Internet Applications

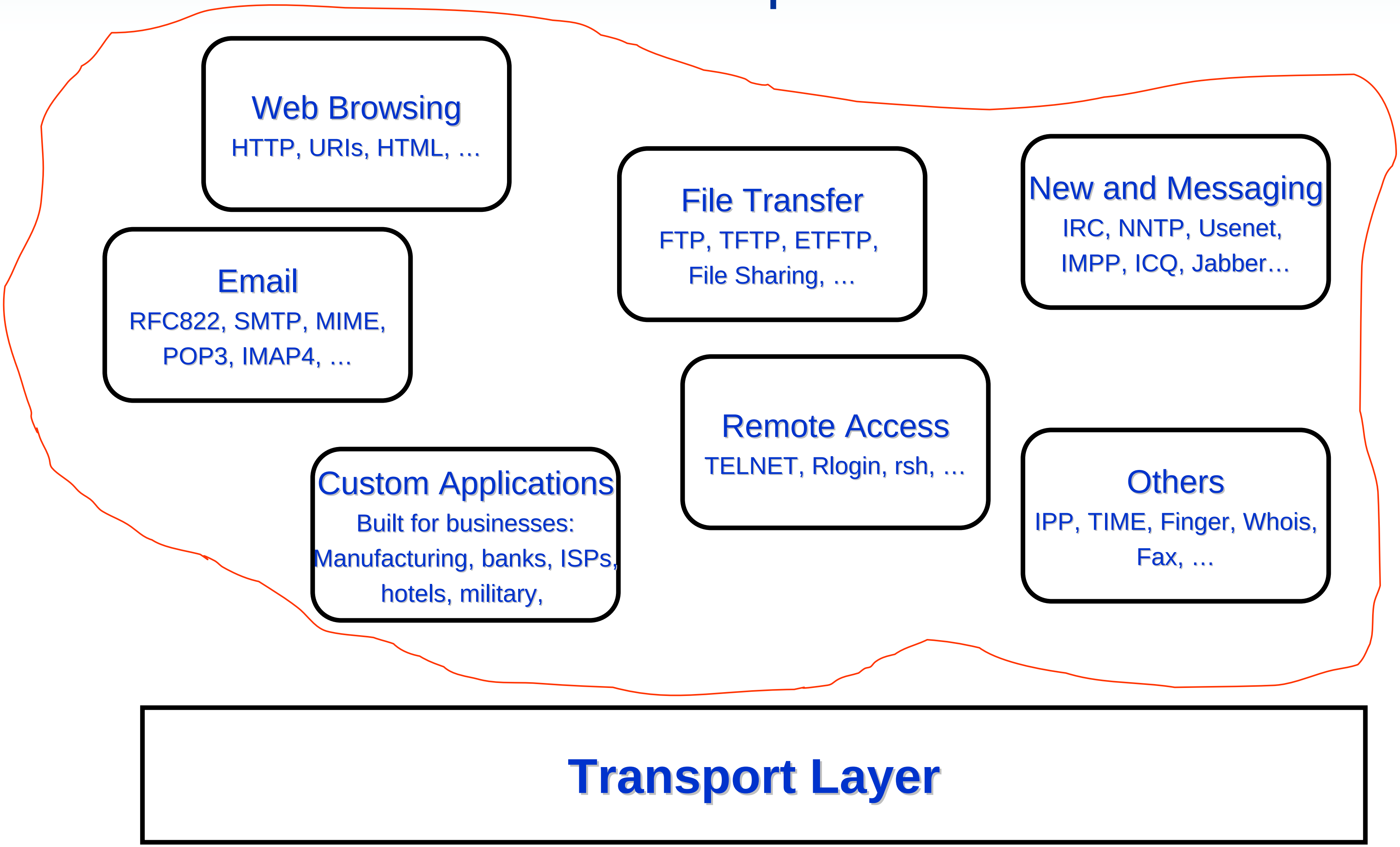
Dr Steve Gordon

ICT, SIIT

Contents

- Network Application Models
- Transport Layer Interface
- Selected Applications and Services
  - Naming Resources
  - Web Access
  - Email
  - Network Management
  - Other Applications

