Internet and Intranet

Lecture by:

Jalauddin Mansur

July 2015

Chapter 5: Designing Internet Systems and Servers

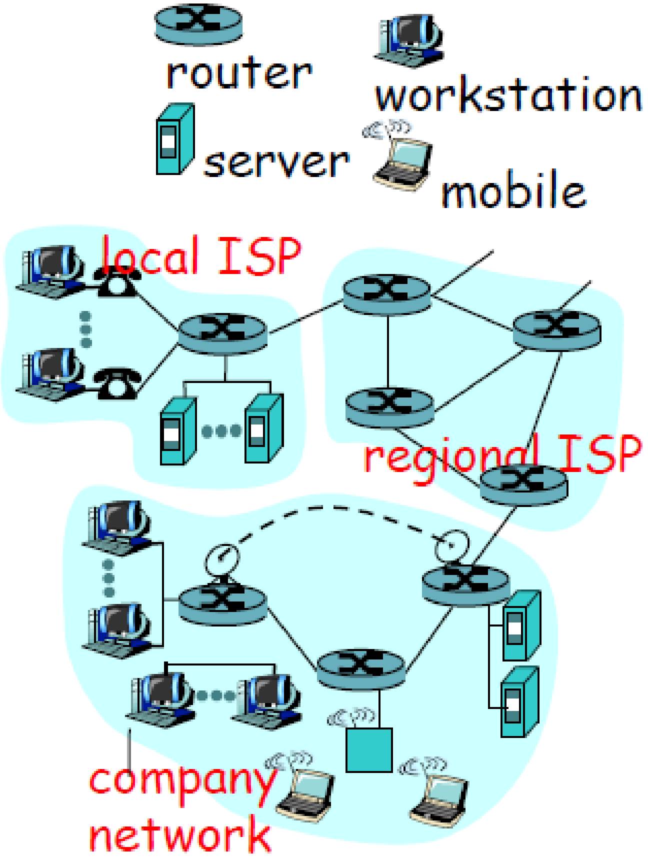
Topics :

- Designing of Internet System Network Architecture
- Choice of Platforms
- Server Concepts: Web, Proxy, RADIUS, MAIL
- Cookies
- Load Balancing: Proxy Arrays
- Server Setup and Configuration Guidelines
- Security and System Administration Issues, Firewalls and Content Filtering

Designing of Internet System Network Architecture

What's the Internet: "nuts and bolts" view

- Millions of connected computing devices: *hosts, end-systems*
  - PCs, servers, PDAs, phones, etc. running *network* apps
- Communication links
  - Fiber, cable, radio, satellite
  - Residential access: modem, DSL, cable modem, satellite
  - Transmission rate = ***bandwidth***
- Routers: forward packets (chunks *of data)*



When those architectural techniques are used in the field of internet networking technology, it is referred as internet network architecture

Network Design and Architecture

- is of critical importance
- contributes directly to the success of the network
- contributes directly to the failure of the network

In the Internet era,

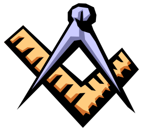
- ▪reliability is something you have to build, not something you buy.
- ▪hard work requires intelligence, skills and budget.
- ▪Reliability is not part of the basic package.

What is a Well-Designed Network?

- A network that takes into consideration these important factors:
  - Physical infrastructure
  - Topological/protocol hierarchy
  - Scaling and Redundancy
  - Addressing aggregation (IGP and BGP)
  - Policy implementation (core/edge)
  - Management/maintenance/operations
  - Cost

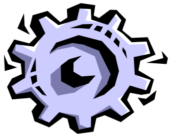
The Three-legged Stool

- Designing the network with resiliency in mind
- Using technology to identify and eliminate single points of failure
- Having processes in place to reduce the risk of human error
- All of these elements are necessary, and all interact with each other
    - One missing leg results in a stool which will not stand



Design

Technology



Process

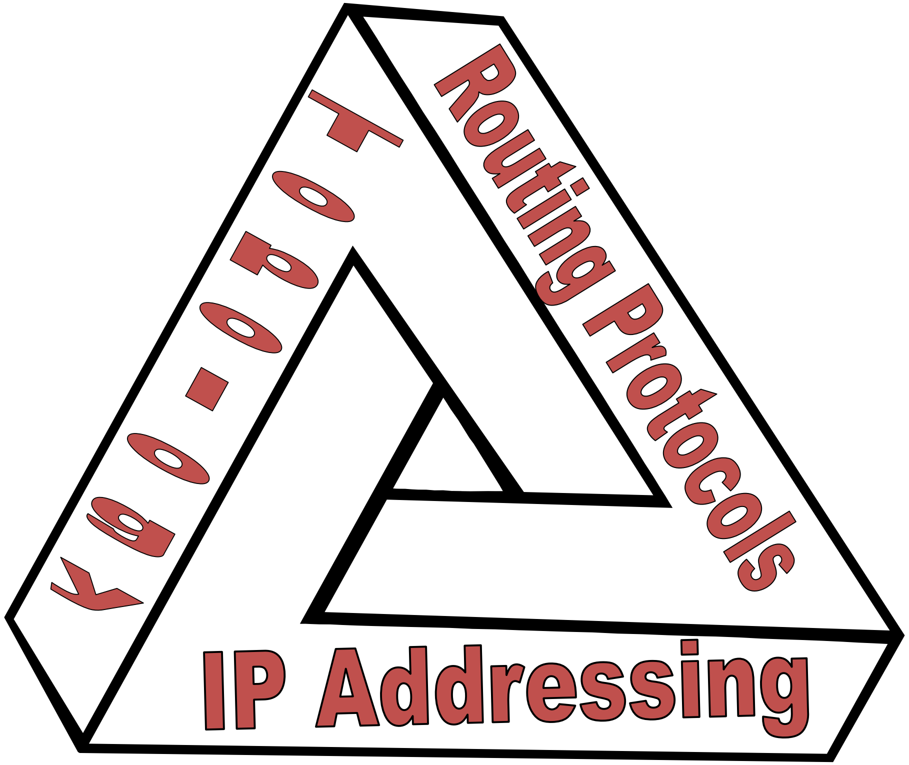
Redundant Network Design

Concepts and Techniques

Basic ISP Scaling Concepts

Design

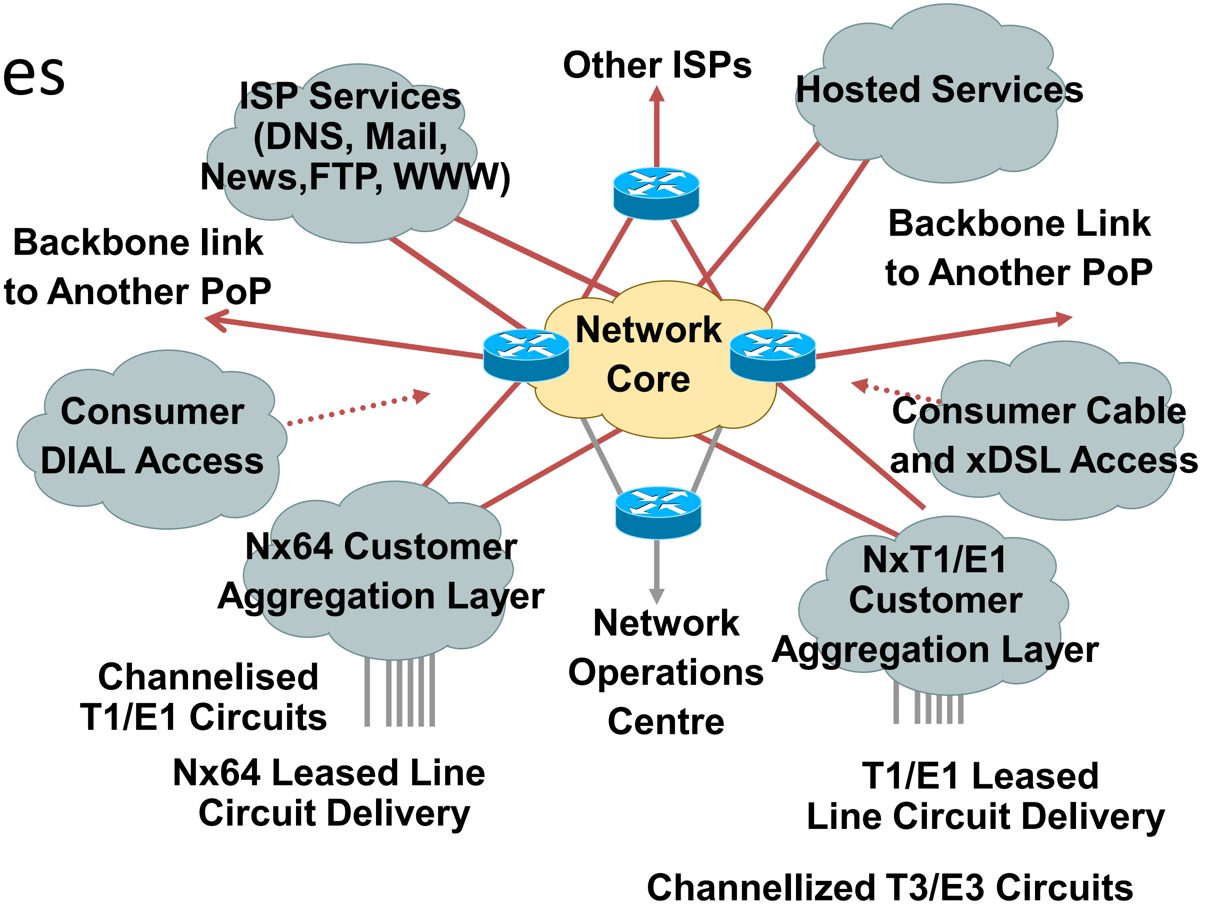
- Modular/Structured Design
- Functional Design
- Tiered/Hierarchical Design Discipline



Modular/Structured Design

Design

- Organize the network into separate and repeatable modules
  - Backbone
  - PoP
  - Hosting services
  - ISP Services
  - Support/NOC


Other ISPs
Hosted Services
ISP Services (DNS, Mail, News,FTP, WWW)
Backbone link to Another PoP
Backbone Link to Another PoP
Network Core
Consumer DIAL Access
Consumer Cable and xDSL Access
Nx64 Customer Aggregation Layer
NxT1/E1 Customer Aggregation Layer
Network Operations Centre
Channelised T1/E1 Circuits
Nx64 Leased Line Circuit Delivery
T1/E1 Leased Line Circuit Delivery
Channellized T3/E3 Circuits

Modular/Structured Design

Design

- Modularity makes it easy to scale a network
  - Design smaller units of the network that are then plugged into each other
  - Each module can be built for a specific function in the network
  - Upgrade paths are built around the modules, not the entire network

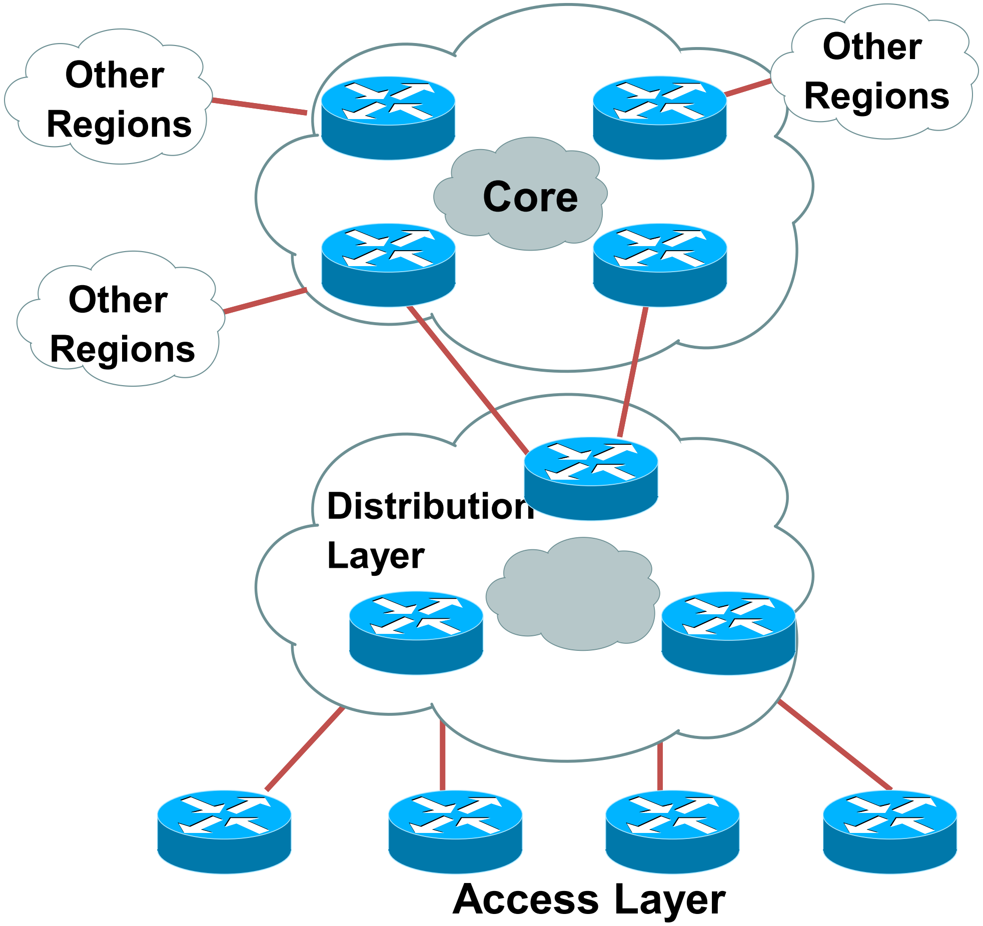
Functional Design

- One Box cannot do everything
  - no matter how hard people have tried in the past
- Each router/switch in a network has a well-defined set of functions
- The various boxes interact with each other
- Equipment can be selected and functionally placed in a network around its strengths
- ISP Networks are a systems approach to design
  - Functions interlink and interact to form a network solution.

Tiered/Hierarchical Design

Design

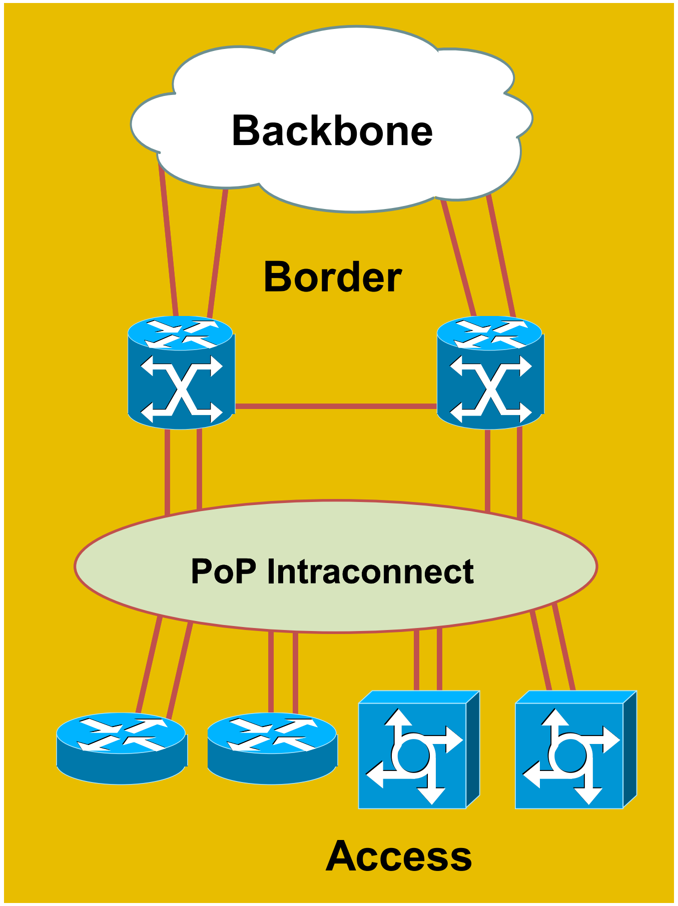
- Flat meshed topologies do not scale
- Hierarchy is used in designs to scale the network
- Good conceptual guideline, but the lines blur when it comes to implementation.



