Application Delivery, Optimization and Security

Marc Van Hoof

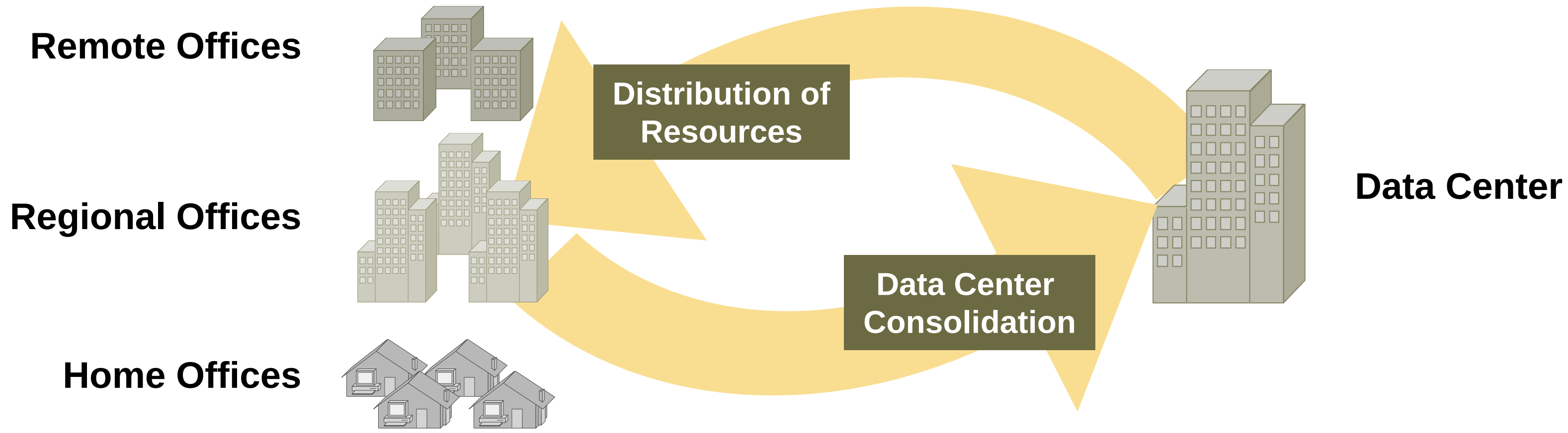
Agenda

- Introduction
  Application Delivery Challenges and Cisco Application Networking Services
- Application Control Engine
  Introduction, Virtual Partitioning, RBAC, App Delivery and Security, Redundancy, Designs
- Application Velocity System
  Optimization
- Application Velocity System
  Security
- Wide Area Application Services

The WAN Application Delivery Problem

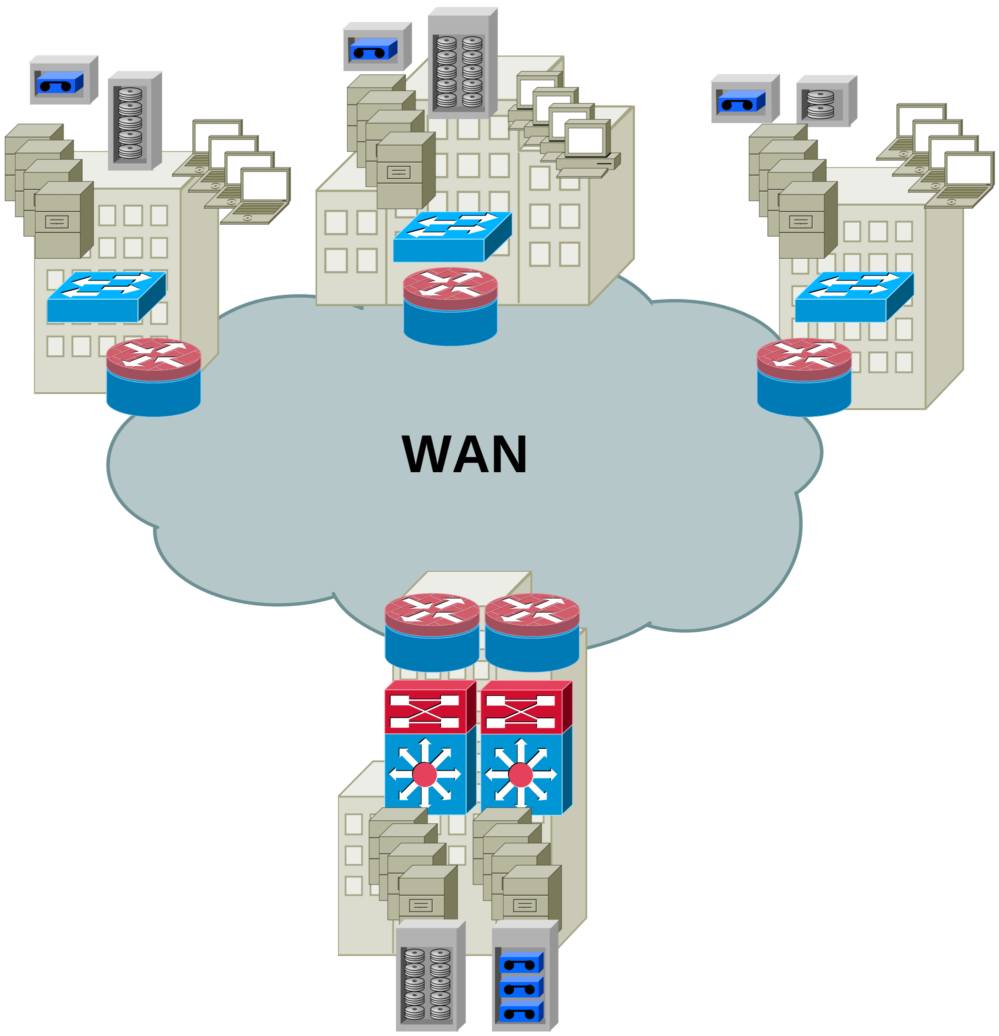
- Increasingly distributed workforce drives need for distribution of I/T resources to remote locations
  - Enable productivity
  - Drive revenue and profits

- Data protection, availability, compliance, and management drives
need for consolidation
  - Fewer devices to manage
  - Fewer points to protect



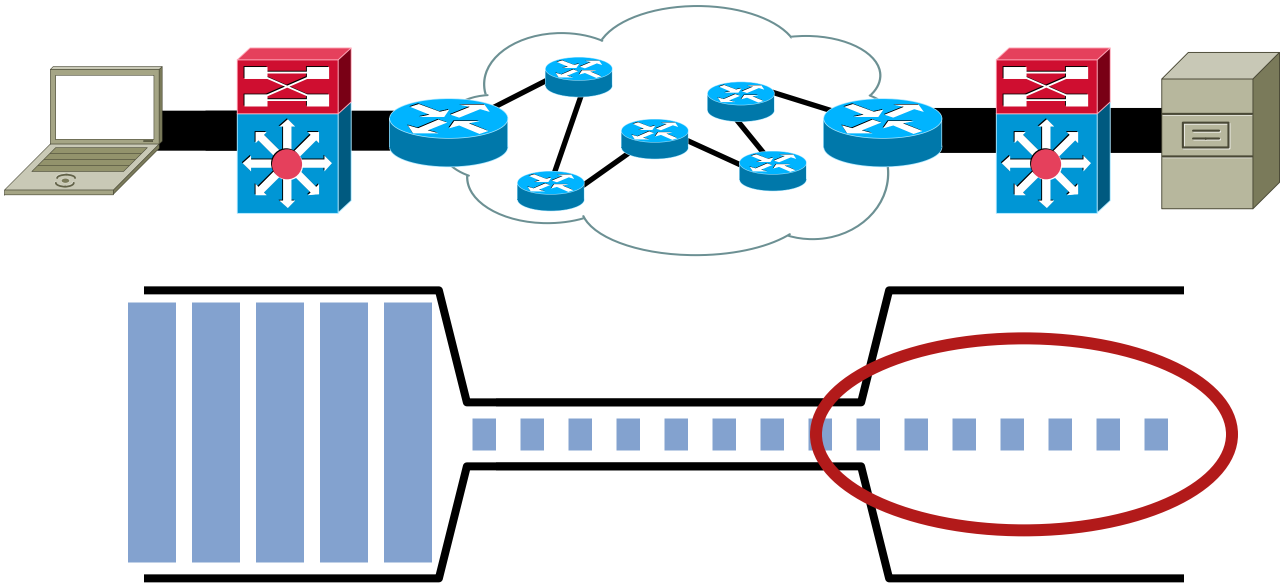
Distributed IT Enables Global Workforce…At a Cost

- Expensive distributed I/T infrastructure
  - File and print servers
  - E-mail servers
  - Tape backup
- Application delivery woes
  - Congested WAN
  - Bandwidth and latency
  - Poor productivity
- Data protection risks
  - Failing backups
  - Costly offsite vaulting
  - Compliance



Bandwidth

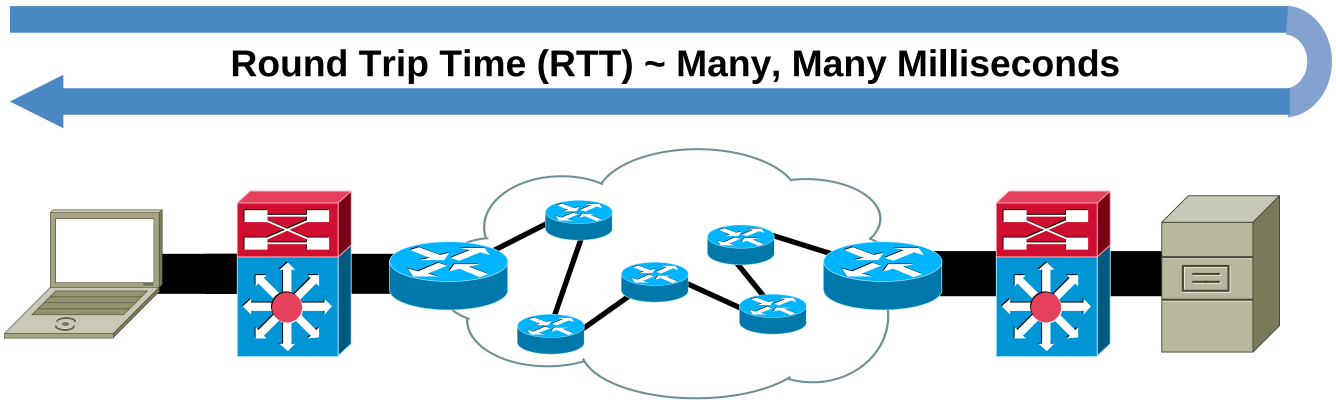
- Bandwidth constraints keep applications from performing well
- Too much data and too small of a pipe causes congestion, packet loss, and backpressure



