

Making the Connection

OR: How does the internet work?

Key Learning Goals

After the networks unit, you will be able to:

- use abstraction to manage complex processes
- describe key abstraction layers in the Internet
- trace the process of communication for a web page request and e-mail message
- describe and resolve key problems for routing information across a point-to-point network
- describe and resolve key problems for sending information over a broadcast network

Bonus: many pieces that make the Internet work
+ our first brush with CS “graphs”

what are the steps in sending a letter?

levels of abstraction

- different *levels of abstraction* are useful in thinking about communication systems
- the higher the level of abstraction, the more details of the system that are hidden

levels of abstraction in the postal system

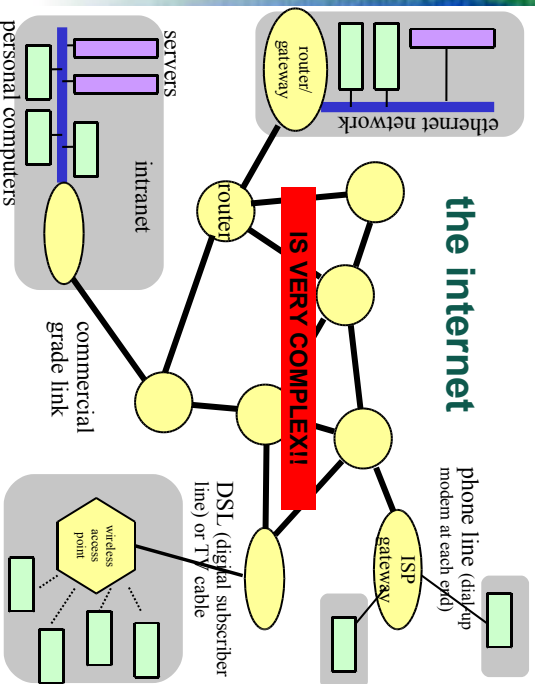
- “*you and me*” *level*: a letter, labeled with address or recipient, is placed in a postbox, arrives at destination
- “*mail carrier*” *level*: a letter is brought to a distribution centre, bar-coded, and sent to the receiving centre
- “*wings and wheels*” *level*: bins of letters are transferred by a combination of plane, truck, and train to their destination centre

Key Learning Goals

After the networks unit, you will be able to:

- use abstraction to manage complex processes
- describe key abstraction layers in the Internet
- trace the process of communication for a web page request and e-mail message
- describe and resolve key problems for routing information across a point-to-point network
- describe and resolve key problems for sending information over a broadcast network

Bonus: many pieces that make the Internet work
+ our first brush with CS “graphs”



levels/layers of abstraction in computer networks

- **application layer:** web browsers, e-mail readers, and other "applications" provide ways to send and receive data
- **transport layer:** puts the data into the right format for routing by the network.
- **network layer:** routes data from source to destination
- **link/physical layers:** move data along one step through the network

levels/layers of abstraction in computer networks

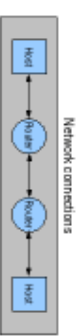
- **application layer:** web browsers, e-mail readers, and other "applications" provide ways to send and receive data
- **transport layer:** puts the data into the right format for routing by the network.
- **network layer:** routes data from source to destination
- **link/physical layers:** move data along one step through the network

what are the steps in accessing a web page?

We use abstraction layers to manage the complexity of communication over the Internet!

abstraction layers of the Internet

Hosts (our computers) communicate through a chain of routers (network computers).



Each level pretends that it is the **whole** Internet to the level above.

Each level ignores the details that make the lower levels work.

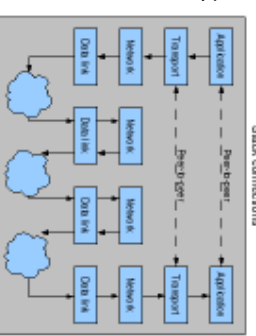


Image released under GNU Free Documentation License v1.2 by the creator.
http://en.wikipedia.org/wiki/Wikipedia:Text_of_the_GNU_Free_Documentation_License

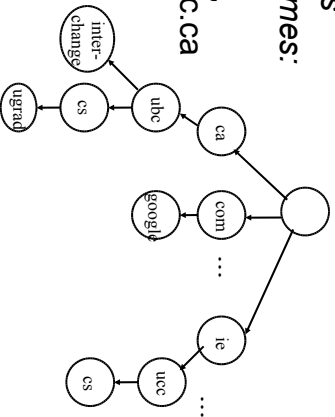
application layer

- What are common communications tasks on computers?
 - send e-mail
 - open web pages
- Addresses are key to performing these tasks:
 - e-mail addresses identify people
 - URLs identify web pages
 - domain names identify computers

domain names

- computer names are hierarchically grouped into *domains*

- **example names:**
cs.ubc.ca,
google.com,
ugrad.cs.ubc.ca



e-mail addresses

- two parts, separated by “@”:
 - userID (resolved at the receiving computer)
 - destination domain name address (resolved at the sending computer)
- **example:** *kkyin@cs.ubc.ca*

URLs (uniform resource locators)

- a URL addresses a web page
- **example:** <http://www.google.ca/index.html>
 - *http* refers to a protocol for transferring files
 - www.google.ca is a ... what?
 - *index.html* is a **file name**

levels/layers of abstraction in computer networks

- *application layer:* communicate by high-level addresses like e-mail addresses and URLs
- **transport layer:** puts the data into the right format for routing by the network.
- *network layer:* routes data from source to destination
- *link/physical layers:* move data along one step through the network

where is kkyin@cs.ubc.ca?

