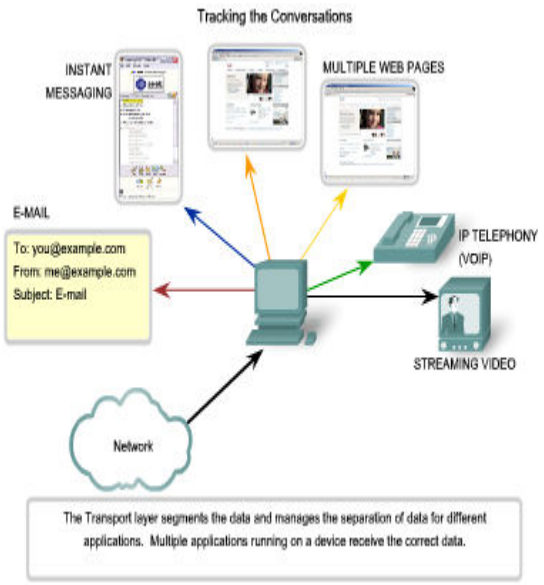
# Transport Layer: Separating Multiple Communications



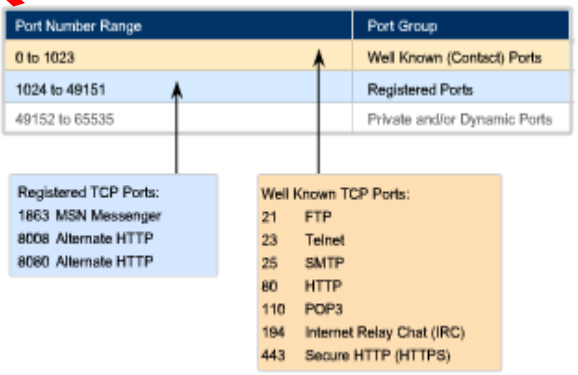
- Consider a computer that is simultaneously receiving and sending e-mail and instant messages, viewing websites, and conducting a VoIP phone call.
  - –Each of these applications is sending and receiving data over the network at the same time.
  - –However, data from the phone call is not directed to the web browser, and text from an instant message does not appear in an e-mail.
- Users require that an e-mail or web page be completely received for the information to be considered useful.
  - –Slight delays are considered acceptable to ensure that the complete information is received and presented.
- In contrast, occasionally missing small parts of a telephone conversation might be considered acceptable.
  - –This is considered preferable to the delays that would result from asking the network to manage and resend missing segments.
  - –One can either infer the missing audio from the context of the conversation or ask the other person to repeat what they said.

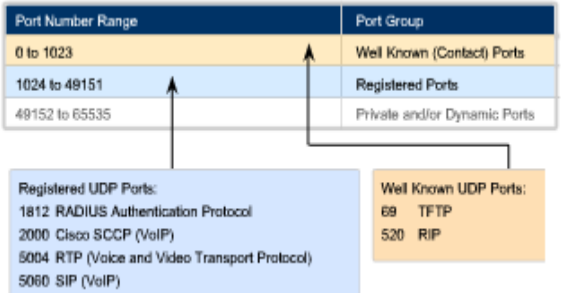


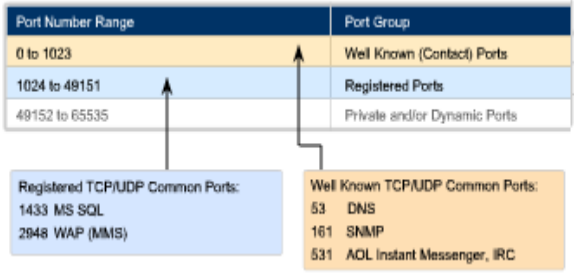
# The IANA assigns port numbers



- Well Known Ports (Numbers 0 to 1023) - These numbers are reserved for services and applications.
  - –HTTP (web server) POP3/SMTP (e-mail server) and Telnet.
- Registered Ports (Numbers 1024 to 49151) - These port numbers are assigned to user processes or applications.
  - –These processes are primarily individual applications that a user has chosen to install.
  - –When not used for a server resource, these ports may also be used dynamically selected by a client as its source port.
- Dynamic or Private Ports (Numbers 49152 to 65535) - Also known as Ephemeral Ports, these are usually assigned dynamically to client applications when initiating a connection.
  - –It is not very common for a client to connect to a service using a Dynamic or Private Port.
- Using both TCP and UDP
  - –Some applications may use both TCP and UDP.
    - •For example, the low overhead of UDP enables DNS to serve many client requests very quickly.
    - •Sometimes, however, sending the requested information may require the reliability of TCP.



