Network Services

Module 8

Network Services

- Enabling
  - Domain Name Service (DNS)
  - Directory Services
  - Network Time Services
  - Routing protocols (ex. BGP, OSPF, RIP)
- User Services
  - Messaging
    - Real time (ex. Instant Messaging)
    - Store and forward (ex. Mail)
  - Interactive Session Services
  - File Services
  - Content Services (ex. Web)
  - Other (ex. Printing, Telephony)

Focus

- Fundamental Services
  - DNS
  - Mail
  - Interactive Session Services
  - File Services
  - Web Services

DNS

- Purpose:
  - Provide name to address resolution services and vice versa (ex. www.iastate.edu => 129.186.23.166)
    - Names are a human convenience but meaningless to the IP layer
  - Provide simple directory services
    - Mail routing is dependent on DNS
      - Entries tell the mail system which host(s) receives mail for a domain
    - Domain naming structure dependent on “name server” records

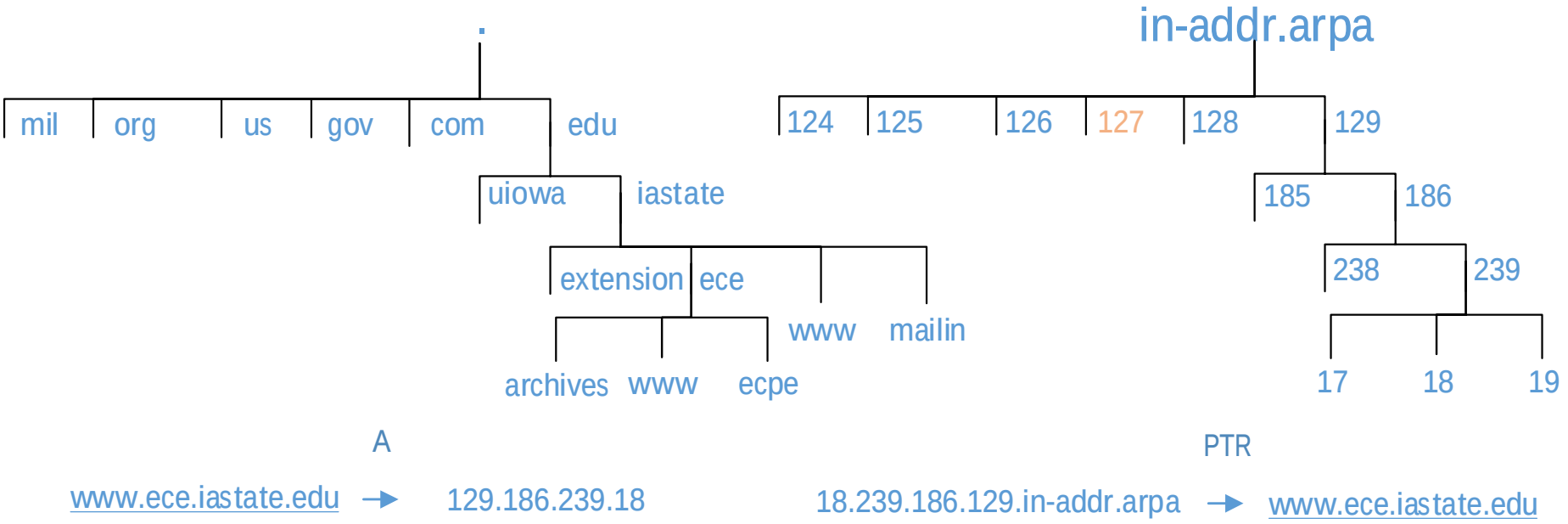
DNS Architecture

- Distributed service managed by many independent organizations
  - The domain name: www.ece.iastate.edu
    - Looks unified, but is actually an assembly of parts
    - DNS responses appear as if a single DNS system provides an answer to the assembled parts
    - Finding the address requires:
      - Finding the name server for edu
      - Finding the name server for iastate
      - Finding the name server for ece
      - Asking the ece name server for www's address

DNS Architecture

- DNS logically structured like an inverted tree very similar to a file system
- The “hidden” domain
  - The root of the DNS “tree” is called the “root domain” and notation for this is “.”
  - The root name servers are extremely busy
    - They provide an answer to many DNS requests
    - They point inquirers to the top level domain name servers they have on record

