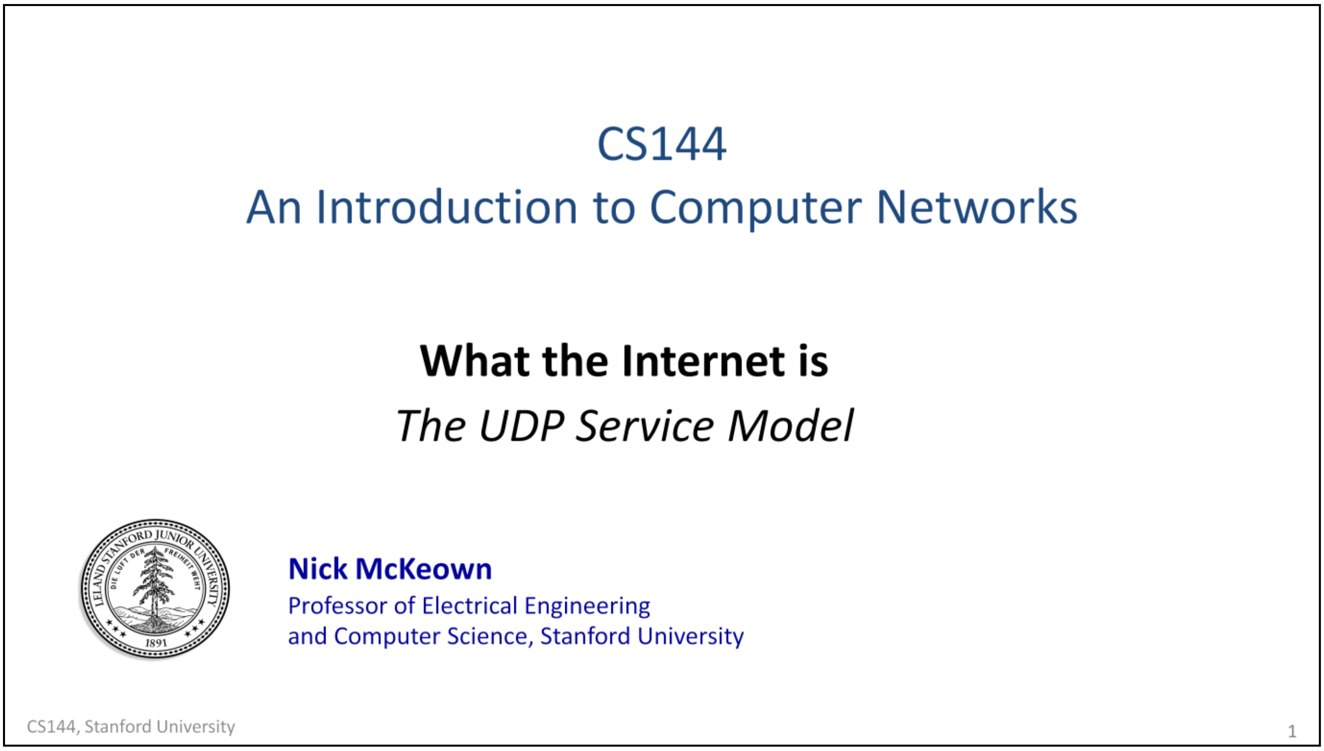
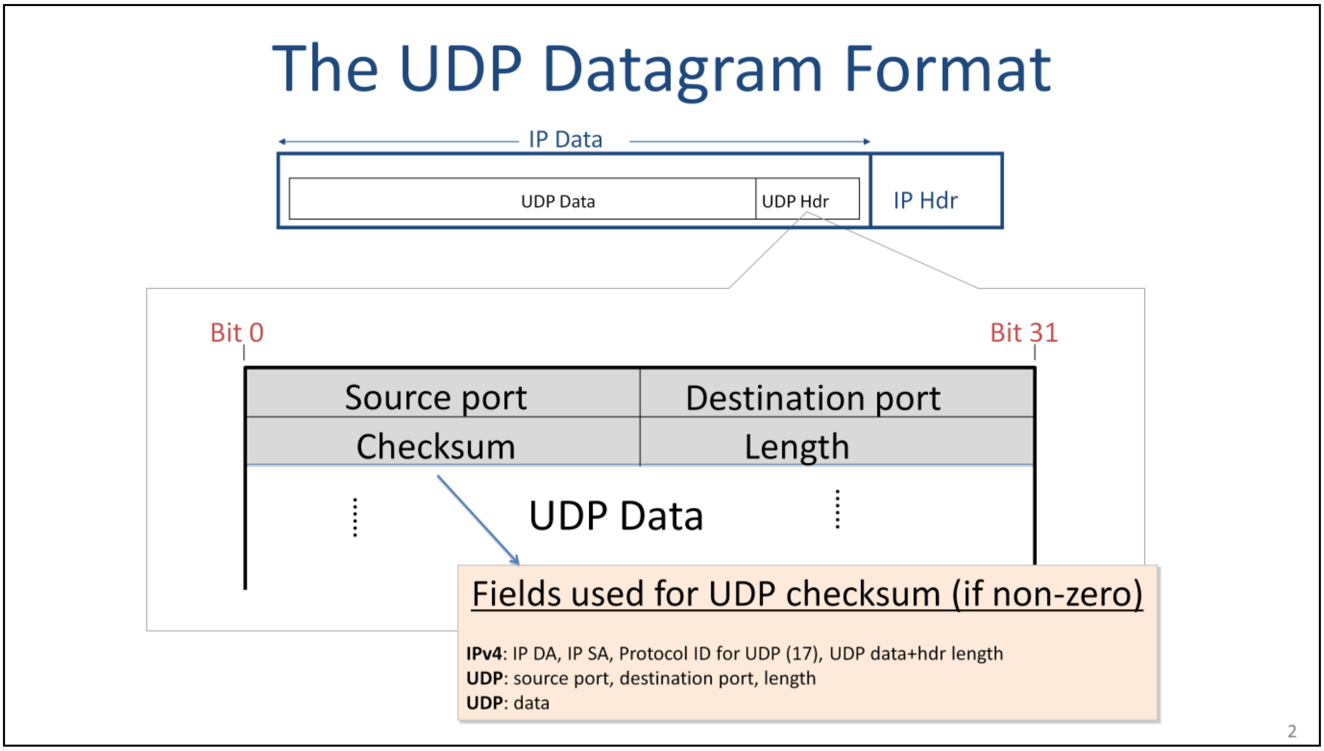