Multiple Levels of Redundancy

Design

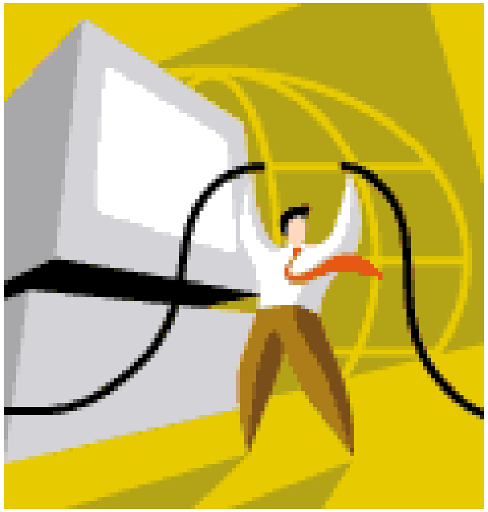
- Triple layered PoP redundancy
  - Lower-level failures are better
  - Lower-level failures may trigger higher-level failures
  - L3: IGP and BGP provide redundancy and load balancing
  - L4: TCP re-transmissions recover during the fail-over



Multiple Levels of Redundancy

Design

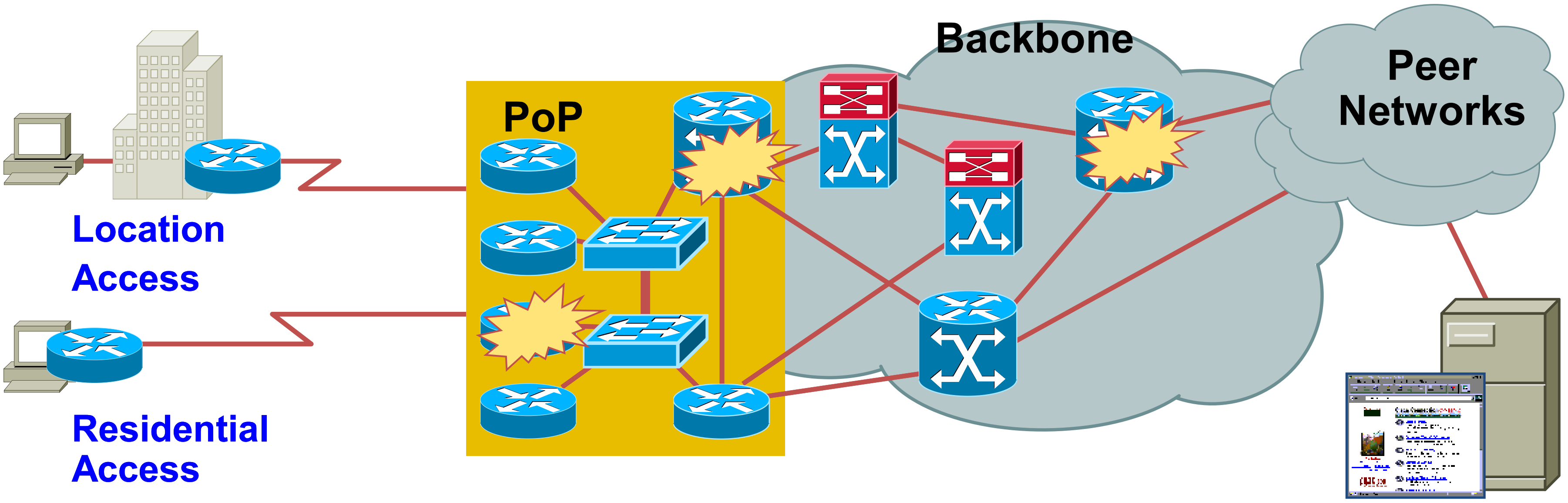
- Multiple levels also mean that one must go deep – for example:
  - Redundant power to the rack – circuit over load and technician trip
- MIT (maintenance injected trouble) is one of the key causes of ISP outage.



Multiple Levels of Redundancy

Design

- Objectives:
  - As little user visibility of a fault as possible
  - Minimize the impact of any fault in any part of the network
  - Network needs to handle L2, L3, L4, and router failure



Redundant Network Design

The Basics

The Basics: Platform

- Redundant Power
  - Two power supplies
- Redundant Cooling
  - What happens if one of the fans fail?
- Redundant route processors
  - Consideration also, but less important
  - Partner router device is better
- Redundant interfaces
  - Redundant link to partner device is better

The Basics: Environment

Design

- Redundant Power
  - UPS source – protects against grid failure
- Redundant cabling
  - Cable break inside facility can be quickly patched by using “spare” cables
  - Facility should have two diversely routed external cable paths
- Redundant Cooling
  - Facility has air-conditioning backup
  - …or some other cooling system?

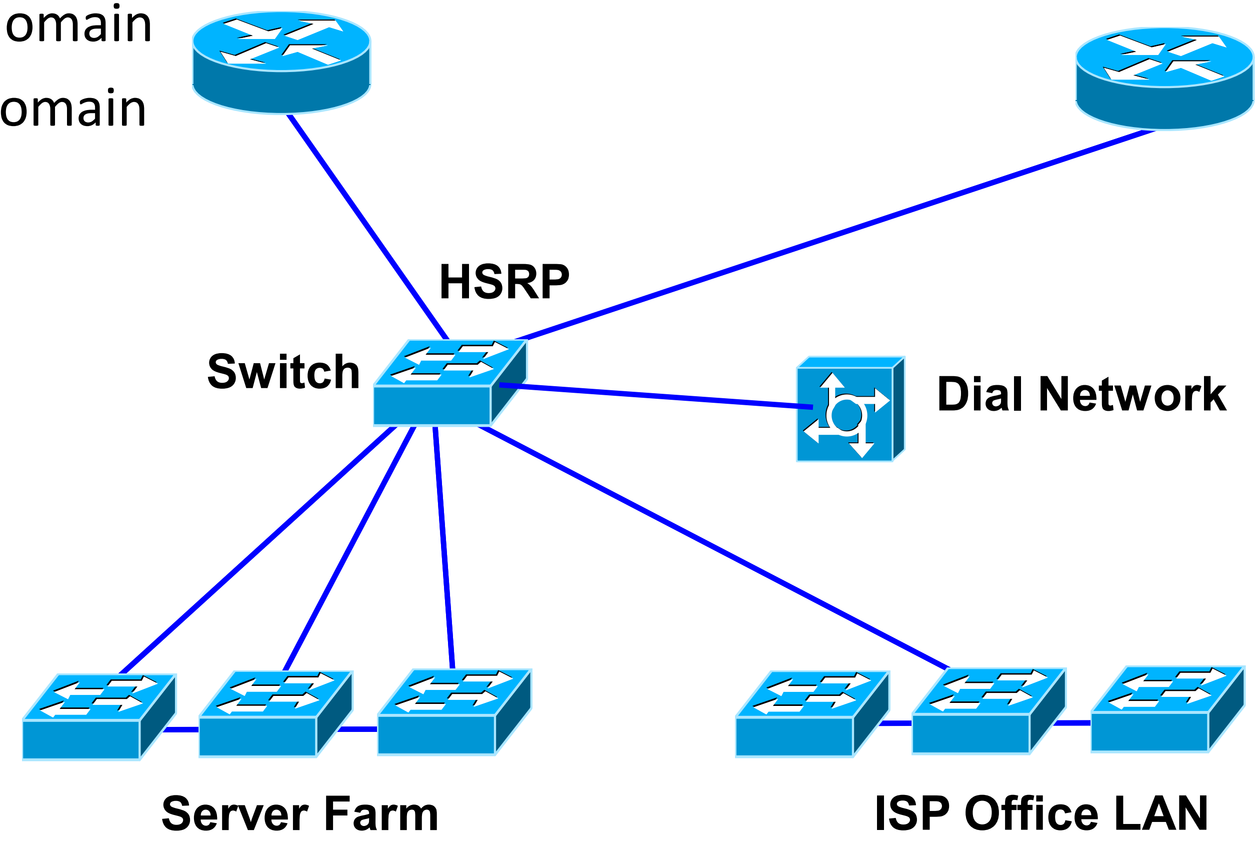
Redundant Network Design

Within the Data-Centre

Bad Architecture

Design

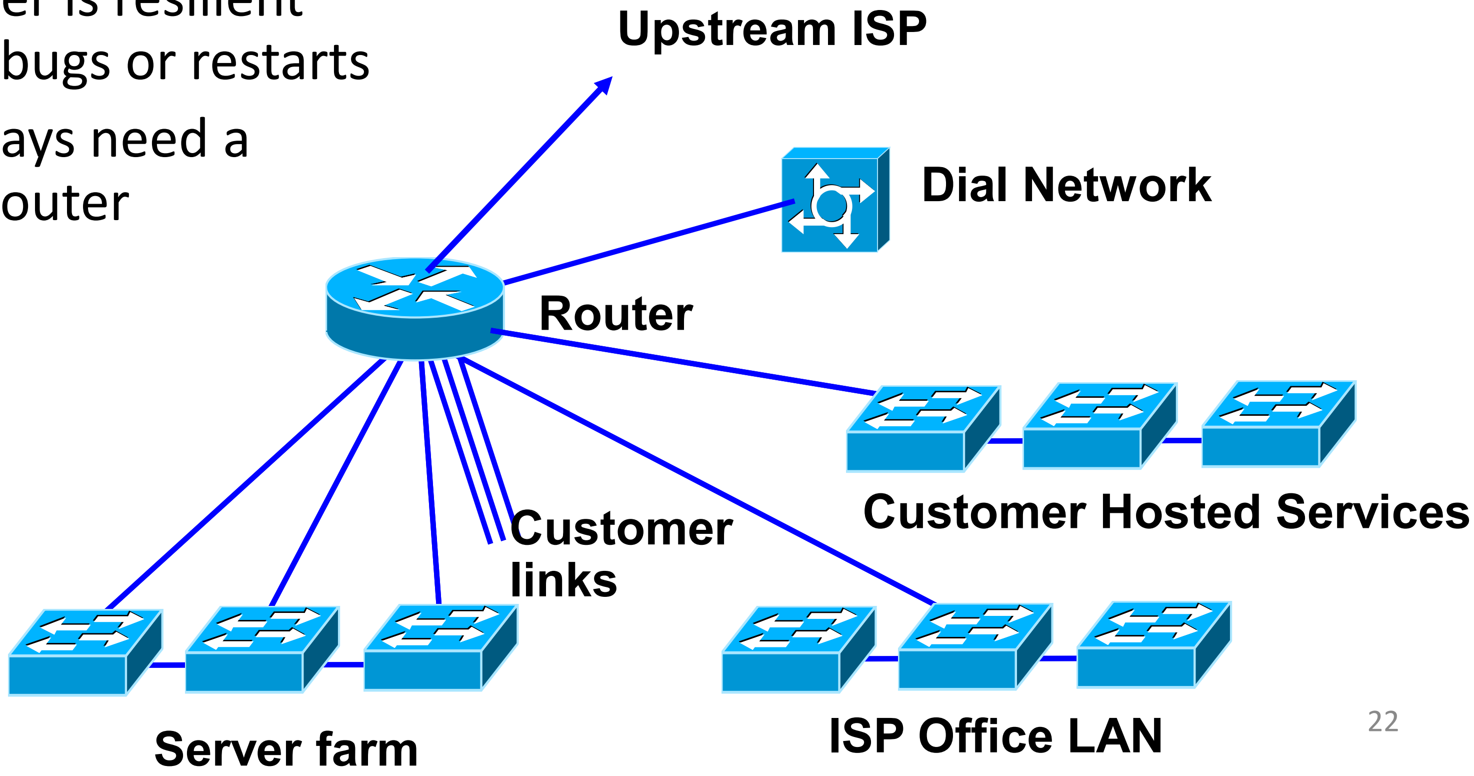
- A single point of failure
  - Single collision domain
  - Single security domain
  - Spanning tree convergence
  - No backup
  - Central switch performance



Bad Architecture

Design

- A central router
  - Simple to build
  - Resilience is the "vendor's problem"
  - More expensive
  - No router is resilient against bugs or restarts
  - You always need a bigger router



Even Worse!!

Design

- **Avoid Highly Meshed, Non-Deterministic Large Scale L2**

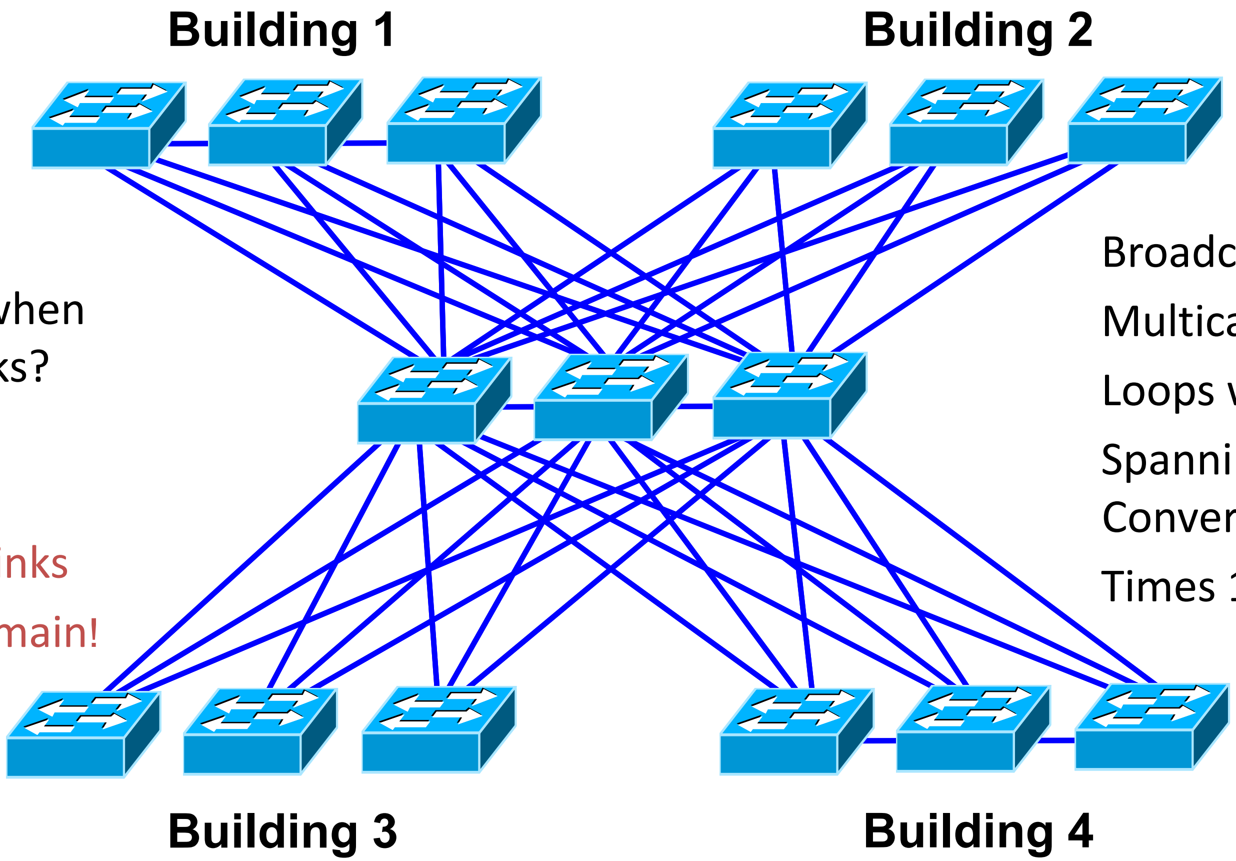
Where Should Root Go?

What Happens when Something Breaks?

How Long to Converge?

Many Blocking Links

Large Failure Domain!