DNS Architecture



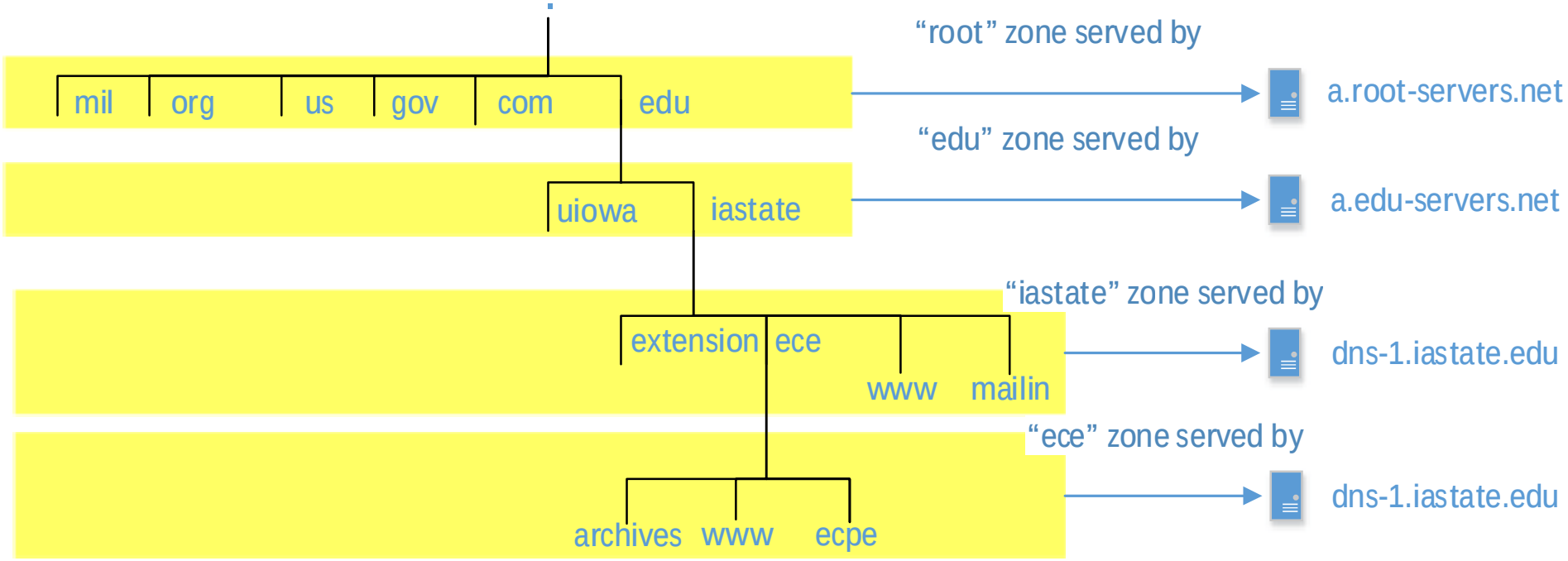
DNS Administration

- Every name server is configured with a list of root servers, so they know where to start
- Traditionally DNS records are stored in text files locally on name servers
  - These files have only records relevant to a portion of the DNS naming structure
  - They are called “zone” files

DNS Administration

- Zone files consist of a variety of record types
  - A record – Maps host name to address
  - NS record – Provides names of hosts serving the current zone or a subdomain
  - MX record – Provides names of hosts accepting mail for a domain
  - CNAME record – Serves as a nickname allowing one name to map to another
  - PTR record – Provides a name for an address (reverse of A record)
  - SOA record – Statement Of Authority record provides important administrative information for a zone

