media technology

course: Web Technology

Maarten Lamers
maarten@proton.nl
www.maartenlamers.com/WT2006/

overview

- mandatory course for students Media Technology, in English
- based on previous course by Joost van Dijk

- introductory level
- you may know about this already, but not everybody does
- experience in surfing and building web pages is not enough

- requirements:
  - knowledge of programming
  - knowledge of HTML is useful

- two lab assignments + final exam
- lab assistance by Amalia Kallergi, starts at February 21
- lab assistance between 14:00 – 17:00h

topics

1. internet and TCP/IP
2. the World Wide Web: HTTP and HTML
3. client side technology: JavaScript
4. XML
5. server side technology: CGI and PHP
6. multimedia, streaming and the internet

Attention: the last class is between 10:00 – 12:00h and 14:00 – 16:00h

today: introduction to the internet and TCP/IP

4

overview

what is...
- a network?
- an internet?
- *the* Internet?
- TCP/IP?
- a client?
- a server?
- the World Wide Web?
- HTTP?

networks: connecting computers

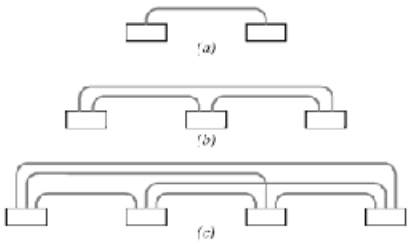
- enables sharing of resources (printers, servers, mail, games, videoconferencing, business, ...)

- a *Local Area Network* (LAN) is a computer network covering a small local area, like a home, office, or group of buildings, in which the computers connect via a shared physical medium

- many possible physical media:
  - direct cable
  - telephone line
  - wireless connection
  - trained monkeys running around with tapes in their hands
  - etc...

connectivity

direct connections between each computer only work for small number of machines

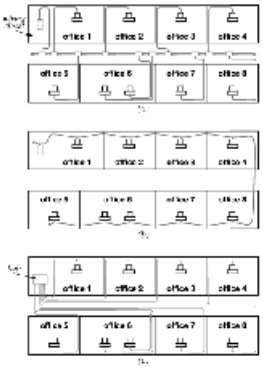


LAN: Local Area Network

actually connects nearby computers e.g. in an office

for example *Ethernet*, but there are also other LAN technologies

all computers on a LAN have a *hardware address* that identifies them, so that they can send messages to one another



packet switching

most networks use *packet switching* to minimize delays

- data is split up and sent in discrete chunks (packets)
- each packet is individually routed to its destination
- multiple data streams can be sent simultaneously by interleaving packet streams

not all packets will necessarily follow the same route, or arrive in similar times

think of small postal cars racing down shared highways, each carrying one box (packet) with data. Each packet may only be a small chunk of a larger data transfer

connecting between networks

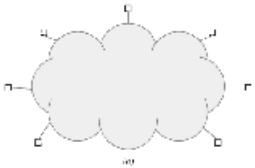
idea:
resource sharing is improved even further by connecting different networks!

but, different LAN technologies are incompatible:
- different addressing schemes
- different packet sizes
- different frame formats, etcetera

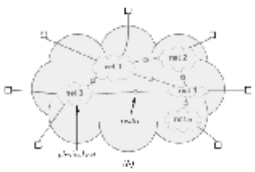
ad hoc solutions exist, but a better solution is to build a *virtual* network on top of *physical* ones (LANs): an *internetwork*

internet (internetwork) concept

(a) the illusion of a single network



(b) the underlying physical network structure



an example internetwork: SURFnet5

SURFnet organization provides a very high quality internetwork between local networks of Dutch higher educational institutions

the Internet

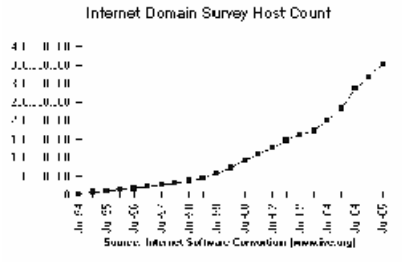
the world's largest public internetwork

'computers' can be...

