

CS2105 Lecture 2

Application Layer

21 January, 2013

Application Layer

21 January, 2013

After this class, you are expected to:

- be able to choose the right architecture and transport-layer services for your own network application (and justify why).
- understand the basic HTTP interactions between the client and the server, including HTTP request (GET and HEAD) and HTTP response.
- understand the concepts of persistent connection, parallel HTTP connections and stateless protocol.
- understand the services provided by DNS and how a query is resolved.

Application

Transport

Network

Link

Physical

Networked applications runs on
hosts and consists of
communicating processes

The **server process** waits to be contacted

The **client process** initiates the communication

Application architecture:
client-server
peer-to-peer
hybrid

Need to identify the source and destination process

Address of a process:
(host, port number)

Host addresses
are **32-bit** integers
known as **IP addresses**,
represented by four numbers

Host addresses
are **32-bit** integers
known as **IP addresses**,
represented by four numbers

we are going to talk a lot more about IP
addresses in Lecture 6.

Ports are **16-bit** integers
(1-1023 are reserved for OS)

IANA coordinates the assignment of port number.

IANA coordinates the assignment of port number.

You can find the list of port numbers at

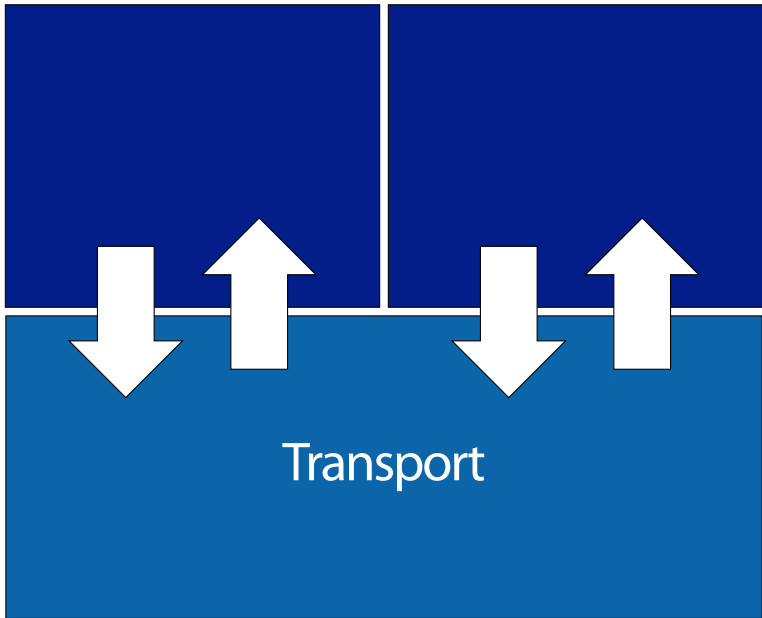
<http://www.ietf.org/assignments/port-numbers>. Look for familiar port numbers, such as HTTP, SSH, Battle.Net. For list of ports, including unofficial ones not registered with IANA, check out http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.

Socket is the software interface
between processes and the
Internet.

initialize a socket
listen for a connection
accept a connection
request a connection
send a message
receive a message
close the socket

initialize a socket
listen for a connection
accept a connection
request a connection
send a message
receive a message
close the socket

The actual APIs for Java will be shown to you
through video-based Lecture 5



Transport service requirements:
loss-tolerance
or
critical?

Transport service requirements:
throughput-sensitive
or
elastic?

Transport service requirements:
time-critical or
not?

Transport service requirements:
time-critical or
not?

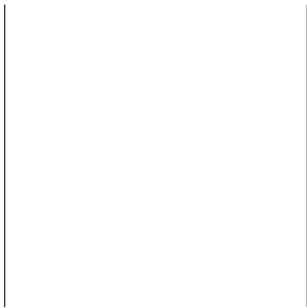
The other transport service requirement mentioned in the textbook is *security*,
but we will move all discussion about security to Lecture 8 (Week 10).

Transport protocols:

TCP
and
UDP

TCP is
**connection-oriented,
congestion-controlled,
and reliable.**

TCP takes one RTT to establish connection.