Latency

Latency Impairs Application Performance in Three Ways

- Network latency: the amount of time necessary for a message to traverse the network
- Transport latency: the amount of time necessary for the transport mechanism (TCP) to acknowledge and retransmit data
- Application latency: “chattiness” of an application protocol causing messages to be exchanged across the network



Application Networking
Business Ready Enterprise

Business-Ready Enterprise

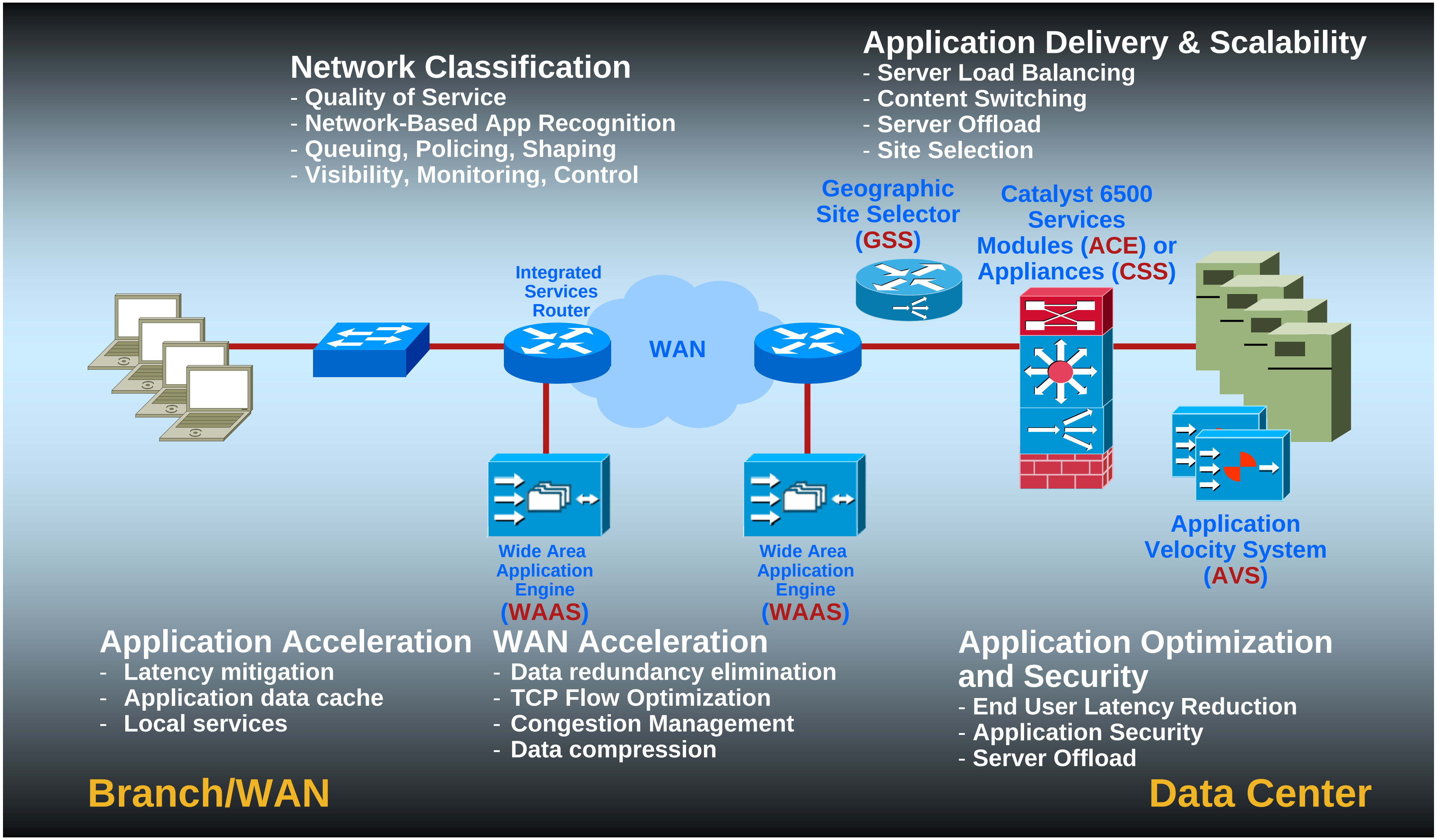
SFA Sales Force Automation	CRM Customer Relationship Management	ERP Enterprise Requirements Planning	ERM Enterprise Resource Management	SCM Supply Chain Management	Com-munications Productivity	Order Processing Vertical

Application Networking Services
Application Delivery and Application-Oriented Networking

Transport Infrastructure Eth, FC, IB, WAN, MAN	Server OS, Hardware	Storage Infrastructure SAN, NAS, DAS

Optimizing Application Performance with Existing Server, Storage, and Network Infrastructure

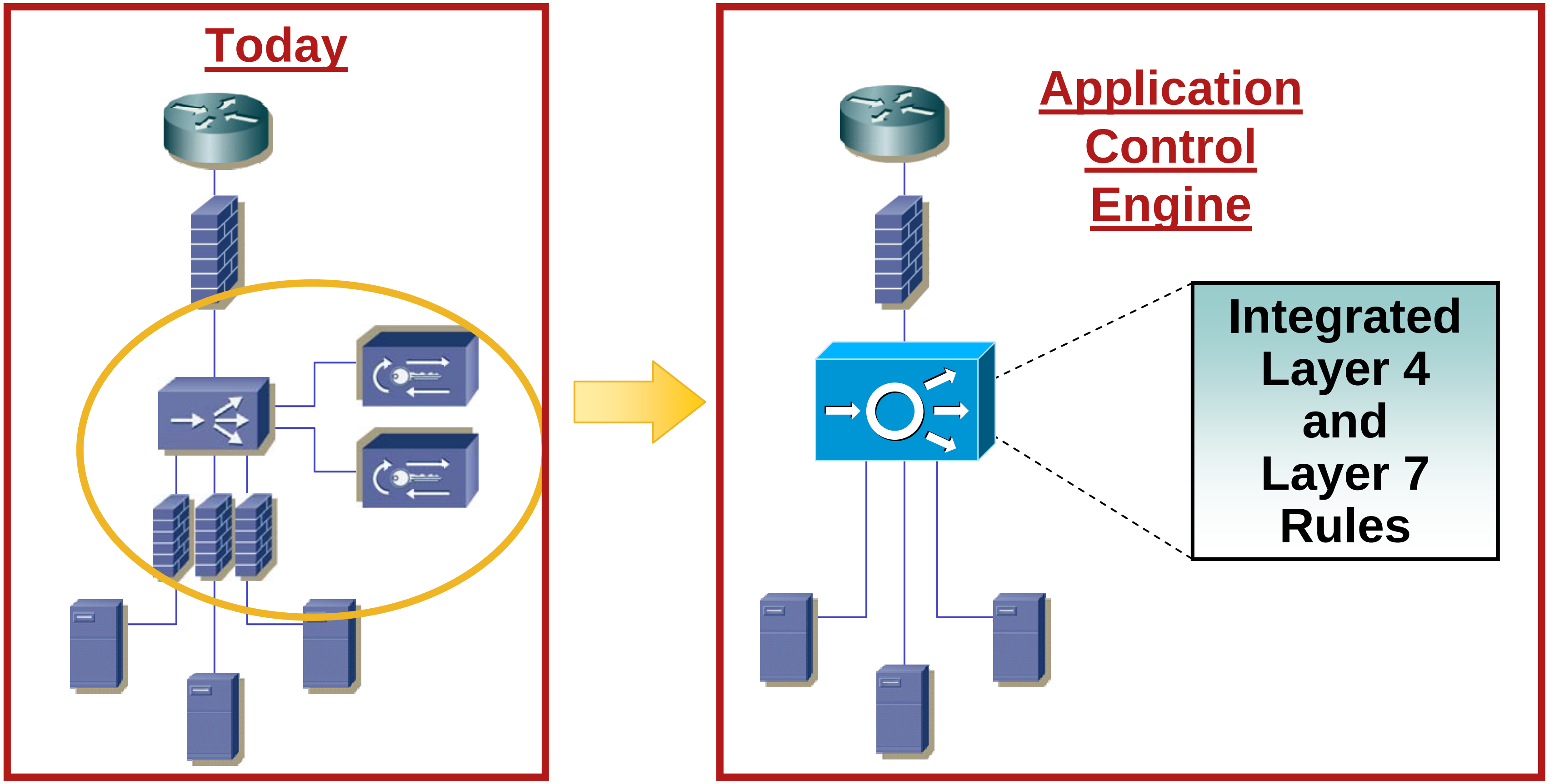
Cisco Application Networking Services



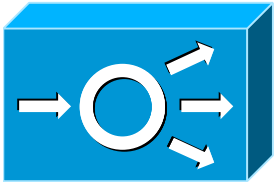
Application Control Engine

A new platform for Application Delivery, Optimization and Security

The Evolution Of L4to7 Services



- **Infrastructure Simplification** with L4-7 Services integration
- **Converged** policy creation, management & troubleshooting
- **Reduced latency** (single TCP termination for all functions)