- PC's, mainframes, PDA's, mobile phones, microwave ovens, radio's, soda machines, trained monkeys, etc...
- running Windows, Unix, Linux, Mac OS X, etc...
- connected by Ethernet, Token Ring, FDDI, ADSL, ATM, Frame Relay, etc...

the Internet

- started as a research networks connecting 4 computers
- has shown exponential growth,
  see the Internet Domain Survey at www.isc.org



how to make an internet

to create the illusion of a single network you need:

1. universal addressing
   to mask differences in underlying addressing schemes
2. routing packets across networks
   to forward packets to neighboring networks
3. handling different packet sizes
   to mask differences in underlying maximum packet sizes

all this is handled by the **Internet Protocol (IP)**

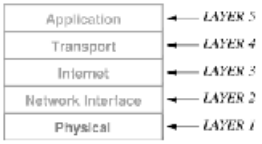
a *protocol* is just a set of standards (rules) that describe how to communicate, so that you and others know what to expect

protocol stack

a *protocol stack* is a partitioning of communication features, in so-called *layers*

every communication system has many features
each layer only solves problems on its own level of abstraction

each layer tries to perform its own task, depending on the layers below it to do their tasks, and unaware of the layers above it



protocol stack

- *Application Layer*
  e.g. HTTP, task: exchange HTML
- *Transport Layer*
  e.g. TCP, task: provide reliability in data transport
- *Internet Layer* or *Network Layer*
  e.g. IP, task: connect networks into a virtual network
- *Networks Interface* or *Link Layer*
  e.g. Ethernet, task: connect computers
- *Physical Layer*
  the wire, or radio signal, or trained monkeys

IP: Internet Protocol

what does IP do?

1. it provides *addressing*
   - give each machine a unique ID
   - map these onto hardware addresses
2. it provides *routing*
   - glue smaller networks together to form one huge network
   - let data traverse networks one by one
3. it provides *fragmentation*
   - where needed divide packets into smaller parts when they are forwarded

IP: Internet Protocol

important: IP is *connectionless* and *unreliable*

- there is no notion of a connection starting and ending
- data is sent in packets called *IP datagrams*
- packets can get lost, duplicated, delayed or corrupted

IP does not care if packets get lost! If a packet is corrupted it is discarded by IP, otherwise it is accepted.

IP addressing

- an IP address is a 32-bit number
- giving only 4,294,967,295 unique addresses

- what we discuss here is IP version 4 (IPv4)

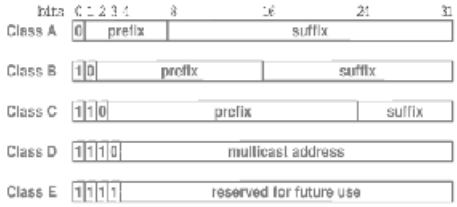
- IPv6 uses 128 bit addresses
- giving us 50.000.000.000.000.000.000.000.000.000 unique addresses for each of the 6.5 billion people alive today

IP addressing

- each IP address contains a network number and host (*machine*) number
- the prefix identifies the network,
  the suffix identifies a host on that network

- length of host and network numbers are variable: divide address space into 5 classes
- address assigned to host is either class A, B or C

IP addressing: structure



IP addressing: dotted address notation

- you can write 32-bit IP addresses using dotted decimal notation: 4 groups of 8 bits (bytes, with values 0 - 255)

32-bit Binary Number	Equivalent Dotted Decimal
10000001 00110100 00000110 00000000	129.52.6.0
11000000 00000101 00110000 00000011	192.5.48.3
00001010 00000010 00000000 00100101	10.2.0.37
10000000 00001010 00000010 00000011	128.10.2.3
10000000 10000000 11111111 00000000	128.128.255.0

- it is nothing more than an easier-to-memorize notation for big (32-bit) numbers. E.g. 2167670272 = 129.52.6.0
- other names are *dotted addressing*, and *dot quad* notation

IP addressing: classes

- since the first bits off an IP address determine its class, the first byte in a dotted address also determines the class

Class	Range of Values
A	0 through 127
B	128 through 191
C	192 through 223
D	224 through 239
E	240 through 255

IP addressing: classes

- the size limits of classes A-C led to large wastes of address space

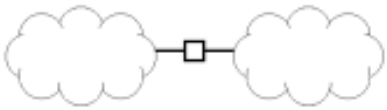
Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

- imagine a company with 300 computers (hosts)
- they would need a class B address
- and as a result, waste 65536 − 300 = 65236 addresses!

