

nixu

How Internet Works

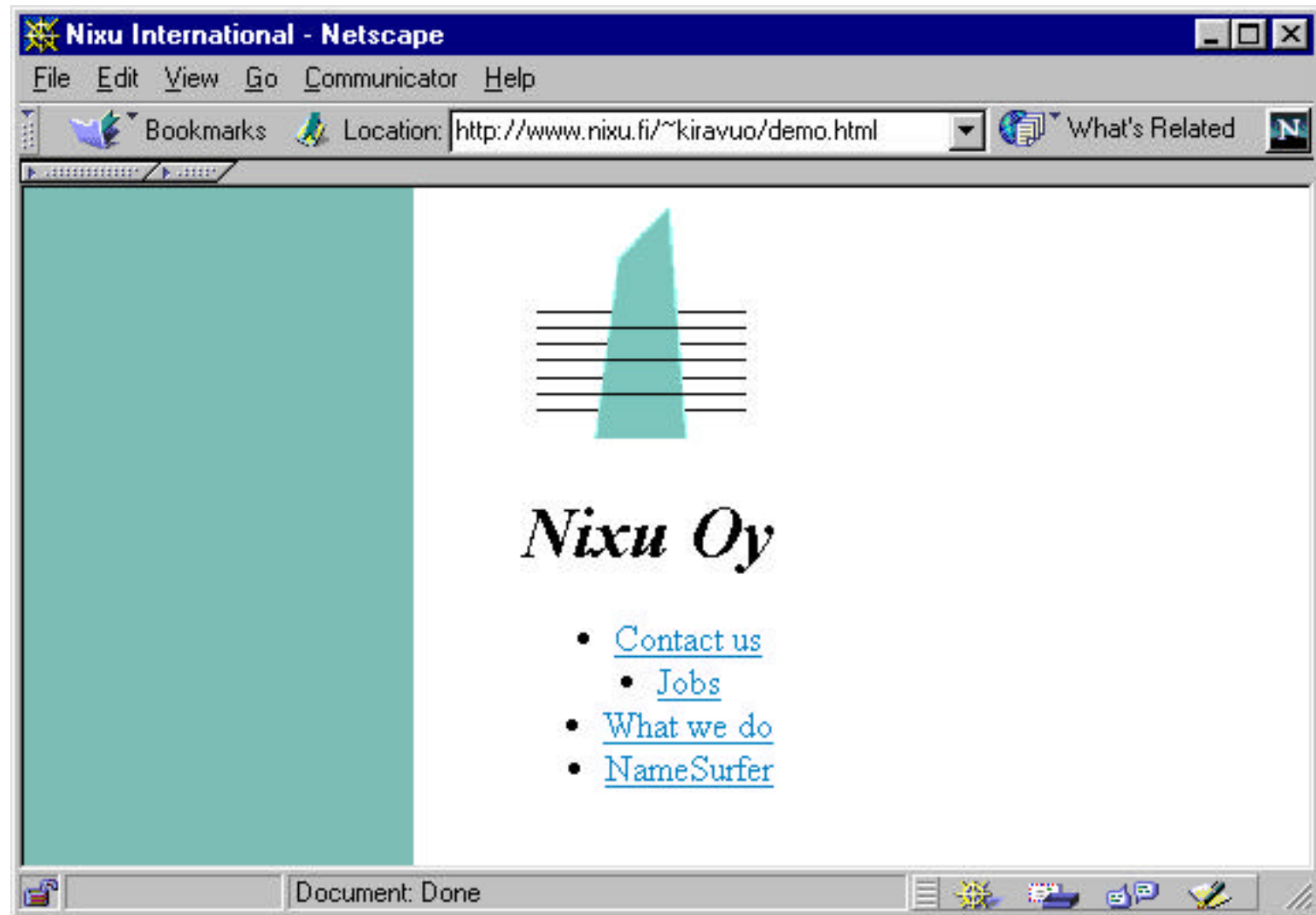
Contents

- World Wide Web, which is made of
- HTML and HTTP and communicates over
- TCP/IP, which uses
- Ethernet and other network media.