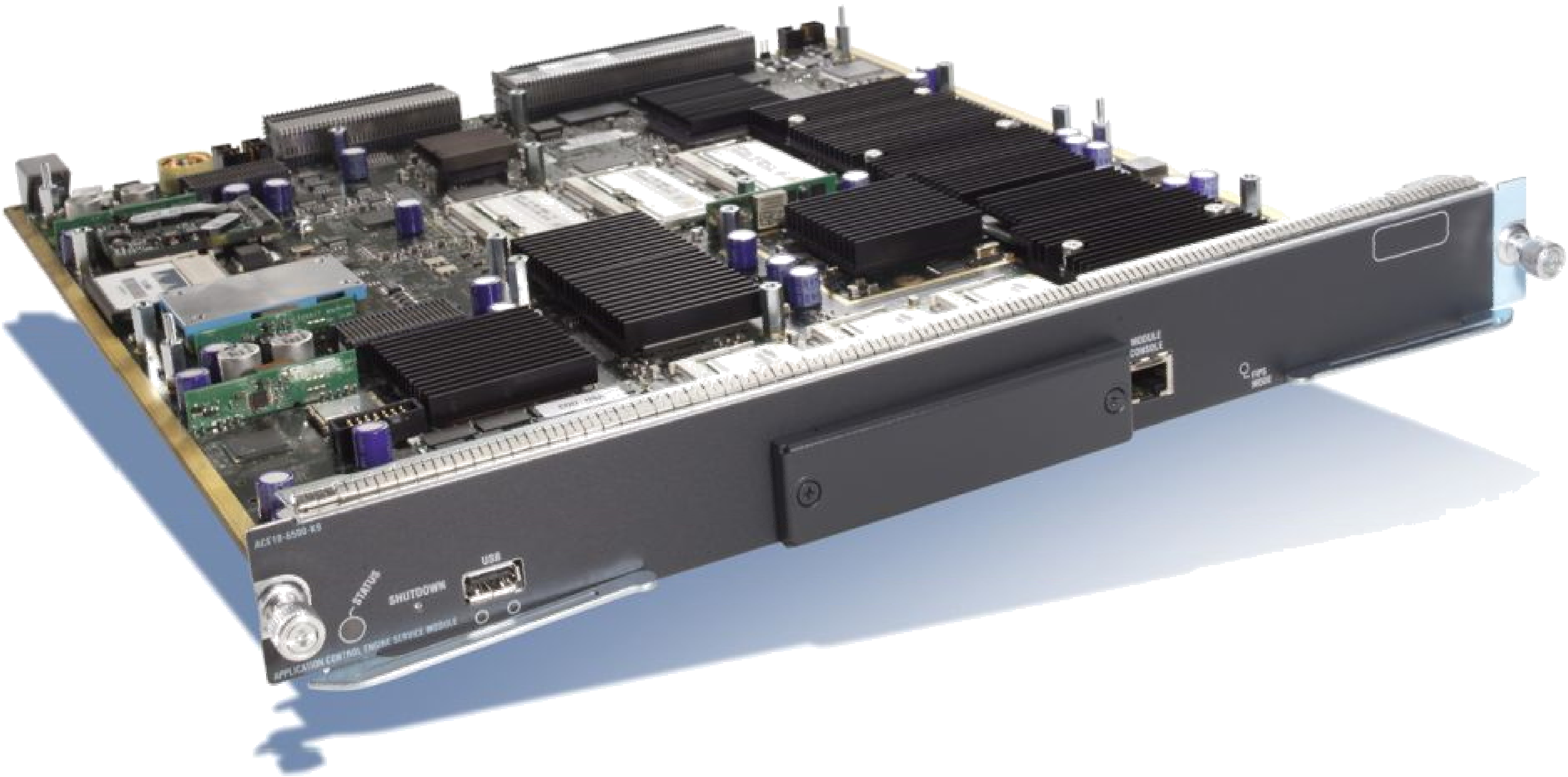
What is ACE ?

Application Control Engine

- Brand new product line in the Cisco ANS portfolio
- Infrastructure Simplicity
  In a single hardware platform, ACE integrates
  - Content Switching
  - SSL Offload
  - Data Center Security features

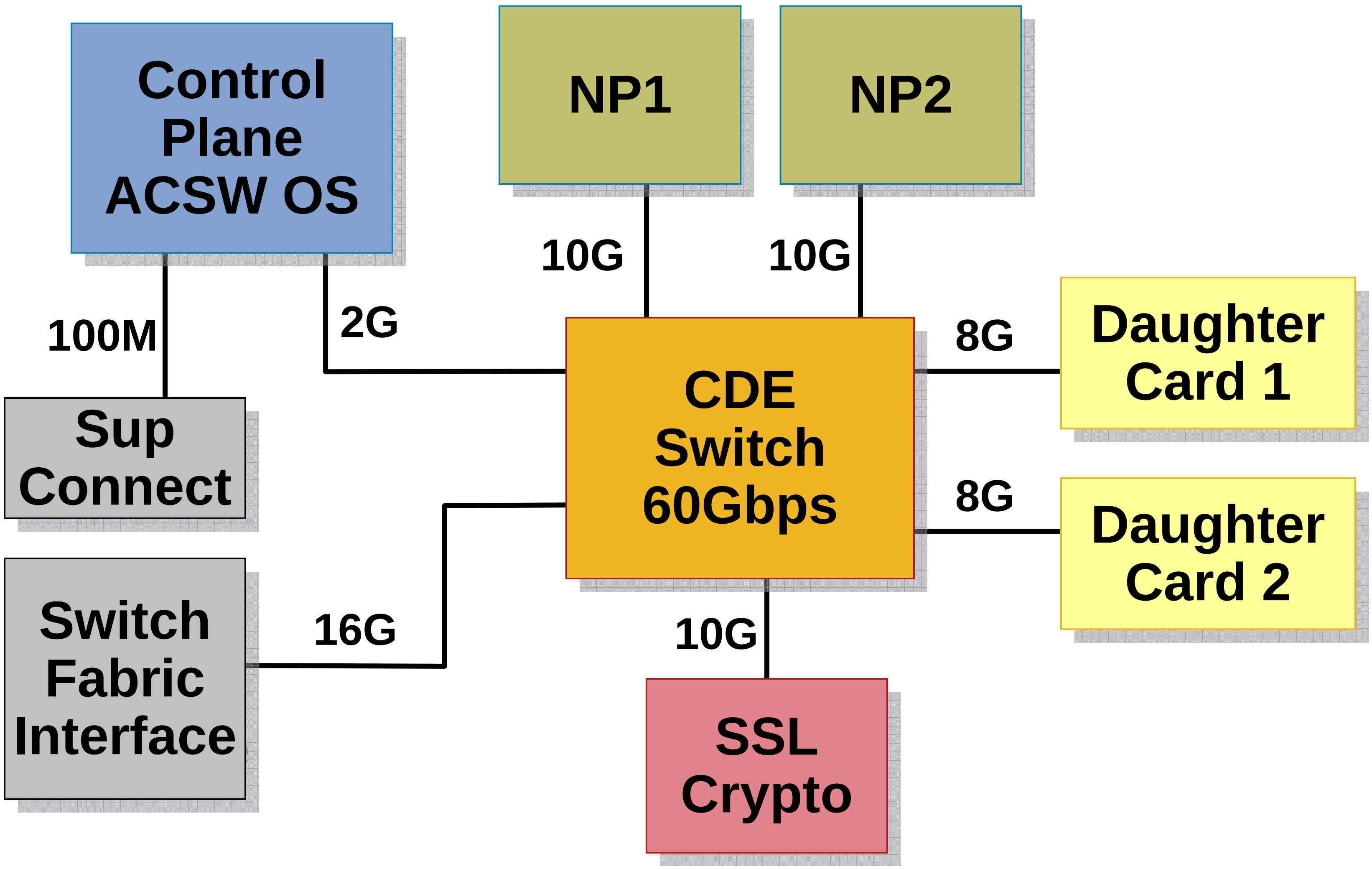
- The first ACE product is a Catalyst 6500 Service Module, which comes in 3 flavours: 4Gbps, 8Gbps and 16Gbps
- The hardware supports 2 field-replaceable daughtercards for future hardware-accelerated application delivery functionality like HTTP compression
- It delivers Application Infrastructure Control, with features like virtual partitions and native Role Based Access Control (RBAC)

Application Control Engine



Parallel network-processor based hardware with separate control and data-path CPU's

ACE – Hardware Architecture



Policy CLI Overview

1. Define match criteria
2. Associate actions to match criteria
   Actions can be access control, NAT, LB, SSL termination, TCP offload, management, protocol inspection, fixups, etc…
3. Activate the classification-action rules on either an interface or “globally”

```
class-map C1
  match <criteria>
```

```
policy-map P1
  class C1
    <action>
```

```
interface vlanX
  service-policy input P1
```

Configuration Rollback

- Allows a user to checkpoint the running configuration and then rollback to that configuration at a later time
  - Quick recovery from configuration errors
  - Fast switching between multiple test or training configurations
- Can also be used to clear the running configuration for any context without requiring a device reload.
- Up to 10 checkpoints per context
  - Maintained as ASCII configuration files and stored as hidden files in the compact flash
- At rollback, a diff is generated between the 2 snapshots and is applied to running configuration to revert back to the check-pointed configuration

ACE Virtual Partitioning and Role Based Access Control

Providing Application Infrastructure Control

Models of "Virtualization"

Abstraction

Physical elements are represented by an abstract entity

- HSRP, VRRP
- VIP, NAT

Pooling

Multiple physical entities appear and treated as one

- link-bundling (etherchannel)
- TCP connection pooling

Partitioning

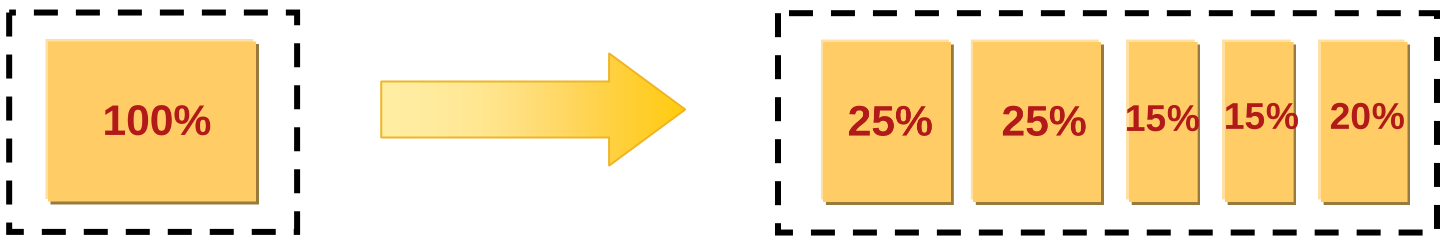
Single physical entity partitioned as multiple distinct entities

- VLAN's (data-path only)
- VRF's (data-path only)
- FWSM virtual contexts (both data- and control-path)