CS144
An Introduction to Computer Networks

What the Internet is
The UDP Service Model

Nick McKeown
Professor of Electrical Engineering
and Computer Science, Stanford University

In this video you are going to learn about the second transport layer UDP, the User Datagram Protocol. UDP is used by applications that don’t need the guaranteed delivery service of TCP, either because the application handles retransmissions in its own private way, or because the application just doesn’t need reliable delivery. UDP is much much simpler than TCP, which is why this video is much shorter. All UDP does is take application data and create a UDP datagram, then hands it to the network layer. The UDP datagram identifies the application that the data should be sent to at the other end. That’s about it.



The UDP Datagram Format

IP Data

UDP Data

UDP Hdr

IP Hdr

Bit 0

Bit 31

Source port

Destination port

Checksum

Length

UDP Data

Fields used for UDP checksum (if non-zero)

IPv4: IP DA, IP SA, Protocol ID for UDP (17), UDP data+hdr length
UDP: source port, destination port, length
UDP: data

As we have come to expect, the UDP datagram is encapsulated inside the data field of the IP datagram.

<click> UDP provides a very simple service, as should be clear from the small number of fields in the UDP header.

Unlike TCP that has over ten header fields, UDP has just four.

<click> 1. The source port indicates which application the data comes from. If the far end replies, it will send a datagram with this port number as the destination so it can find its way back to the correct application.