Network models

TCP/IP	OSI	UColorado	UKUUG
		Political	Confusion
		Financial	Financial
Application	Application	Application	Application
	Presentation	Presentation	
	Session	Session	
TCP, UDP	Transport	Transport	Transport
IP	Network	Network	IP
Data link	Data link	Data link	Network
Physical	Physical	Physical	

WWW-page



HTML code

```
<HTML>
  <HEAD>
    <TITLE>Nixu International</TITLE>
  </HEAD>
  <BODY background="/gfx/back1.gif">
    <CENTER>
      <IMG SRC="/gfx/logo.gif" ALT="Nixu Oy"><BR>
      <H1><I>Nixu Oy</I></H1>
      <UL>
        <LI><A HREF="/yhteysti.html">Contact us</A>
        <LI><A HREF="/avoimett/">Jobs</A>
        <LI><A HREF="/palvelut/">What we do</A>
        <LI><A HREF="http://www.namesurfer.com/">
          NameSurfer</A>
      </UL>
    </CENTER>
  </BODY> </HTML>
```

World Wide Web

- Combines text, pictures and hyperlinks to a hypertext document
 - a hyperlink points to another document
- Can also contain moving pictures, active programs etc.
- Hypertext document is written in HTML language
 - HyperText Markup Language
 - based on SGML (Standard Generalized Markup Language)
 - describes the structure of a document
- WWW browser decides how to display the document based on the structure description in HTML

URL

- Universal Resource Locator
- URL is an address to the current location of information

```
protocol://host[:server port]/path/file.html
protocol://host[:server port]/path/
```
- For example

```
http://www.hut.fi/
http://www.nixu.fi/~kiravuo/etiketti/index.html
news:sfnet.harrastus.retkeily
ftp://ftp.funet.fi/pub/netinfo/rfc/
```
- WWW browser uses the URL address to retrieve a document over the network

HTTP session

- Connection to the HTTP server program in `www.nixu.fi`
— equivalent to: `telnet www.nixu.fi 80`

```
GET /~kiravuo/demo.html HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.06 [en] (Win95; I)
Accept: image/gif, image/x-xbitmap, image/jpeg,
image/pjpeg, image/png, */*
Accept-Charset: iso-8859-1,*,utf-8
...
```

```
HTTP/1.1 200 OK
Server: Apache/1.2.6
Last-Modified: Tue, 24 Nov 1998 06:35:48 GMT
Content-Length: 466
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
```

```
Content-Type: text/html
...
<HTML>
  <HEAD>
    <TITLE>Nixu International</TITLE>
  ...

GET /gfx/back1.gif HTTP/1.0
Referer: http://www.nixu.fi/~kiravuo/demo.html
...

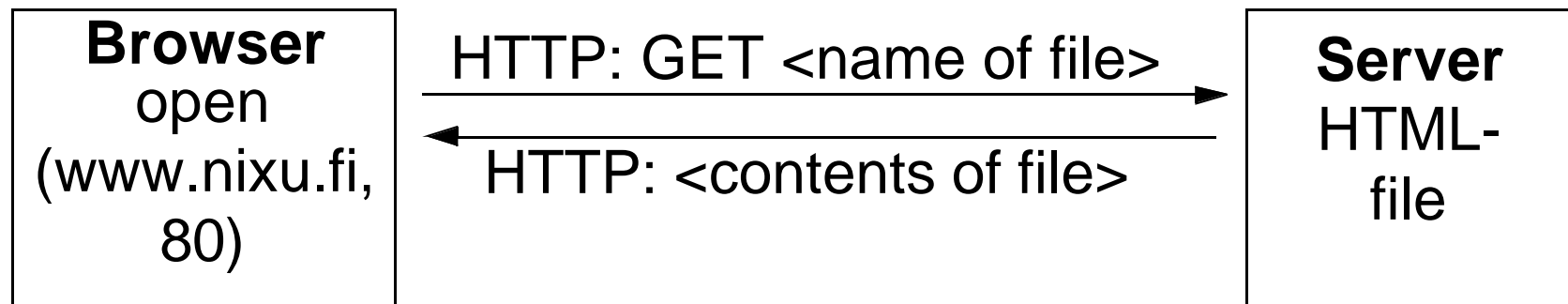
HTTP/1.1 200 OK
Content-Length: 902
Content-Type: image/gif
..GIF89a...
...
```