Virtual Partitioning

One physical device

Multiple virtual systems (partitioned control and data path)



<u>Traditional device</u>

Single configuration file

Single routing table

Limited RBAC

Limited resource allocation

<u>Cisco Application Services Virtualization</u>

Distinct configuration files

Separate routing tables

RBAC with Contexts, Roles, Domains

Management and data resource control

Independent application rule sets

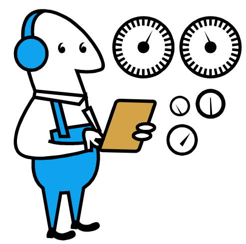
Global administration and monitoring

Virtual Partitioning
Resource Control

Per context Control

- Resource levels for each context
- Support for over-subscription

Rates



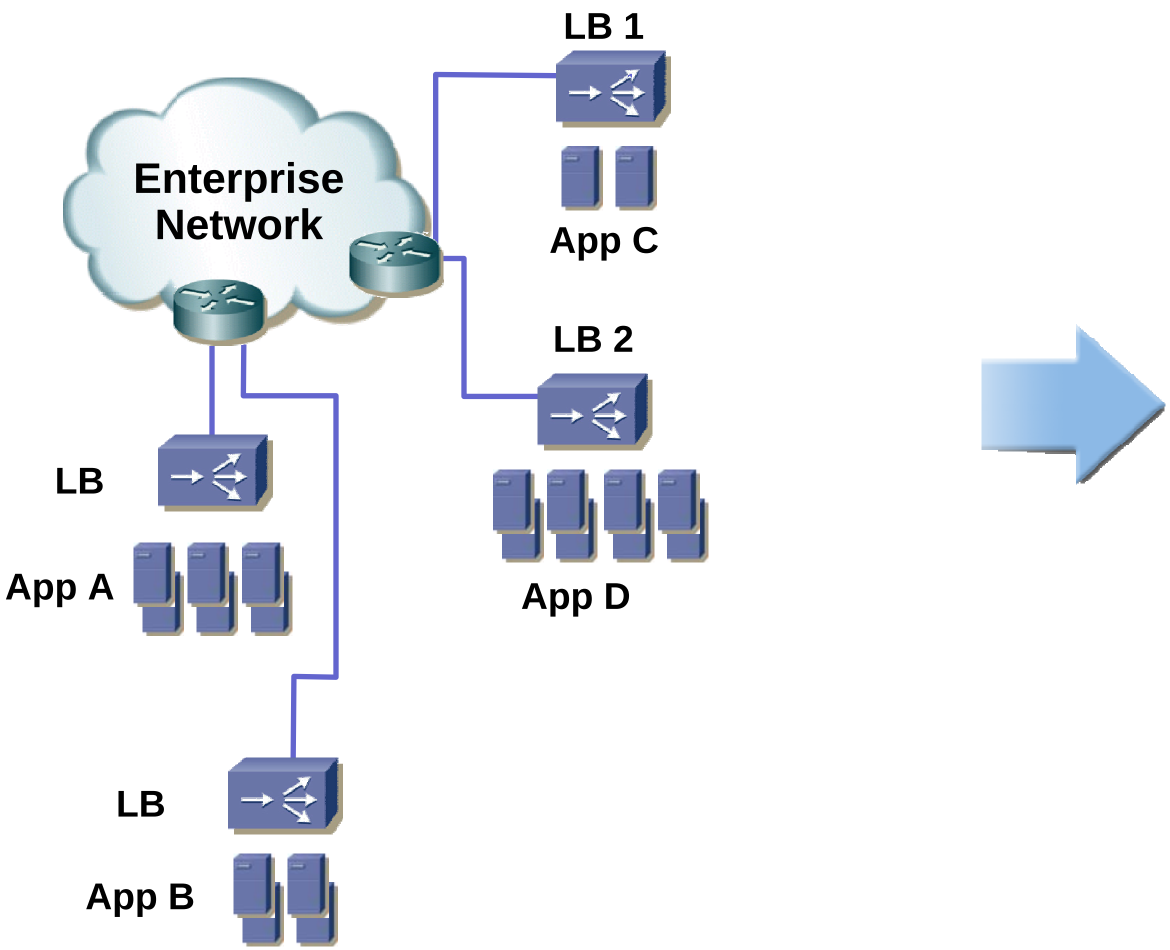
Memory

Bandwidth
Data connections / sec
Management connections / sec
Ssl-bandwidth
Syslogs / sec

Access Lists
Regular Expressions
Data connections
Management connections
SSL connections
Xlates
Sticky entries

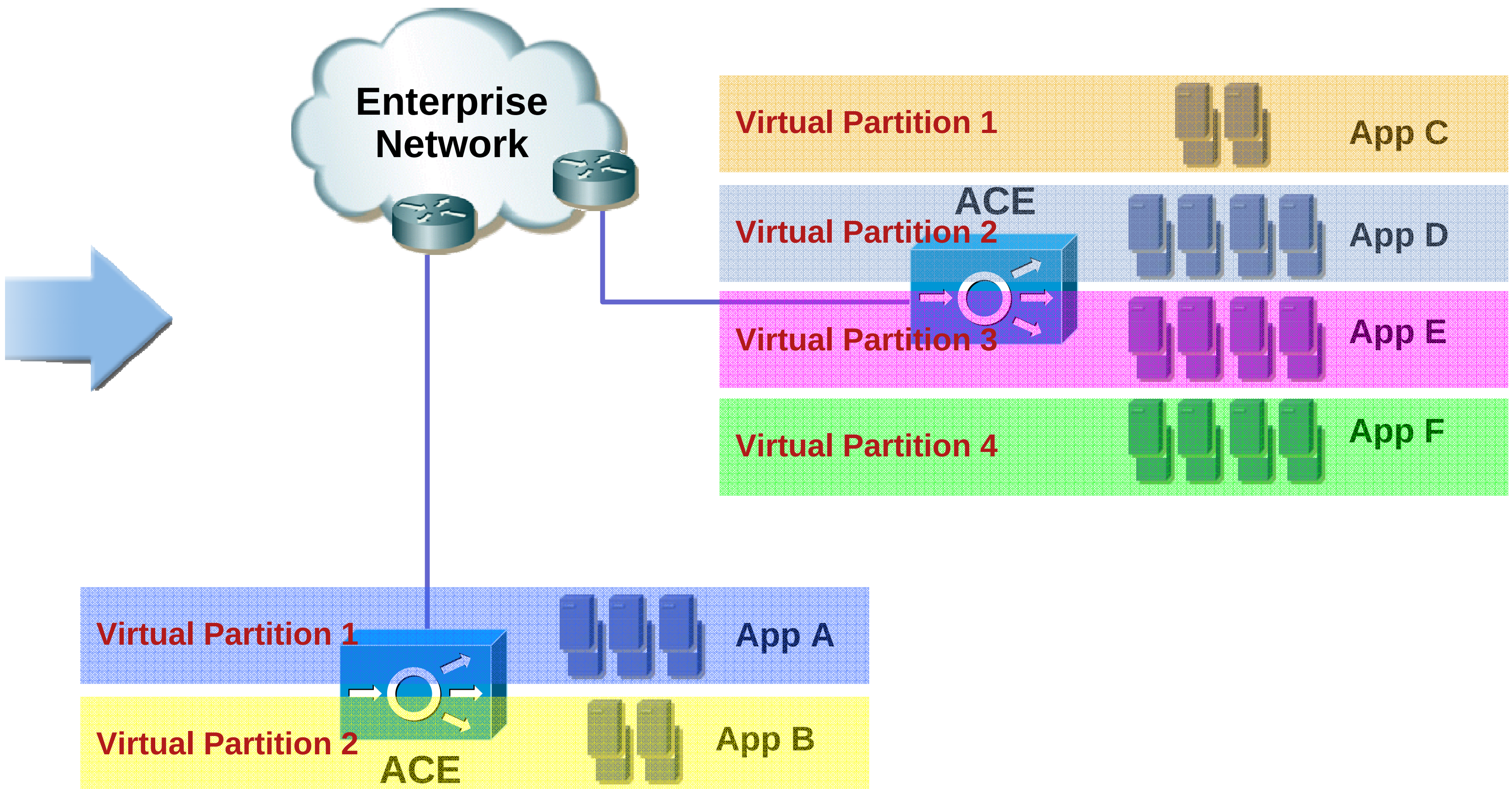
ACE in Action
Applications over Multiple Load Balancers

