CSC 402 – Internet Technology

Recap

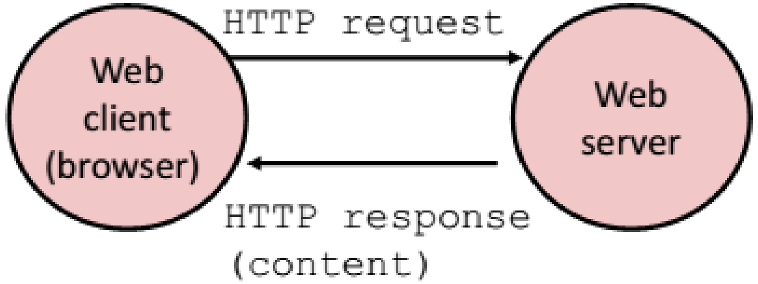
- Labels – scope and uniqueness
- LER
- LSR
- LSP
- LDP
- LIB

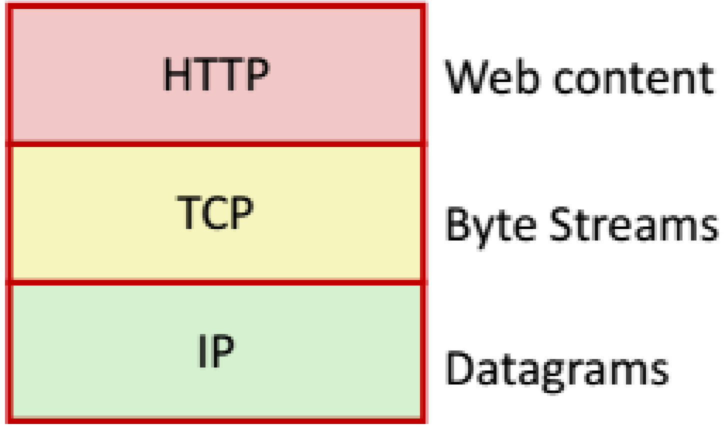
Web

- A Web site is a location on the Internet where an individual, company, or organization keeps its Web pages and related files (such as graphic and video files).
- A Web page is displayed on your computer screen by a Web browser.
  - First web browser NSCA Mosaic (1993, University of Illinois), then came Navigator (1994, Netscape), and then Internet Explorer (1996, Microsoft). The initiation of “Browser War”.
- A person can retrieve and open a Web page in a Web browser either by entering a URL in the Web browser’s address box or by clicking a hypertext link.
- When a user wants to access a Web page using either method, the user’s Web browser sends a Web server a request for the Web page.
- A Web server is a computer that delivers Web pages.
- The Web server’s reaction to the user’s request is called the response.

Web Communication

- In 1989 Tim Berners-Lee (CERN) writes proposal on a distributed hypertext system.
  - Connects "a web of notes with links". Text with links to other text.
- Clients and servers communicate using Hyper Text Transfer Protocol (HTTP).
  - Current version is HTTP/1.1.
  - RFC 2616, June, 1999.





Web Communication

- A common misconception is that the words “Web” and Internet are synonymous.
- The Web is only one part of the Internet, and is a means of communicating on the Internet. The Internet is composed of other communication elements, such as e-mail systems that send and receive messages.
- Because of its enormous influence on computing, communications, and the economy, the World Wide Web is arguably the most important part of the Internet today.

Web Service

- Web services are software components that communicate using pervasive, standards-based Web technologies including HTTP and XML-based messaging.
  - An HTML program (page) contains instructions (tags) that tell the browser how to display various text and graphical objects in the page.
- They are designed to be accessed by other applications and vary in complexity from simple operations, such as checking a banking account balance online, to complex processes running CRM (customer relationship management) or enterprise resource planning (ERP) systems.
- Web services are hardware, programming language, and operating system independent.
  - This means that applications written in different programming languages and running on different platforms can seamlessly exchange data over intranets or the Internet using Web services.
  - Web services are XML-based information exchange systems that use the Internet for direct application-to-application interaction. These systems can include programs, objects, messages, or documents.
- Web services are self-contained, modular, distributed, dynamic applications that can be described, published, located, or invoked over the network to create products, processes, and supply chains.
  - These applications can be local, distributed, or web-based. Web services are built on top of open standards such as TCP/IP, HTTP, Java, HTML, and XML.
- Web services are powered by XML and three other core technologies:
  - **WSDL** (Web Service Description Language), **SOAP** (Simple Object Access Protocol), and **UDDI** (Universal Description, Discovery, and Integration).