TCP provides
**no gurantees on
throughput and delay**

UDP is
connection-less,
not congestion-controlled,
and not reliable.

when writing network application, ask:

- what architecture?**
- what type of services?**
- how messages are exchanged?**

HTTP

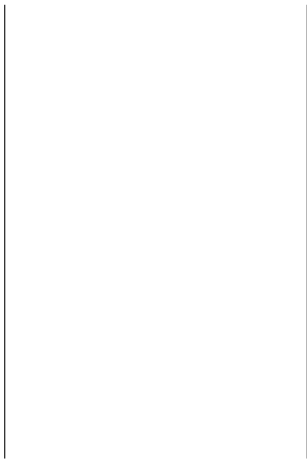
Hyper-Text Transfer Protocol

HTTP

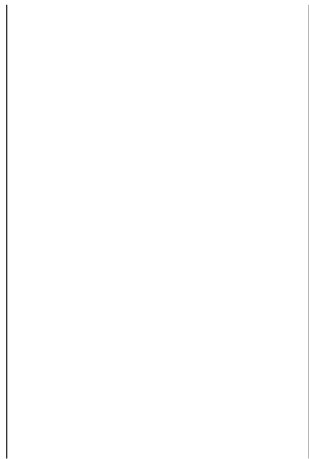
Hyper-Text Transfer Protocol

We are not able to cover everything about HTTP here. You may want to read up on your own about cookie (Section 2.2.4) and Web caching (Sections 2.2.5 and 2.2.6) to learn more about these two important topics.

Web page
HTML file
Web object
URL

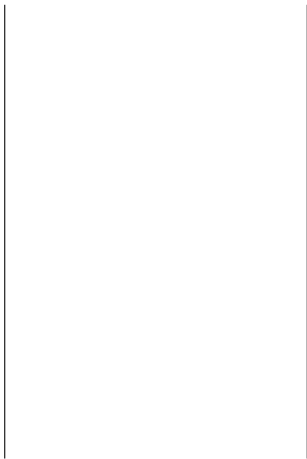


persistent
vs.
non-persistent



stateless
vs.
stateful

pipeline
vs.
sequential



```
GET /~cs2105/ HTTP/1.1
Host: www.comp.nus.edu.sg
User-Agent: Mozilla/5.0
Connection: close
```

HTTP/1.1 200 OK

Date: Wed, 19 Jan 2011 06:58:35 GMT

Server: Apache/2.2.6 (Unix)

Connection: close

Content-Type: text/html

Demo

with telnet and curl

Demo
with telnet and curl

Sample commands:

```
- telnet <hostname> 80
```

```
- curl -I <URL>
```

You can get curl from

```
http://curl.haxx.se/download.html
```

DNS

Domain Name Service

Two ways to identify a host:

domain name

(e.g., `www.nus.edu.sg`)

IP address

(e.g., `137.132.39.133`)

**DNS translates between the
two**

Demo

with nslookup and dig

Demo
with nslookup and dig

dig is installed on many UNIX-based systems. For windows-based OS, dig is available at <http://members.shaw.ca/nicholas.fong/dig/>.

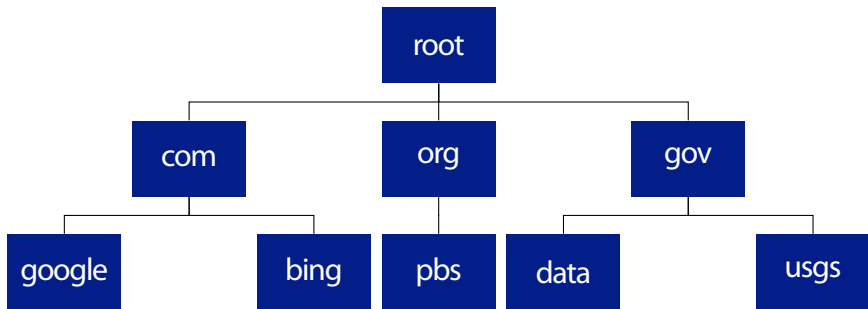
Useful dig options include +trace and +short.

DNS resource record (name, value, type, TTL)

DNS record types

A, MX, CNAME, NS

DNS uses a **hierarchical distributed databases**



Root servers

Root servers

The list of all DNS root servers can be found on
<http://www.root-servers.org/>.

TLD servers

Authoritative servers

Local DNS servers

Root

TLD

Auth

Host

Local

Root

TLD

Auth

Host

Local

DNS Caching

DNS Registrar