# GIVING DATA TO THE APP

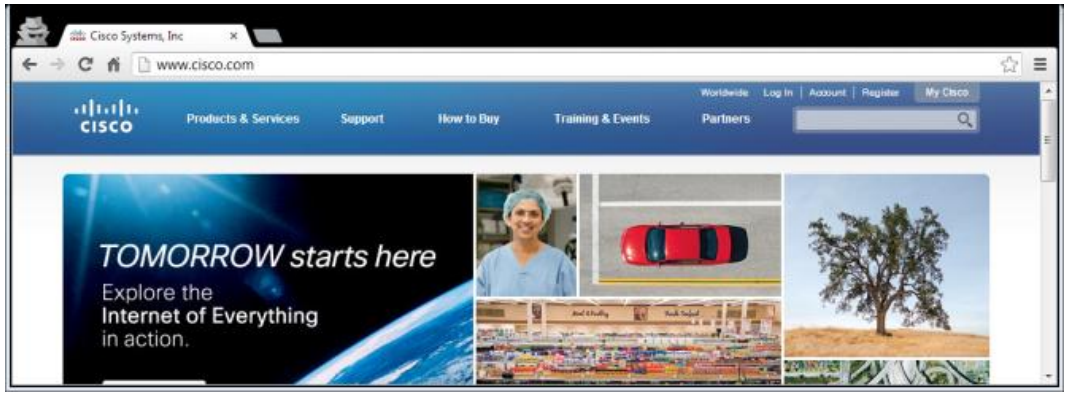
Ensuring that the data arrives at the correct application

# Host to Host transport

- Till now, we are able to transmit data from one device to another via hardware and logical addressing
- Once the data arrives the destination device must deliver it to the correct process,
- Each application running, be it on a router, a firewall or a computer as a separate process for each instance of a program,
- We need to find a way to get the data to the correct instance

# Reversing the problem

- Look at the problem from another angle, you open your browser, which launches “browser.exe”
- You then ask this browser to open cisco.com, which is translated by the command “give me your index page”
- Your browser say to the operating system, “please deliver this command/message to www.cisco.com”



# Sending the command

- We now have the following information:
  - The Source (The browser window)
  - The Destination (www.cisco.com)
  - The message (Give me your index page)
- On the Cisco side, the website is run by a web server application, it's job is to give you the pages requested
- This application, that run just as our browser, is referred to by a number, in a web server's case this is 80
- This number enables us to send a message to that specific process (eg. webserver.exe) on the Cisco server

# Sending the command Cont'd

- We now have the following information:
  - The Source (The browser window)
  - The Destination (www.cisco.com)
  - The message (Give me your index page)
  - The destination process identification number (80)
- So we could now send the request, but how would the reply get back to our browser window ?
- Our computer is going to select a number to identify the "browser.exe" process that generated that request
- This number will enable our computer to display the result in the correct process

# Sending the command Cont'd

- We now have the following information:
  - The Source (The browser window)
  - The Destination (www.cisco.com)
  - The message (Give me your index page)
  - The destination process identification number (80)
  - The Source process identification number (randomly picked amongst available numbers in between 49152-65535)

- Our OS then builds the request in the following way

| Source Port | Destination Port | Data |
|---|---|---|
| 2 bytes | 2 bytes | n-bytes |
| random | 80 | Give me your index page |

# Next Step

- The host to host transport layer ensures communication in between target processes at each end of the communication
- Getting the request to the destination device is the logical addressing's job
- The created Data, called Segment, is passed to the next process which will prepend logical addressing information (eg. Cisco.com's IP address)

# Communication Reliability

- We now understand the end to end communication process,
  - First we identify the source and destination processes, (source and destination ports)
  - Then we identify the source and destination logical addressing, (IP addresses)
  - Finally, the source hardware address and the next-hop hardware address are added
- Notice that at no point have we discussed the eventuality of the data not getting to the destination

# Communication Reliability Cont'd

- Reliability is the responsibility of the transport protocol, some protocols ensure data delivery and data retransmission, others don't

- TCP, Transport control protocol, is the most used reliable transport protocol :
  - each segment sent is tracked by a sequence number
  - the receiving host must acknowledge the reception of one segement to receive another
  - The first segment is not sent until the destination as not acknowledge being ready

# Syn, Syn-Ack, Ack

- The Syn, Syn-Ack, Ack process, known as the TCP three way handshake, is the TCP communication set up method

- No data leaves the device unless this process is completed :
  - The source host sends a TCP SYN message to the device it wishes to communicate with
  - The destination replies with a SYN Acknowledgement message (SYN-ACK)
  - The source confirms the end to end communication process by sending an Acknowledgement

# Three-way handshake