Web Service

- Uses a standardized XML messaging system and is available over the Internet or private (intranet) networks.
- Is not tied to any one operating system or programming language.
- Is self-describing via a common XML grammar.
- Is discoverable via a simple find mechanism.
- How it works:
  - XML to tag the data.
  - SOAP to transfer a message.
  - WSDL to describe the availability of service.
- Why Web Service:
  - **Interoperability**: e.g. VB or .NET application can talk to Java web services and vice versa.
  - **Standardized Protocol**.
  - **Low Cost communication**: uses use SOAP over HTTP protocol and can also use FTP. Cheap as compared to their expensive proprietary cousins EDI/B2B (Electronic data interchange/Business-to-Business).
  - **XML based**: Using XML eliminates any networking, operating system, or platform binding.
  - **Flexibility**: Ability to be Synchronous or Asynchronous. In synchronous invocations, the client blocks and waits for the service to complete its operation before continuing. Asynchronous operations allow a client to invoke a service and then execute other functions.

Web Service

- Example: Consider an account-management and order processing system.
  - The client program bundles the account registration information into a SOAP message.
  - This SOAP message is sent to the web service as the body of an HTTP POST request.
  - The web service unpacks the SOAP request and converts it into a command that the application can understand.
  - The application processes the information as required and responds with a new unique account number for that customer.
  - Next, the web service packages the response into another SOAP message, which it sends back to the client program in response to its HTTP request.
  - The client program unpacks the SOAP message to obtain the results of the account registration process.

Web Service vs Data File Transfer

- Traditionally data file transfers are done using FTP. Some of the disadvantages of using FTP (or similar) for data transfer are listed below.
  - Keep an extra user/access administration.
  - Create file/directory structures.
  - Manage queueing and cleanup of files that you need to copy into those locations.
  - Poll for new files.
  - Most companies with connections to the Internet have implemented firewall solutions which disable FTP access.
  - Custom scripts need to be developed for the automated exchange of data.

Web Service vs Data File Transfer

- Using web services for data transfer has lot of advantages, below are few
  - Based on industry standards HTTP, XML etc.
  - Secure access to data. Web services can tightly control access to the data and services they make available to other applications.
  - Widely supported across all major ERP’s and business solutions.
  - No custom scripts needed for automated data exchange.
  - Near real time data transfer.
  - No special user/access administration needed.
  - Corporate firewalls rarely restrict HTTP access.

Web Content

- Web servers return content to clients
  - content: a sequence of bytes with an associated MIME (Multipurpose Internet Mail Extensions) type
- Example MIME types
  - text/html: HTML document
  - text/plain: Unformatted text
  - application/postscript: PostScript document
  - image/gif: Binary image encoded in GIF format
  - image/jpeg: Binary image encoded in JPEG format
- Content returned in HTTP responses can be static or dynamic:
  - Static content
  - Dynamic content

Static and Dynamic Content

- Static: content stored in files
  - Examples: HTML files, images, audio clips.
  - Request identifies content file
- Dynamic: content produced on-the-fly in response to an HTTP request
  - Example: content produced by a program executed by the server on behalf of the client.
  - Request identifies file containing executable code
- All Web content is associated with a file that is managed by the server.

URL (Universal Resource Locator)

- A.k.a "Web address".
- Every piece of content returned by a Web server is associated with some file that it manages.
- Each of these files has a unique name known as a URL(Universal Resource Locator) i.e. name for a web content.
- URLs for static content:
  - http://www.deerwalk.edu.np/courses/spring16/CSC-40201-003/index.html
- URLs for dynamic content:
  - http://www.deerwalk.edu.np/~th1133/cgi-bin/adder?235&123
  - Here, "cgi-bin" identifies an executables called adder with arguments as "235" and "123".
- URL schemes:
  - **http**: is the most common scheme; it means use the HTTP protocol.
  - **https**: is similar to http: except that it uses SSL encryption.
  - **file**: means read a file from the local disk.
  - **mailto**: means open an email program composing a message.

