering packets from other users in the network. My packets meet their packets in the router buffers along the way. How long my packets have to wait depends on how many other packets happen to be waiting there when mine arrive. With a range of about 200ms, the variable queueing delay makes up almost half of the overall end to end delay.



<Read from slide>

This is the end of the video on end to end delay.