UNIT – I

Why networked Java

* In the last ten years, network coding has stopped being the province of a few specialists and became a core part of every developers toolbox.
* Today more programs are network aware than aren’t. Beside classic apps like email, web browsers and Telnet clients. Most major apps have some level of networking built.

For e.g.,

* Text editors like BEdit save and open files directly from FTP server.
* IDEs like Eclipse and Intellij IDEA communicate with CVS repositories.
* Word processors like Microsoft word open files from URLS.
* Antivirus programs like NORTON Antivirus check for new virus by connecting to the vendors website every time the computer is started.
* Music players like winamp and itunes upload CD Track length to CDDB to download the corresponding track titles.
* Super marker cash register
* Schedule Apps like Microsoft outlook automatically.

What can a network program do?

* Network adds a lot of power to a simple program
* Single program can retrieve information stored in millions of computer located anywhere in the world.
* A single program can communicate with tens and millions of people.
* Network apps generally take one of several forms.
* In the simplest case clients retrieve data from a server and display it. More complex clients filter and reorganize data.
* It repeatedly retrieve changing data send data to other people and computers and Interact with peers in Real Time for chat, Multiplayer games or collaboration.

Retrieve Data:

* Basic level of network client retrieves data from a server
* It can format the data for display to a user store it in a local database, combine it with other data sources both local and Remote.

Security:

* Not all network programs need to run code uploaded from remote systems. Code loaded from the network is usually prohibited from doing.
* Remotely loaded code cannot access arbitrary addresses in memory.
* Remotely loaded code cannot access the local file system.
* Cannot print documents.
* Remotely loaded code cannot read from or write to the system.

Finally and most importantly:

* Remotely loaded code can only open network connections to the host from which the code itself was downloaded.
* Remotely loaded code cannot listen on ports below 1024.

Basic web concepts:

* Java can do a lot more than create flashly web pages. Nonetheless many of your programs will be applets on web pages, servelets running on the server or web services that need to talk on other web servers therefore it is important to have a solid understanding of interaction between web servers and web browsers.
* HTTP: Hyper Text Transfer Protocol is a standard that defines how a web client talks to a server and how data is transferred from server back to the client.

Basic Network Concepts:

* A network is a collection of computers and other devices that can send data and receive data.
* Networks are often connected by wires, and bits of data are turned into Electromagnetic waves that move through wires.
* However wireless network transmit data through infrared light and microwaves.
* Long distance transmission are now carried over fibre optic cable .
* Theoritically it is told the data is transmitted by coal powered computers that send smoke signal to each other. Response time of such computers are poor.
* Every network node has a address. Addresses are assigned differently on different kinds of networks.
* Apple talk addresses are chosen randomly to start up each host. The host check to see if any other machine are using that address.
* Some kind of networks, nodes also have names that help human being s identify them. Names are not locked to address, names can change when addresses remain the same or addresses can change when name remains the same.

The layers of a Network:

* There are 7 layers in the model, hence the name is 7 layer Model
* The model acts as a framework of reference in the design of communications and networking products.

7. APPLICATION L AYER

6. PRESENTATION LAYER

5. SESSION LAYER

4. TRANSPORT LAYER

3. NETWORK LAYER

2. DATA LINK LAYER

1. PHYSICAL LAYER

The function of a layer:

* Each layer deals with one aspect of networking.
* Layer1 deals with communication media.
* Each layer communicates with the adjacent layers

-in both directions

-e.g network layer communicates with

\*Transport layer

\*data Link layer

\*each layer formats the data packet

e.g adds or deletes addresses

7. Application layer

\* Purpose: user application to network service interface

* E.g –File Request from server

-Email services etc.

Function:

* general netwok access
* - Flow control
* - Error Recovery

6. Presentation Layer

\* Purpose: Formats data for exchange between points of communicatioin

\* E.g. Redirector software

\* Formats for transmission to the server.

Function:

* Protocol conversion
* Data translation
* Encryption
* Character set conversion
* Expansion of graphics command

e.g F:/PUR/ORDER c:/CORRES/USDA

(to server) redirects (to local disk)

5.Session Layer:

Purpose: oversee a communication session

-Establish

-Maintain

-Terminate

E.g.

Function :

* Performs name recognition and related security
* Synchronization between sender and receiver
* Assignment of time for transmission

-Start time

-End time etc.

4. Transport Layer:

Purpose:

* Repackage proper and efficient delivery of package
* Error free
* Insequence
* Without duplication

e.g.

\* for sending data

- repackaging the message to fit into packets .

\* split long message

\* assemble small messages

-On receiving data

\* perform the reverse

\* send an acknowledgement to the sender

- solve packet problems

\* during transmission and reception.

3. Network Layer

Purpose:

* Addressing and routing the packets.

e.g. Application at the Router.

- If the packet size is large, splits into small packets

Function:

* Address messages
* Address Translation from Logical to Physical.

