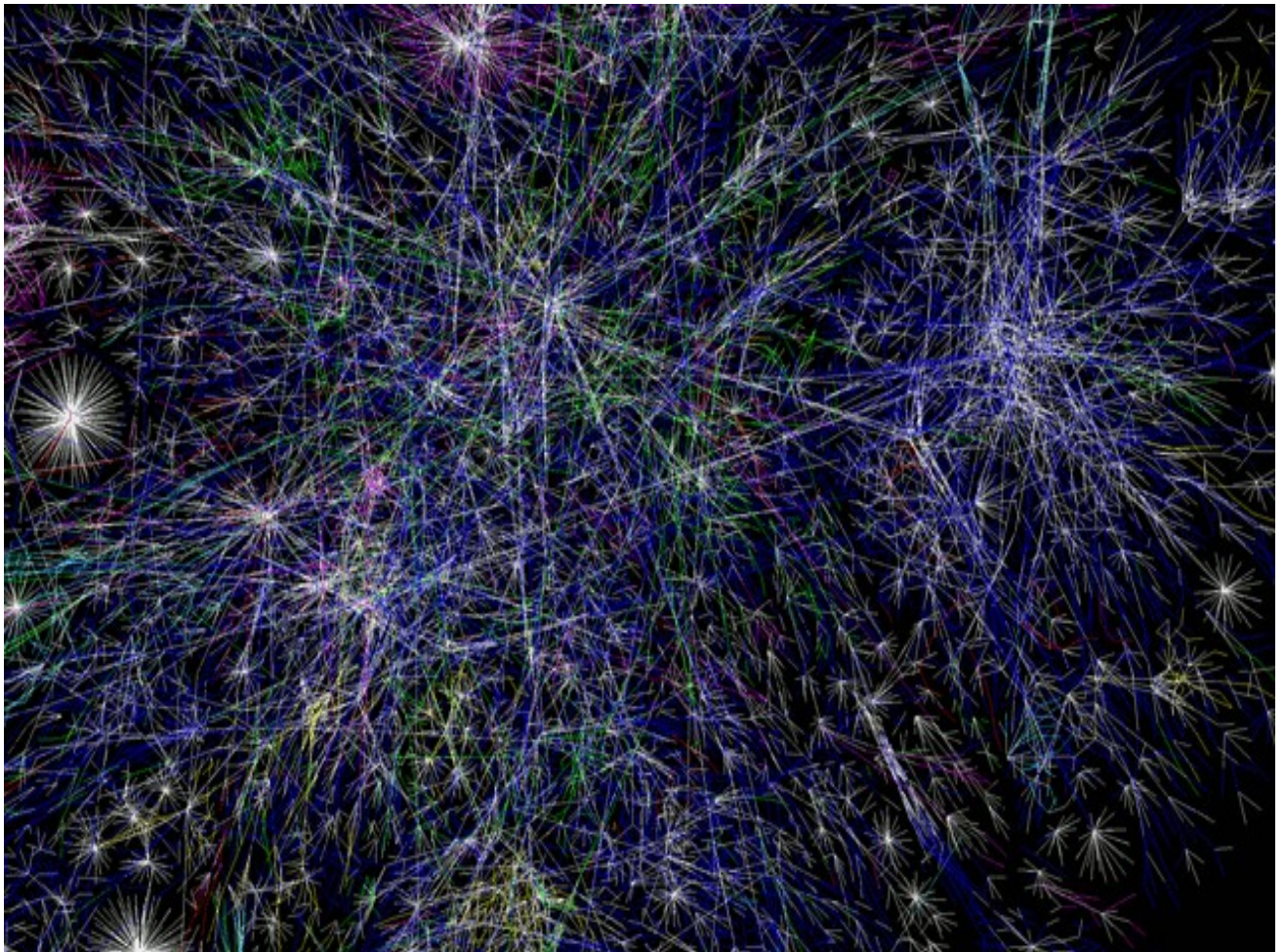
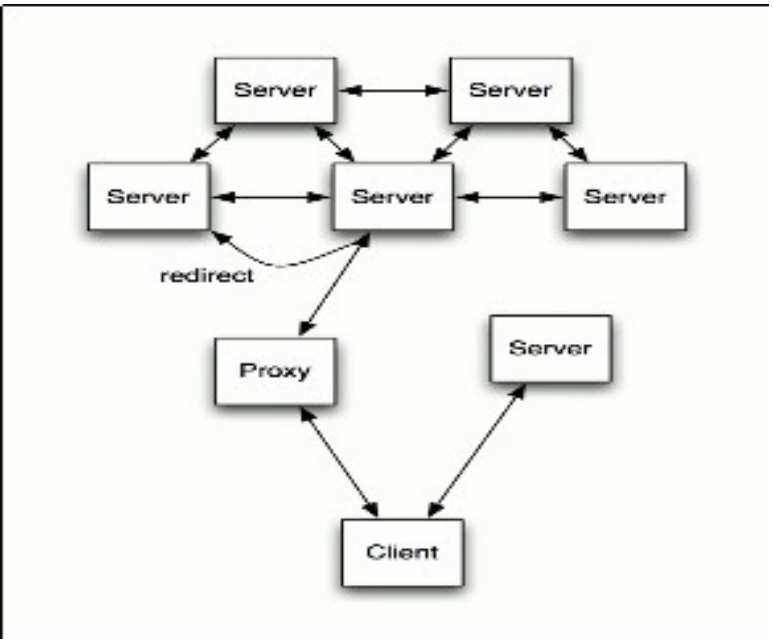
Networking