Building 1

Building 2

Building 3

Building 4

Broadcast Flooding

Multicast Flooding

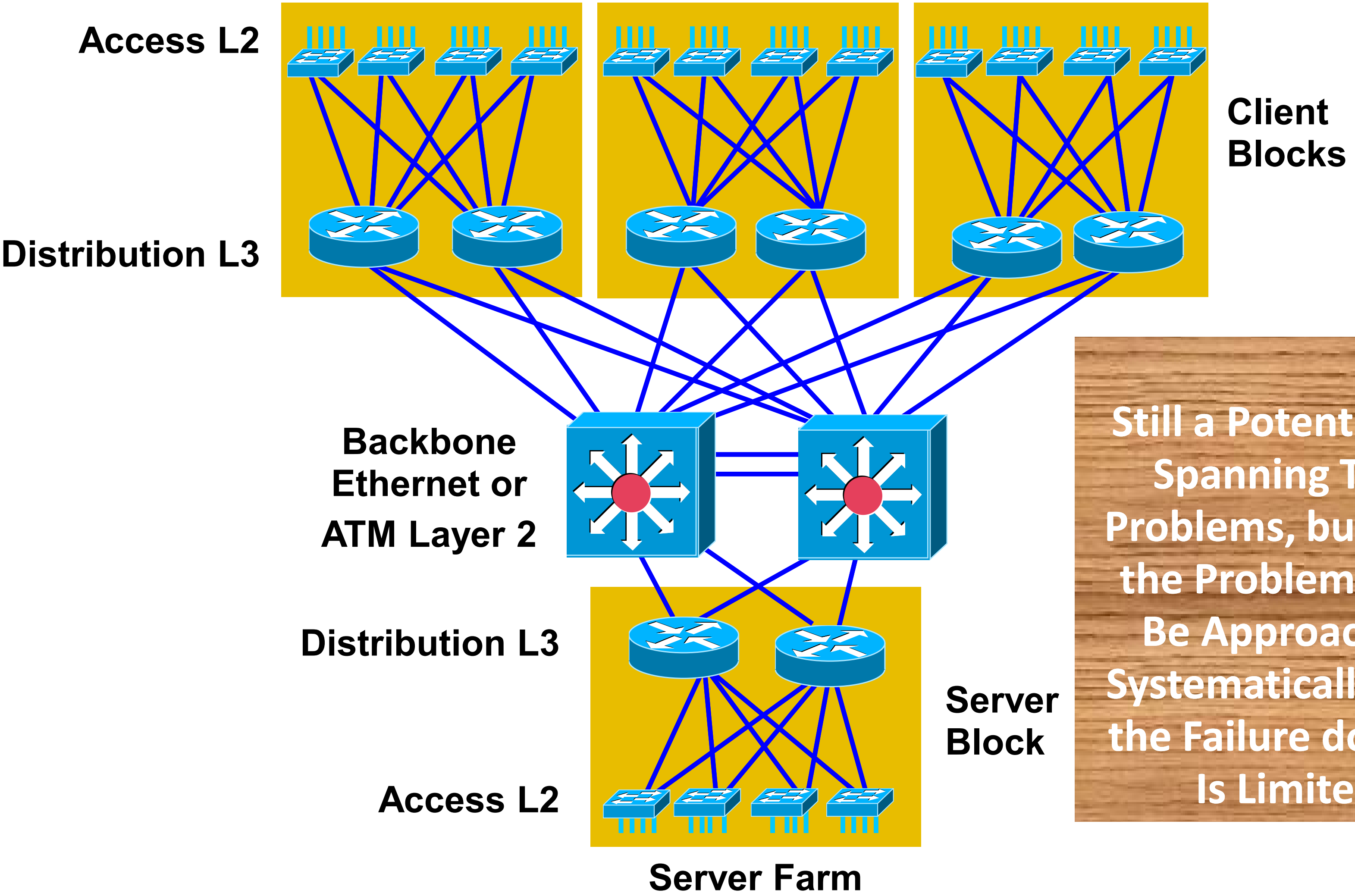
Loops within Loops

Spanning Tree Convergence Time

Times 100 VLANs?

Typical (Better) Backbone

Design



Still a Potential for Spanning Tree Problems, but Now the Problems Can Be Approached Systematically, and the Failure domain Is Limited

Benefits of Layer 3 backbone

Technology

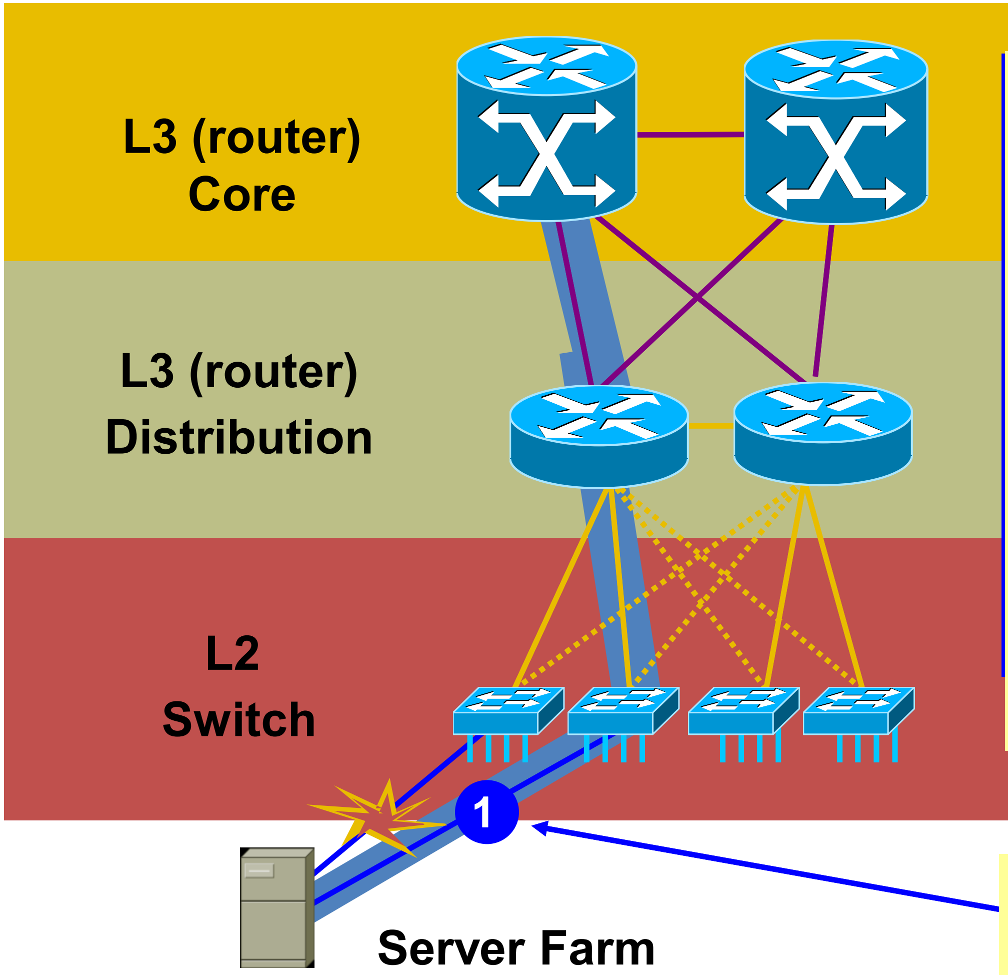
- Multicast PIM routing control
- Load balancing
- No blocked links
- Fast convergence OSPF/ISIS/EIGRP
- Greater scalability overall
- Router peering reduced

Redundant Network Design

Server Availability

Multi-homed Servers

Technology



Using Adaptive Fault Tolerant” Drivers and NICs

NIC Has a Single IP/MAC Address (Active on one NIC at a Time)

When Faulty Link Repaired, Does Not Fail Back to Avoid Flapping

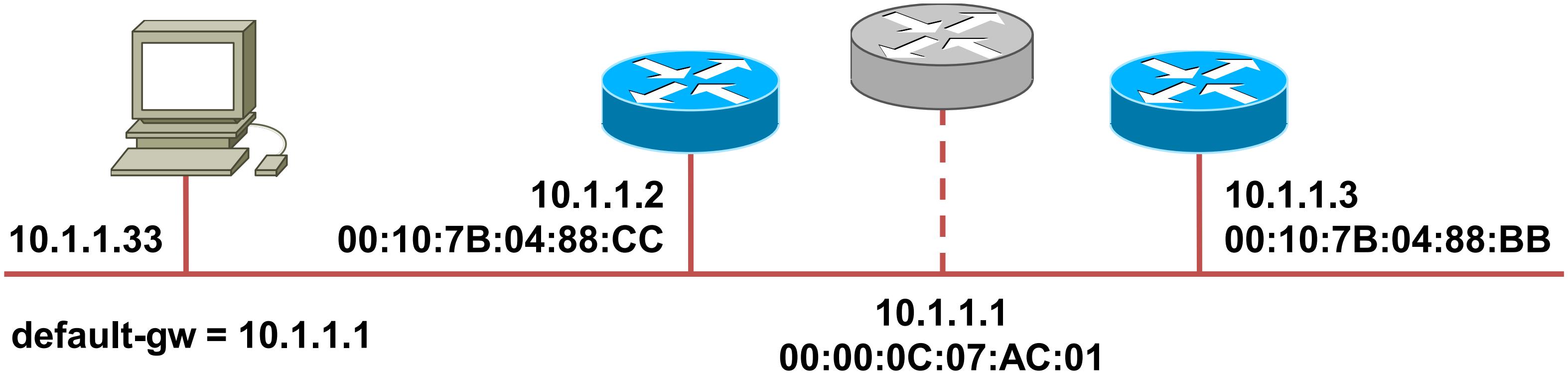
Fault-tolerant Drivers Available from Many Vendors: Intel, Compaq, HP, Sun

Many Vendors also Have Drivers that also Support ether channel

Dual-homed Server—Primary NIC Recovery (Time 1–2 Seconds)

HSRP – Hot Standby Router Protocol

Technology

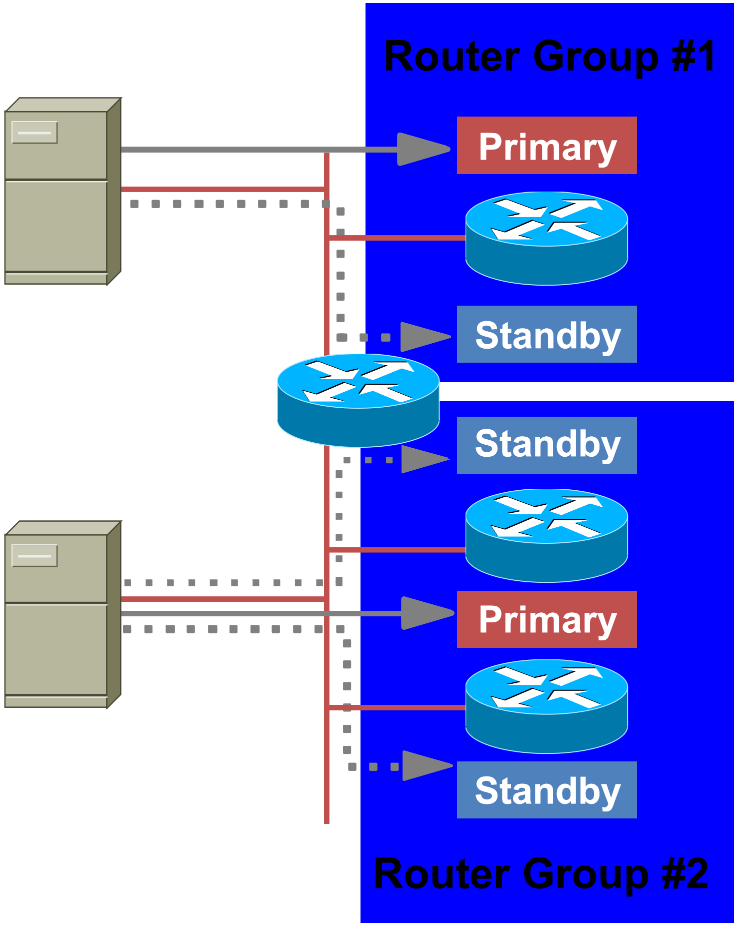


- Transparent failover of default router
- “Phantom” router created
- One router is active, responds to phantom L2 and L3 addresses
- Others monitor and take over phantom addresses

HSRP – RFC 2281

Technology

- HSR multicasts hellos every 3 sec with a default priority of 100
- HSR will assume control if it has the highest priority and preempt configured after delay (default=0) seconds
- HSR will deduct 10 from its priority if the tracked interface goes down



Redundant Network Design

WAN Availability

Circuit Diversity

Design

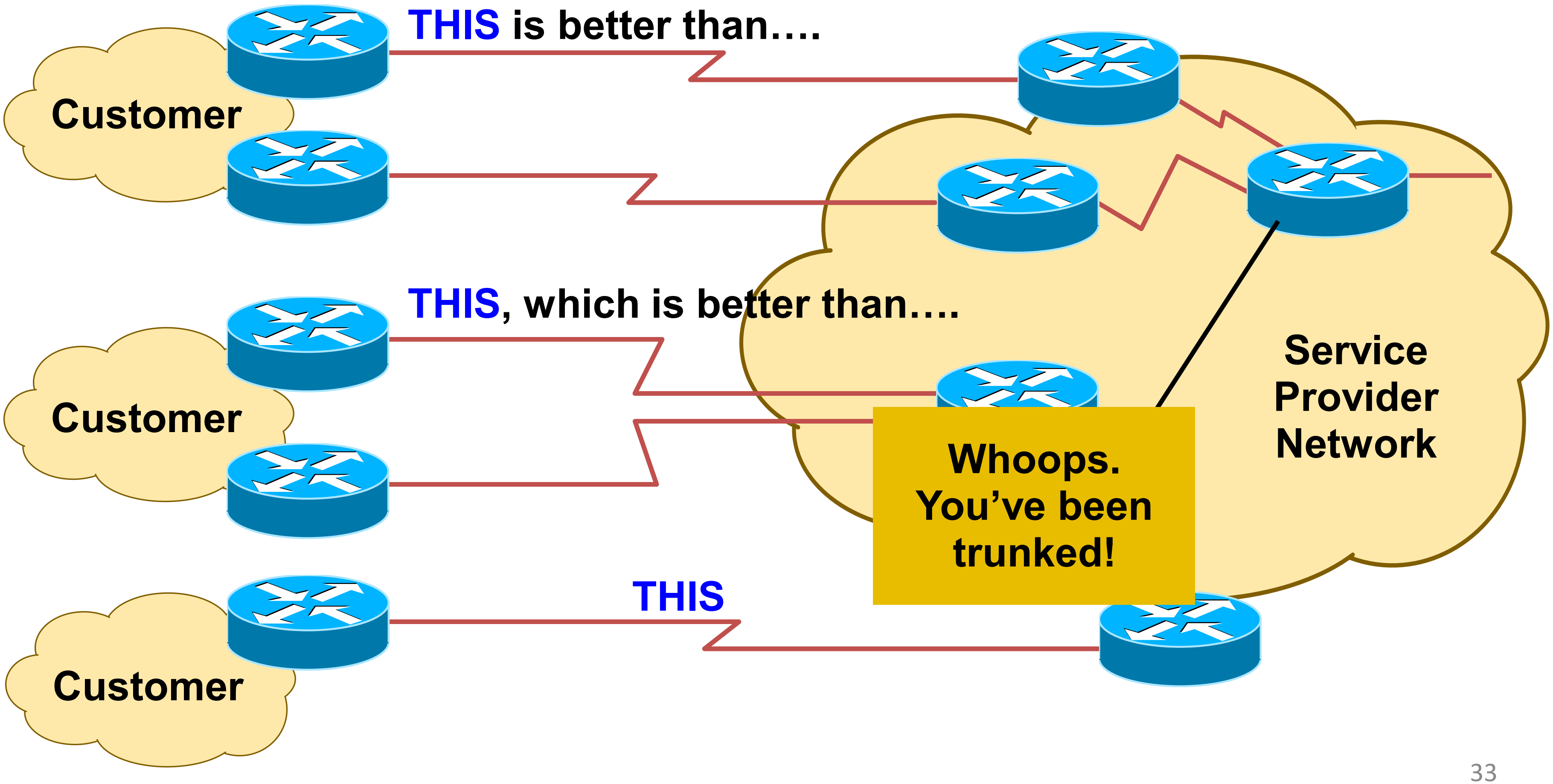
- Having backup PVCs through the same physical port accomplishes little or nothing
  - Port is more likely to fail than any individual PVC
  - Use separate ports
- Having backup connections on the same router doesn’t give router independence
  - Use separate routers
- Use different circuit provider (if available)
  - Problems in one provider network won’t mean a problem for your network

Circuit Diversity

Design

- Ensure that facility has diverse circuit paths to providers
- Make sure your backup path terminates into separate equipment at the service provider
- Make sure that your lines are not trunked into the same paths as they traverse the network
- Try and write this into your Service Level Agreement with providers

Circuit Diversity



THIS is better than….