- kkyin is a user ID; the computer addressed by *cs.ubc.ca* will figure it out for us
- *cs.ubc.ca* is an address... but it's designed for humans

The transport layer switches from human-readable **domain names** to machine-friendly “**IP addresses**”

an IP address

- identifies nodes (computers) on a network
- consists of four numbers, ranging from 0-255, such as 128.95.1.4
 - future version will have 16 numbers... why?
- may be permanent, or assigned temporarily (e.g. DHCP) by an Internet Service Provider (ISP)

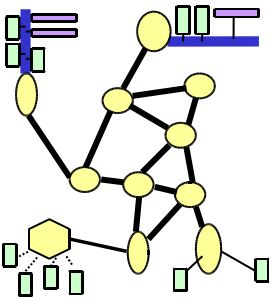
domain name servers: bridging the application and transport layers

- domain name servers (DNSS) keep a directory connecting domain names to IP addresses
- every computer connected to the internet knows the IP address of its nearest DNS
- this DNS is used to *resolve*, or translate, a domain name to an IP address
- DNS's need to be constantly updated

levels/layers of abstraction in computer networks

- *application layer*: communicate by high-level addresses like e-mail addresses and URLs
- *transport layer*: changes domain names into IP addresses (among other things)
- *network layer*: **routes data from source to destination**
- *link/physical layers*: move data along one step through the network

network layer



- the internet is a collection of *nodes* and *links*
- data is routed along links from node to node in the network

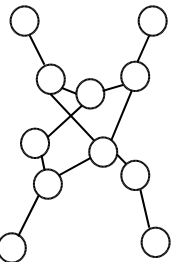
Key Learning Goals

- After the networks unit, you will be able to:
- use abstraction to manage complex processes
 - describe key abstraction layers in the Internet
 - trace the process of communication for a web page request and e-mail message
 - **describe and resolve key problems for routing information across a point-to-point network**
 - describe and resolve key problems for sending information over a broadcast network

Bonus: many pieces that make the Internet work + our first brush with CS “graphs”

point-to-point routing

- The network layer communicates “point-to-point”, like the postal network:
- computers send information to their neighbours, which send information to their neighbours, ...
 - somehow, information gets from one computer across the network to another



designing a routing scheme

- How to choose a good route between two points in the network?
- How can nodes “know” where to send a message?

problems with simple scheme?

solutions?

Key Learning Goals

After the networks unit, you will be able to:

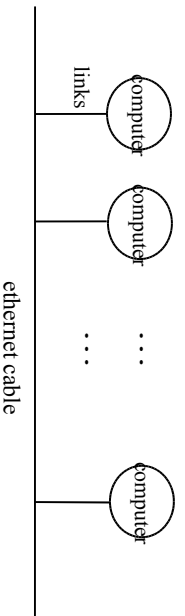
- use abstraction to manage complex processes
- describe key abstraction layers in the Internet
- trace the process of communication for a web page request and e-mail message
- describe and resolve key problems for routing information across a point-to-point network
- **describe and resolve key problems for sending information over a broadcast network**

Bonus: many pieces that make the Internet work + our first brush with CS “graphs”

broadcast systems

- Unlike a point-to-point system, in a *broadcast system*, each message is delivered to all recipients within range, e.g., voice, radio.
- Norm Abramson at U. Hawaii linked together a computer and terminals on the Hawaiian islands using a radio channel in 1970.
- Today, *ethernet channels* (wires or optical fibres) link together computers in a small area.

the ethernet



- the ethernet provides broadcast communication
- the *party protocol* implements point-to-point communication on the ethernet

a simple party protocol

1. **repeat**
2. send packet on channel
3. listen for packet on channel
4. **if** packet is detected on channel **then**
 5. you are done!
 6. **else** (noise is detected on channel)
 7. wait a random number of time steps
 8. **until** you are done

observations on party protocol

- Randomness helps! (Why?)
- It works well when there is little traffic on the channel.
- There has been much research on improved protocols that can do better in the face of high traffic – ideas?

levels/layers of abstraction in computer networks

- *application layer*: communicate by high-level addresses like e-mail addresses and URLs
- *transport layer*: changes domain names into IP addresses (among other things)
- *network layer*: finds a way to move data from the source computer to the destination
- *link/physical layers*: move data along one step through the network

abstraction layers of the Internet

Message to domain name: cs.ubc.ca

Sent to IP address: 142.103.6.5

Routed through the network

Transferred as electrical signals (or light or radio waves or...) from one router to the next.

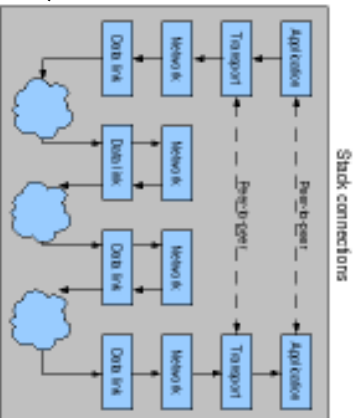


Image released under GNU Free Documentation License v1.2 by the creator.
http://en.wikipedia.org/wiki/Wikipedia:Text_of_the_GNU_Free_Documentation_License