Enterprise with growing number of applications



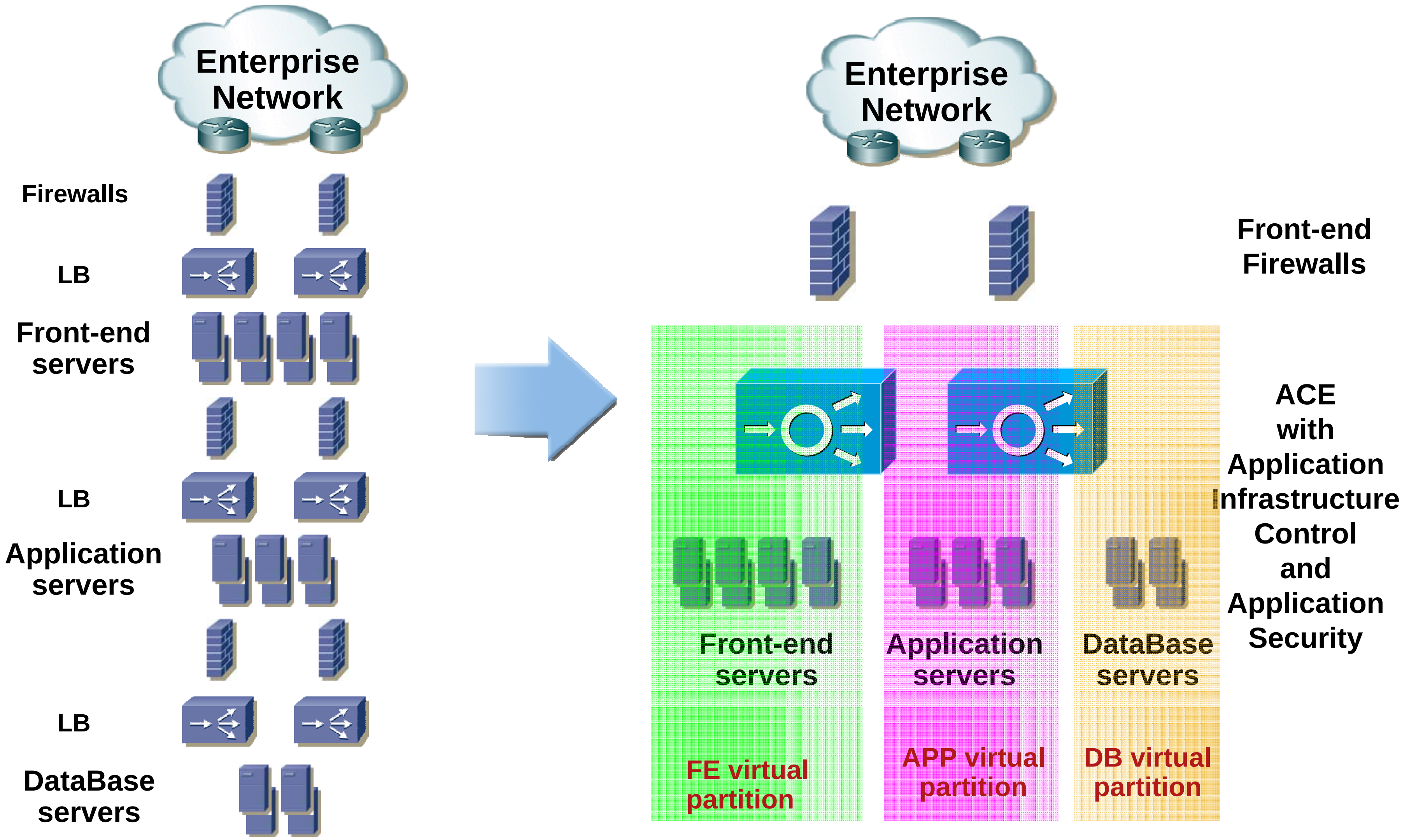
ACE in Action
Applications over Multiple Load Balancers

Enterprise with growing number of applications

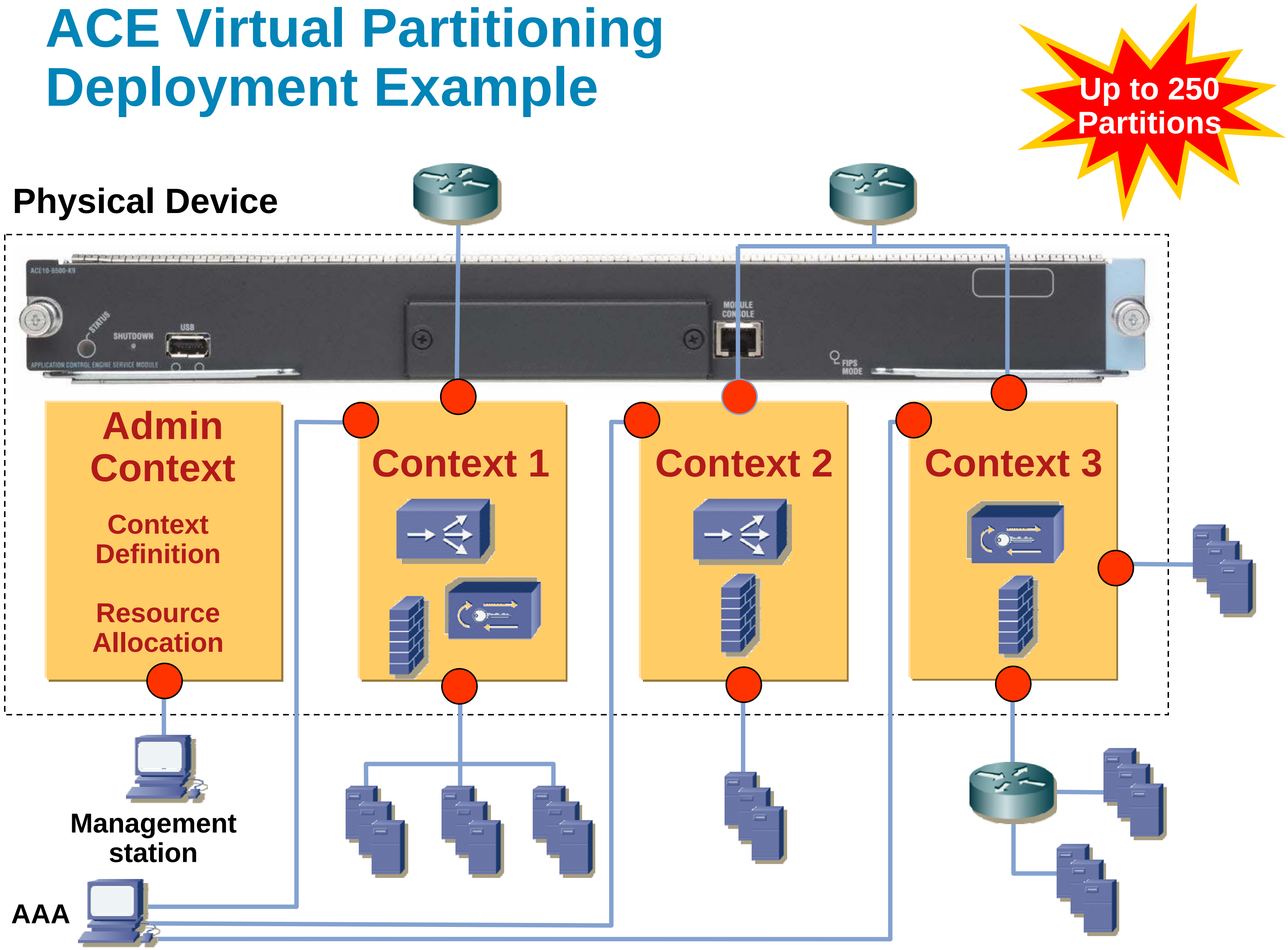


ACE Virtual Patitioning and App Security in Action

Multi-tier Applications



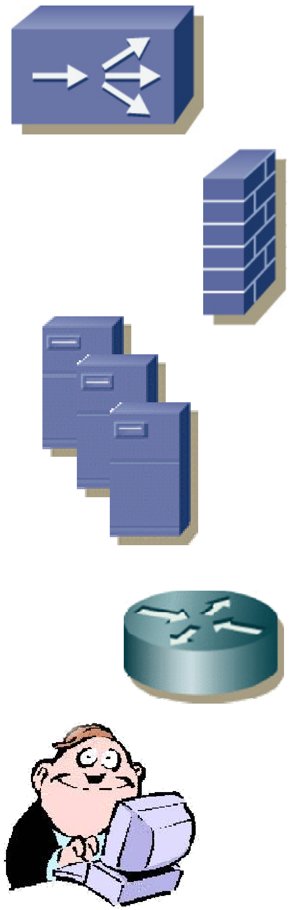
ACE Virtual Partitioning Deployment Example



Role Based Access Control (RBAC)

- Fully integrated Role Based Access Control
- Four main levels of actions over categories of commands
    1. Create/Delete
    2. Modify
    3. Debug
    4. Monitor
- Roles are defined by specifying which actions can be performed on the sets of commands
- Pre-defined roles
- New roles can be created to adapt to different organization structures

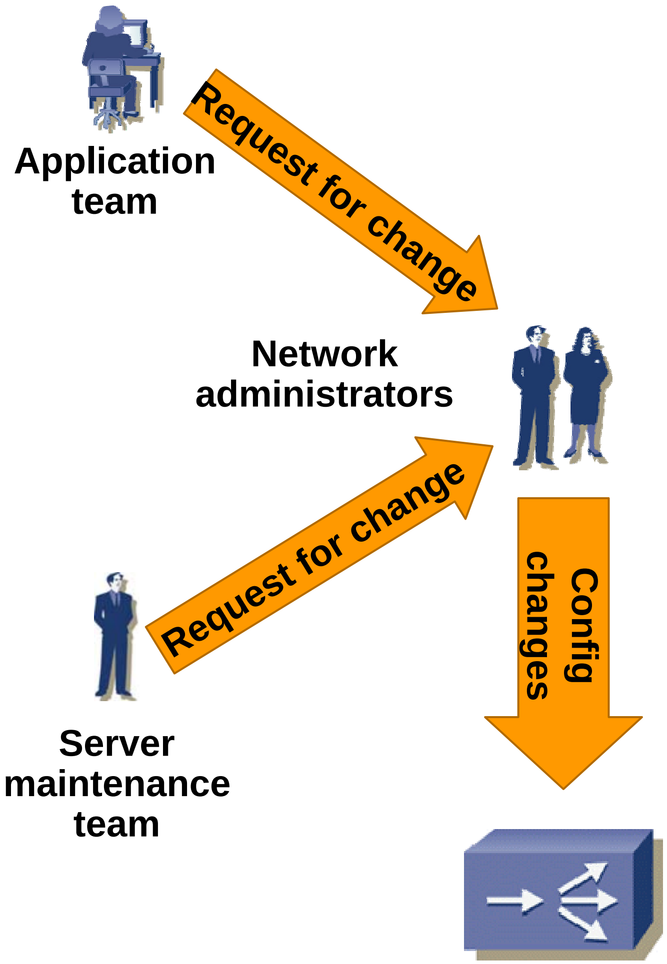
Default Roles in the System

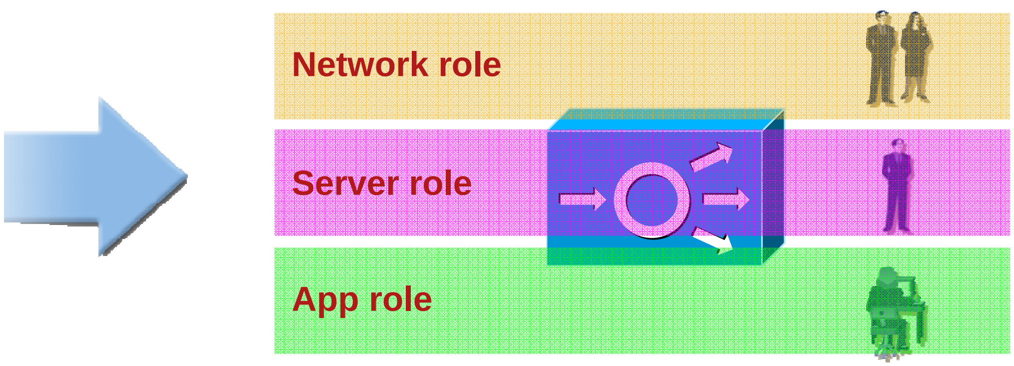


- Admin
  Access to all functions in the context/device.
- SLB-Admin
  Serverfarm, Servers, Health Monitoring
- Security-Admin
  Access Control, Inspection, AAA, NAT
- Server-Maintenance
  Servers in/out of rotation, debug of SLB functions
- Server-Application-Maintenance
  Servers, Health Monitoring, Load Balancing Rules
- Network-Admin
  Interfaces, Routing, NAT, TCP
- Network-Monitor
  Access to all show commands only