THIS, which is better than….

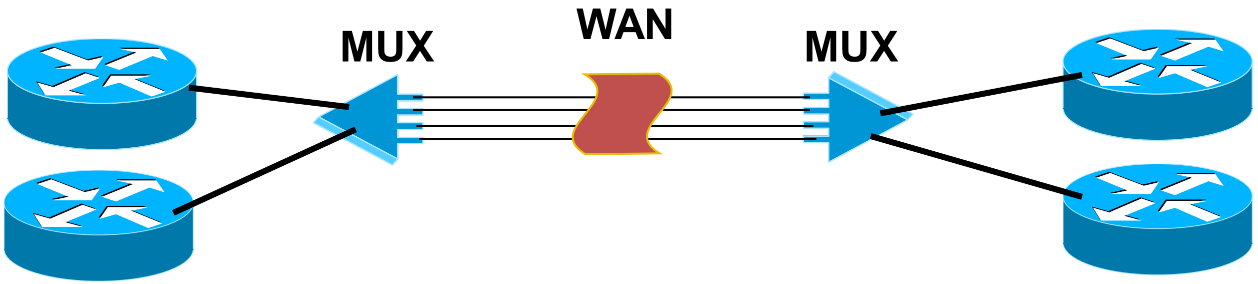
Whoops. You've been trunked!

THIS

Circuit Bundling – MUX

- Use hardware MUX
  - Hardware MUXes can bundle multiple circuits, providing L1 redundancy
  - Need a similar MUX on other end of link
  - Router sees circuits as one link
    - Failures are taken care of by the MUX

- Using redundant routers helps



Load Sharing

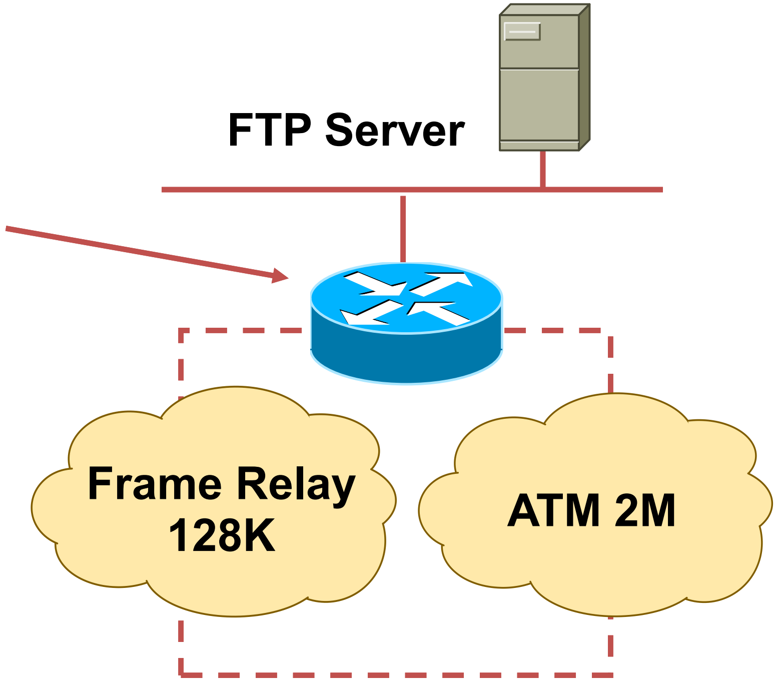
Design

- Load sharing occurs when a router has two (or more) equal cost paths to the same destination
- EIGRP also allows unequal-cost load sharing
- Load sharing can be on a per-packet or per-destination basis (default: per-destination)
- Load sharing can be a powerful redundancy technique, since it provides an alternate path should a router/path fail

Policy-based Routing

Technology

- If you have unequal cost paths, and you don’t want to use unequal-cost load sharing (you don’t!), you can use PBR to send lower priority traffic down the slower path



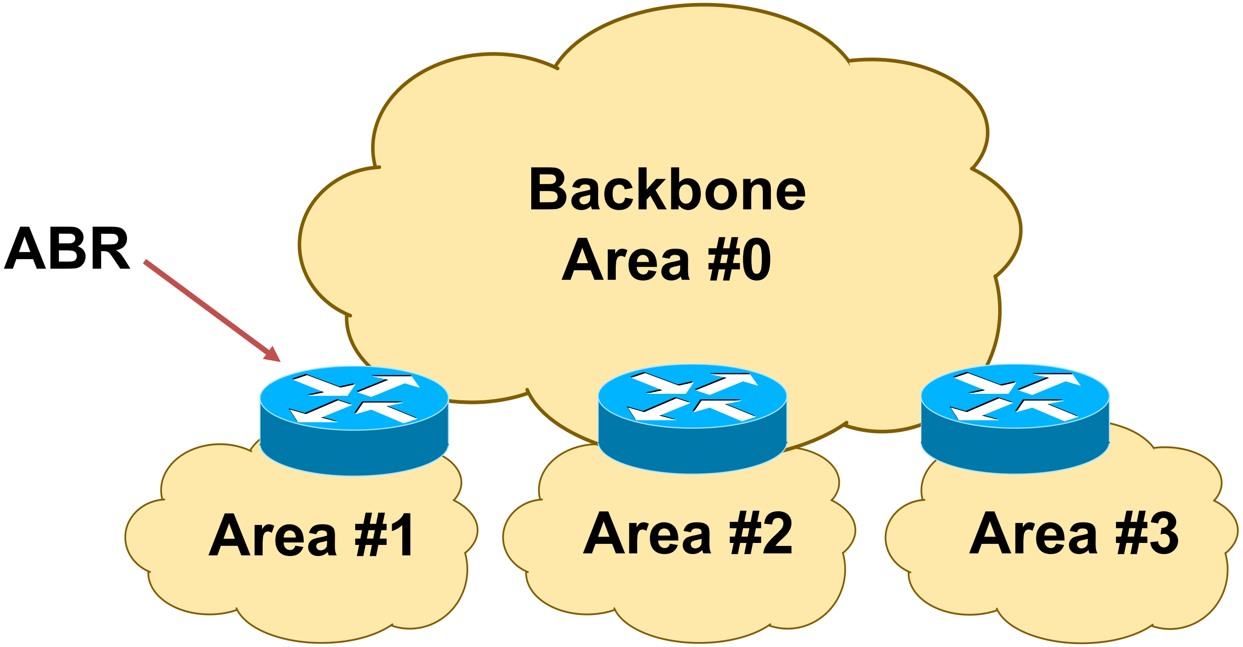
Convergence

Design

- The convergence time of the routing protocol chosen will affect overall availability of your WAN
- Main area to examine is L2 design impact on L3 efficiency
- **Factors Determining Protocol Convergence**
  - Network size
  - Hop count limitations
  - Peering arrangements (edge, core)
  - Speed of change detection
  - Propagation of change information
  - Network design: hierarchy, summarization, redundancy

OSPF – Hierarchical Structure

Design



- Topology of an area is invisible from outside of the area
  - LSA flooding is bounded by area
  - SPF calculation is performed separately for each area

Factors Assisting Protocol Convergence

Design

- Keep number of routing devices in each topology area small (15 – 20 or so)
  - Reduces convergence time required
- Avoid complex meshing between devices in an area
  - Two links are usually all that are necessary
- Keep prefix count in interior routing protocols small
  - Large numbers means longer time to compute shortest path
- Use vendor defaults for routing protocol unless you understand the impact of “twiddling the knobs”
  - Knobs are there to improve performance in certain conditions only

Redundant Network Design

Internet Availability

PoP Design

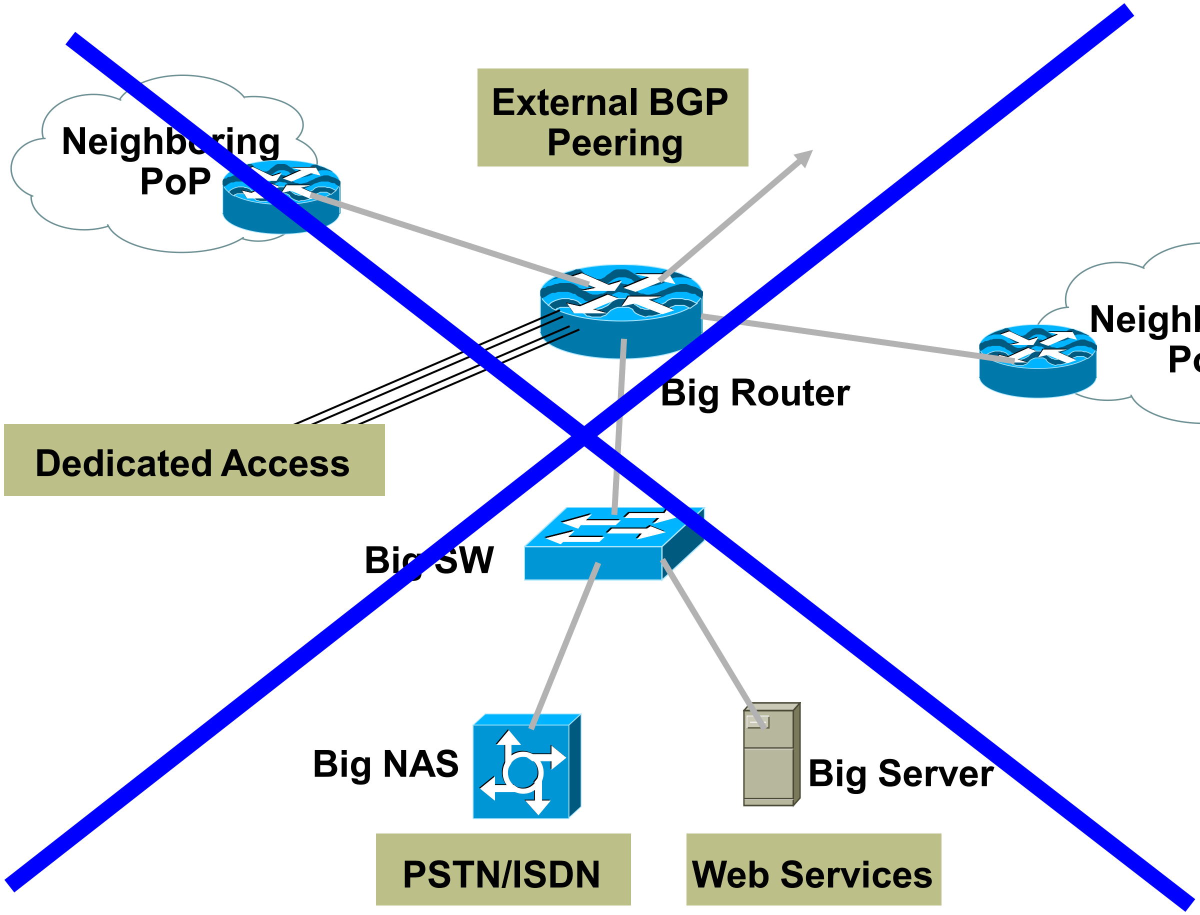
- One router cannot do it all
- Redundancy redundancy redundancy
- Most successful ISPs build two of everything
- Two smaller devices in place of one larger device:
  - Two routers for one function
  - Two switches for one function
  - Two links for one function

PoP Design

- Two of everything does not mean complexity
- Avoid complex highly meshed network designs
  - Hard to run
  - Hard to debug
  - Hard to scale
  - Usually demonstrate poor performance

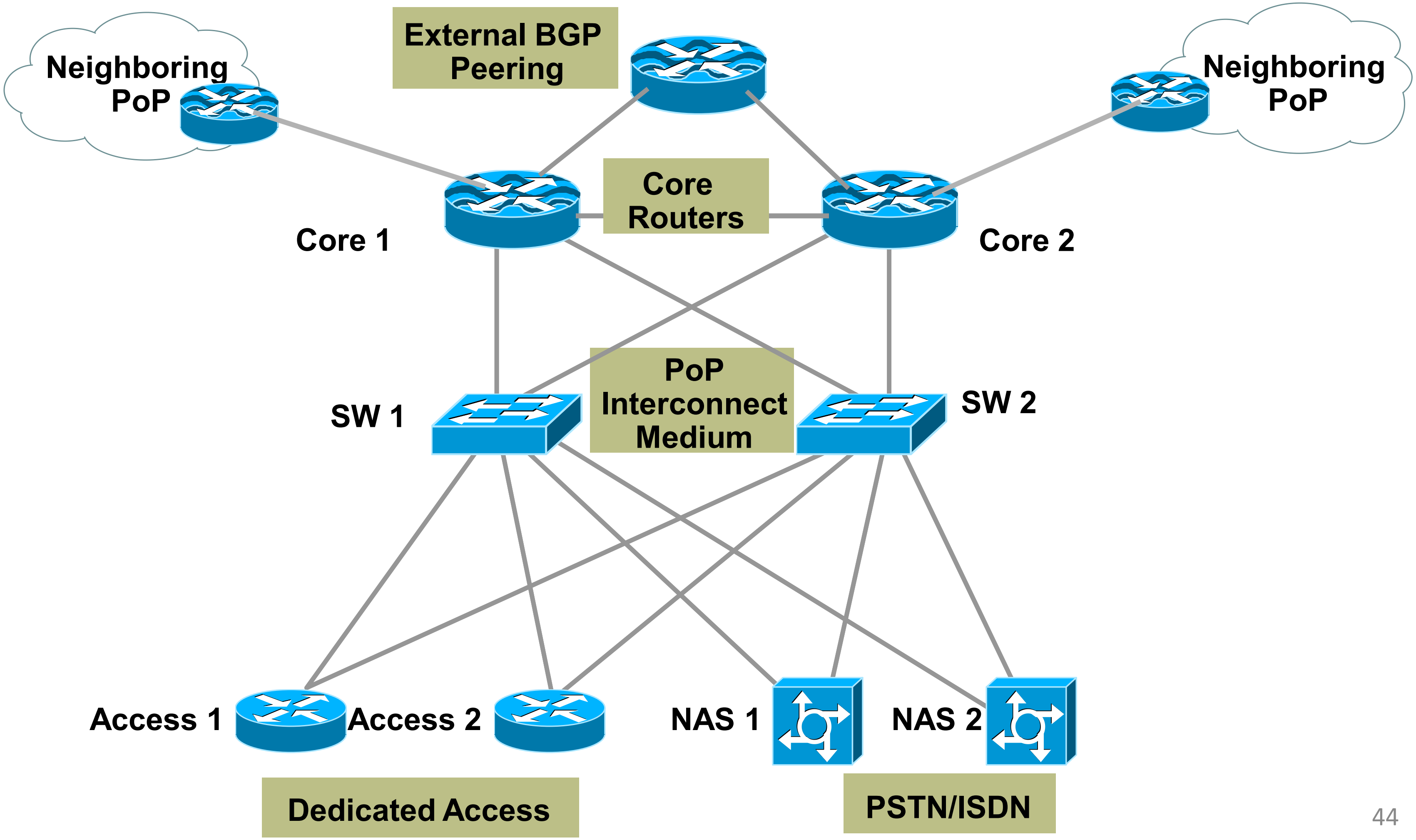
PoP Design – Wrong

Design



PoP Design – Correct

Design



Hubs vs. Switches

- Hubs
  - These are obsolete
    - Switches cost little more
  - Traffic on hub is visible on all ports
    - It’s really a replacement for coax ethernet
    - Security!?
  - Performance is very low
    - 10Mbps shared between all devices on LAN
    - High traffic from one device impacts all the others
  - Usually non-existent management

Hubs vs. Switches

- Switches
  - Each port is masked from the other
  - High performance
    - ➢10/100/1000Mbps per port
    - ➢Traffic load on one port does not impact other ports
  - 10/100/1000 switches are commonplace and cheap
  - Choose non-blocking switches in core
    - ➢Packet doesn’t have to wait for switch
  - Management capability (SNMP via IP, CLI)
  - Redundant power supplies are useful to have

Beware Static IP Dial

Design

- Problems
  - Does NOT scale
  - More customers, slower IGP convergence
  - Support becomes expensive
- Solutions
  - Route “Static Dial” customers to same RAS or RAS group behind distribution router
  - Use contiguous address block
  - Make it very expensive – it costs you money to implement and support

Redundant Network Design

Operations!