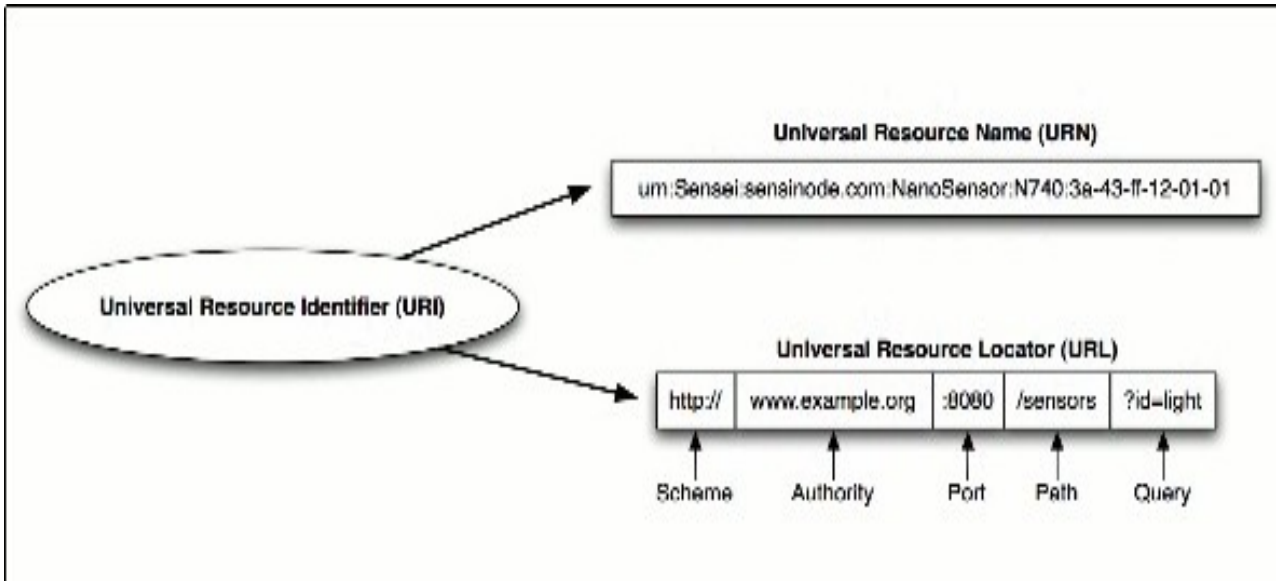
IP
DHCP server
WiFi

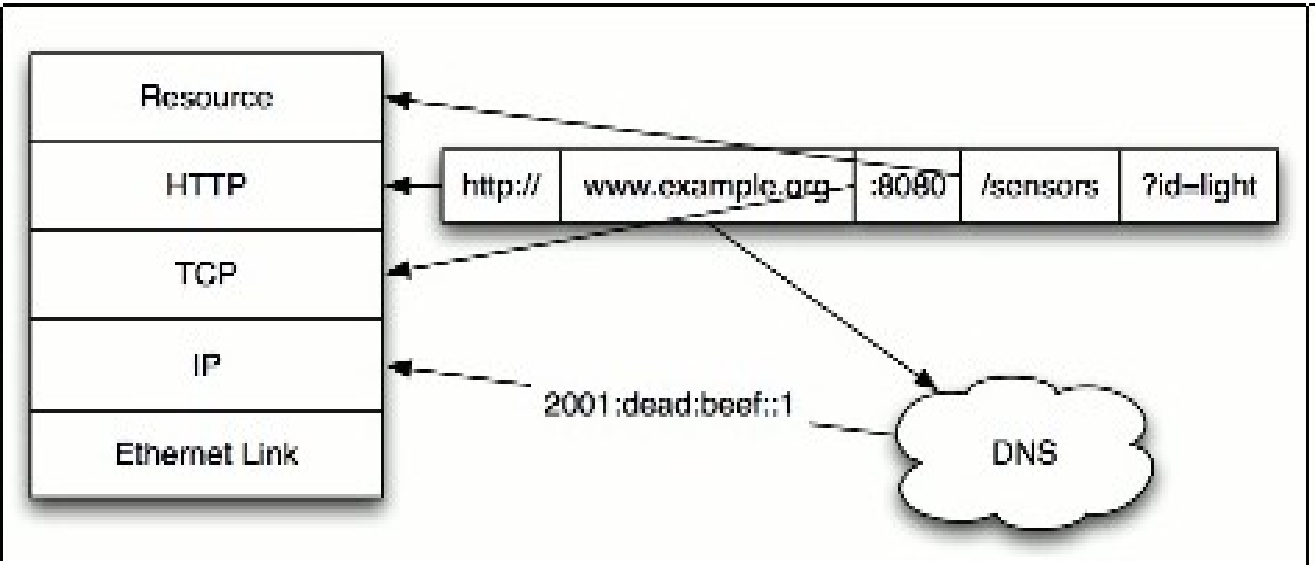


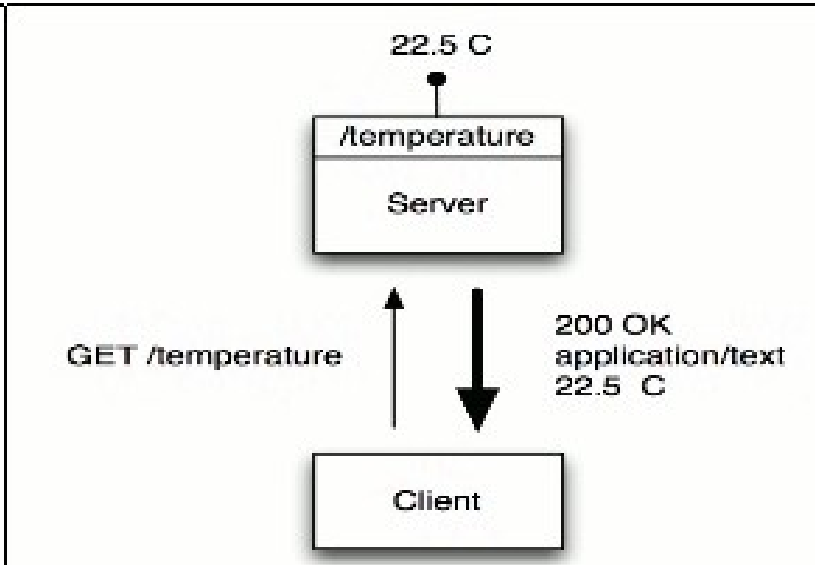
Review: The Web and REST