ACE Role Based Access Control in Action
Addresses Network Management Inefficiencies

Lack of delegation – Continuous requests for change

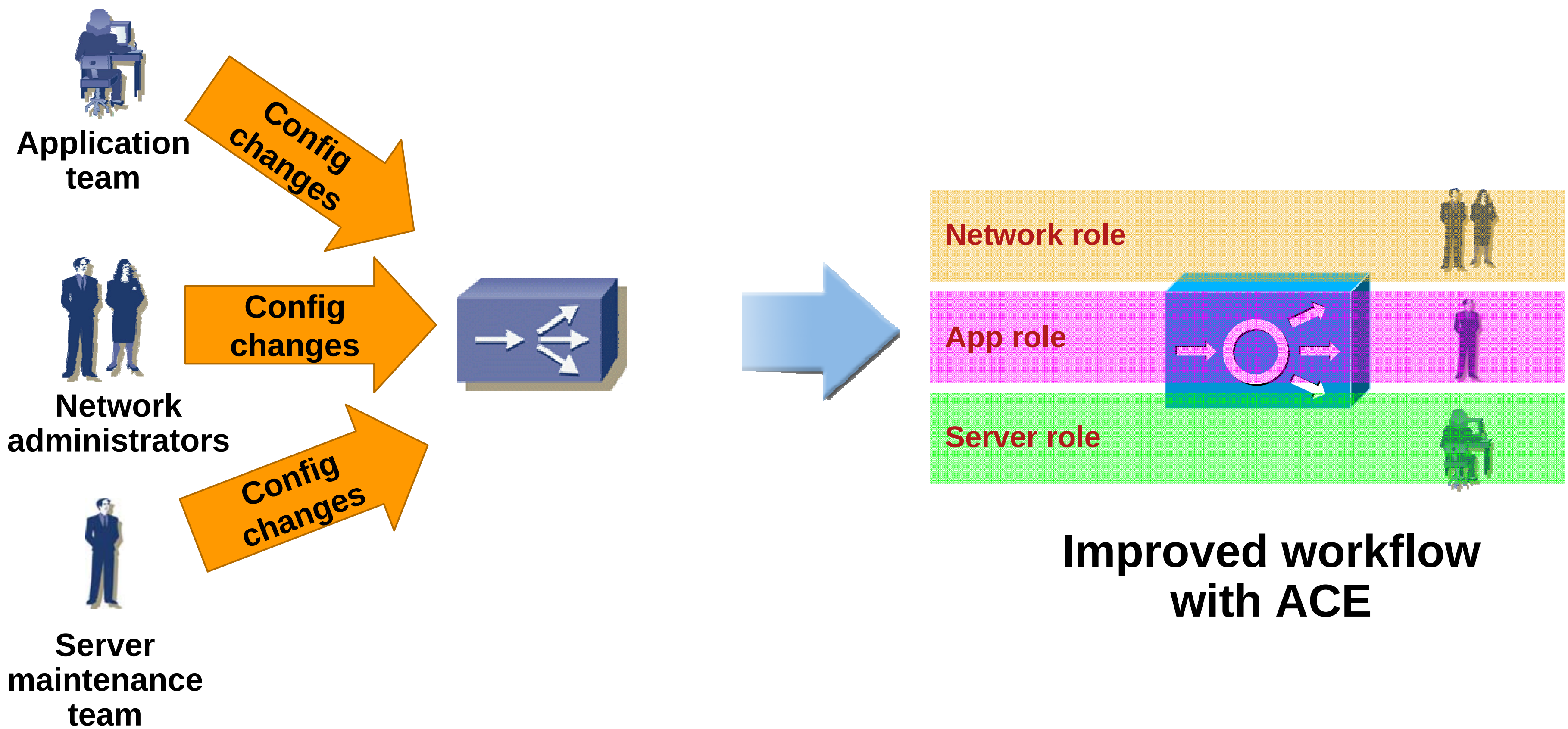




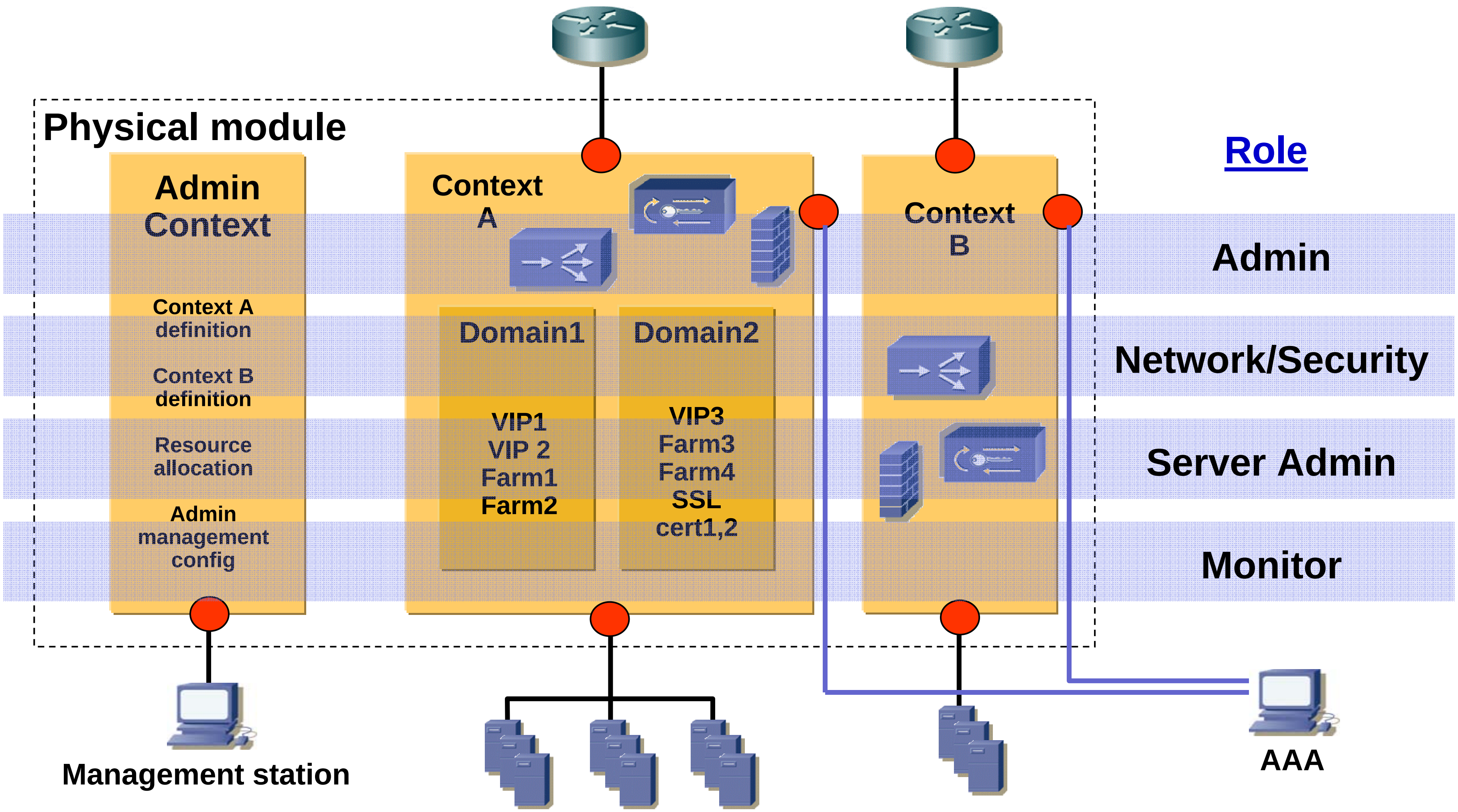
Improved workflow with ACE

ACE Role Based Access Control in Action
Addresses Network Management Inefficiencies