DNS Architecture



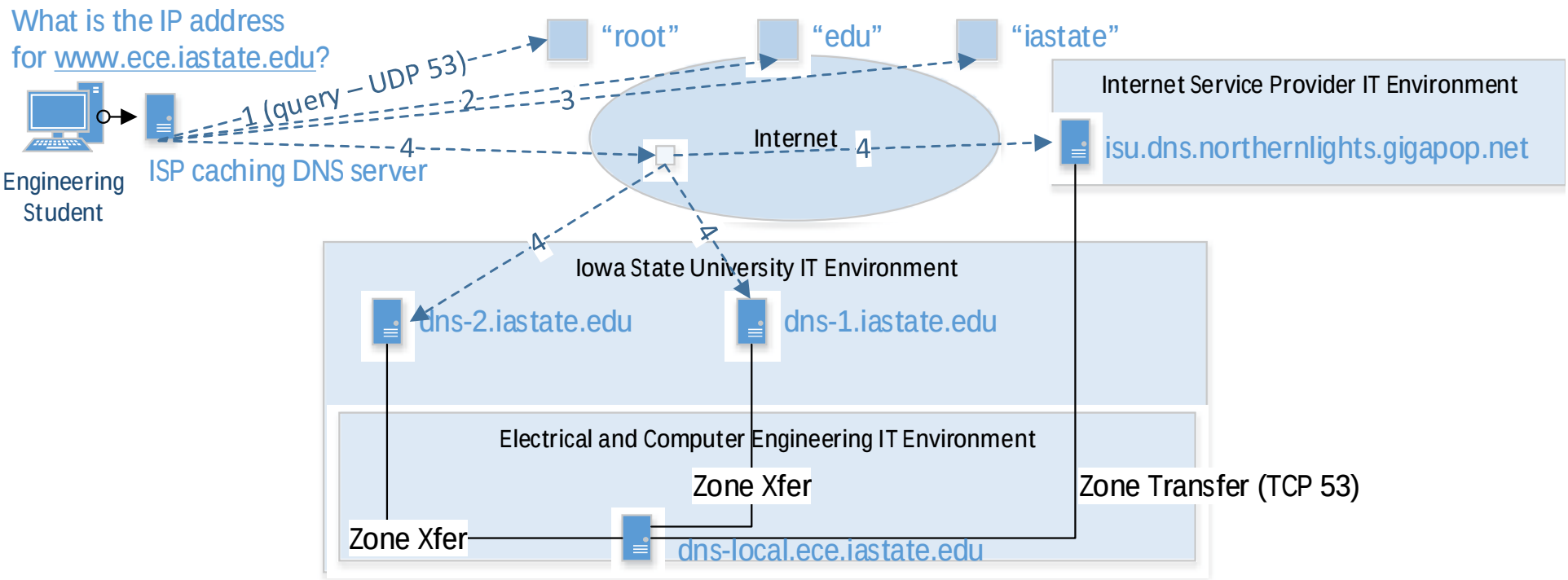
DNS Administration

- Name servers have a configuration file that map zone files to each portion of the naming hierarchy it serves.
  - a.edu-servers.net serves “edu”
    - The zone file for “edu” contains NS records for “iastate”, which point to names not addresses
    - If “iastate” name servers’ names are in subdomain of “edu” (ex. dns-1.iastate.edu), then zone file will contain A records for those servers
      - These A records are called “glue” records

DNS Architecture

- 3 Name server roles
  - Primary – Holds the authoritative records
  - Secondary – Serves zone with records obtained from Primary
  - Caching – Serves no zone, but seeks out answers to client queries
    - Responses to queries are saved for a period of time in case another client asks the same question
      - Speeds up name resolution
      - Reduces burden on Primary and Secondary servers

DNS Architecture



DNS Client Tools

- Applications use resolver libraries
- Testing or troubleshooting tools available
  - nslookup – once common on Unix, but now primarily a Windows command line tool
  - dig - common tool on Unix
- By default, these tools use the same configured name servers as those applications/services using resolver libraries
- These tools can be instructed to use a different server – very useful for troubleshooting
  - Remember firewall must allow UDP 53 outbound if query is for server beyond firewall

Split DNS

- Purpose
  - Standard DNS assumes the Internet based name servers are accessible and only one copy of a domain exists
    - There is only one ece.iastate.edu
  - Perimeters established using firewalls and the use of NAT complicates things

Split DNS

- DNS clients on the Internet and clients behind the firewall will access same zone information by default
  - But, two different zones are needed
  - Outside clients have access to public systems
  - Inside clients have access to internal and public systems
    - Private addresses make internal systems hard to get to for a client on the Internet
    - We do not want to reveal internal systems information to the public

Split DNS

- Function
  - Using features of standard DNS servers a special architecture can be implemented
    - Original and reference implementation of DNS protocols is a program called “bind”.
  - The world will only have access to records related to publicly available servers
  - Internal clients will access records for internal and public systems

Split DNS

- Design
  - Public Zone
    - Name servers registered with parent domain will be accessible to Internet users
    - Zone information contains records for systems designated as Internet accessible
  - Private/Internal Zone
    - Internal name servers will be authoritative for same domain
    - Internal clients will be configured to resolve to these internal servers