(from: http://www.slideshare.net/zdshelby/coap-tutorial)







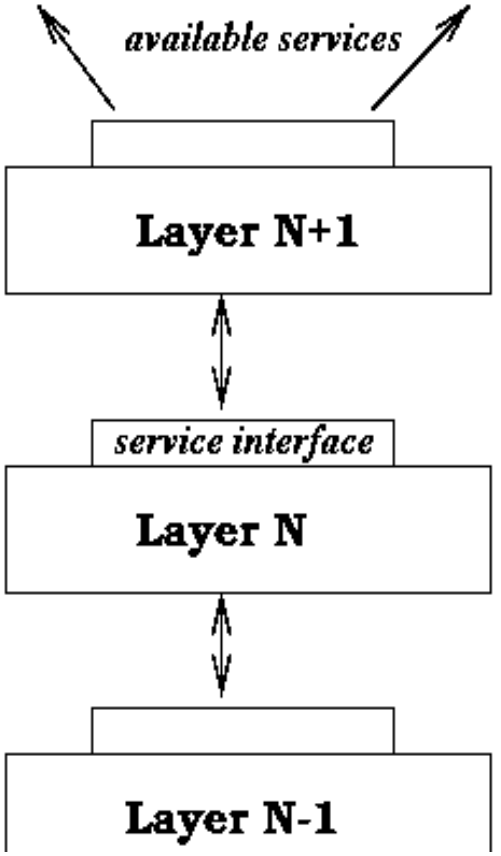


Layered Architecture

(from University of Minnesota, CSCI 4211, Intro to Computer Networks)