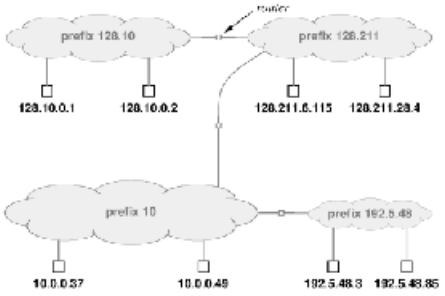
- nowadays additional techniques are used to save IP addresses

internetwork routing

- a *router* connects two or more physical networks
- a router is part of each physical network that it connects
- data packets move from one network to the next network through a router
- the router is responsible for forwarding traffic to the right destination



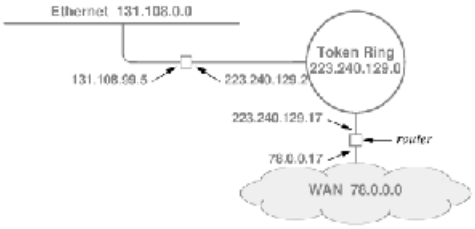
routing



see: http://netbook.cs.purdue.edu/anmtions/anim09_3.htm

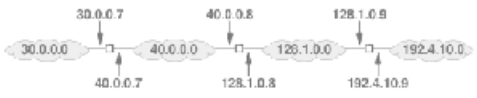
routing

- a router is always part of the networks that it connects



routing tables

- shown is the routing table for the middle router in the diagram
- if a packet is received for a host on the router's network, it is delivered to the destination host directly

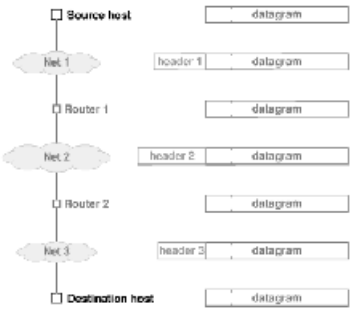


(a)

Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

(b)

hop-by-hop routing



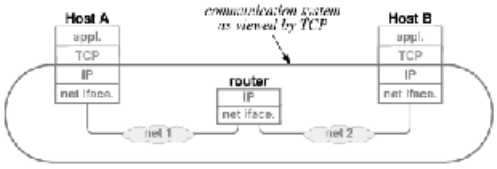
see: http://netbook.cs.purdue.edu/anmtions/anim17_1.htm

TCP: Transmission Control Protocol

1. it provides *reliability*
   - resend lost, damaged, delayed packets
   - sort packets into the original order they were sent in
2. it provides *virtual connections*
   - note: underlying Internet Layer is connectionless
   - therefore, there cannot be a physical connection
   - only virtual
3. it addresses applications on machines using *port numbers*, for example:
   - 80 = HTTP server (WWW)
   - 25 = SMTP server (e-mail)
   - 21 = FTP server (file transfer)
   - etc...

TCP: abstraction

TCP uses IP's services to abstract from underlying networks:
- it sees only one virtual internetwork
- it doesn't know / care about the underlying infrastructure

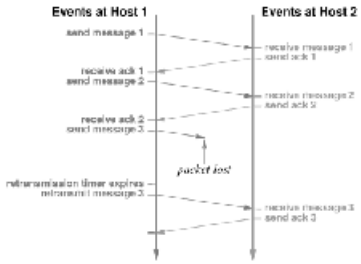


TCP: reliability

- TCP uses *acknowledgements* to track delivery of packets
- when a packet is not delivered in time, it is resent by TCP

fact: in TCP packets are called *segments*

Q: why does this make sense?



basics: client / server

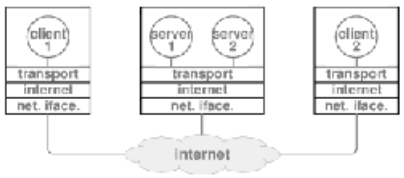
server: provider of a service
client: consumer of a service

for example:
- a web client a.k.a. *browser* (e.g. Safari) retrieving pages from a web server (e.g. Apache)
- a mail client (e.g. Outlook) retrieving e-mail from a mail server (e.g. Exchange)
- an FTP client sending files to an FTP server
- a telnet client connecting to a telnet server
- etc...

servers are usually quite passive, waiting for requests from clients to come in.

TCP: client / server

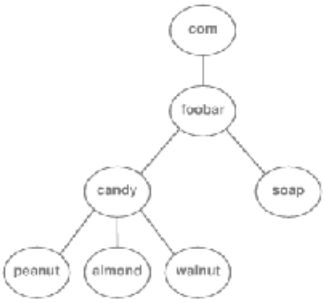
- a machine can run multiple servers (application, programs) on different ports
- each server uses a different Application Layer protocol (e.g. telnet, ftp, HTTP)
- multiple clients can contact multiple servers on one machine



DNS: Domain Name System

- designed to give computers a name that is easier to remember
- maps computer names to IP addresses using a distributed database
  (e.g. krypton.liacs.nl is name for 132.229.132.11)
- computer names are organized into hierarchies called *domains*
- example top-level domains (TLD's):
  - .com .net .org .gov .mil .edu
  - .nl .be .tv .fr (countries)
  - .info .biz .museum .name .pro (since 2001)