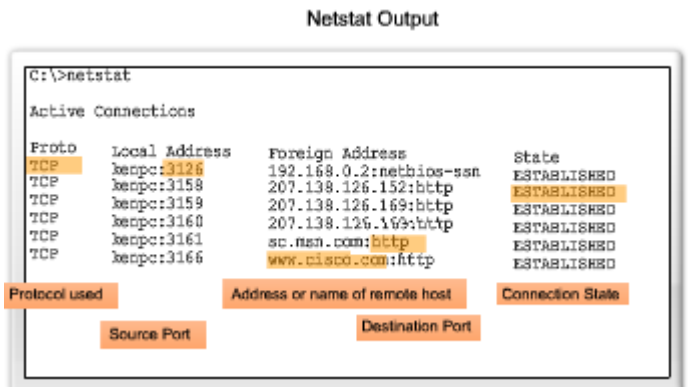



# Port Addressing: netstat command



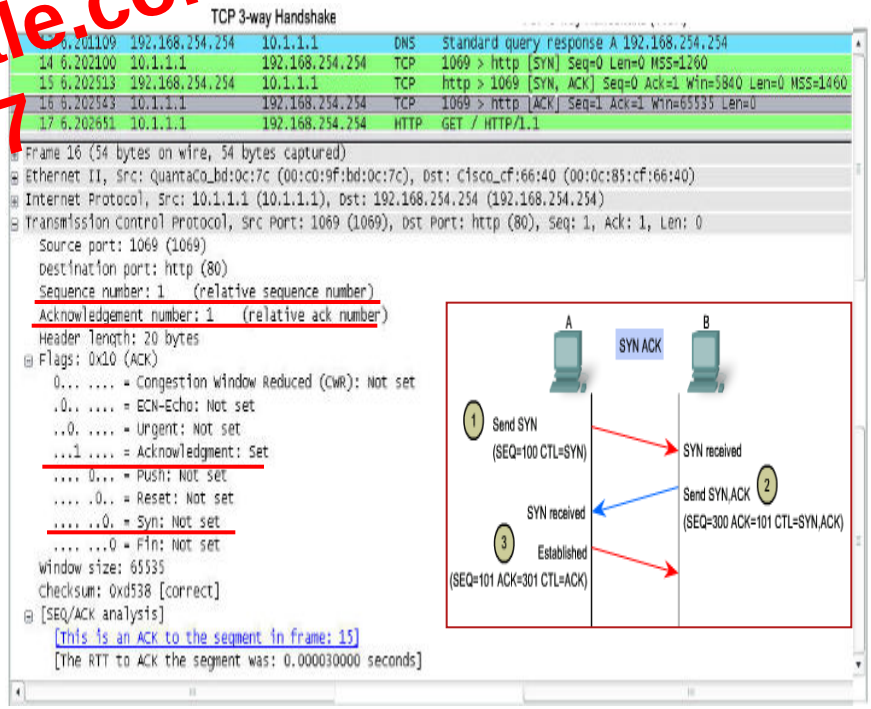
- Sometimes it is necessary to know which active TCP connections are open and running on a networked host.
- Netstat is an important network utility that can be used to verify those connections. Netstat lists:
  - –the protocol in use,
  - –the local address and port number,
  - –the foreign address and port number,
  - –the state of the connection.



# TCP three-Way Handshake - Step 3



- Finally, the TCP client responds with a segment containing an ACK that is the response to the TCP SYN sent by the server.
  - –There is no user data in this segment.
  - –The value in the acknowledgement number field contains one more than the initial sequence number received from the server.
- Once both sessions are established between client and server, all additional segments exchanged in this communication will have the ACK flag set.
- Security can be added to the network by:
  - –Denying the establishment of TCP sessions
  - –Only allowing sessions to be established for specific services
  - –Only allowing traffic as a part of already established sessions