Example Internet Applications and Techniques



Web Browsing
HTTP, URIs, HTML, …

Email
RFC822, SMTP, MIME, POP3, IMAP4, …

File Transfer
FTP, TFTP, ETFTP, File Sharing, …

New and Messaging
IRC, NNTP, Usenet, IMPP, ICQ, Jabber…

Custom Applications
Built for businesses: Manufacturing, banks, ISPs, hotels, military,

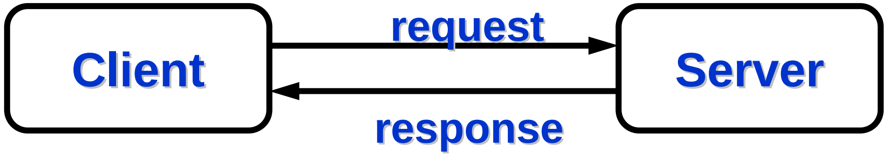
Remote Access
TELNET, Rlogin, rsh, …

Others
IPP, TIME, Finger, Whois, Fax, …

Transport Layer

Network Application Models

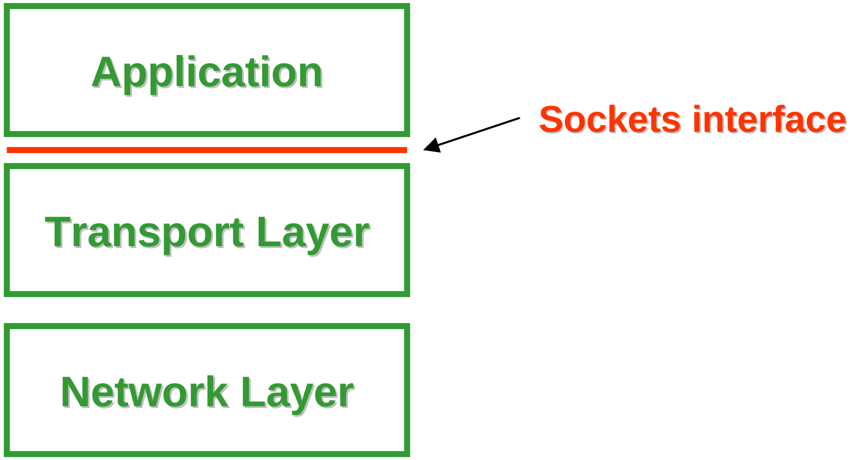
- Most network applications follow a Client/Server model



  - Servers implemented as application programs (not a physical computer)
    - Wait (or listen) for requests on well-known ports
    - May create a slave process to handle the request, i.e. allow multiple concurrent connections
  - Use TCP/IP for communication
  - Client application program sends request to server well-known port
    - A connection may be established, and then data transfer takes place
  - Once the connection is established, both client and server can send data
- Other models:
  - Pre-collection of information: a software process runs all the time and collects information. When a user application needs some information, it uses the pre-collected information. Inefficient use of network and CPU
  - Peer-to-peer:
    - From a programmers view, most peer-to-peer applications use a Client/Server model (that is, one software process will listen, and the other will initiate a connection)

Transport Layer Interface

- Transport layer (e.g. TCP, UDP) is implemented in the operating system
- Applications use the transport layer to send data
- There is a common programming interface between applications and transport protocols, called sockets
  - Based on Berkeley Unix sockets, although most operating systems have the same or similar concepts
    - Forms the basis of Microsoft Windows Sockets (Winsock)
  - Designed as generic interface to support many protocols
- We will study and use sockets in Network Lab (ITS332)