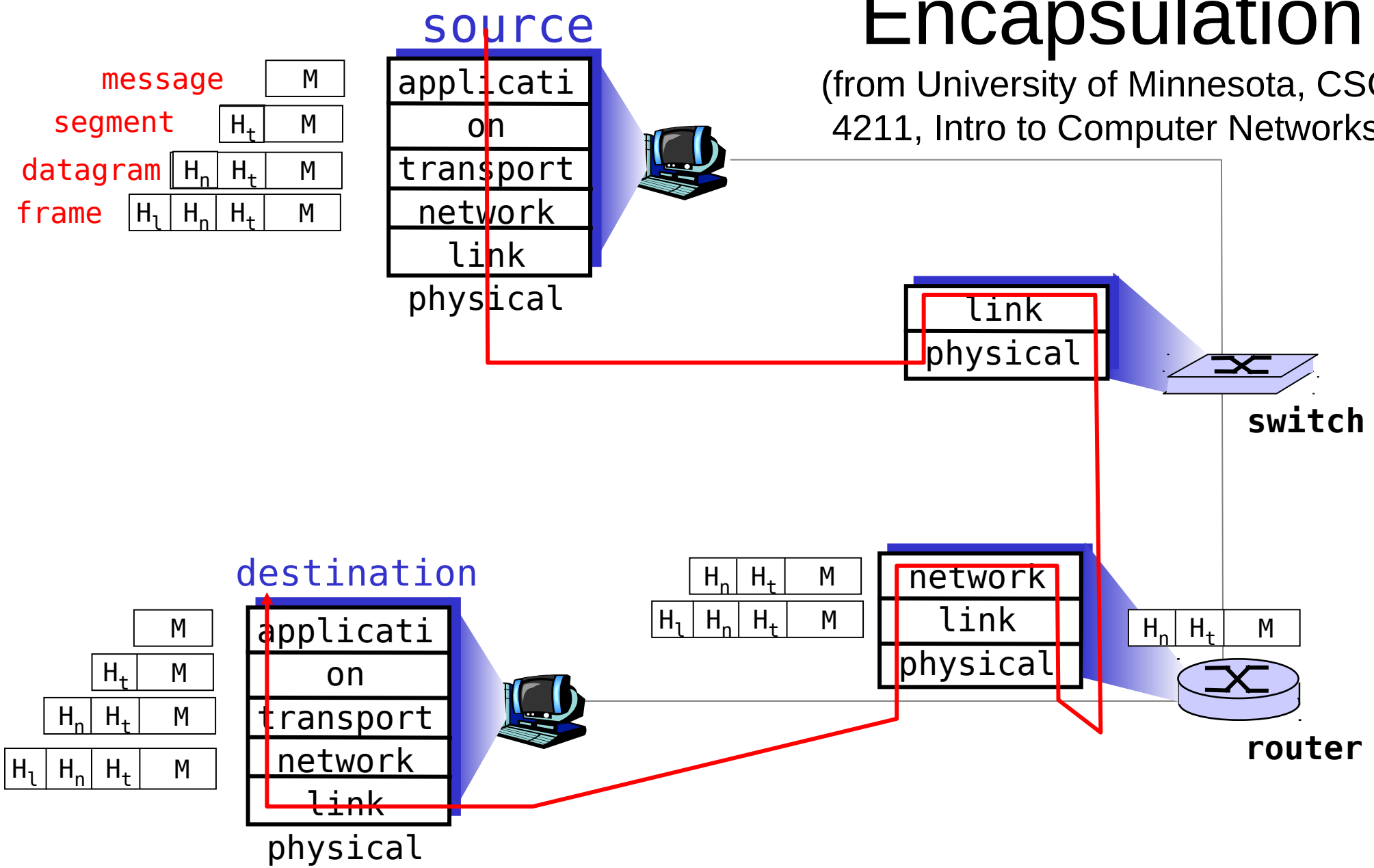
- *Layering* simplifies the architecture of complex system
- Layer N relies on *services* from layer N-1 to provide a *service* to layer N+1
- *Interfaces* define the services offered
- Service required from a lower layer is independent of its implementation
  - Layer N change doesn't affect other layers
  - Information/complexity hiding
  - Similar to object oriented methodology