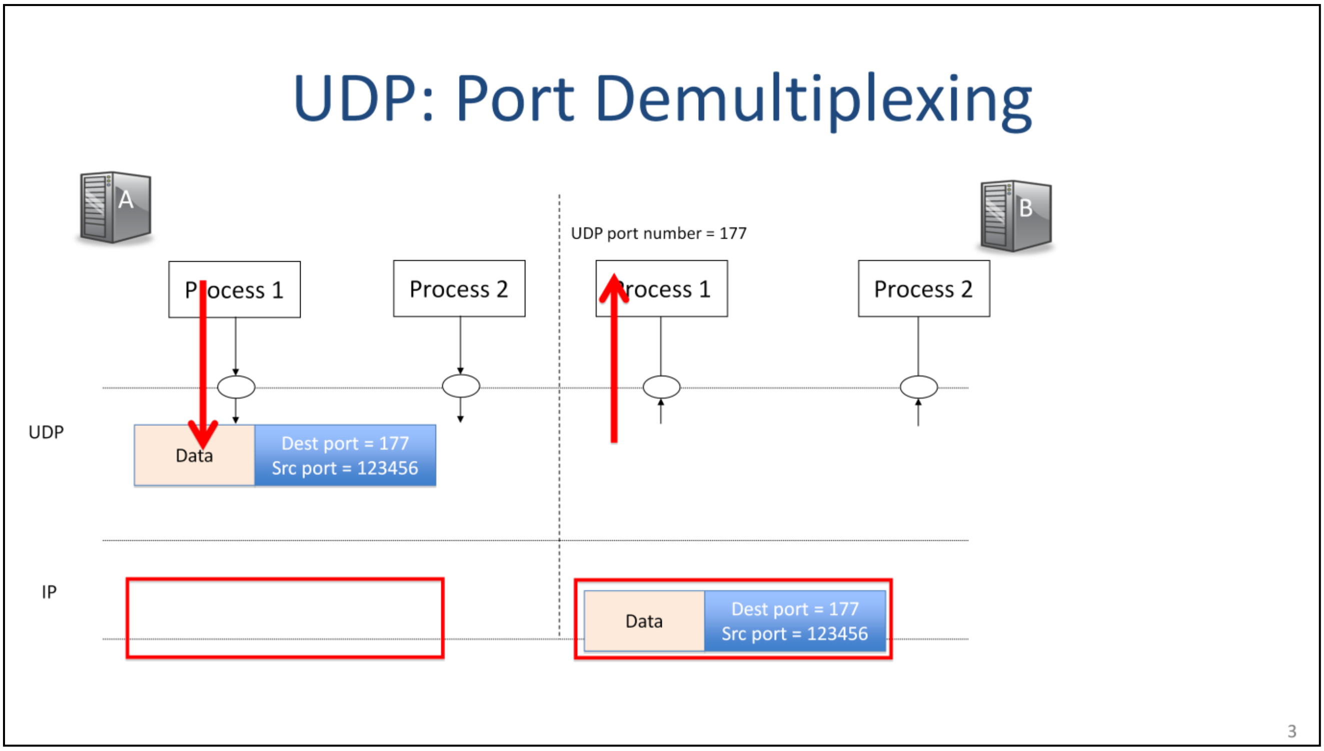
<click> 2. The destination port indicates which application the data should be delivered to at the other end host. The port numbers in UDP serve the same purpose as in TCP - they direct incoming packets to the correct application process.

<click> 3. The 16-bit Length field specifies the length of the whole UDP datagram – header plus data – in bytes. The value must be at least 8 bytes, because that is the length of the UDP header.

<click> 4. The UDP checksum is optional when using IPv4. If the sender doesn't include a checksum, the field is filled with all zeroes. If a UDP checksum is used, then it is calculated over the UDP header and data.

In fact, the UDP checksum calculation also includes a portion of the IPv4 header as well, as shown here <click>. The calculation includes the IP source and destination addresses and the protocol ID which has the value of 17 and tells us that the IP datagram carries UDP data. You might be wondering why the UDP checksum includes part of the IP header – doesn't that violate the clean separation of layers? Yes, it does. The rationale for violating the layering principle and using information from the layer below is that it allows the UDP layer to detect datagrams that were delivered to the wrong destination.

In summary, the UDP header is small, because the service it offers the application is very simple. It provides a simple message protocol for sending data from an application on one host that may or may not be delivered to an application on a remote host.