HTTP protocol

- HTTP is a simple protocol for file transfer over the network
- Client (browser) initiates the session by connecting to the server
 - pull protocol, server does not transmit on its own
- Client issues the GET-command to retrieve a file from the server
- Additional information is also passed
 - client type
 - acceptable data formats
- Server sends information about the file and the contents of the file

Connections

- HTML is a data format
- HTTP protocol transfers files over a reliable connection
 - no error checks or checksums in HTTP
- On the Internet we use TCP/IP for the reliable connection
 - TCP/IP provides a reliable byte-stream as a service to HTTP
- Browser uses HTTP to get the required HTML file specified by the URL



Addresses

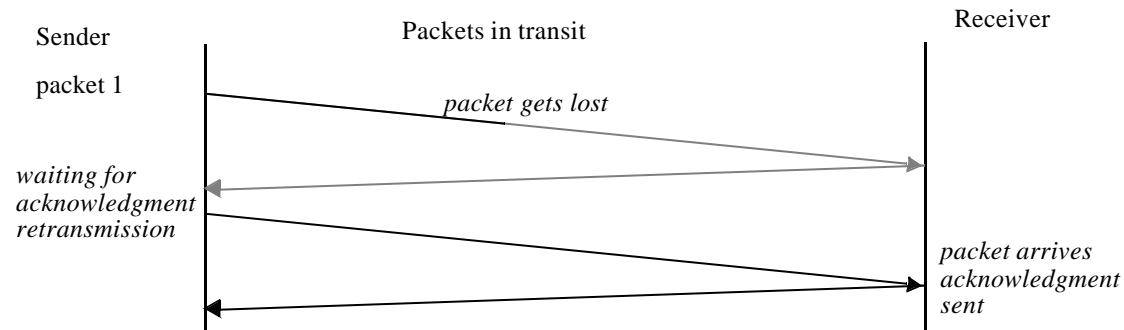
- Each host has a name
 - eg. www.hut.fi
- The network uses numeric addresses
 - eg. 130.233.224.28
- Translation is provided by the DNS (Domain Name Service)
- A host has also numbered ports, which are connected to server programs
 - port addresses are standardized, eg port 80 for HTTP, port 25 for SMTP email

TCP

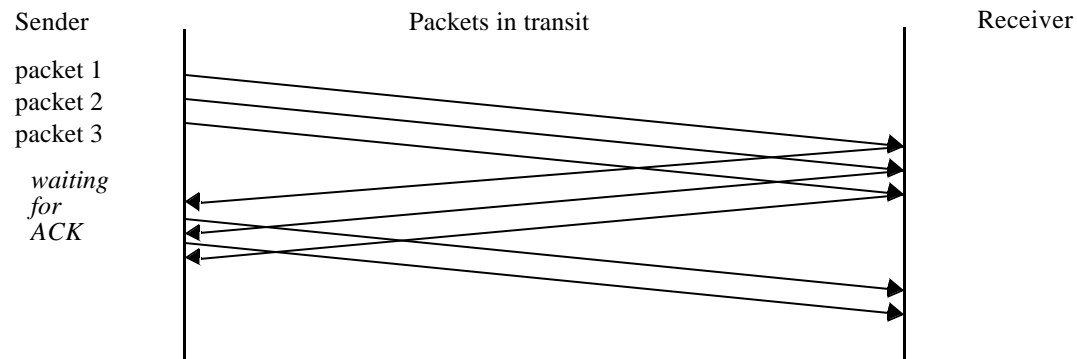
- TCP = Transmission Control Protocol
- Defined in RFC-793
- Connection-oriented, reliable, **byte-stream** service
- Application data is broken into **segments**, which are sent as IP datagrams.
- Features:
 - checksums, timeouts and flow control
 - segment reassembly in correct order, discarding duplicate packets