E.g. nsamp – 102.13.345.25

\* Routing of data

- Based on Priority

- Best path at the time of transmission

\* Congestion control

2. DataLink Layer

\* Purpose:

- Manages the flow of data over the physical media

- Responsible for error free data submission to the network layer

1. Physical layer function

\* Purpose:

- Deals with the transmission of 0’s and 1’s the physical media

\* Translation of bits into signals.

e.g –Pulse duration determination

-Transmission synchronization etc

Physical layer function:

* Encode bits into signals
* Carry data from the h higher layers
* Define interface to the card

-Electrical

-Mechanical

-Funcational

-e.g Pin count on the connection

What is Internet?

* It is a global network of computing resources
* It can be thought as a physical collection of routers and circuits as a set of shared resources

Internet Based services:

Email: fast, easy and inexpensive way to communicate with other internet users.

Telnet: allow users to transfer virtually every kind of file that can be stored on a computer from one Internet connected computer to another.

UseNet news: A distributed bulletin board that offers a combination news and discussion service on thousands of topics

World wide web: A hypertext interface to internet information resources.

What is WWW? World wide web

* All the resources and users on the internet that are using the hypertext transfer protocol (HTTP)
* TIM BERNERS LEE helped found, the world wide web consortium (W3c) : the world wide web is the universe of network- accessible information, an embodiment of human knowledge an embodiment of human knowledge.
* www is a way of exchanging information between computers on the Internet.

What is HTTP?

* Hypertext protocol
* This protocol is used to transfer hypertext documents that makes the world wide possible
* A standard web address such as yahoo.com is called a URL and here the prefix http indicates its protocol.

Basic Web concepts:

* Basic web concepts URL’s, HTML, SGML and XML
* HTTP
* MIME Media types
* Server side Programs

URL’s: A uniform resource Identifier is a string of characters in a particular syntax that identifies the Resource.

-A file on a server

-An Email address

-A news Message

Scheme: scheme –specific part

* Current scheme includes : data, file, ftp, http, mailto, telnet, urn
* The syntax of scheme specific part depends on the scheme being used.
* There is no specific syntax it has a hierarchial form like :

//authority/path?Query

URL e.g

<ftp://mp3:mp3@ci43198-a.ashvil1.nc.home.com:33/VanHalen-Jump.mp3>

Authority: mp3

Username:mp3

Password:mp3

Host:@ci43198-a.ashvil1.nc.home.com

Port:33

It has the scheme ftp

Path:/VanHalen-Jump.mp3

There are two types of URL’s

\*URL-UNFORM RESOURCE LOCATORS

\*URN-UNIFORM RESOURCE NAME

-URL is a pointer to a particular resource on the Internet at a particular location.

-URN is a name for a particular resource but without reference to a particular location.

URN:-

* URN has a general form

URN has a general form

Urn:namespace:resource\_name

* The namespace is the name of a collection of certain kinds of Resources maintained by some authority.
* The resource name is the name of a resource within that collection.

E.g.

The URN urn:ISBN:1565924851 identifies a resource in the ISBN namespace with the identifier 1565924851 of all books published this one selects the first edition of Java I/O

URL,s:-

* A URL identifies the location of a resource on the Internet
* It specifies the protocol used to access a server(e.g, FTP, HTTP), the name of the server, and the location of a file on that server

Syntax of URL

Protocol|| username@hostname:port||path/filename?query#fragment

* Protocol is another word for what was called the scheme of the URL
* The host name part of URL is the name of the server that provides the resource you want
* The username is an optional username for the server
* The port number is also optional
* The path points to a particular directory on the server specified.
* The path is relative to the document root of the server, not necessary to the root of the filesystem on the server.
* The filename point to a particular file in the directory specified by the path
* The query string provides additional arguments for the server.
* It is commonly used only in http URL’s where it contains Form data for Input to programs running on the server.
* Finally the fragment references a particular part of the remote resources.

Relative URL’s:

* URL’s that are not complete but inherit pieces from their parent are called relative URL’s
* In contrast a completely specified URL is called an absolute URL.

E.g.

<http://www.ibiblio.org/java.faq/Javatutorial.html>

you can click on this hyperlink

<ahref=”Javafaq.html”>

* If the relative link begins with a/, then it is relative to the document root instead of relative to current file.

HTML,SGML and XML

\*HTML is the primary format used for web documents

\*HTML is simple standard for describing the semantic content of textual data.

SGML:

* It is the idea of describing Text semantics rather than its appearance comes from the older standard called standard generalized markup language (SGML).
* Standard HTML is an instance of SGML
* SGML and by inheritance, HTML is an instance of SGML
* SGML and by in inheritance , HTML are based on the notation of design by meanin rather than design by appearance.
* H1 in HTML-> Top level heading (you don’t say 18 point type)
* <EM> in HTML-> Emphasize (you don’t say placed in Italics)

TAG:

* The tags used to markup the text, text are case sensitive. Thus,<STRONG> is same as <strong> is the same as<StROnG>
* Some tags have a matching end-Tag to define a region of text.
* An end-tag is the same as the start tag, except that the opening angle bracket is followed by a/
* For e.g <STRONG>this text is strong</STRONG>
* <EM>this text is emphasized </EM>

Element Attributes:

* Some elements have additional attributes that are encoded as name value pairs on the start Tag.

For e.g.