The OSI model



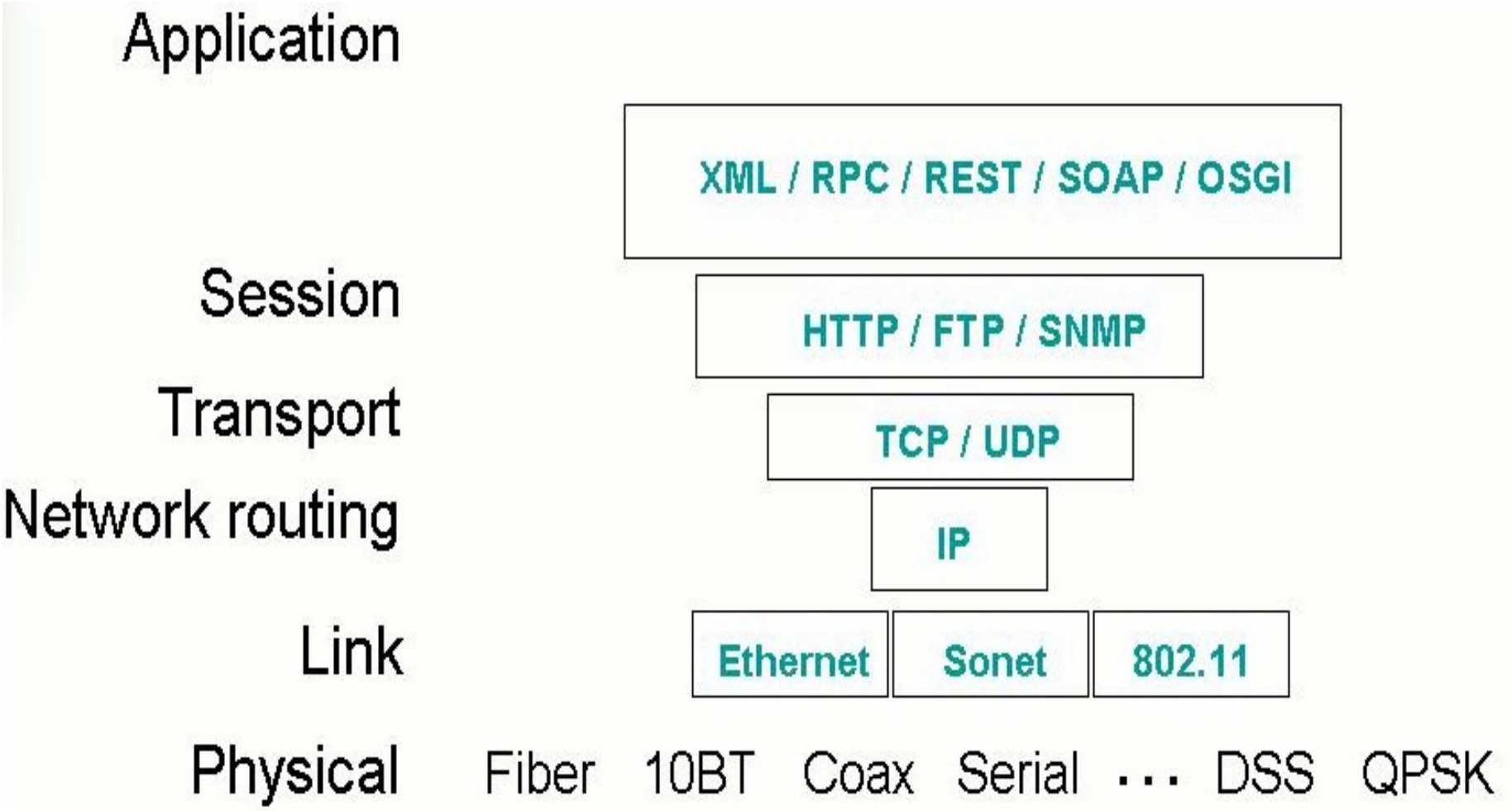
Encapsulation

(from University of Minnesota, CSCI 4211, Intro to Computer Networks)



Network protocol header

The IP layer



- The IP layer provides homogeneous naming and routing over different physical layers, different media access control (MAC) protocols, different frame formats, and different physical media.
- The IP protocol header