Reliable TCP operation

- Receiver sends acknowledgment for each segment.
- If a packet gets lost, timeout will ensure it's retransmitted



— Normally a sliding window technique



UDP

- UDP = User Datagram Protocol
- Defined in RFC-768
- Unreliable datagram-oriented transportation layer protocol
 - offers little extra functionality besides port numbers
 - light-weight, easy to implement
- Applications using UDP: DNS, Radius, NTP, SNMP

IP

- IP = The Internet Protocol
- Defined in RFC-791
- IP sends simple **datagrams** over network.
- It provides **unreliable** and **connectionless** delivery service.
 - unreliable = no guarantees, best effort only
 - connectionless = each packet is routed separately

Ethernet

- Defined in IEEE 802.3
- Local area network
 - limited to short range connections (100 m - 1 km)
 - different cabling options (coaxial cabling, twisted pair etc.)

Data client -> server

```
ETHER: Destination = 8:0:20:74:f1:2c, Sun
ETHER: Source       = 0:0:3b:80:e:93,
ETHER: Ethertype    = 0800 (IP)
IP: Protocol        = 6 (TCP)
IP: Source address  = 194.197.118.74, chili.nixu.fi
IP: Dest. address   = 194.197.118.20, jalopeno.nixu.fi
TCP: Source port    = 35620
TCP: Destination port = 80 (HTTP)
TCP: Sequence number = 760000273
TCP: Acknowledgement number = 2370000258
HTTP: "GET /~kiravuo/demo.html HTTP/1.0
      Connection: Keep-Alive
      User-Agent: Mozilla/4.06..."
```

Ack server -> client

```
ETHER: Destination = 0:0:3b:80:e:93,  
ETHER: Source      = 8:0:20:74:f1:2c, Sun  
ETHER: Ethertype   = 0800 (IP)  
IP:   Flags = 0x4 (do not fragment)  
IP:   Protocol = 6 (TCP)  
IP:   Source address = 194.197.118.20, jalopeno.nixu.fi  
IP:   Dest. address = 194.197.118.74, chili.nixu.fi  
TCP:  Source port = 80 (HTTP)  
TCP:  Destination port = 35620  
TCP:  Sequence number = 2370000258  
TCP:  Acknowledgement number = 760000379  
TCP:  Flags = 0x10 (ACK)  
HTTP:  ""
```

Data server -> client

```
ETHER: Destination = 0:0:3b:80:e:93,  
ETHER: Source      = 8:0:20:74:f1:2c, Sun  
ETHER: Ethertype   = 0800 (IP)  
IP: Protocol      = 6 (TCP)  
IP: Source address = 194.197.118.20, jalopeno.nixu.fi  
IP: Dest. address = 194.197.118.74, chili.nixu.fi  
TCP: Source port  = 80 (HTTP)  
TCP: Destination port = 35620  
TCP: Sequence number = 2370000299  
TCP: Acknowledgement number = 760000379  
TCP: Data offset = 20 bytes  
TCP: Flags = 0x10 (ACK)  
HTTP: "HTTP/1.1 200 OK  
      Server: Apache/1.2.6  
      Last-Modified: Tue, 2..."
```