BSD Sockets

- Create a socket:
  - socketid = socket (protocol_family, comms_type, protocol)
- Bind socket to address:
  - bind (socketid, addr, addr_length)
- Connect socket to a destination address:
  - connect (socketid, destaddr, addr_length)
- Prepare socket for incoming connections (server):
  - listen (socketid, queue_length)
- Wait for incoming connection (server):
  - newsocketid = accept (socketid, addr, addr_length)
- Sending and receiving data:
  - write (socketid, data, length)
  - read (socketid, buffer, length)
- Close a connection:
  - close (socketid)

BSD Socket Example

- Client Application

```
s = socket(PF_INET,SOCK_STREAM,0);
connect(s, 172.17.3.12:23, len);

write(s, “Request”, 7);


read(s, buffer, 5);
close(s);
```

- Server Application

```
s = socket(PF_INET,SOCK_STREAM,0);
bind(s, 172.17.3.12:23, len);
listen(s, 5);
while(1) {


   snew = accept(s,clientaddr,len);

   read(snew, buffer, 7);
   write(snew, “Reply”, 5);
   close(snew);
}
```

Selected Applications and Services

Resource Identification

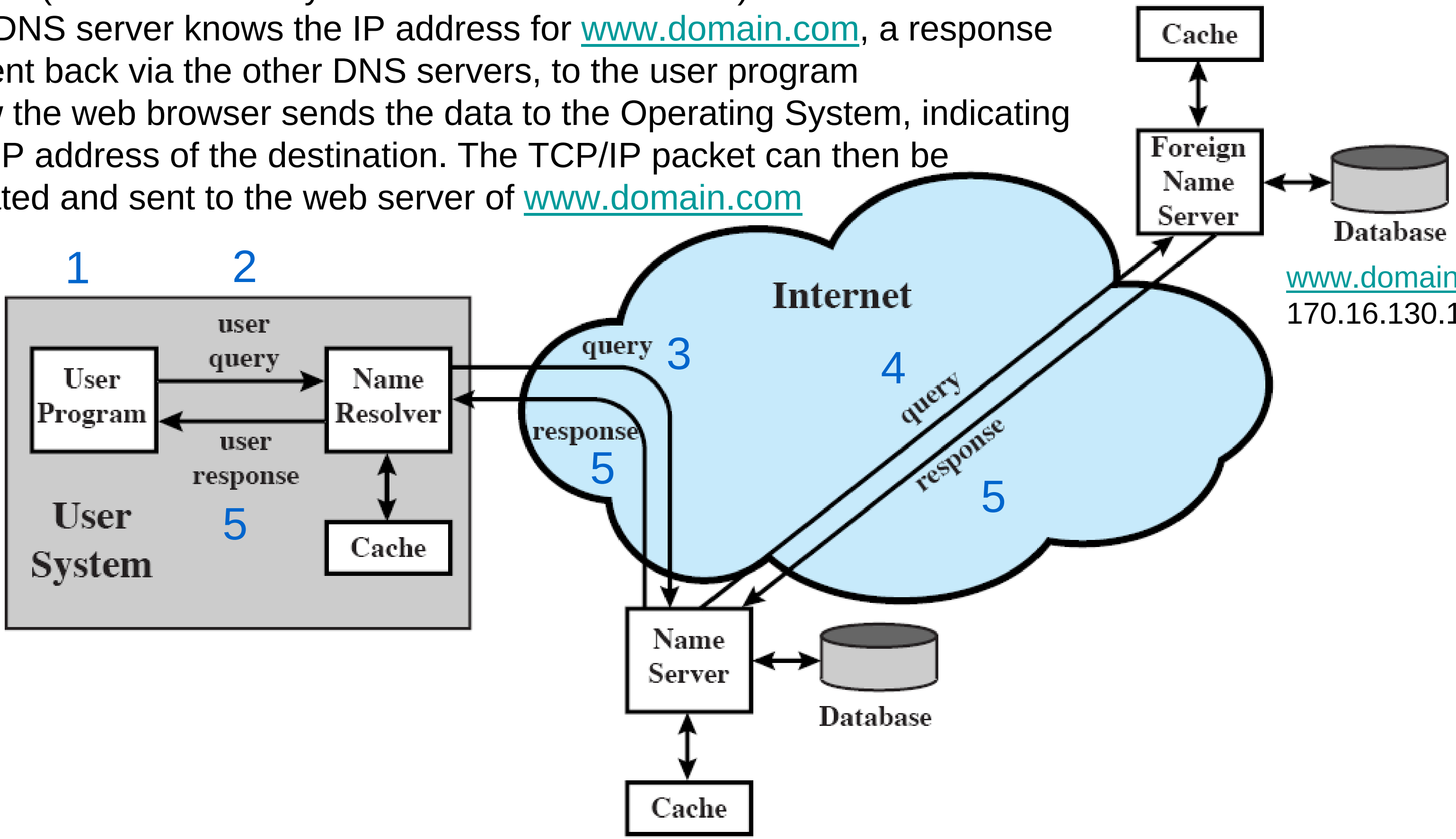
- We need some method of identifying resources in networks
  - Should be consistent, unique and user-friendly
- For computers in Internet, domains names are used
  - Hierarchy: an organisation manages a domain, and can allocate sub-domains to others
    - E.g. THNIC manages .th domain (including .ac.th)
    - Thammasat University obtains .tu.ac.th from THNIC
    - SIIT obtains .siit.tu.ac.th from TU
    - The SIIT computer centre allocates names for different services: www.siit.tu.ac.th, reg.siit.tu.ac.th, ict.siit.tu.ac.th, …
  - Domains are to identify computers (or IP addresses), not just web servers
    - E.g. possible to have stevespc.siit.tu.ac.th
- A more general format for resource identification is Uniform Resource Identifiers (URIs): a Uniform Resource Locator (URL) is a URI
  - Examples:
    - http://domain/path?query (may also contain username, password, port, …)
    - mailto:user@domain?header=value