Trust model - Prone to conflicting changes and errors



Contexts, Roles, Domains



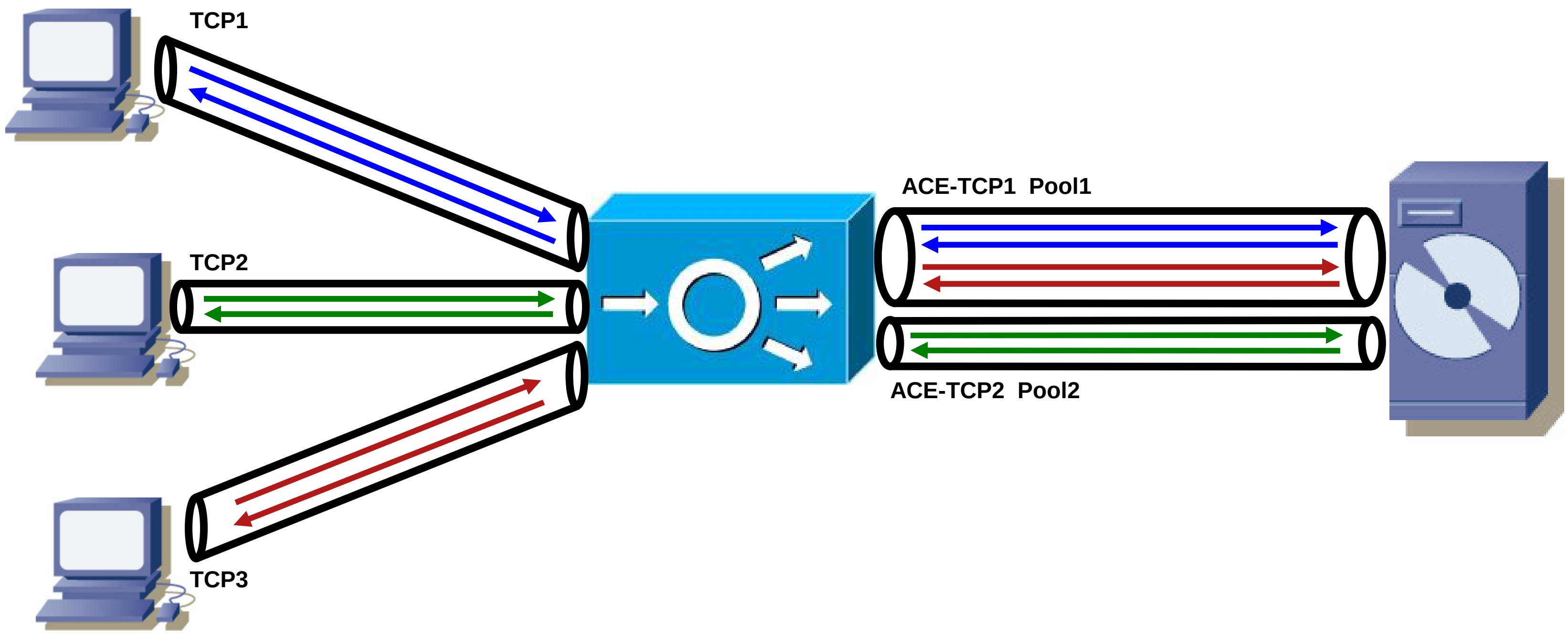
ACE Application Delivery and Security

Accelerating and Securing Applications

TCP Reuse (Offload)

- Offload TCP (HTTP) setup processing from server
- TCP connections to the server are kept open (HTTP 1.1 Persistence)
- Client requests multiplexed to existing server connections
- TCP Reuse can be enabled on per virtual server basis.
- Creates a connection pool on the reals [ip:port] associated to the virtual server
  - Per rserver per serverfarm
  - Client connections matched to server connections based on TCP options
    - **Sack**
    - **timestamp**
    - **window_scale**
    - **MSS**

TCP Reuse (Offload)



Hardware-based IP Normalization

- Always Enabled
- Entirely performed in hardware
- Following packets are dropped
  1. src IP == dest IP
  2. src IP or dest IP == 127.x.x.x
  3. dest IP >= 240.0.0.0
  4. src IP == 0.x.x.x
  5. src IP >= 224.0.0.0
- src IP == 0.0.0.0 and dest IP == 255.255.255.255 allowed for DHCP requests

Hardware-based TCP Normalization

TCP standard header checks

Always performed

I. **src port and dest port != 0**
II. **Only SYN packet allowed to create connection**
III. **TCP header >= of 20 bytes**
IV. **TCP header <= ip->length – ip->header_length**
V. **urg flag cleared if urg_pointer is zero**
VI. **If urg flag not present urg_pointer is cleared**
VII. **Illegal flags combinations dropped (SYN|RST etc.)**

- TCP option processing
- TCP state tracking
- TCP window checking

Configurable

I. **reserved bits allow/clear/drop**
II. **urg flag allow/clear/drop**
III. **syn-data allow/drop**
IV. **exceed-mss allow/drop**
V. **random-seq-num-disable**

User configurable
Random Sequence Numbers

Inspection in ACE

- Protocol-specific inpection supported for
  - FTP
  - Strict FTP
  - RTSP
  - ICMP
  - DNS
  - HTTP / S

Performed on NP CPU

Performed on NP Micro Engines

HTTP Inspection Overview

- HTTP Inspection is a special case of Application Firewall in which the focus is mainly on HTTP attributes such as
  HTTP header, URL, the payload itself
- Enables users to validate, filter and log the HTTP transactions by matching the traffic against the policies configured.
- Shares the HTTP stack and the REGEX engine with L7 SLB with added features for inspection
- Can work in conjunction with L7 Loadbalancing for the same flow
- User defined REGEX can be used in a limited way to detect offending traffic by searching for “signatures”

HTTP Inspection Components

- **RFC 2616 Compliance and filtering**

  Protocol Conformance: the 1st line of a REQUEST is "Method SP" and that of RESPONSE is "HTTP-Version SP". etc
  De-obfuscation: override attempts to avoid regex searches by encoding the URL
  Methods: OPTIONS, GET, POST, HEAD, PUT, DELETE, TRACE, CONNECT
  Extensions: INDEX, MOVE, MKDIR, COPY, EDIT, UNEDIT, SAVE, LOCK, NLOCK, REVLABEL, REVLOG, REVNUM, SETATTRIBUTE, GETATTRIBUTE, GETATTRIBUTENAMES, GETPROPERTIES, STARTREV, STOPREV

- **Length and Encoding checks**

  Length: Configurable range for URL and URL Header requests and responses
  Encoding: chunked | compress | deflate | gzip | identity

- **Detect HTTP misuse**

  Peer-to-peer (p2p) applications: KAZAA, GNUTELLA
  Tunneling applications: HTTPort/HTTHost, FireThru
  Instant Messaging: (IMI - YAHOO Messenger)

- **MIME type validation and filtering**

  audio: /*, /midi, /basic, /mpeg, /x-adpcm, /x-aiff, /x-ogg, x-wav (8)
  image: /*, /cgf, /gif, /jpeg, /png, /tiff, /x-3ds, /x-bitmap, /x-niff, /x-portable, /x-xpm (11)
  text: /*, /css, /html, /plain, /richtext, /sgml, /xmcd, /xml (8)
  video: /*, /-flc, /mpeg, /quicktime, /sgi, /x-avi, /x-fli, /x-mng, /x-msvideo (9)
  application: /msword, /octet-stream, /pdf, /postscript, /vnd.ms-excel, /vnd.ms-powerpoint, /x-gzip, /x-java-archive, /x-java-vm, /zip (10)

- **Regex filtering on HTTP messages (headers and payload)**

  Detect protocol running on top of HTTP – i.e. to detect YAHOO MESSENGER, look for YMSG in the first 4 bytes