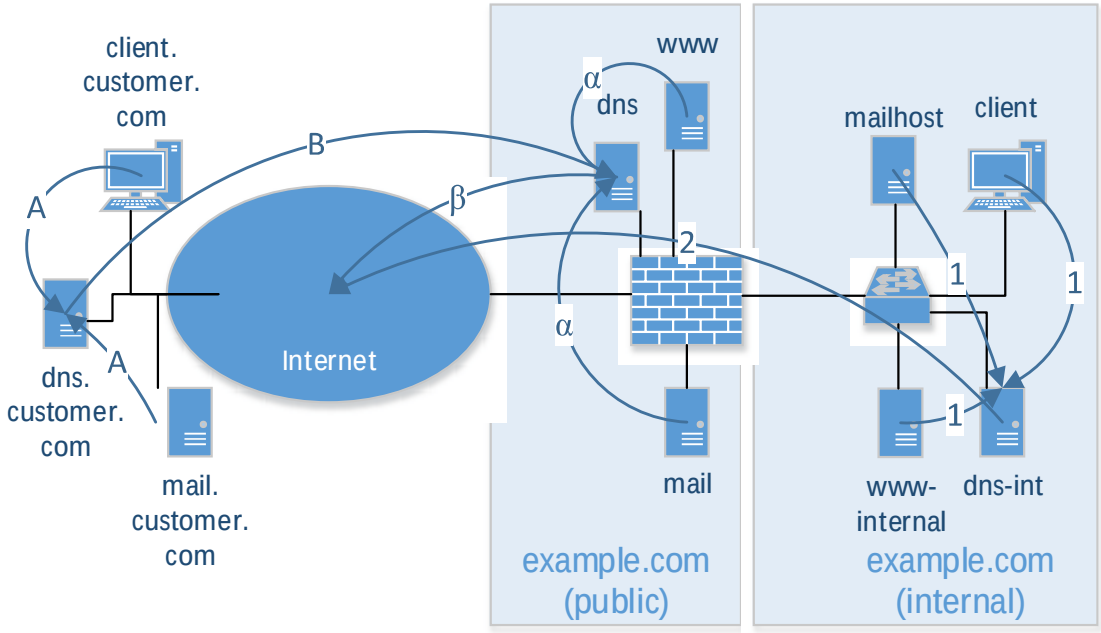
Split DNS

- Private/Internal Zone
  - An organization has a choice
    - Do we let any name server query Internet servers?
      - Performance can be improved if a common server is queried, because it will cache frequently requested records
      - Security vulnerabilities in DNS name servers can be exploited by Internet query replies
      - Limiting servers means the designated servers must be reliable and handle a possibly large query load
      - Fewer systems serving the internal zone is easier to manage

Split DNS

- Private/Internal Zone
  - If restricting access to Internet name servers is chosen, DNS can comply
    - Name servers can be configured with one or more “forwarder” directive as well as the “slave” directive
      - Forwarder tells server “If answer to query is not available, pass query to server specified in ‘forwarder’ directive”
      - Slave tells server “Hold tight. Wait for reply by ‘forwarder’, and give up if no reply is received”

Split DNS



example.com (internal zone)

	IN	NS	dns-int.example.com
www	IN	A	128.89.10.100
dns	IN	A	128.89.10.101
mail	IN	A	128.89.10.102
mailhost	IN	A	10.10.10.10
www-internal	IN	A	10.10.10.11
dns-int	IN	A	10.10.10.12
client	IN	A	10.10.10.221

example.com (external zone)

	IN	NS	dns.example.com
www	IN	A	128.89.10.100
dns	IN	A	128.89.10.101
mail	IN	A	128.89.10.102
mailhost	IN	A	10.10.10.10

Mail

- Purpose:
  - Provide a service like the postal system
  - Send messages consisting of ASCII text to a remote server on a best effort basis
- Design
  - Accommodate unreliable and slow connections connecting institutions to the ARPANET
  - Users will logon to servers to read mail

Mail Today

- Expectation
  - Highly reliable service with which organizations can conduct business
  - Large multimedia content in messages
  - Read, compose and send messages from any available platform in any location

Mail Today

- Design
  - Additional mail protocols needed
  - Merging of services – ex. Web mail
  - Dedicated servers hosting user mailboxes
    - Mail has become an archive
    - Mechanism for routing content like contracts, presentations and spreadsheets
  - Mail client software on personal computers, devices and for web access
  - Message authenticity, confidentiality, integrity are not built in services

Mail Protocols

- SMTP – simple mail transfer protocol – used to send email
  - Traditionally unauthenticated and no encryption
- POP3 – 3rd version of Post Office Protocol – used for email retrieval by mail clients
- IMAP – Internet Message Access Protocol – used for email retrieval by mail clients