Ports and Services

- There is a connection between networking and operating systems
  - the network protocols are implemented in the operating system
- Ports associate services with the applications that provide the service.
  - The port number is the address of the service on the particular server
    - standard port numbers are generally chosen from 1 to 1023
      - http is on port 80; SSH is on port 22
    - Client port numbers are short lived and generally range from 1024 to 5000
    - Port number higher than 5000 are reserved for non-standard servers
  - Use command ***cat /etc/services*** to list services on your machine
- On a machine, a port number coupled with the IP address of the machine is known as a socket.
- A combination of IP and port on both client and server is known as four tuple. This four tuple uniquely identifies a connection.

IP Address

- We think of IP addresses as identifying machines
  - e.g., www.unca.edu translates to IP address 152.18.68.26
- But a machine can be attached to several network links through interfaces
  - Each interface on a machine has an IP address
  - Use the command ***ifconfig*** to see the interfaces on your computer

IP Tools

- IP includes a collection of low level tools/protocols including the Internet Control Message Protocol (ICMP):
  - ping, traceroute, netstat, route, nc

In-Class Experiment 1

- A chat network between two RPis using sockets in python
  - Download network.py and chat.py:

    wget www.cs.unca.edu/~bruce/Fall14/networks.tar.gz

    tar xfvz networks.tar.gz
  - Follow these instructions to configure a static IP for each Pi
  - Connect pairs of Pis with a single ethernet cable
  - cd to the networks directory & make one Pi the server by typing:

    python3 chat.py
  - cd to the network directory on the other Pi and make it the client by typing:

    python3 chat.py <server IP address> (e.g., 192.168.0.2)
  - Type messages on either Pi, and they will appear on the other Pi's screen when you press the enter key
  - When done, change the network configuration back to a dynamic IP address as shown in the "Clean up" section of the guide

In-Class Experiment 2

- Using a DHCP server on an isolated network
  - Decide which Pi will be the DHCP server in stall software

    sudo apt-get install dnsmasq
  - Make the DHCP server Raspberry Pi have the static IP address: 192.168.0.1 by editing /etc/network/interfaces
  - Restart the network:

    sudo service networking restart
  - Disconnect the switch from the internet
  - Configure the DHCP server software, dnsmasq, that was installed earlier:

    cd /etc

    sudo mv dnsmasq.conf dnsmasq.default

    sudo nano dnsmasq.conf
  - You should now be editing a blank file. Copy and paste the following into it:

    interface=eth0

    dhcp-range=192.168.0.2,192.168.0.254,255.255.255.0,12h
  - Save the file and restart the server:

    sudo service dnsmasq restart
  - Test the network and then clean-up as described in on-line exercise
- More on DHCP

Connecting to the UNCA WiFi network, Part 1

- Install wicd

  sudo apt-get update

  sudo apt-get install wicd-curses

- Shutdown Pi

  sudo halt

- Disconnect the ethernet cable & connect the EDIMAX adapter
  - Check that drives exist using command: ***lsusb***
  - The wifi adapter should be in the list
  - Check ***ifconfig*** and record MAC address
- Start wicd-curses and enter wlan0 as wireless interface
  - ***sudo wicd-curses***
  - enter P and enter wlan0 in Wireless Interfaces field
  - Press F10 to save

Connecting to the UNCA WiFi network, Part 2

1. Establish a connection to the ***UNCA_Setup*** network using wicd-curses:
   - start X and open a browser (Midori should work)
   - move to a terminal window WITHOUT X using Ctrl-Alt-F4 (press simultaniously)
   - in terminal without X start wicd-curses

     sudo wicd-curses
   - Use the arrow keys to select the UNCA_Setup network
   - Enter C to connect to that network
2. Use Ctrl-Alt-F8 (press simultaniously) to go back to X
   - load wifi.unca.edu in browser
   - follow directions to get passkey for UNCA_PSK

- An on-line wicd-curses tutorial

Connecting to the UNCA WiFi network, Part 3

- After initial connection when you get your passkey establish a connection to the UNCA_PSK network using wicd-curses:
  - DO NOT INITIATE X
  - ***sudo wicd-curses***
    - Select the UNCA_PSK network
    - Use → key (i.e., configure) to enter the passkey
    - Enter C to connect after entering the passkey
    - Enter Q to leave wicd-curses
- Check connection using ***ifconfig***
  - Should see an IP address under wlan0