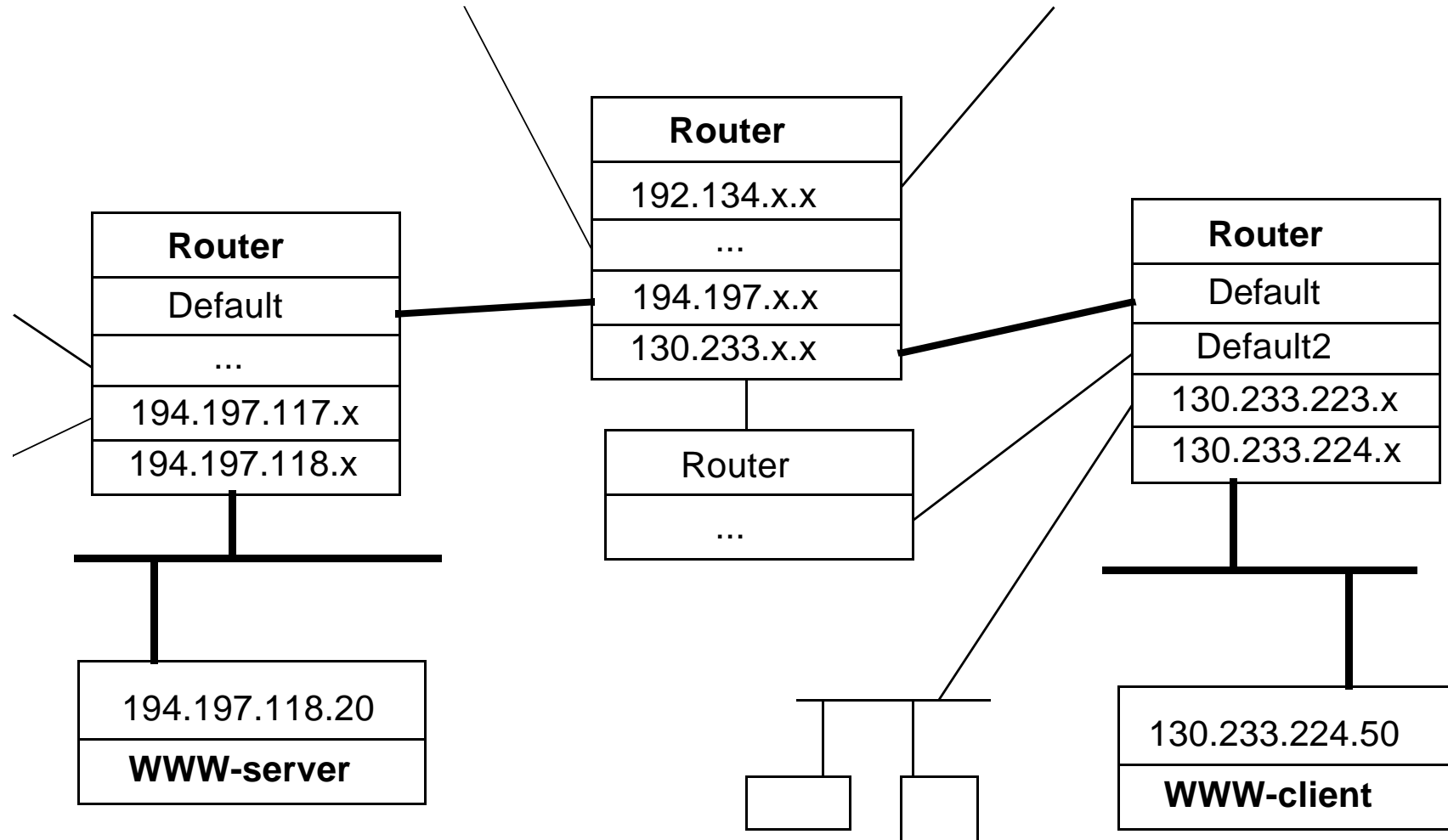
Ack client -> server

```
ETHER: Destination = 8:0:20:74:f1:2c, Sun
ETHER: Source       = 0:0:3b:80:e:93,
ETHER: Ethertype    = 0800 (IP)
IP: Protocol        = 6 (TCP)
IP: Source address  = 194.197.118.74, chili.nixu.fi
IP: Dest. address   = 194.197.118.20, jalopeno.nixu.fi
TCP: Source port    = 35620
TCP: Destination port = 80 (HTTP)
TCP: Sequence number = 760000379
TCP: Acknowledgement number = 2370000627
TCP: Flags = 0x10 (ACK)
HTTP:  ""
```

Routing

- LANs are connected by routers
- Each router has a table that tells which network address can be reached by which connection
- Backbone routers update their information automatically and can route around breaks in the network