Message Structure

- Two Parts
  - SMTP Envelope
    - Used for routing
  - Message body
    - Headers MTAs insert during routing
    - Payload of message
      - ASCII text
      - Binary content (ex. images, sound, documents) is encoded into ASCII using various schemes
        - » Encoding is essentially a tunnel to satisfy protocol limitations in order to achieve a modern need

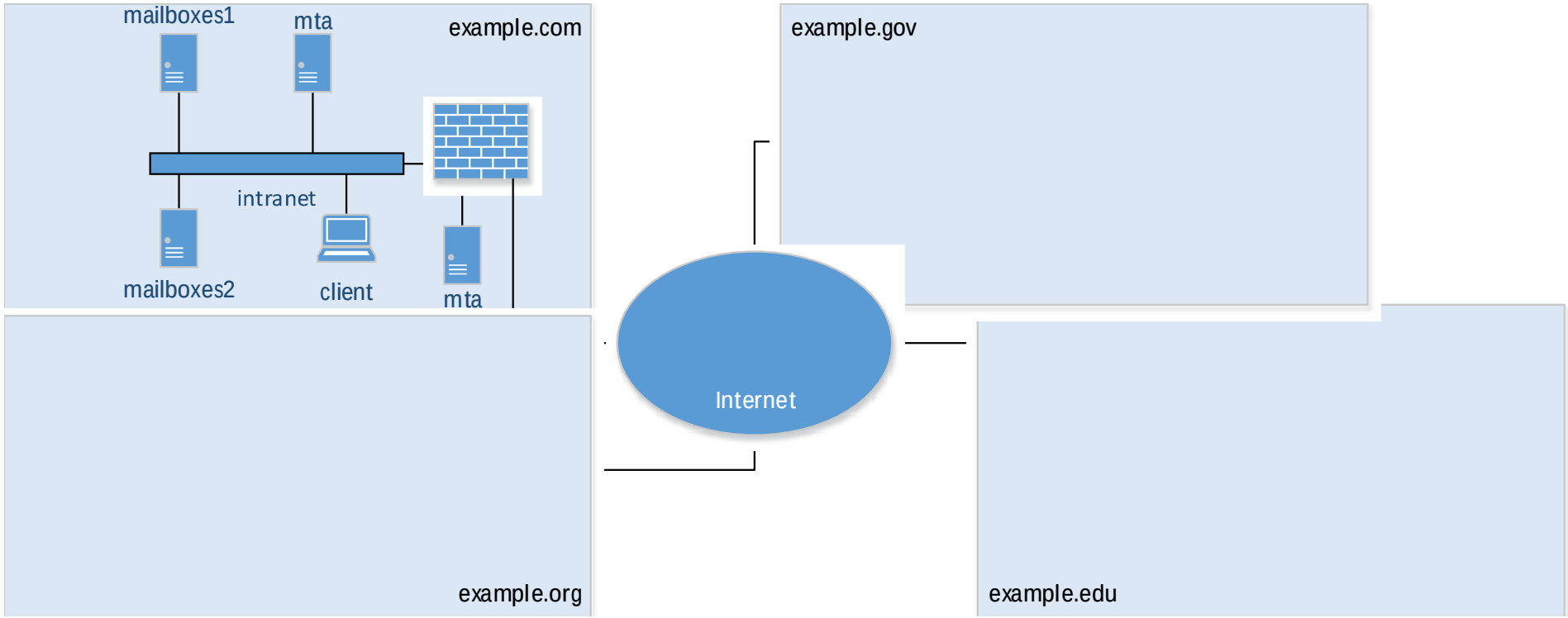
Mail Routing

- Email addresses are in the form: *recipient@domain*
  - Recipient can be a process, person or group
- Domains as defined in DNS are the key to message routing
  - The recipient is ignored until last stage of routing
- Message Transfer Agent – Service directs message to another intermediate MTA or final destination host
  - An MTA can be the final destination host