Naming Service

- Domain Name Service (DNS)
  - URIs identify resources – user friendly names
  - IP addresses identify network interfaces – unique identifiers used for TCP/IP communications
  - DNS maps URIs (an in particular, domain names) to IP addresses
- Basic Operation of DNS
  - Domain names (e.g. www.domain.com) and their corresponding IP address are registered at DNS servers
    - Registration may be manual (e.g. if the IP/domain does not change often) or automatic (e.g. if your IP address changes often, such as on a home ADSL internet connection)
  - When applications have a domain name, the application uses DNS protocol to retrieve the corresponding IP address from the DNS server
  - Then the IP address is used by TCP or UDP to send data to the destination computer
- There is a hierarchy of DNS servers across the globe so that your requests are fast

DNS Example

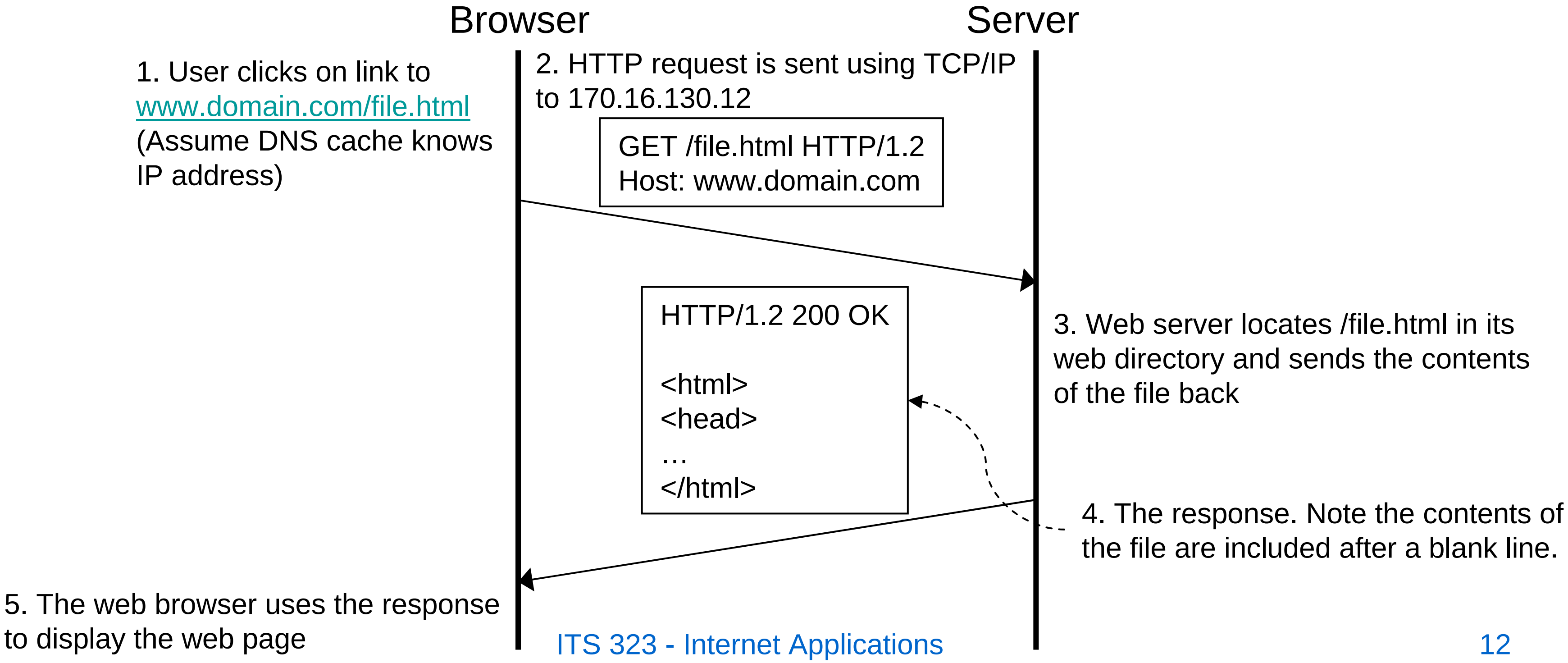
1. User clicks on a link to www.domain.com
2. Web browser (User program) queries local DNS cache (name resolver) for IP address
3. If IP address is not in cache, name resolver queries the local DNS server
4. If IP address is not known by local DNS server, it queries another DNS server (there is hierarchy of DNS servers in Internet)
5. If a DNS server knows the IP address for www.domain.com, a response is sent back via the other DNS servers, to the user program
6. Now the web browser sends the data to the Operating System, indicating the IP address of the destination. The TCP/IP packet can then be created and sent to the web server of www.domain.com



www.domain.com is at 170.16.130.12

Web Access

- HyperText Transfer Protocol (HTTP)
  - Request/response protocol using TCP: default port 80 for servers
  - Typically implemented by web browsers and web servers
  - Stateless: there is no connection between one request and the next
    - Although the browser/server may cache some information, HTTP does not



Browser

Server

1. User clicks on link to www.domain.com/file.html (Assume DNS cache knows IP address)

2. HTTP request is sent using TCP/IP to 170.16.130.12

```
GET /file.html HTTP/1.2
Host: www.domain.com
```

3. Web server locates /file.html in its web directory and sends the contents of the file back

```
HTTP/1.2 200 OK

<html>
<head>
…
</html>
```

4. The response. Note the contents of the file are included after a blank line.

5. The web browser uses the response to display the web page

Web Access

- HTTP Format
  - Different types of request messages may be sent (called *methods)*:
    - Methods: GET, POST, PUT, DELETE, …