DNS: hierarchical

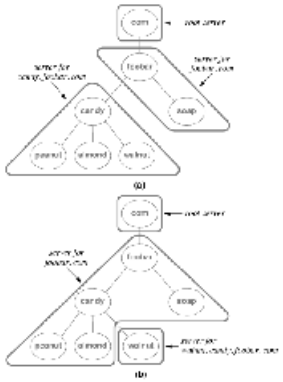


DNS: distributed database

suppose you want to know the IP address associated with almond.candy.foobar.com

if your local DNS server does not hold the requested domain name, the request is forwarded to a DNS *root server.*

the DNS root server may tell you which DNS server holds the requested domain name.



WWW: World Wide Web

- the World Wide Web is a network of *hypertext documents*, not a network of computers!
- hypertext is text containing hyperlinks
- a hyperlink is a pointer from a document to another one
- WWW actually uses *hypermedia*, not just *text* documents
- the application layer protocol used is called *HTTP*
- a *web browser* is a HTTP client that can retrieve, display and navigate hypertext documents
- a *web server* is a HTTP server that can return hypertext documents on request

HTTP: HyperText Transfer Protocol

- transfers hypertext documents between clients and servers
- Hypertext documents are written in *HyperText Markup Language* (HTML)
- HTTP servers run on TCP port 80

Scenario:

- client browser connects to HTTP server
- client browser send a request to the HTTP server
- HTTP server reacts by sending a response
- HTTP server disconnects

HTTP transaction

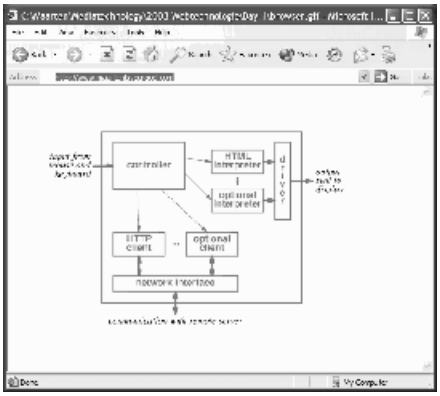
> telnet www 80

```
% telnetwww 80
Trying132.229.44.15...
Connected to www.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Wed, 07 Nov 2001 23:52:42 GMT
Server: Apache/1.3.12 (Unix)  (Red Hat/Linux) PHP/3.0.18 mod_perl/1.23
Last-Modified: Tue, 23 Oct 2001 11:44:04 GMT
ETag: "31aaf-1a2a-3bd55804"
Accept-Ranges: bytes
Content-Length: 6698
Connection: close
Content-Type: text/html

<HTML> <HEAD> <META name="description" content="LIACS, Leiden Institute of
   Advanced Computer Science, Universiteit  Leiden">
...
```

web browser internals



SMTP: Simple Mail Transfer Protocol

- used to send e-mail
- SMTP servers run on TCP port 25

Scenario:

- sender's mail client (e.g. Outlook) connects to SMTP server and specifies sender, recipient, message, subject, etc...
- sender's mail client disconnects
- SMTP server connects to recipient's mail server and forwards the message
- recipient's mail server stores message in recipient's mailbox
- recipient's mail client connects to mailbox (usually using *Post Office Protocol 3*, POP3) and retrieves the message

> telnet noc.liacs.nl

```
> telnet noc 25
Trying 132.229.44.11...
Connected to noc.liacs.nl.
Escape character is '^]'.
220 noc.liacs.nl ESMTP Sendmail 8.9.3/8.9.3/LIACS 1.0; Wed, 7 Nov 2001 14:44:45 +0100 (MET)
HELO liacs.nl
250 noc.liacs.nl Hello joostd@krypton.wi.leidenuniv.nl [132.229.132.11], pleased to meet you
MAIL FROM: george@whitehouse.gov
250 george@whitehouse.gov... Sender ok
RCPT TO: joostd@liacs.nl
250 joostd@liacs.nl... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Hi Joost! Wanna grab a beer after lunch?
See ya!
GWB
.
250 OAA11521 Message accepted for delivery
QUIT
221 noc.liacs.nl closing connection
Connection closed by foreign host.
```

references

- The World Wide Web Consortium: www.w3.org
- Internet Assigned Numbers Authority (IANA): www.iana.org
- ICANN: www.icann.org
- Stichting Internet Domeinregistratie (SIDN): www.sidn.nl

- www.visualroute.nl
- www.whois.org
- www.whatismyip.com