<H1 ALIGN=CENTER>This is centered H1 heading </H1>

The value of attributes may be enclosed in double or single.

<H1 ALIGN=”CENTER”>

<H2 Heading</H2>

* Quotes are requested only if it has embedded spaces.

XML:

* It is a semantic language that allows page authors to create the elements rather than relying on a few fixed Elements such as P and LI.

For e.g. if you are writing a web page with price list, you would have an SKU element, a Price element a manufacturer element a product element and so forth.

E.g.

<PRODUCT MANUFACTURER = “IBM”>

<NAME>LOTUS SMART SUIT</NAME>

<VERSION> 9.8</VERSION>

<PLATFORM> windows</PLATFORM>

<PRICE CURRENCY=”us”>299.95</PRICE>

<SKU>Do5WGML</SKU>

</PRODUCT>

* This looks like HTML
* There are elements and attributes
* Tags are setoff by <and>
* Attributes are enclosed in quotation marks
* You can create all the new unique tags you need.
* Since no browser can know in advance all the different elements that may appear, a stylesheet is used to described how each of the items should be displayed.

Advantages:

* HTML can be quite sloppy elements are opened but not closed
* Attribute values may or may not be enclosed in quotes
* XML lays out very strict requirements for the syntax of a well formed XML document, and it requires that browsers reject all malformed documents.

Document type Definition (DTD):

* An XML document may have a DTD which can improve additional constraints on valid document.

For e.g.

DTD may require that every product elememnt contain exactly one name element

* Main advantage is it is much easier to work with XML rather than HTML

HTTP:

* It is a standard protocol for communication between web browsers and web servers .
* HTTP specifies how a client and server establish a connection, how client requests data from the server. Responds to the request and finally how the connection is closed
* HTTP uses the TCP/IO protocol for data Transfer.

Four steps:For each request from client to server there is a sequence fo four steps:

* Making the connection
* Making the request
* The response
* Closing the connection

Making the connection:

* The client establishes a TCP connection to the server on PORT 80, by default other ports may be specified in the URL.

HTTP Request Headers:

Keyword :value

* The most common keyword is Accept which tells the server what kind of data the client can handle (though servers often ignore this)
* For E.g.

The MIME media types, corresponding to HTML documents

Accept:text/html,text/plain, image/gif, image/jpeg.

Plain text, JPEG and GIF images

HTTP Request Headers:

* User-Agent is another common keyword that lets the server know what type. The line below says the request comes from version 2.4 of Lynx Browser.

Browser:

User-agent:Lynx/2,4 libwww/2.1.4

* All but the oldest first generation browsers also include a Host field specifying the servers name, which allows web servers to distinguish between different named hosts served from the same IP address.

e.g.

Host:www.cafeaulait.org

The Response:

* The server sends a response to client.
* The response begins with a response code, followed by a header full of metadata, a blank line, and the requested document or an error message.

HTTP Response Example:

HTTP/1.1 2000k

Date:Mon,13Jan2015 21:06:50GMT

Server:Appache/2.0.40(RedHat Linux)

Connection:close

Content type :text/html;charset=ISO-8859-1

Content Length:107

<html><head><title A sample HTML file>

</title>

</head><body>the rest of the document here

</body></html>

Server side Program:

* The content may be pulled from a database or generated algorithmically by a program
* In java server side programs are written using servlets or java server pages(JSP)
* They can also be written in other languages such as C and Perl, or frameworks, such as ASP and PHP.

HTTP 1.1:

* Opens a new connector for every requests
* The primary improvement in HTTP1.1 allows a browser to send may different requests over a single connection the connection remains open until it is explicitly closed
* The requests and responses are all asynchronous a browser doesn’t need to wait for a response to its first request before sending a second or third.

MIME media types:

* An open standard for sending multipart, multimedia data through internet mail.
* Describes a file content so that client software can tell the difference between different kinds of data

For e.g.,

A web browser uses MIME to tell whether a file is a GIF image or a printable post script file.

Type and Subtype:

* MIME supports more than 100 predefined types of content.
* Content types are classified at two levels a type and a subtype
* The type shows generally what kind of data is contained : is it a picture text or movie
* The subtype identifies the specific types of data: GIF image, JPEG Image, TIFF image

e.g.

\* HTML’s content type is text.html the type is text and the subtype is html

\* the content type for a GIF image is image/gig; the type is image, and the subtype is gif.

\* web servers use MIME to identify the kind of data they are sending

\* web clients use MIME to identify the kind of data they are willing to accept

\* most web servers and clients understand atleast two MIMe text content types text/html and text/plain and two image formats image/gif and image/jpeg

\* more recent browsers also understand application/xml and several other image formats.

Server-side programs:

* These days many web pages not served from static files on the hard drive
* The server generates them dynamically to meet user requests.

Response Code:

* A response code from 200to299 always indicates success.
* A response code from 300to 399 always indicates redirection.
* A response code from 400to499 always indicates a client error.
* A response code from 500to599 indicates a server error.

Closing the connection:

* Either the client or the server or both close the connection.
* Thus, a separate network, connection is used for each request.
* If the client reconnects the server retains no memory of the previous connection or its results.