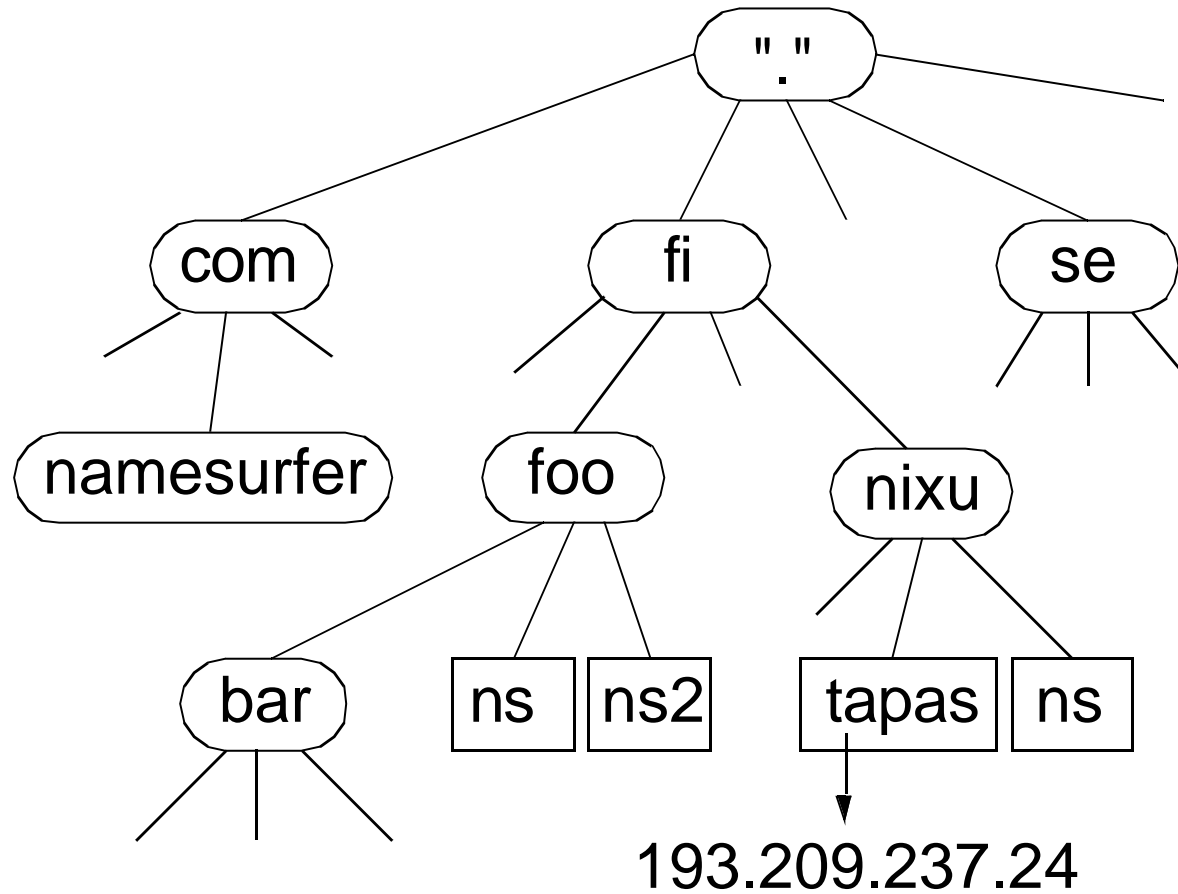
Routing



Domain Name System

- Distributed database with local caching
- Distributed administration (adding or deleting data)
- Services provided:
 - Hostname to IP address
 - IP address to hostname
 - Mail exchange records

Name space structure



Network models

TCP/IP	OSI	UColorado	UKUUG
		Political	Confusion
		Financial	Financial
Application	Application	Application	Application
	Presentation	Presentation	
	Session	Session	
TCP, UDP	Transport	Transport	Transport
IP	Network	Network	IP
Data link	Data link	Data link	Network
Physical	Physical	Physical	

Conclusion