UDP: Port Demultiplexing

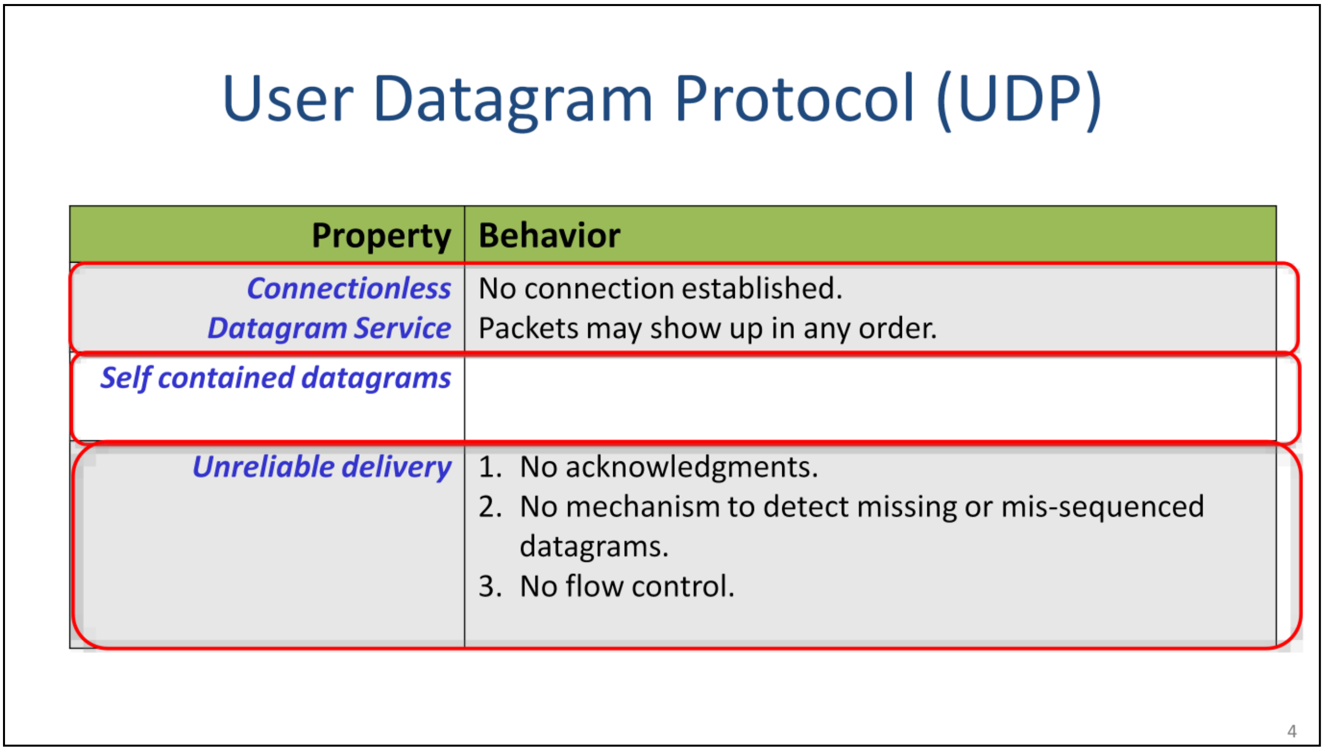
Port numbers in UDP work the same way as in TCP.

<click> If process 1 on Host A has data to send to process 1 on Host B that uses port 177, the data is placed into a new UDP datagram with destination port 177. Host A adds its own source port number so any replies can be sent to Process 1 on Host A.

<click> The datagram is encapsulated in an IP datagram and sent to Host B.

<click> Host B removes the UDP datagram and directs the data to Process 1 <click>

It is useful to think of UDP as merely a Demultiplexing mechanism to divide up the stream of UDP datagrams and send them to the correct process. In fact, some people call it the User Demultiplexing Protcol for this reason --- it is essentially all UDP does.



User Datagram Protocol (UDP)

Property	Behavior
Connectionless Datagram Service	No connection established. Packets may show up in any order.
Self contained datagrams	
Unreliable delivery	1. No acknowledgments. 2. No mechanism to detect missing or mis-sequenced datagrams. 3. No flow control.

To sum up UDP's Service Model, we say that it has the following three properties shown in the table.