**Protocol Analyzer shows client response to session in frame 16**

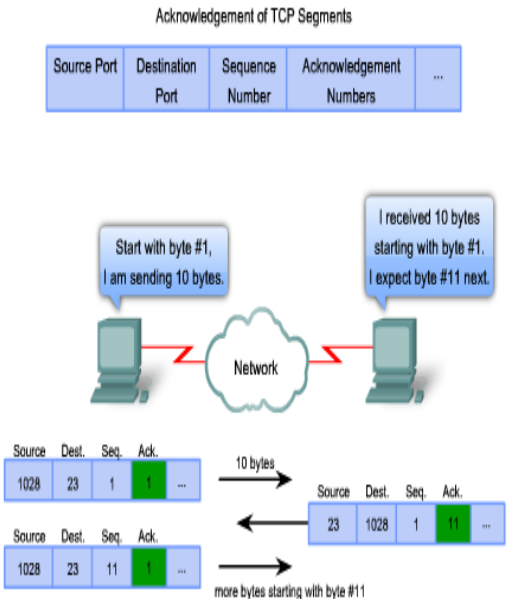
The TCP segment in this frame shows:

- ACK flag set to indicate a valid Acknowledgement number
- Acknowledgement number response to initial sequence number as relative value of 1
- Source port number of 1069 to corresponding
- Destination port number of 80 (HTTP) indicating the web server service (httpd)

# TCP Acknowledgement with Windowing



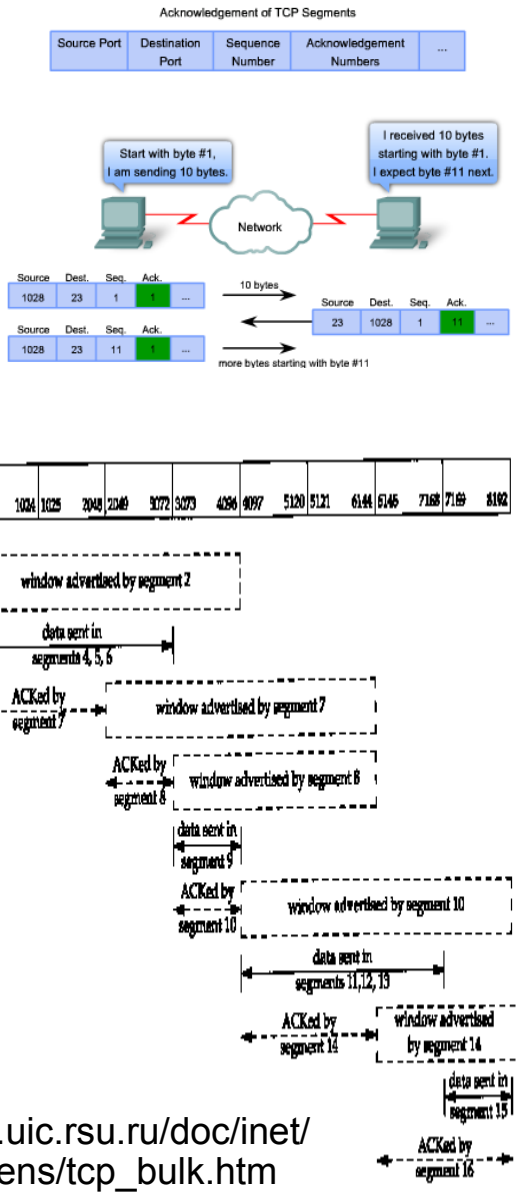
- In the example,
  - –the host on the left is sending data containing 10 bytes of data and a sequence number equal to 1 in the header.
  - –The on the right receives the segment and determines that the sequence number is 1 and that it has 10 bytes of data.
    - •The host then sends a segment back to the host on the left to acknowledge the receipt of this data.
    - •the host sets the acknowledgement number to 11 to indicate that the next byte of data it expects to receive is byte number 11.
  - –When the sending host on the left receives this acknowledgement, it can now send the next segment starting with byte number 11.



# TCP Acknowledgement: Sliding Window



- Looking at this example, if the sending host had to wait for acknowledgement of the receipt of each 10 bytes, the network would have a lot of overhead.
  - –To reduce the overhead of these acknowledgements, multiple segments of data can be sent before and acknowledged with a single TCP message.
  - –This acknowledgement contains an acknowledgement number based on the total number of bytes received.
- For example, starting with a sequence number of 2000, if 10 segments of 1000 bytes each were received, an acknowledgement number of 12001 would be returned to the source.
- The amount of data that a source can transmit before an acknowledgement must be received is called window size.
- [Tony]: What this slide is trying to describe is called: Sliding Window
- See:




http://www.rhyshaden.com/tcp.htm

http://www.uic.rsu.ru/doc/inet/tcp_stevens/tcp_bulk.htm

# TCP Acknowledgement with Windowing

- Length = 1460

No 1: SYN

No 2: SYN. ACK

No 3: ACK

No 4: SEQ = 1

No 5: SEQ = 566

No 6: ACT = 566 (for No. 5)