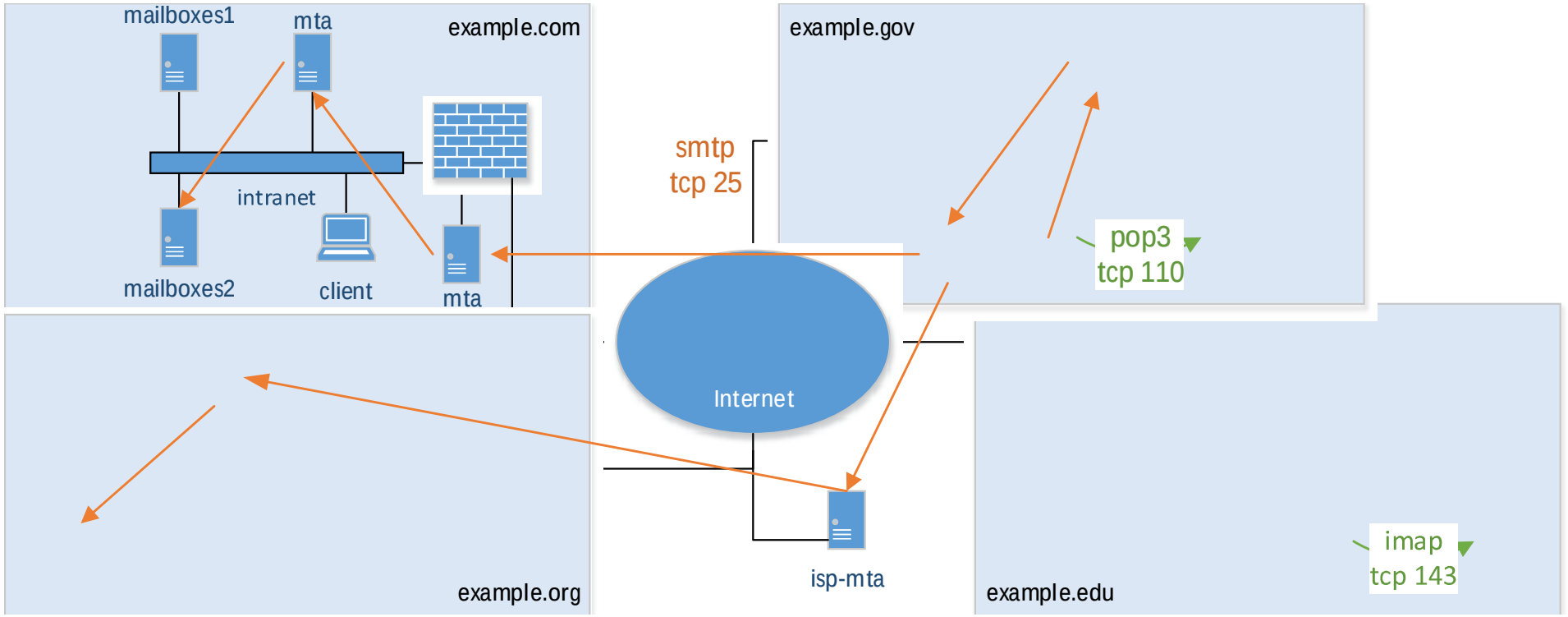
Mail Routing

- Sendmail – an MTA service that is traditionally available on Unix systems
  - Highly flexible routing service
    - Once used to route mail to and from a variety of mail systems, which had different addressing conventions, content conventions and routing schemes
    - Flexibility comes from configuration files
  - Traditionally this service listened to port 25, made routing decision and passed message via SMTP to next MTA (if necessary)
  - Long history of vulnerabilities due to its complexity
    - Various approaches attempt to minimize risk

Mail Architecture



Mail Architecture



Mail and DNS

- example.com

  IN MX 10 mta.example.com

- example.org

  IN MX 10 mta.example.org

  IN MX 100 isp-mta.isp.net

Mail Configuration

- Coordinate DNS to reflect desired mail address convention
- Configure mail services to receive mail for established address convention
  - Be sure inbound mail gets directed to mailbox host
- Configure server(s) to host mailboxes
  - May need imap and pop3 to support mail clients
- Configure mail services to route outbound mail per desired data flow

Mail Administration

- Managing mailboxes
  - Create, migrate, disable and remove
  - Size monitoring in space limited environments
- Managing aliases
  - Mapping generic recipients to real people
    - Ex. info@example.com, midwestsales@example.com
- Security concerns
  - Malware
  - Spam
  - Open mail relay

- Spoofing
- Phishing
- Message confidentiality

Interactive Session Services

- Two broad categories
  - Console – accessing system using attached user interface devices (keyboard,mouse,etc)
  - Remote – accessing system from a remote computer or user device
    - Focus of this section
- Purpose
  - Provide administrators and users a means to interact with OS and applications residing on a system.