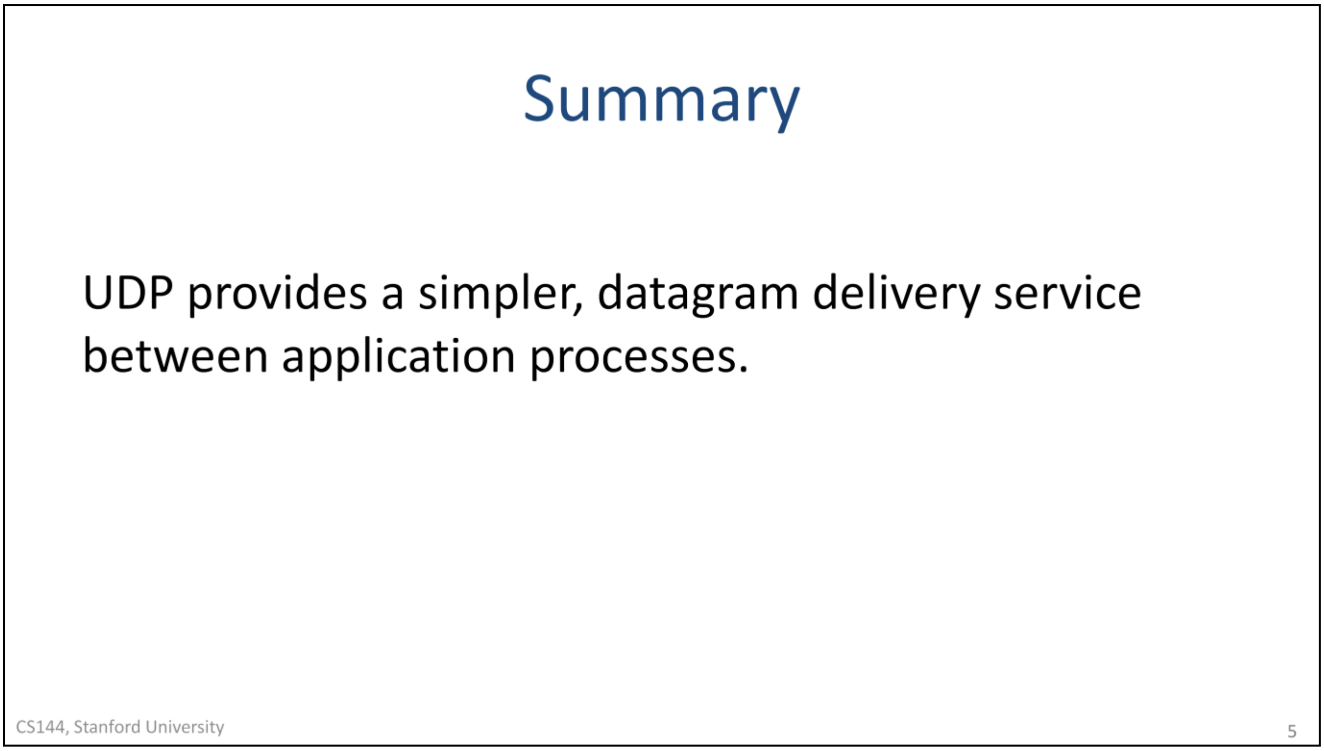
<click> First, it provides a connectionless datagram service. No connection is established, because none is needed – all of the information is self-contained in the datagram <click>. It means packets may show up in an any order, so if the application cares about in order delivery, it will need to resequence the data itself.

<click> UDP is an unreliable delivery service. It doesn't send any acknowledgments to let us know data reached the other end. It has no mechanism to detect missing datagrams. If an entire datagram is dropped along the way, UDP will not inform the application, and it will not ask the source to resend the datagram. However, the application might choose to ask for the data again by itself, essentially building a retransmission mechanism on top of UDP. Early versions of NFS, the network file system did exactly this. They decided they didn't want to use the sliding window used by TCP, so they created their own inside the application.

UDP should sound very much like the service provided by the IP layer. That's because UDP is offering little more than a simple wrapper on top of the IP layer, with the means to direct the arriving data to the correct application at the other end.

So why do we have UDP? It is used by applications that don't need reliable delivery, such as simple request-response applications. DNS – the domain name system used by the Internet to turn a hostname into an IP address uses UDP because the request is fully contained in one UDP datagram. You'll learn how DNS works later, but for now you just need to know that if we send a DNS request containing a hostname, the DNS server will respond with an IP address we can use to send IP datagrams to the host. If the request is successful, then using UDP is lightweight and fast – there is no need to setup a connection before making the query. If the request is unsuccessful, it simply times out and is resent. This makes DNS simple and fast most of the time. The DHCP or Dynamic Host Configuration Protocol also uses UDP. DHCP helps a new host find out its IP address when it joins a network. Your laptop probably uses DHCP when it connects to WiFi. We'll learn more about DHCP later, but it's enough to know here that DHCP is also a request-response application making a single, self-contained request in one UDP datagram. The Network Time Protocol or NTP also uses UDP for the same reason.

Some applications use UDP because they have their own special needs for retransmission, congestion control, in-sequence delivery. A few real-time streaming audio and video services use UDP. This is much less common that it used to be, because most video and audio streams of http today, which uses TCP instead of UDP.



Summary

UDP provides a simpler, datagram delivery service between application processes.

In summary, UDP provides a simpler, datagram delivery service between application processes.

<The End>