Network Operations Centre

Process

- NOC is necessary for a small ISP
  - It may be just a PC called NOC, on UPS, in equipment room.
  - Provides last resort access to the network
  - Captures log information from the network
  - Has remote access from outside
    - ➢Dialup, SSH,…
  - Train staff to operate it
  - Scale up the PC and support as the business grows

Operations

Process

- A NOC is essential for all ISPs
- Operational Procedures are necessary
  - Monitor fixed circuits, access devices, servers
  - If something fails, someone has to be told
- Escalation path is necessary
  - Ignoring a problem won’t help fixing it.
  - Decide on time-to-fix, escalate up reporting chain until someone can fix it

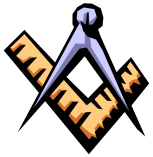
Operations

Process

- Modifications to network
  - A well designed network only runs as well as those who operate it
  - Decide and publish maintenance schedules
  - And then STICK TO THEM
  - Don’t make changes outside the maintenance period, no matter how trivial they may appear

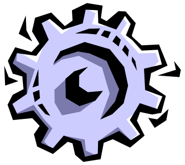
In Summary

- Implementing a highly resilient IP network requires a combination of the proper process, design and technology



Design

Technology

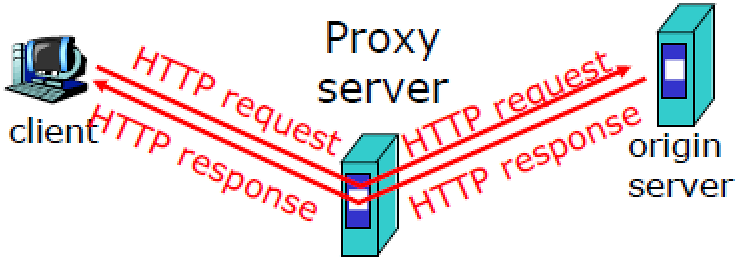


Process

Proxy

- Proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers.
- Proxy server stores all the data it receives as a result of placing requests for information on the internet in it’s *cache.*
  - Cache simply means memory
  - The cache is typically hard disk space, but it could be RAM
  - Caching documents means keeping a copy of internet documents so the server doesn’t need to request them over again
  - With proxy caching, clients make requests to servers, but the requests first go through a proxy cache

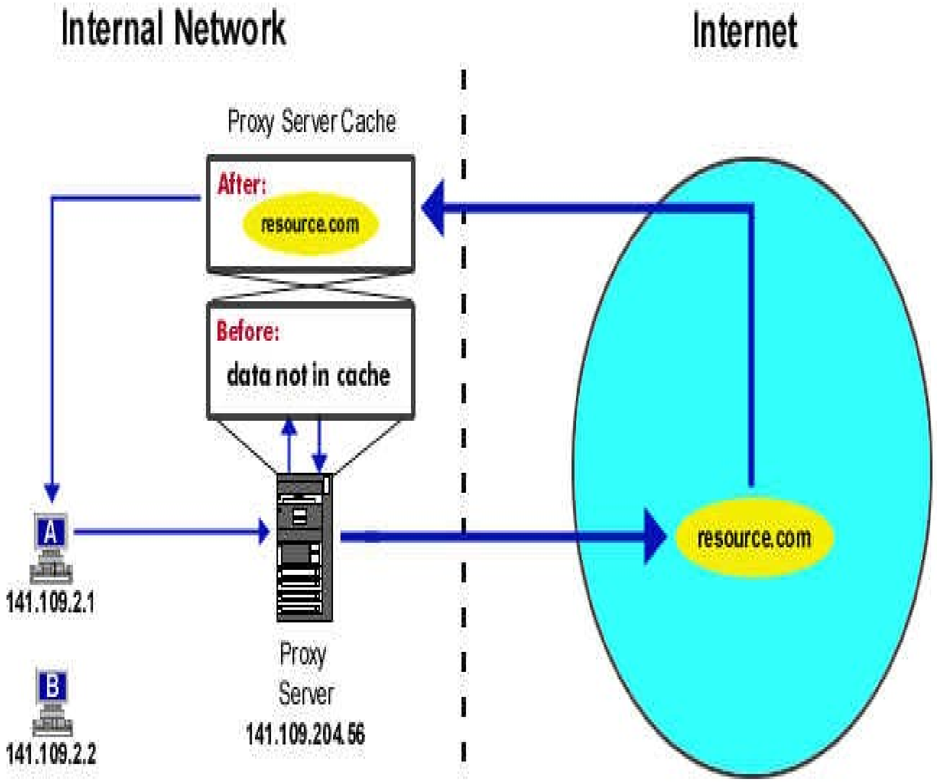
Proxy



- Origin server: Original source of an object
- Proxy server: Supplies object

  instead of origin server
- Some (caches) are demand-driven: Acts as both a
  - server: responding to client’s requests
  - client: forwarding requests that it cannot respond to towards the origin server
- Some are driven by supply and demand (e.g. proactive caching)
- Some are supply-driven: Content Distribution Networks

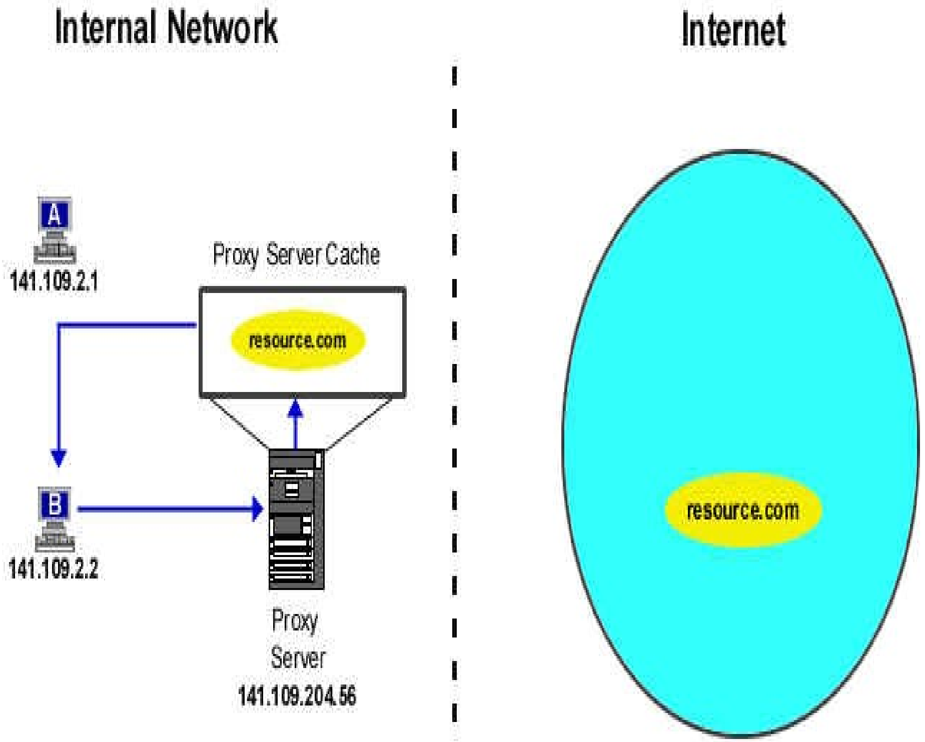
Caching a Document on a Proxy Server



<u>Scenario 1</u>:

- user **A** request a web page
- the request goes to the proxy server
- the proxy server checks to see if the document is stored in cache
- the document is not in cache so the request is sent to the Internet
- the proxy server receives the request, stores (or caches) the page
- the page is sent to user **A** where is viewed

Retrieving Cached Documents



<u>Scenario 2</u>:

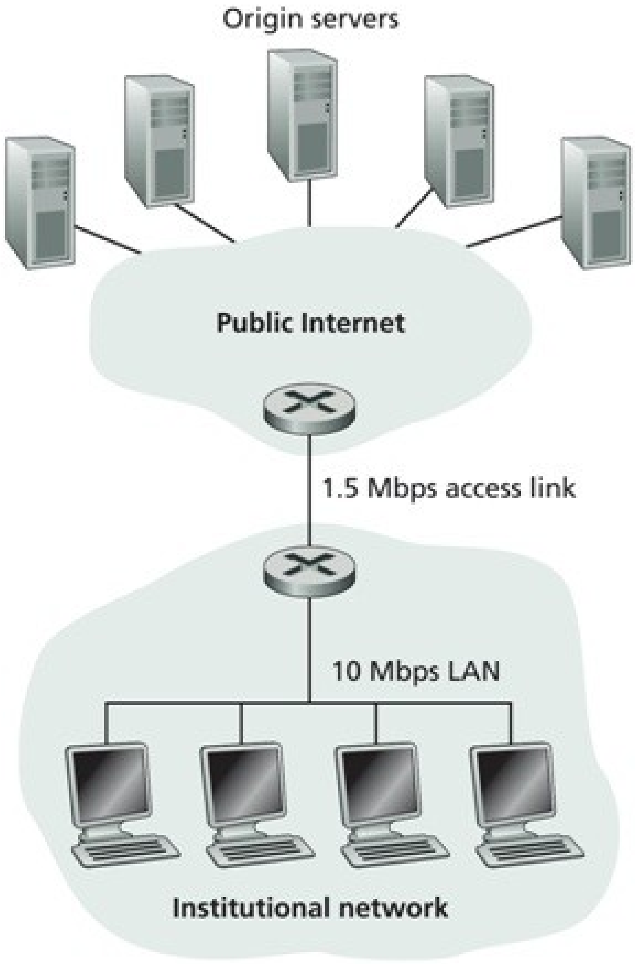
- user **B** request the same page as user **A** (ie. resource.com)
- the request goes to the proxy server
- the proxy server checks its cache for the page
- the page is stored in cache
- the proxy server sends the page to user **B** where it is viewed
- no connection to the Internet is required

Managing Cached Documents

- Many documents available on the Internet are “living” documents
- Determining when documents should be updated or deleted can be difficult task
    - Some documents can remain stable for a very long time and then suddenly change.
    - Other documents can change weekly or a daily basis.
- You need to decide carefully how often to refresh or delete the documents held in cache.

Caching Example

- Assumptions
  - average object size = 100,000 bits
  - avg. request rate from institution’s browser to origin servers = 15req/sec
  - Access bandwidth = 1.5Mbps
  - Traffic Intensity = (15 req/sec)*(100,000bits/request)/(1.5 Mbps) =1

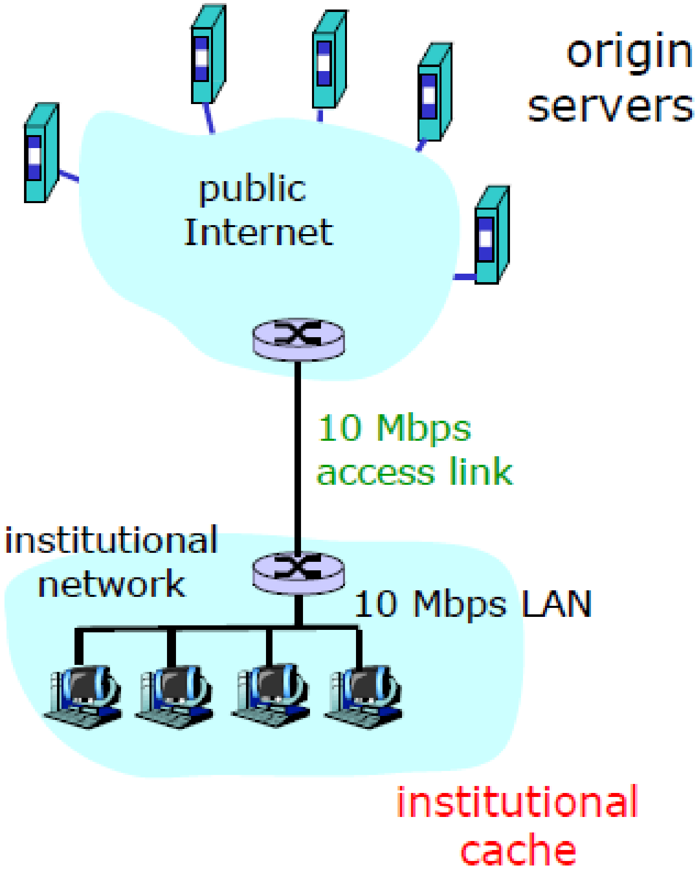


Caching Example

- Assumptions
  - average object size = 100,000 bits
  - avg. request rate from institution’s browser to origin servers = 15/sec
  - delay from institutional router to any origin server and back to router = 2 sec
- Consequences
  - utilization on LAN = 15%
  - utilization on access link = 100%
  - total delay = Internet delay + access delay + LAN delay = 2 sec + minutes + milliseconds
  - Generally LAN delays are very less approx 10ms or less in 10 Mbps

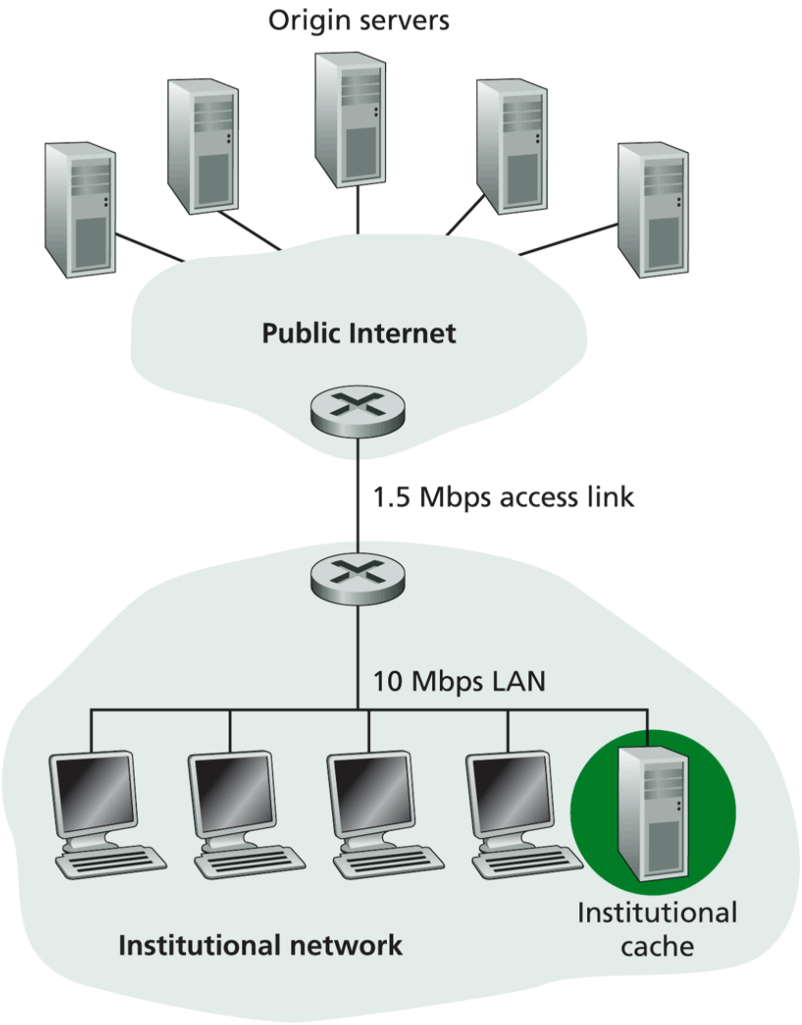
Caching Example

- Possible solution
  - increase bandwidth of access link to, say, 10 Mbps
- Consequences
  - utilization on LAN = 15%
  - utilization on access link =15%
  - Total delay = Internet delay + access delay + LAN delay= 2 sec + msecs + msecs
  - often a costly upgrade



Caching Example

- Assumptions
  - average object size = 100,000 bits
  - avg. request rate from institution’s browser to origin servers = 15req/sec
  - Access bandwidth = 1.5Mbps
  - LAN Bandwidth = 10Mbps
  - Traffic Intensity = (15 req/sec)*(100,000bits/request)/(10 Mbps) =0.15



♦ Adding a cache to the institutional network

Caching Example

- Install cache
  - suppose hit rate is 0.4
- Consequence
  - 40% requests will be satisfied almost immediately 60% requests satisfied by origin server
  - utilization of access link reduced to 60%, resulting in negligible delays (say 10 msec)
  - total delay = Internet delay + access delay + LAN delay = 0.6*2 sec + 0.6*0.01 secs + milliseconds < 1.3 secs

Benefits of Caching

- Eliminate the need (in many cases) to:
  - Send request to origin server (reducing delay, and link use)
  - Send full response from origin server (reducing link use)
- Reduced delay directly benefits end-user.
- May benefit service providers (ISPs or web servers) by making their service more popular to end-users.
- Reduced traffic
  - Reduces load on network links
- Reduces load on server
  - e.g. reducing “flash crowds”
- Mask unavailability of origin server e.g. when working offline, or during faults

Corporate Caching

- General Proxy is suitable for Company or LAN only
- For ISP General Proxy not suitable
- May have problem with load distribution

Load Balancing

- Problem: Single physical Origin or Proxy Server may not be able to handle its load
- Solution : install multiple servers and distribute the requests.
- How do we distribute requests among the Servers?