Interactive Session Services

- Architecture
  - Point-to-point communications
  - Client – server service design
  - Authentication
    - Commonly relies OS to provide auth. services
    - Can provide alternatives – not always better
  - Confidentiality is provided on a per service basis

Interactive Session Services

- User experience
  - Text interface – interaction limited to keyboard
    - Ex. telnet, rlogin, ssh
  - Graphical user interface – interaction with mouse and keyboard
    - Ex. Remote Desktop Client (Windows), Virtual Network Computing (VNC), X Windows

Focus

- Text interface
  - telnet, ssh
- Graphical user interface
  - Remote desktop client – uses remote desktop protocol

Telnet

- TCP port 23
- Service provides command line access to remote server
- Provides no confidentiality services
  - All traffic is readable
- Authentication relies on OS to provide related services
- On Unix – “telnetd” is the service, “telnet” is the client

SSH

- Secure Shell typically runs on TCP port 22
- Fairly standard service on Unix
- Provides communications confidentiality
- Designed to be a secure substitute for rlogin, rexec, FTP and telnet
- Supports variety of user authentication
- Client authenticates server to ensure user knows if a substitute server has been introduced
  - Substitution is suspected to be malicious by default
- On Unix – “sshd” is the service, “ssh” is the client

Remote Desktop Connection

- TCP port 3389
- Microsoft protocol
  - Service is built into Windows OS
- User experiences nearly complete Windows interactivity and functionality
- Session is encrypted using Transport Layer Security (TLS) 1.0
  - Server side authentication is one benefit

File Services

- Purpose
  - Share files with other people or computers/devices to which you have access
  - Relocate files to a desirable location
  - Offload or extend local disk storage capacity by having files stored on a remote readily accessible storage system
    - OS mounts storage regularly and integrates it in such a way that physical location loses significance

File Services

- Two broad categories
  - Intermittent – files storage is accessed as the need arises. No connectivity persists after need is addressed.
  - Continuous – file storage is mounted prior to the need arises. Connectivity persists whether or not the the storage is accessed.
    - Services that support this feature can be mounted temporary as the need dictates

Architecture

- Point-to-point communications
  - Common usage: Service only involves a service requestor and service provider
  - Proxies are an exception for intermittent services
- Client-server design

File Services

- Continuous
  - NFS
  - Windows File Services/SMB/CIFS
- Intermittent
  - FTP
  - SFTP
  - HTTP
  - rcp, scp

Focus

- Continuous
  - NFS
  - Windows File Services/SMB/CIFS
- Intermittent
  - FTP
  - SFTP

NFS

- TCP port 2049
- Unix file sharing solution
- Current version 4 specified in RFC 3530
- Server shares portion of its local file system
- Client associates those shares to mount points similar to local disk partitions or removable media
- File level permissions managed by server file system
  - User ID of user on client system must match user id on server
- Historically
  - Authentication of user done by client computer
  - Access control by server done by client IP address and consistent use of client's source port

SMB/CIFS File Services

- TCP port 445, with NetBIOS UDP ports 137, 138 & TCP ports 137, 139
- Samba is Unix project to support CIFS
  - cifs-utils is package on Linux
- Native to Windows –Windows Explorer provides interfaces for sharing and using network shares
  - Servers in Windows domain will use Active Directory for authentication
  - Windows systems not in domain will use local user management
- Command line tools in Windows available
  - Ex. net use, net share,

FTP

- TCP 21 – control channel, 2nd TCP port for data channel
- Default data transfer type is ASCII
  - Binary files should be transferred in “binary” mode
- All communication is visible on network
  - User credentials and file contents
- Clients – command line, browsers, WinSCP
- Server – available for Unix and Windows (part of IIS services bundle)
- Anonymous FTP – Server does not require user credentials to be issued, so anyone can use

SFTP – SSH File Transfer Protocol

- Uses SSH channel on TCP port 22
  - Relies on SSH to negotiate authentication
- Similar user commands to FTP, but very different protocol
- Transfer is always in binary mode
- All communications are encrypted
- Servers available for Unix, Windows versions commonly free for only personal or non-commercial use
  - sshd commonly configured during Unix installation

Administrative Concerns

- Authentication – Do we really know who is accessing the files?
- Access Control – Are only authorized people or processes doing only authorized things to files?
- Availability – Are the files accessible when they are needed?
- Confidentiality – Are people able to see sensitive content without having to authenticate or needing access permissions?