METHOD URI Version	GET /file.html HTTP/1.2
Headers	User-Agent: Firefox

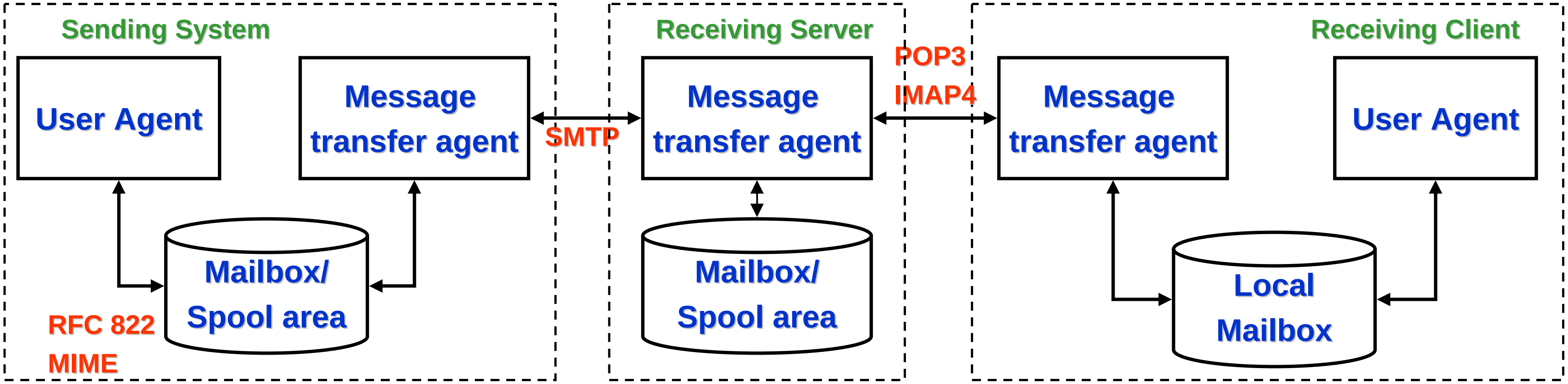
  - Server responds with a status code and possibly content
    - Status codes: 200 OK; 404 Not Found; 304 Not Modified; …
    - Content: text (e.g. the HTML code), audio, video, …
      - The content type is normally given in a header field

Version StatusCode StatusText	HTTP/1.2 200 OK
Headers	Date: 7 Sep 2007
Content	<html>
	<head> …

  - Both requests and responses can have header fields
    - E.g. date, caching information, content description, …

Email Protocols

- A User Agent (software for user to write email, e.g. MS Outlook, Eudora, web browser) creates email and saves in mailbox (also called spool area)
- Simple Mail Transfer Protocol (SMTP)
  - A Message Transfer Agent (MTA) retrieves the email from the mailbox and uses SMTP to send the email to the receiver SMTP server
  - SMTP uses TCP, with the server listening on port 25
  - Receiving SMTP server stores the email in its mailbox
  - User Agent can access mail from mailbox
- Post Office Protocol (POP), Interactive Mail Access Protocol (IMAP)
  - Many computers do not run SMTP server (needs to be running all the time to receive email)
  - Hence use POP or IMAP for a Receiving MTA to retrieve the email from SMTP server