DNS Round Robin

- DNS is configured so multiple IP Addresses correspond to a single host name
- multiple type “A” records in DNS Database
  - Example.com 120.89.100.1
  - Example.com 120.89.100.2
  - Example.com 120.89.100.3
  - Example.com 120.89.100.4
- Modify the DNS server to round-robin through the IP addresses for each new request
- This way, different clients are pointed to different servers

Problems with DNS Round Robin

- Not Optimal for proxy servers
  - Cache content is duplicated
  - Multi-tier proxy arrangement won’t work if cookies are used
  - Load is not truly balanced
    - Assignment is at DNS lookup level, not HTTP request level

Internet Cache Protocol (ICP)

- Used for querying proxy servers for cached documents
- Typically used by proxy servers to check other proxy server's cache
- Could be used by clients
- ICP request has desired URL in it
- Send via UDP to other proxy servers
- Other proxy servers respond "HIT" or "MISS"
- Works better in LANs than Internet

Problems with ICP

- ICP queries generate extra network traffic
- Does not scale well
  - More proxy servers= more querying
- Caches become redundant

Non-redundant Proxy Load Balancing

- Proxy Selection based on a hash function
- Hash value is calculated from the URL
- Use resulting hash value to choose proxy
- Use Host name in hash function to ensure request routed so same proxy server

Cache Array Routing Protocol(CARP)

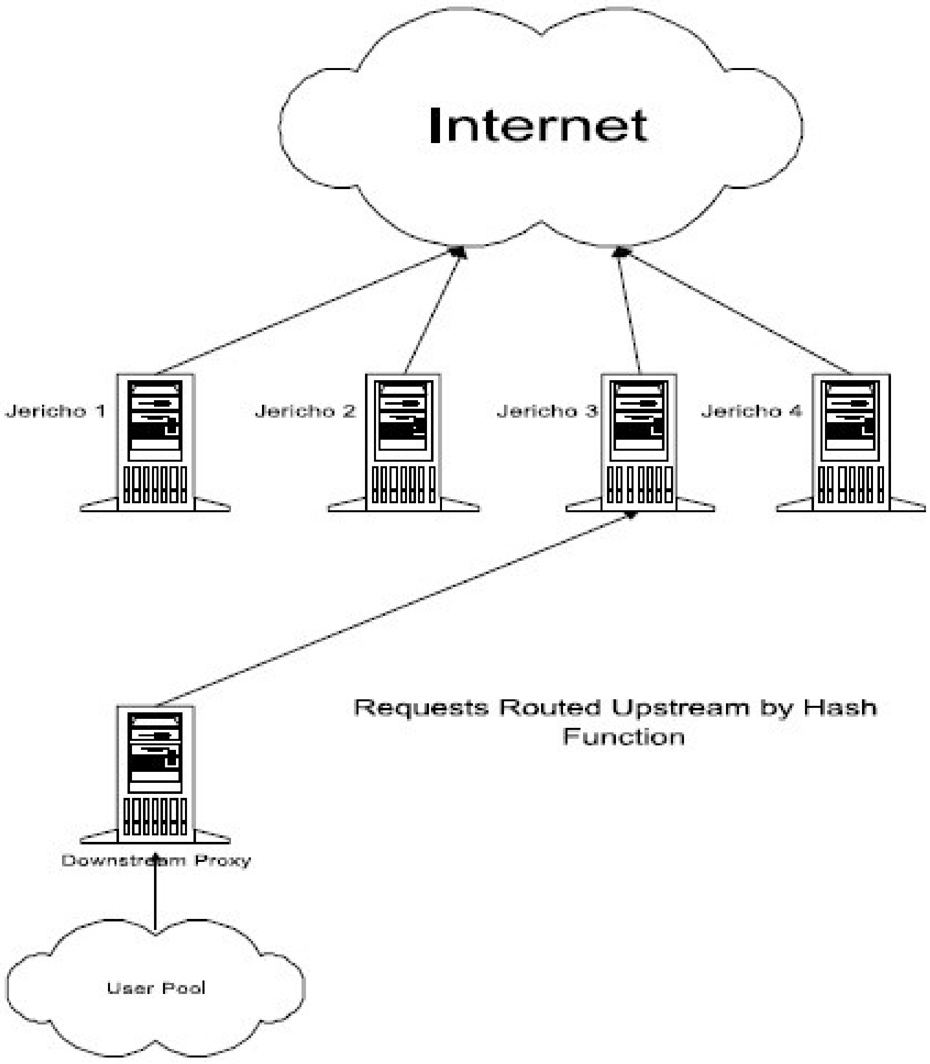
- Hash-based Proxy Selection mechanism
  - No queries
- Hashing used to select server
- Highly Scalable
  - Performance improves as size of array increases
  - Automatically adjusts to additions/deletions of servers
- Eliminates cache redundancy

How CARP works

- Given an array of Proxy servers
- Assume array membership is tracked using a membership list
- A hash value H_s is computed for the name of each proxy server in list (only when list changes)
- A hash value H_u is computed for the name of each requested URL
- For each request, a combined hash value $H_c=F(H_s,H_u)$ is computed for all servers
- Use highest H_c to select server

CARP: Hierarchical Routing

- One Server acts as director using Hash routing.
- Cache hit rate is maximized
- Single point of Failure

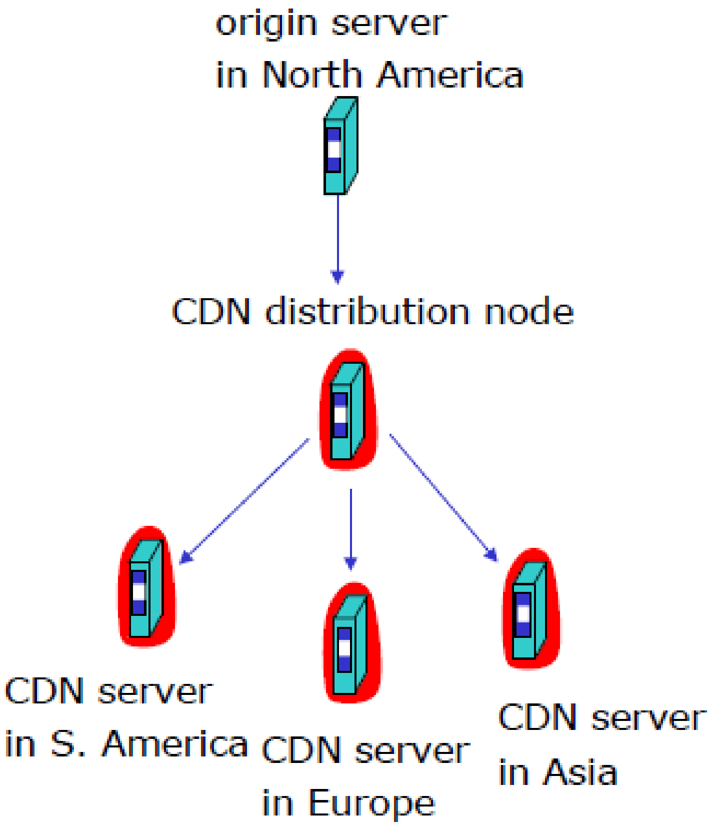


CARP: Distributed Routing

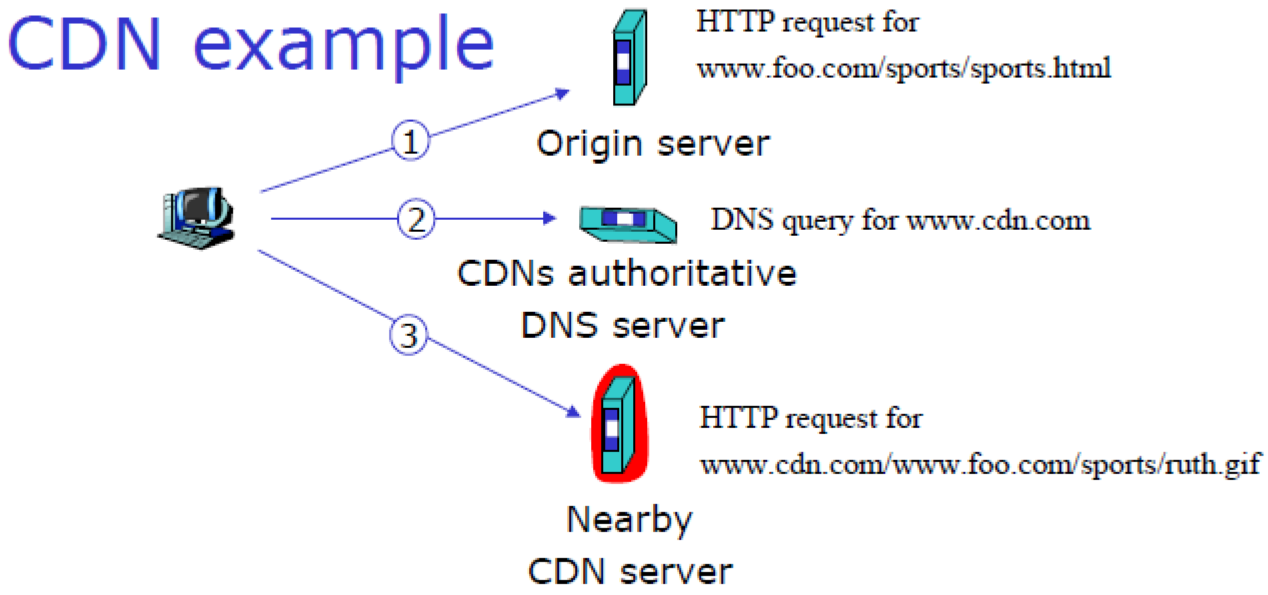
- Requests can be sent to ANY member of the Array.
- Route request to best score if not me.
- Don’t cache response if redirected

Content distribution networks (CDNs)

- The content providers are the CDN customers.
- Content replication
  - CDN company installs hundreds of CDN servers throughout Internet
    - ➢in lower-tier ISPs, close to users
  - CDN replicates its customers' content in CDN servers. When provider updates content, CDN updates servers



Content distribution networks (CDNs)



- origin server
- www.foo.com
- distributes HTML
- Replaces:

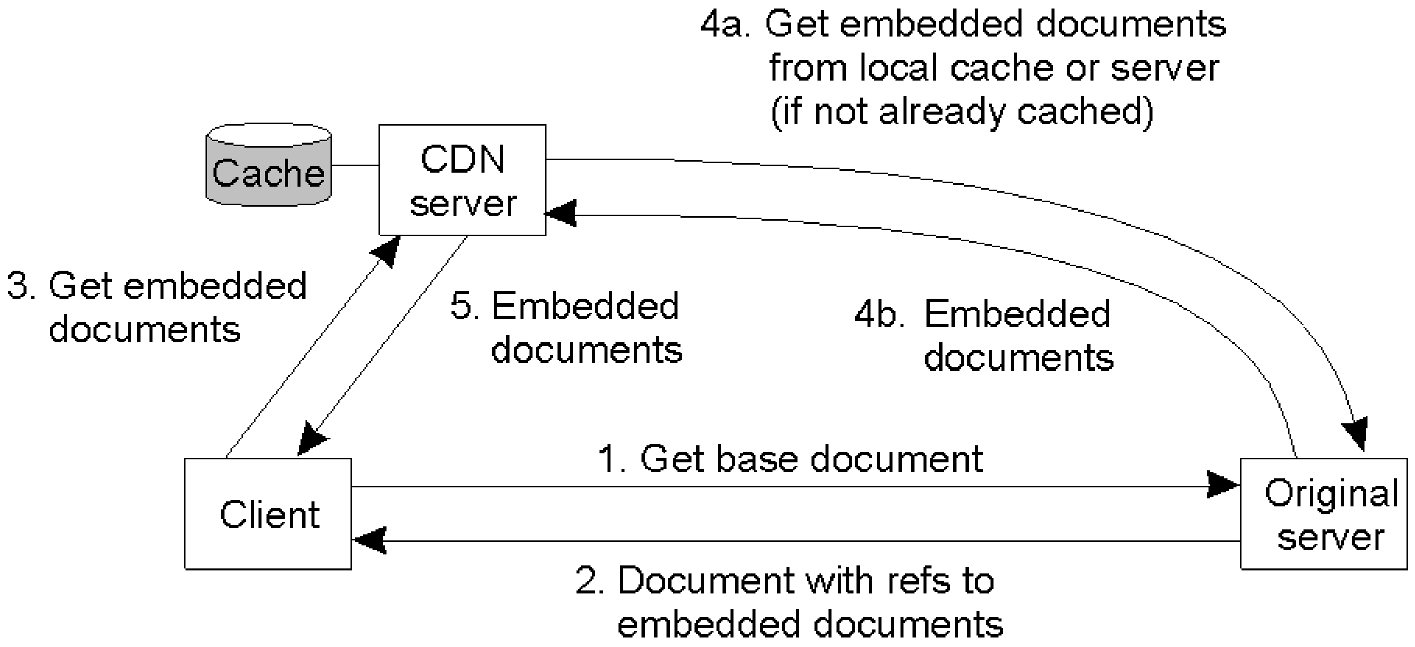
  http://www.foo.com/sports.ruth.gif

  With http://www.cdn.com/www.foo.com/sports/ruth.gif

CDN company

- cdn.com
- distributes gif files
- uses its authoritative DNS server to route redirect requests