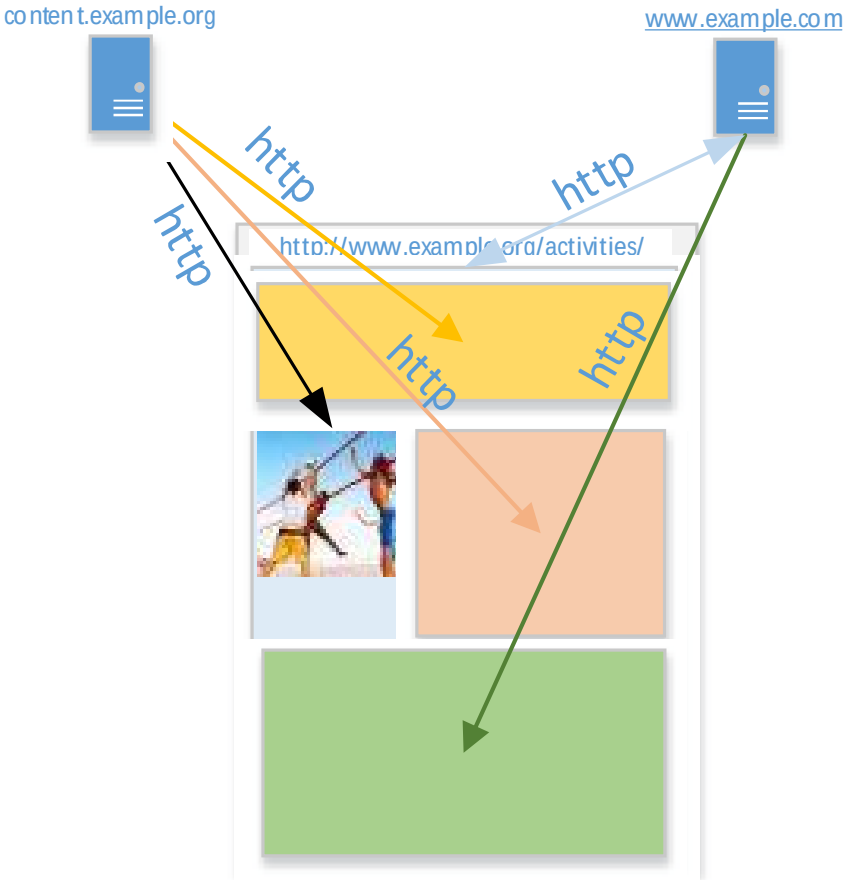
Web – World Wide Web

- Purpose
  - Make conceptually interrelated content accessible by interlinking content by reference on documents or pages that can be read and presented on various user platforms.
    - Content need not be limited to text

World Wide Web

- Architecture
  - Client-server application design
  - Communications between client and server are point-to-point
  - Browser's automatic fetching of embedded content results in multiple potential connections for one page.
    - Content need not be located on same server
  - Relies on multiple protocols and standards

Web Page Access



http://www.example.org/activities/

WWW Protocols

- HTTP
  - Service registered to use TCP port 80
    - In mid to late 1990's, it was common to have web servers listen on other ports (ex. 8080)
  - An unauthenticated plaintext file transfer protocol that passes commands and data in the same channel
  - This application layer protocol considers session ended after content retrieval is complete
  - Protocol provides no inter-page session continuity
    - A users "session" with a web site must be maintained above the application layer by browser and web server

WWW Protocols

- HTTPS = HTTP + SSL/TLS
  - Service typically assigned TCP port 443
  - https is a combination of two protocols that both client and server are prepared to handle
    - TLS provides authenticated encrypted tunnel
    - HTTP functions normally after TLS tunnel is established
  - TLS authentication requires certificates and means to verify certificates are trustworthy
    - Authentication is typically limited to server-side authentication

WWW Protocols

- HTML
  - Not a communications protocol
  - An inline markup language that instructs the client on how the author intended for the content to be presented
  - Language provides the linkage instructions that allows content from multiple files to be retrieved and integrated in order for a user to see a unified multimedia document
  - Originally designed for static content management
    - Today, HTML is commonly generated dynamically by the application prior to providing the page to the browser
  - Allows for scripting to be embedded and referenced within the page enabling pages to be interactive

WWW Protocols

- URI – Uniform Resource Identifier
  - Identifies an information resource and the means to get access to it
  - Uniform Resource Locator (URL) and Uniform Resource Name (URN) are URIs as well as the result of concatenating the URL and URN
  - 

  - URL – Identifies location and access method
  - URN – Identifies the name of information resource

Administrative Concerns

- Availability – Keeping web presence online
- Content integrity – Content available to the public has not be altered improperly
- User security and privacy – User is not put at risk by using web application
- Content management – Providing users with engaging and current content
- Performance – Providing a user experience that is responsive to most user connections