URL Parts

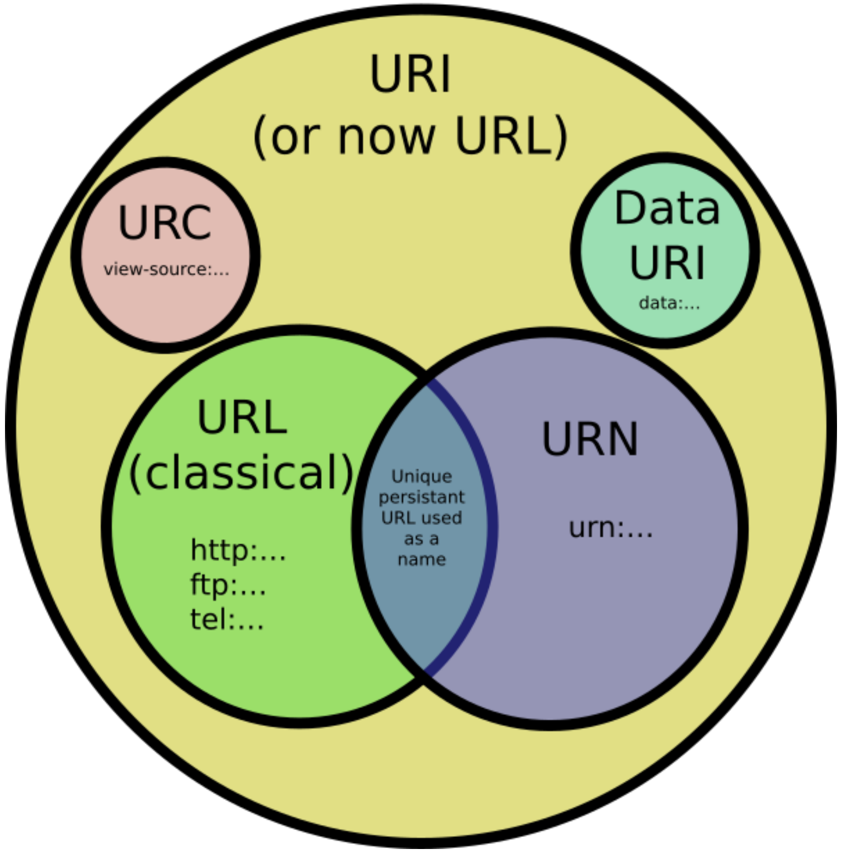
- http://host.company.com:80/a/b/c.html?user=Alice&year=2014#p2
- **Scheme (http:)**: identifies protocol used to fetch the content.
- **Host name (//host.company.com)**: name of a machine to connect to.
- **Server's port number (:80)**: allows multiple servers to run on the same machine. Optional
- **Hierarchical portion (/a/b/c.html)**: used by server to find content.
- **Query parameters (?user=Alice&year=2014)**: provides additional parameters.
- **Fragment (#p2)**: Have browser scroll page to fragment (html: p2 is anchor tag)
  - Used on the browser only; not sent to the server

URI, URN, and URL

- **Resource**: A resource can be anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), and a collection of other resources.
  - A resource can remain constant even when its content—the entities to which it currently corresponds—changes over time, provided that the conceptual mapping is not changed in the process.
- **URLocator**: A general resource locator is an object that describes the location of a resource.
  - An object: the structure of the sequence of bytes that comprise the locator.
  - A descriptor of location: location can mean many things – lat/long coordinates, a (protocol, server address, pathname) triple. The same resource can have multiple location. Descriptors (ftp and http URLs)
- **URName**: A persistent identifier for a resource. Key idea is that a name will not change over the life of the resource, whereas a locator might. A name is not necessarily usable to retrieve the resource.
- **URIdentifier**: An identifier is an object that can act as a reference to something that has identity (i.e. a resource). In the case of URI, the object is a sequence of characters with a restricted syntax.
  - URI is the most general kind of identifier. A URI may also be a locator (URL), or a name (URN), or both.
  - {scheme}:{scheme-specific parts}

URI, URN, and URL

- URL is a type of URI. All pigeons can fly but everything that flies is not a pigeon. Whereas a URN is globally unique part of identification.
    - “Access mechanism” makes URI, a URL. E.g. http:// or ftp://.
- URI example: URLs and URNs are all types of URI.
- URN example:
    - urn:isbn:0451450523 to identify a book by its ISBN number.
    - urn:uuid:6e8bc430-9c3a-11d9-9669-0800200c9a66 a globally unique identifier.
    - urn:publishing:book - An XML namespace that identifies the document as a type of book.
- URL example:
    - http://example.com/mypage.html
    - ftp://example.com/download.zip
    - mailto:user@example.com
    - file:///home/user/file.txt



Venn diagram of URIs as defined by the W3C