Akamai CDN

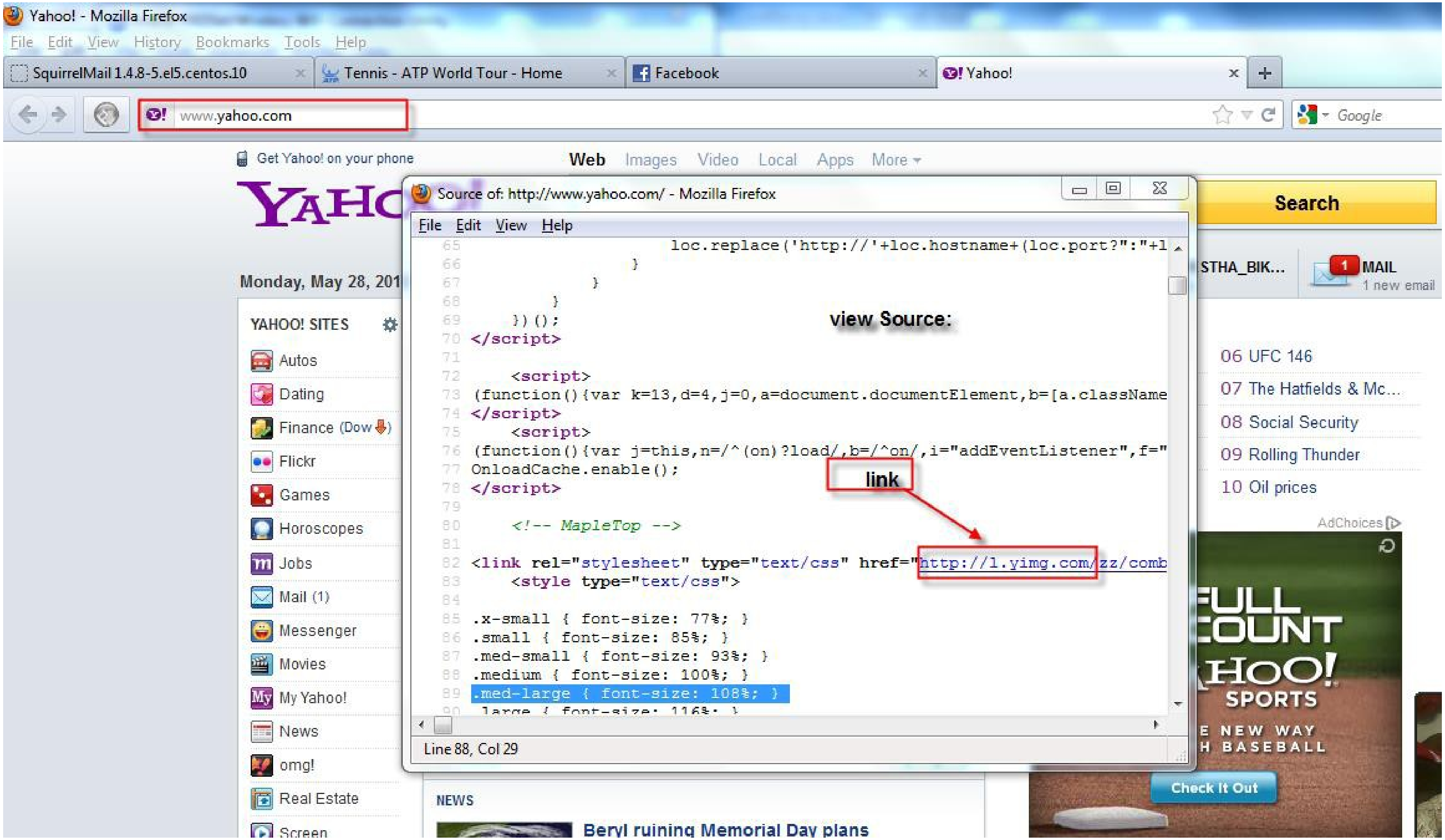

4a. Get embedded documents from local cache or server (if not already cached)
Cache
CDN server
3. Get embedded documents
5. Embedded documents
4b. Embedded documents
1. Get base document
Client
Original server
2. Document with refs to embedded documents

- Embedded documents have names that are resolved by Akamai DNS to local CDN server
    - Use Internet "map" to determine local server
- Local server gets copy from original server
- Akamai has many CDN servers "close" to clients

Content distribution networks (CDNs)

- routing requests
- CDN creates a “map”, indicating distances from leaf ISPs and CDN nodes
- when query arrives at authoritative DNS server:
  - server determine ISP from which query originates
  - uses “map” to determine best CDN server
  - not just Web pages
    - streaming stored audio/video
    - streaming real-time audio/video
      - CDN nodes create application-layer overlay network

Caching and Content Distribution(CDN)



view Source:

link

```
                        loc.replace('http://'+loc.hostname+(loc.port?":"+l
                    }
                }
            }
        })();
</script>

        <script>
(function(){var k=13,d=4,j=0,a=document.documentElement,b=[a.className
</script>
        <script>
(function(){var j=this,n=/^(on)?load/,b=/^on/,i="addEventListener",f="
OnloadCache.enable();
</script>

        <!-- MapleTop -->

<link rel="stylesheet" type="text/css" href="http://l.yimg.com/zz/comb
        <style type="text/css">

.x-small { font-size: 77%; }
.small { font-size: 85%; }
.med-small { font-size: 93%; }
.medium { font-size: 100%; }
.med-large { font-size: 108%; }
.large { font-size: 116%; }
```

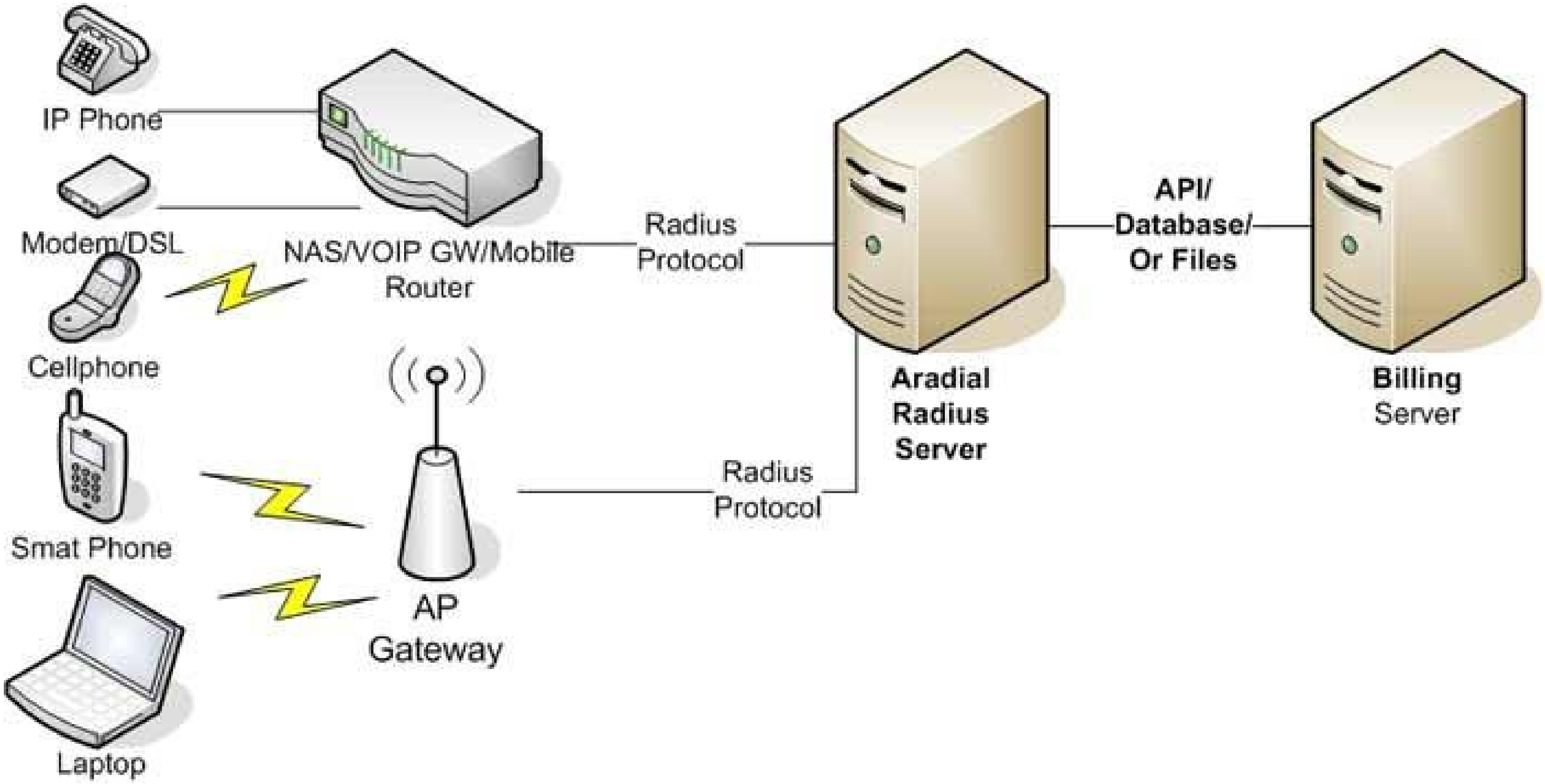
RADIUS

- Remote Authentication Dial In User Service (RADIUS) is a networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for computers to connect and use a network service.
- RADIUS was developed by Livingston Enterprises, Inc., in 1991 as an access server authentication and accounting protocol and later brought in
- Because of the broad support and the universal nature of the RADIUS protocol, it is often used by ISPs and enterprises to manage access to the Internet or internal networks, wireless networks, and integrated e-mail services.

RADIUS Contd..

- These networks may incorporate modems, DSL, access points, VPNs, network ports, web servers, etc to the Internet Engineering Task Force (IETF) standards.
- RADIUS is a remote authentication protocol.
- RADIUS is a de-facto standard for remote authentication.
- RADIUS is an extensible protocol, and can support many authentication methods

RADIUS


IP Phone
Modem/DSL
Cellphone
Smat Phone
Laptop
NAS/VOIP GW/Mobile Router
AP Gateway
Radius Protocol
Radius Protocol
Aradial Radius Server
API/ Database/ Or Files
Billing Server

RADIUS

- RADIUS serves three functions:
  - to authenticate users or devices before granting them access to a network,
  - to authorize those users or devices for certain network services and
  - to account for usage of those services.

Authentication

- Verify the user is who he/she claims to be
  - Use Password, Special Token card, Caller-ID, etc.
  - May issue additional ‘challenge’

RADIUS

- **Authorization**
  - Check that the user may access the services he/she wishes.
  - Check database or file information about the user
- **Accounting**
  - Record what the user has done.
  - Time online. Bytes sent/received. Services accessed. Files downloaded. etc.

Cookies

- A cookie, also known as an HTTP cookie, web cookie, or browser cookie
- small piece of data sent from a website and stored in a user's web browser while the user is browsing that website.
- Every time the user loads the website, the browser sends the cookie back to the server to notify the website of the user's previous activity.

Cookies Contd..

- Cookies were designed to be a reliable mechanism for websites to remember stateful information (such as items in a shopping cart) or to record the user's browsing activity (including clicking particular buttons, logging in, or recording which pages were visited by the user as far back as months or years ago).

Structure of Cookies

- Browsers are expected to support, at least, cookies with a size of 4KB. It consists of seven components:
  - Name of the cookie
  - Value of the cookie
  - Expiry of the cookie
  - Path the cookie is good for
  - Domain the cookie is good for
  - Need for a secure connection to use the cookie
  - Whether or not the cookie can be accessed through other means than HTTP (i.e., JavaScript)
- The first two components (name and value) are required to be explicitly set.

Types of Cookie

- Session Cookie
- Persistent cookie
- Secure cookie
- HttpOnly cookie
- Third-party cookie
- Supercookie
- Zombie cookie

Types of Cookie

- Session cookie
  - A session cookie, also known as an in-memory cookie or transient cookie, exists only in temporary memory while the user navigates the website.
  - When an expiry date or validity interval is not set at cookie creation time, a session cookie is created.
  - Web browsers normally delete session cookies when the user closes the browser.

Types of Cookie

- Persistent cookie
  - A persistent cookie outlast user sessions.
  - If a persistent cookie has its Max-Age set to one year (for example), then, during that year, the initial value set in that cookie would be sent back to the server every time the user visited the server.
  - This could be used to record a vital piece of information such as how the user initially came to this website.
  - For this reason, persistent cookies are also called tracking cookies

Types of Cookie

- Secure cookie
  - A secure cookie has the secure attribute enabled and is only used via HTTPS, ensuring that the cookie is always encrypted when transmitting from client to server.
  - This makes the cookie less likely to be exposed to cookie theft via eavesdropping.
  - In addition to that, all cookies are subject to browser's same-origin policy.

Types of Cookie

- HttpOnly cookie
  - The HttpOnly attribute is supported by most modern browsers.
  - On a supported browser, an HttpOnly session cookie will be used only when transmitting HTTP (or HTTPS) requests, thus restricting access from other, non-HTTP APIs such as JavaScript.
  - This restriction mitigates but does not eliminate the threat of session cookie theft via cross-site scripting (XSS).
  - This feature applies only to session-management cookies, and not other browser cookies.

Types of Cookie

- Third-party Cookie
  - First-party cookies are cookies that belong to the same domain that is shown in the browser's address bar (or that belong to the sub domain of the domain in the address bar).
  - Third-party cookies are cookies that belong to domains different from the one shown in the address bar.
  - Web pages can feature content from third-party domains (such as banner adverts), which opens up the potential for tracking the user's browsing history.
  - Privacy setting options in most modern browsers allow the blocking of third-party tracking cookies.

Types of Cookie

- Supercookie
  - is a cookie with an origin of a Top-Level Domain (such as .com) or a Public Suffix (such as .co.uk).
  - It is important that supercookies are blocked by browsers, due to the security holes they introduce.
  - If unblocked, an attacker in control of a malicious website could set a supercookie and potentially disrupt or impersonate legitimate user requests to another website that shares the same Top-Level Domain or Public Suffix as the malicious website.
    - For example, a supercookie with an origin of .com, could maliciously affect a request made to example.com, even if the cookie did not originate from example.com. This can be used to fake logins or change user information.

Types of Cookie

- Zombie cookie
  - Some cookies are automatically recreated after a user has deleted them; these are called zombie cookies.
  - This is accomplished by a script storing the content of the cookie in some other locations, such as the local storage available to Flash content, HTML5 storages and other client-side mechanisms, and then recreating the cookie from backup stores when the cookie's absence is detected

Server Setup and Configuration Guidelines

- Hardware/The Basics: Environment
- Operating System / Firewall
- Number of sessions and load balancing

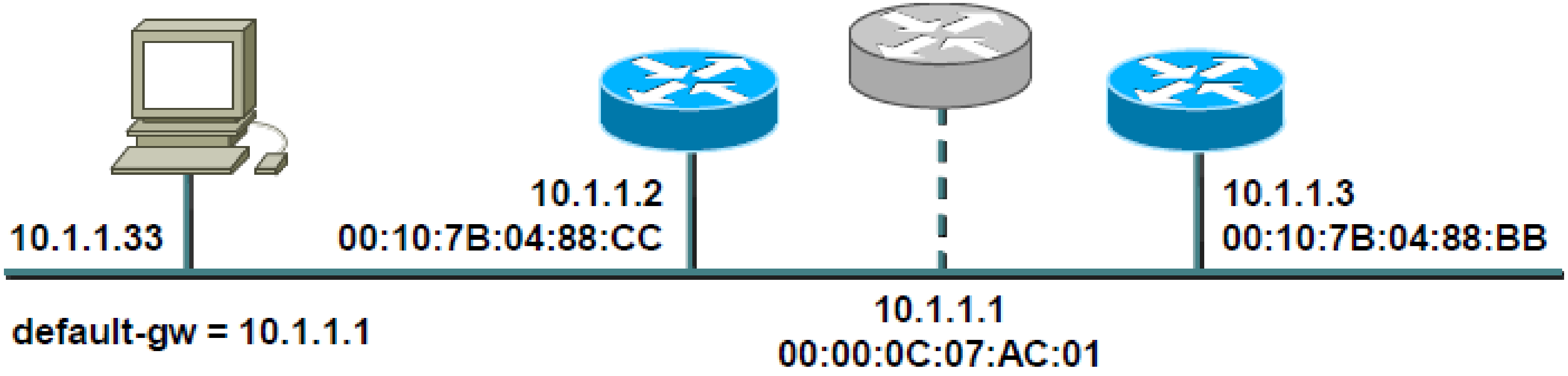
Hardware/The Basics: Environment

- Redundant Power
  - Two power supplies
- Redundant Cooling
  - What happens if one of the fans fail?
- Redundant processors
  - Consideration also, but less important
  - Partner router device is better
- Redundant interfaces
  - Redundant link to partner device is better

Hardware/The Basics: Environment

- Redundant Power
  - UPS source – protects against grid failure
  - “Dirty” source – protects against UPS failure
- Redundant cabling
  - Cable break inside facility can be quickly patched by using “spare” cables
  - Facility should have two diversely routed external cable paths
- Redundant Cooling
  - Facility has air-conditioning backup
  - …or some other cooling system?