Email Formats

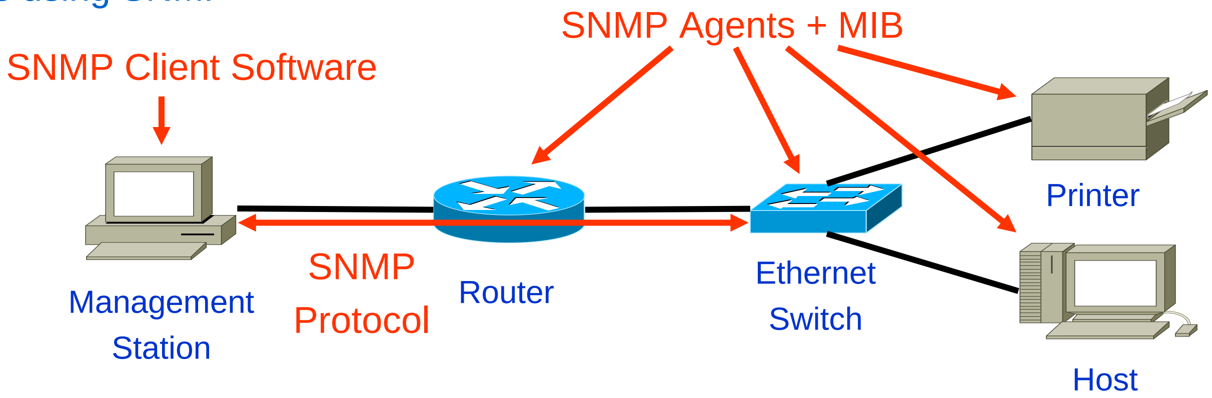
- RFC822
  - The original format defined by IETF is ASCII text, with:
  - Set of header lines with field: value
    - E.g. From, Date, To, Cc, Subject, Message-ID, …
  - The body of the email
- Multipurpose Internet Mail Extension (MIME)
  - Allows different types of content to be included in email
    - Text, images, audio, video, documents, ...
  - Defines how to format the content to be transferred in SMTP

Network Management

- Networks have become a critical part of operations in businesses, governments and other organisations
  - Therefore important that the network can be easily managed to avoid problems (and diagnose/fix them as quickly as possible)
- Network management systems should allow:
  - Single, easy-to-use interface to manage entire network
  - Perform a large set of diverse tasks
  - Not require significant additional hardware/software in the network
- Examples of network management tasks:
  - View and edit the current configuration of network devices (e.g. switches, routers, computers, …)
  - Collect statistics about the usage of the network (e.g. bits per second processed by a switch, utilisation of a link, CPU usage of a router, …)
  - Receive notifications of important events occurring (e.g. a switch fails, a router starts dropping many packets, link utilisation approaches capacity, …)

Simple Network Management Protocol

- SNMP is used to manage devices in an internet
- There are two types of entities in SNMP:
  - Managed nodes, that is, the devices we want to manage
    - Routers, computers, printers, switches, bridges, …
    - These devices execute SNMP agents, which is software that allows a SNMP client to send/receive commands on that device
    - The devices also store a database about its status, called a Management Information Base (MIB), e.g. a router MIB may include counts packets forwarded per second, dropped packets, route not found, …
  - Management station (computer with SNMP management software)
    - This is typically the interface to the human manager; a client program presents the information from all devices in the network to the human user
    - The client on the management station collects information from SNMP agents on devices using SNMP



Other Applications

- Instant Messaging
  - Every user creates a TCP connection from computer to MSN/Yahoo/AIM server
  - Instant messages are sent from your computer to server, and then from server to destinations computer
    - This is easy for company to manage, but puts heavy load on servers
  - Some applications (such as voice calls in MSN) use a client-to-client TCP connection (e.g. bypass the MSN server)
    - Voice calls (which generate a lot of traffic compared to text IMs) are more efficiently handled this way
  - Covered in ITS 413
- Voice over Internet (VoIP)
  - Use UDP for data transfer
    - TCP retransmission and flow control is inefficient for real-time transfer needed for voice (too much delay and jitter)
  - Use protocols like RTCP and SIP for call connection and control
    - Skype uses a proprietary protocol
  - Covered in ITS 327

Other Applications

- File Transfer
  - Client/server based
  - TCP for reliable file downloads
- File Sharing
  - Peer-to-peer architecture
    - Setup TCP connection from one user’s computer to another user’s computer
    - However may use a centralised server to locate files
  - Covered in ITS 413
- Video conversations, Video streaming, IPTV, Network games, …
  - Maybe covered in ITS 413 and ITS 327
- Custom applications
  - Created for specific uses in industry such as:
    - Control of manufacturing systems, database access, military command and control, embedded applications in vehicles, machinery and electronic devices, …