No 7: SEQ = 566 +1460 = 2026

No 8: SEQ = 2026 + 1460 = 3486

No 9: ACT = 2026 (for No. 7)

No 10: SEQ = 3486 + 1460 = 4946

No 11: SEQ = 4946 + 1460 = 6406

No 12: ACT = 3486 (for No. 8)

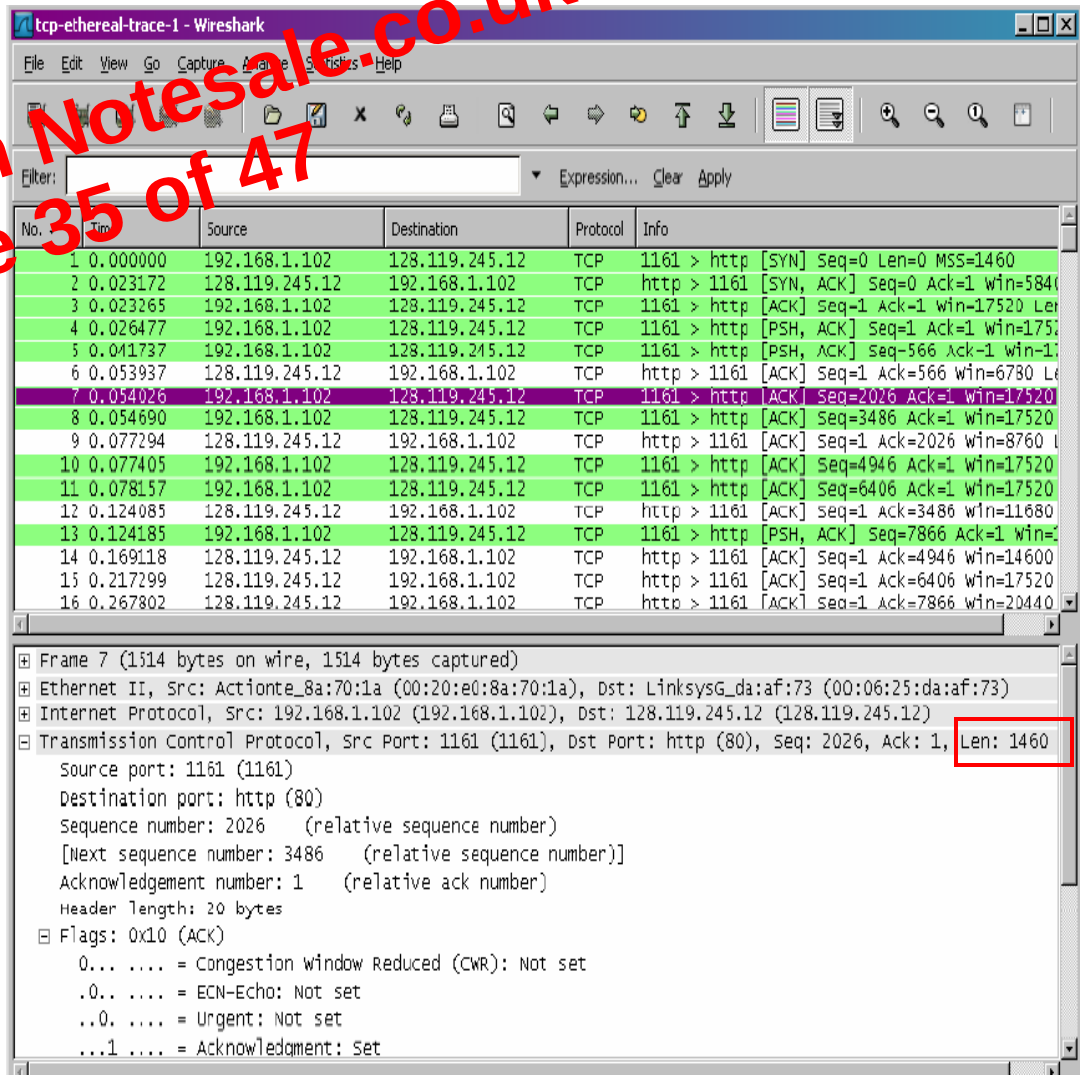
No 13: SEQ = 6406 + 1460 = 7866

No 14: ACT = 4946 (for No. 10)

No 15: ACT = 6406 (for No. 11)

No 16: ACT = 7866 (for No. 13)






zoo.cs.yale.edu/classes/cs433/assignments/assign2/TCP.pdf