HSRP – Hot Standby Router Protocol



- Transparent failover of default router
- “Phantom” router created
- One router is active, responds to phantom L2 and L3 addresses
- Others monitor and take over phantom addresses

Operating System and Security

- Platform
  - Windows
  - Linux
- Should have corporate level firewall
  - Packet filtering
  - Application level
  - IDS

Number of sessions and load balancing

- Threading should be increased
- Beside Threading there should be load balancing Hardware that should be responsible for load balancing in the server either packet wise or session wise in the replicated server.

Security and System Administration Issues

- Security administration is the process of maintaining a safe computing environment.
- System Administrators are the people responsible for uninterrupted operation of the computers
- Administrator's knowledge on System security loopholes and their implications on business they are managing, is a good asset to any Enterprise/Company

Internet security threats

Mapping:

- before attacking: “case the joint” – find out what services are implemented on network
- Use ping to determine what hosts have addresses on network
- Port-scanning: try to establish TCP connection to each port in sequence
- nmap (http://www.insecure.org/nmap/) mapper: “network exploration and security auditing”

Internet security threats

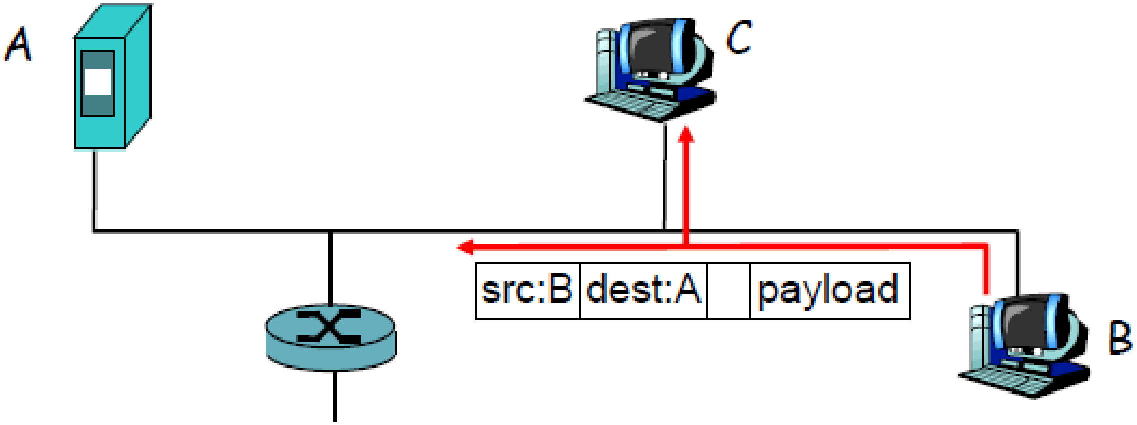
Mapping: countermeasures

- record traffic entering network
- look for suspicious activity (IP addresses, ports being scanned sequentially)

Internet security threats

Packet sniffing:

- broadcast media
- promiscuous NIC reads all packets passing by
- can read all unencrypted data (e.g. passwords)
- e.g.: C sniffs B's packets



Internet security threats

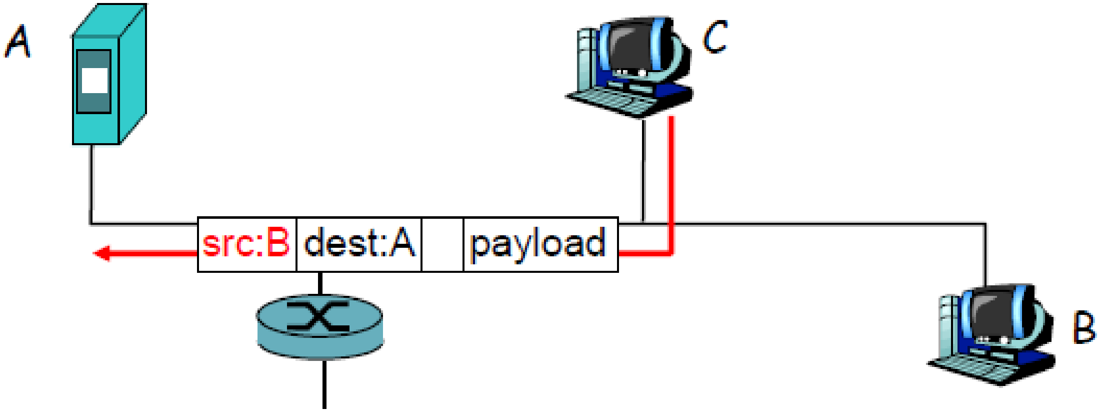
Packet sniffing: countermeasures

- all hosts in organization run software that checks periodically if host interface in promiscuous mode.
- one host per segment of broadcast media (switched Ethernet at hub)

Internet security threats

IP Spoofing:

- can generate "raw" IP packets directly from application, putting any value into IP source address field
- receiver can't tell if source is spoofed
- e.g.: C pretends to be B



Internet security threats

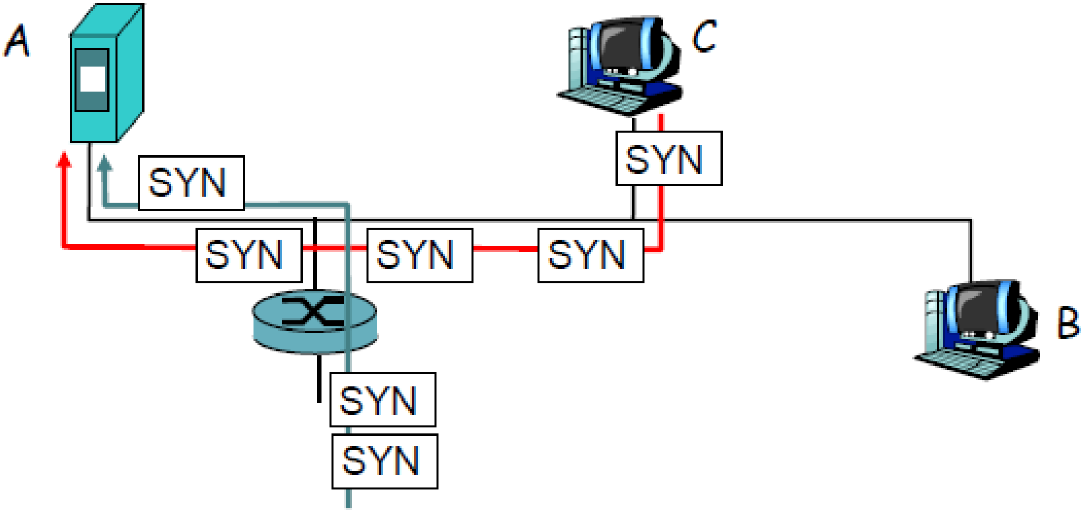
IP Spoofing: ingress filtering

- routers should not forward outgoing packets with invalid source addresses (e.g., datagram source address not in router’s network)
- great, but ingress filtering can not be mandated for all networks

Internet security threats

Denial of service (DOS):

- flood of maliciously generated packets "swamp" receiver
- Distributed DOS (DDOS): multiple coordinated sources swamp receiver
- e.g., C and remote host SYN-attack A



Internet security threats

Denial of service (DOS): countermeasures

- filter out flooded packets (e.g., SYN) before reaching host: throw out good with bad
- traceback to source of floods (most likely an innocent, compromised machine)

Firewall

- Isolates organization’s internal network from larger Internet, allowing some packets to pass, blocking others.
  - Acts as a security gateway between two networks
- A Firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, and provides protection from various kinds of IP spoofing and routing attacks.

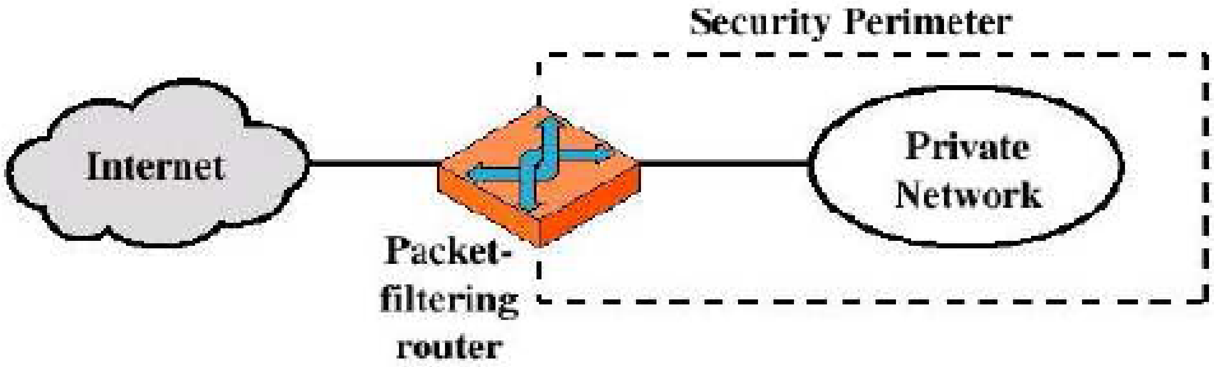
Firewall: Why

- Prevent denial of service attacks:
  - SYN flooding: attacker establishes many bogus TCP connections, no resources left for “real” connections.
- Prevent illegal modification/access of internal data.
  - e.g., attacker replaces CIA’s homepage with something else
- Allow only authorized access to inside network (set of authenticated users/hosts)
- Preserve customer and partner confidence
- Prevent viruses and worms on your network
- Prevent malicious attackers from getting into your network

Types of Firewall

1. Packet Filtering

- It applies a set of rules to each incoming IP Packet.
- The router is configured to filter packets going in both directions.
- Router filters packet-by-packet, decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits



2. Application Level Gateway

- In order to have a finer level security, firewalls must combine packet filters with application gateways.
- Application gateways look beyond the IP/TCP/UDP headers and actually make policy decisions based on application data.
- An application gateway is an application-specific server through which all application data (inbound and outbound) must pass.
- Multiple application gateways can run on the same host, but each gateway is a separate server with its own processes.

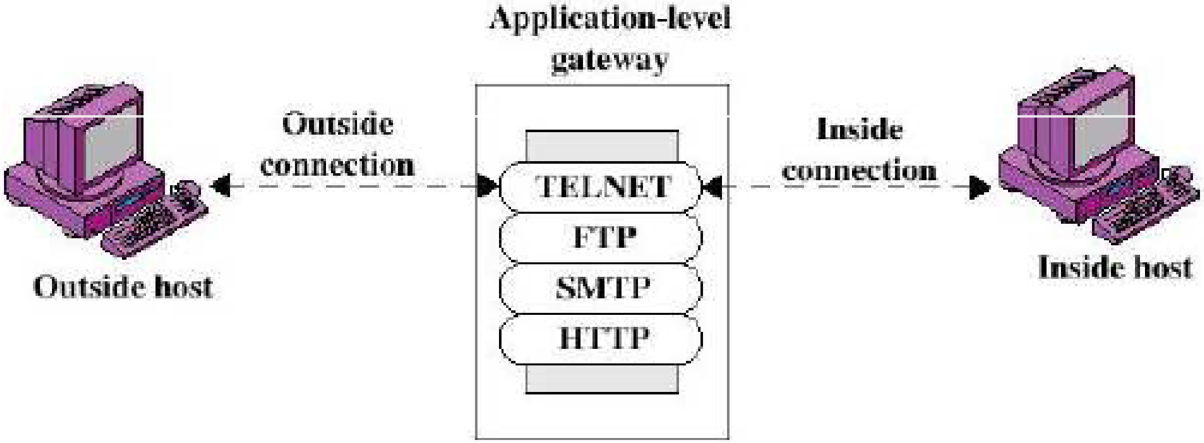
2. Application Level Gateway

- Example: allow selected internal users to telnet outside.

  1. Require all telnet users to telnet through gateway.

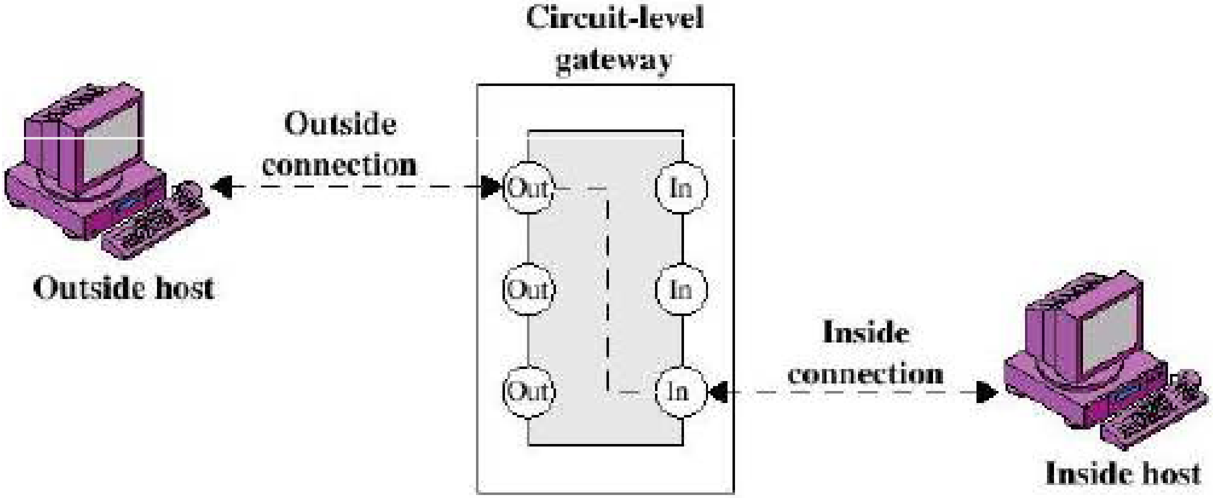
  2. For authorized users, gateway sets up telnet connection to destination host. Gateway relays data between 2 connections

  3. Router filter blocks all telnet connections not originating from gateway.



3. Circuit Level Gateway

- It does not permit an end to end TCP Connection directly.
- The gateway setups two TCP Connections (IN and OUT).
    - monitor TCP handshaking between packets to determine whether a requested session is legitimate
- Once two connections are established => Gateway Relays



Limitations of firewall and gateways

- IP spoofing: router can’t know if data “really” comes from claimed source
- The firewall cannot protect against attacks that bypass the firewall.
  - Internal systems may have dial-out capability to connect to an ISP.
- The firewall does not protect against internal threats,
  - such as a dishonest employee
  - or an employee who unwittingly cooperates with an external attacker.
- The firewall cannot protect against the transfer of virus-infected programs or files.
- Impractical and perhaps impossible for the firewall to scan all incoming files, e-mail, and messages for viruses.

Content Filtering

- Block and allow application, website, etc
- Choose which categories of Web sites to block
  - Adult/Mature Content
  - Alcohol/Tobacco
  - Gambling
  - Illegal Drugs
  - Pornography
  - Violence/Hate/Racism
  - Weapons

- The rules to block traffic are based on the traffic’s category of service.
  - **Inbound Rules (port forwarding)**
    - Inbound traffic is normally blocked by the firewall unless the traffic is in response to a request from the LAN side. The firewall can be configured to allow this otherwise blocked traffic.
  - **Outbound Rules (service blocking)**
    - Outbound traffic is normally allowed unless the firewall is configured to disallow it.
  - **Customized Services**
    - Additional services can be added to the list of services in the factory default list. These added services can then have rules defined for them to either allow or block that traffic.

Thank You

If you have any Queries write to me

@

Jalauddin.mansur@gmail.com

jalawdarling@hotmail.com