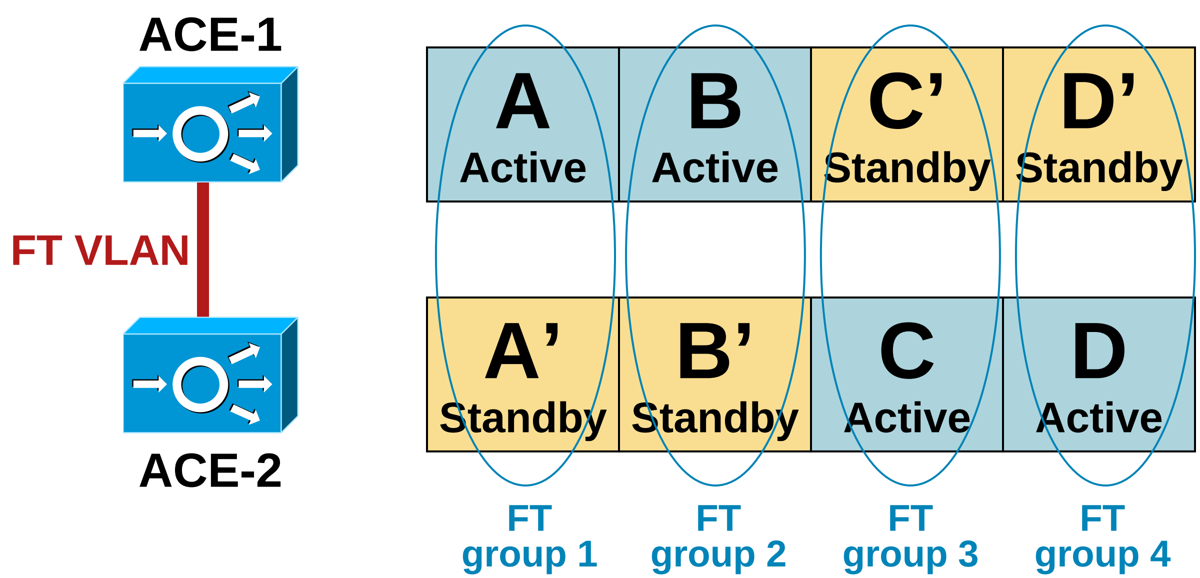
ACE High Availability

Providing the highest reliability for Data-Center Applications

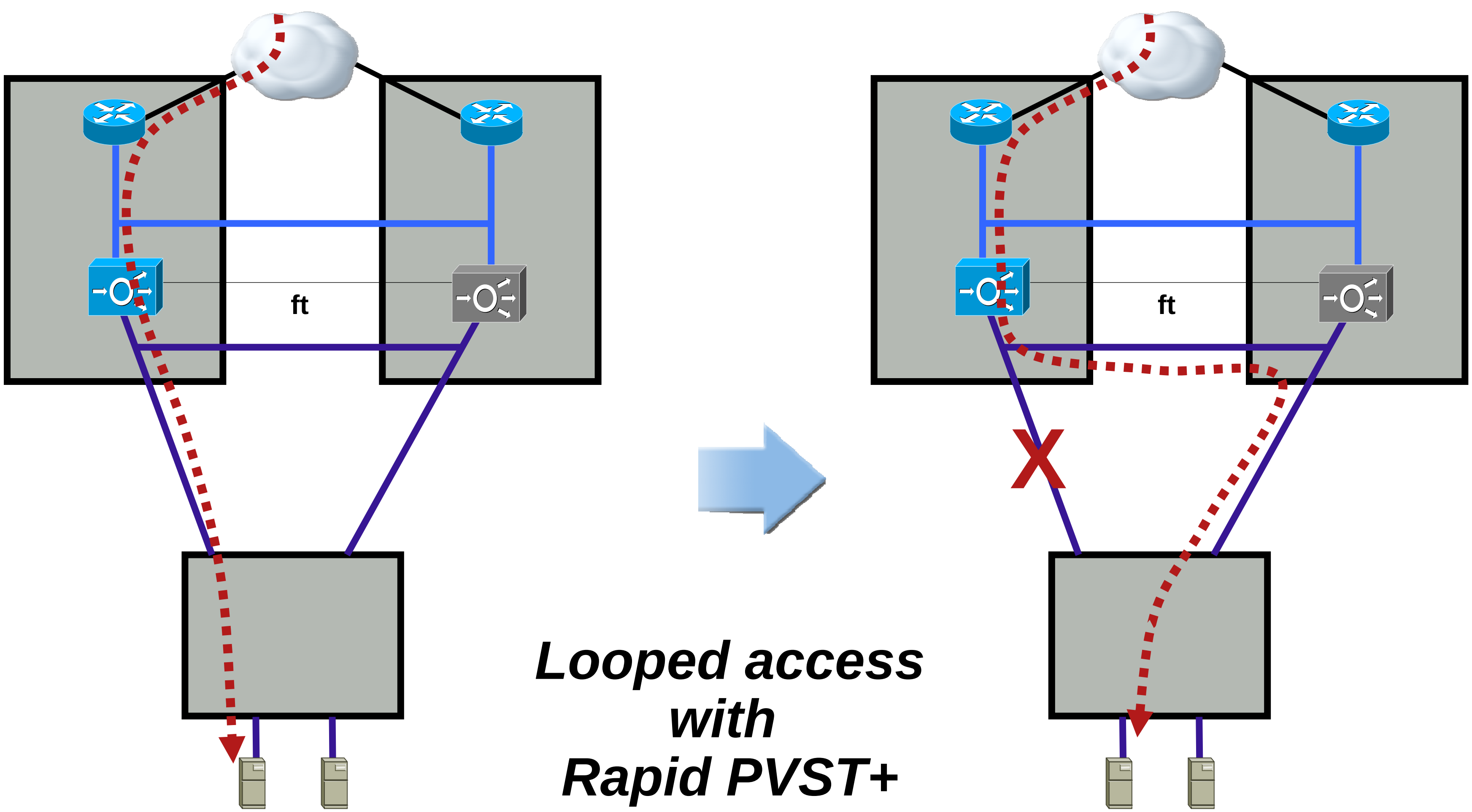
Redundancy Model

- Redundancy groups (Fault Tolerance, FT groups) are configured based on virtual contexts.
- Two instances of the same context (on two distinct ACE modules) form a redundancy group, one being active and the other standby.
- The peer ACE can be in the same or different Catalyst 6k chassis.
- Both ACE modules can be active at the same time, processing traffic for distinct contexts, and backing-up each other (stateful redundancy)

Example:
2 ACE modules
4 FT groups
4 Virtual Contexts (A,B,C,D)



Typical Looped Access Topology



Looped access with Rapid PVST+

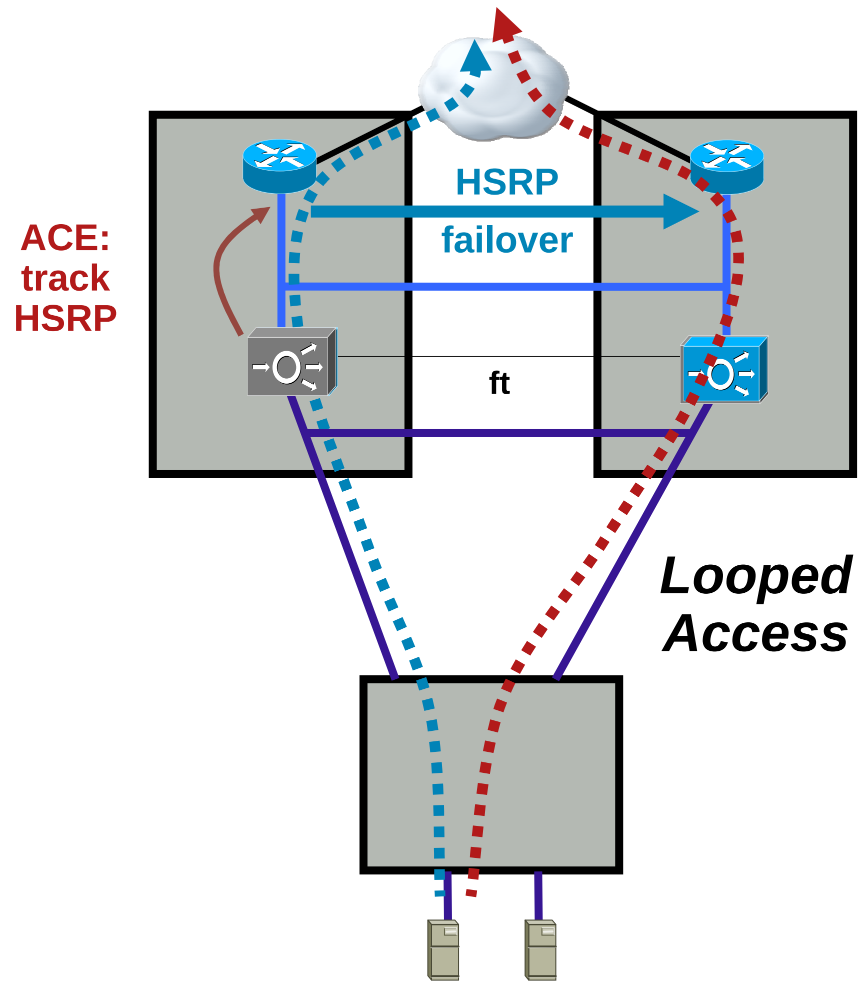
Failover tracking
ACE Tracking HSRP and Hosts

Reduced ISL load

```
ft track hsrp access-trunk
  track-hsrp ACE
  peer track-hsrp ACE
  priority 150
  peer priority 150
```

```
interface Vlan104
 ip address 12.20.40.2 255.255.255.0
 standby 1 ip 12.20.40.1
 standby 1 timers 1 3
 standby 1 priority 112
 standby 1 preempt
 standby 1 name ACE
 standby 1 track …
```

Hosts can be tracked
via generic probes
with priority mechanism

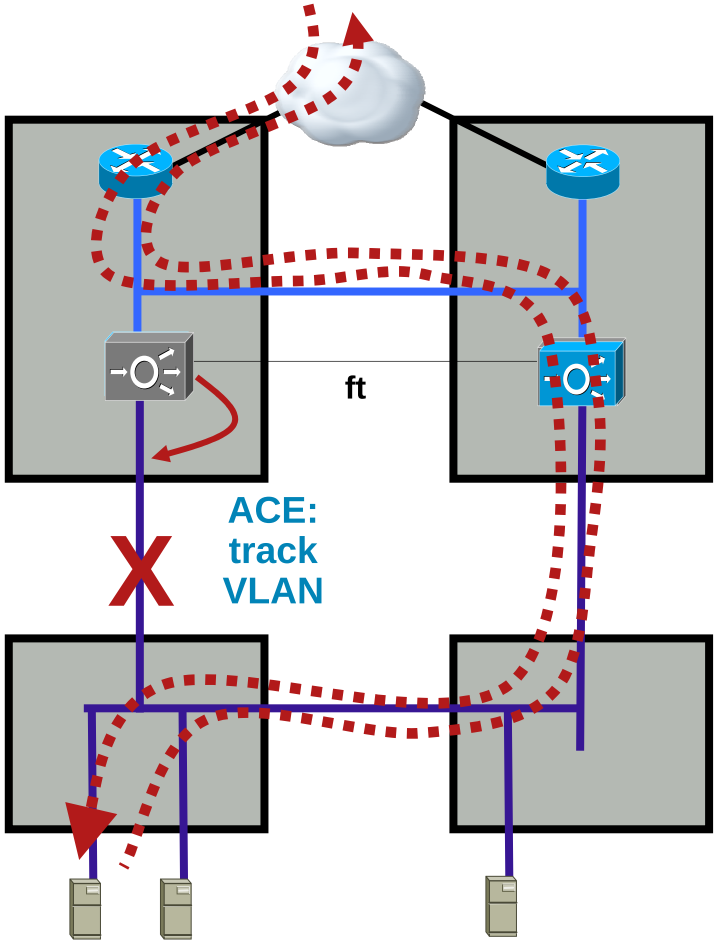


Failover tracking
ACE Tracking VLAN

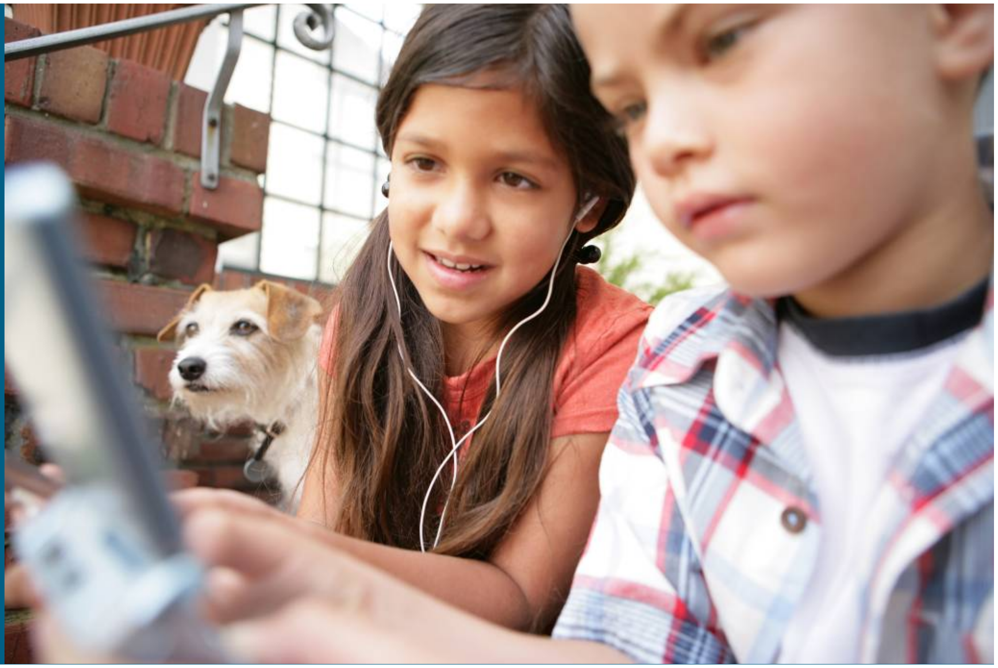
Loop free access

ft track interface vlan-example
  track-interface vlan 204
  peer track-interface vlan 204
  priority 150
  peer priority 150

note: VLAN tracking requires 6500 global command: ***svclc autostate***

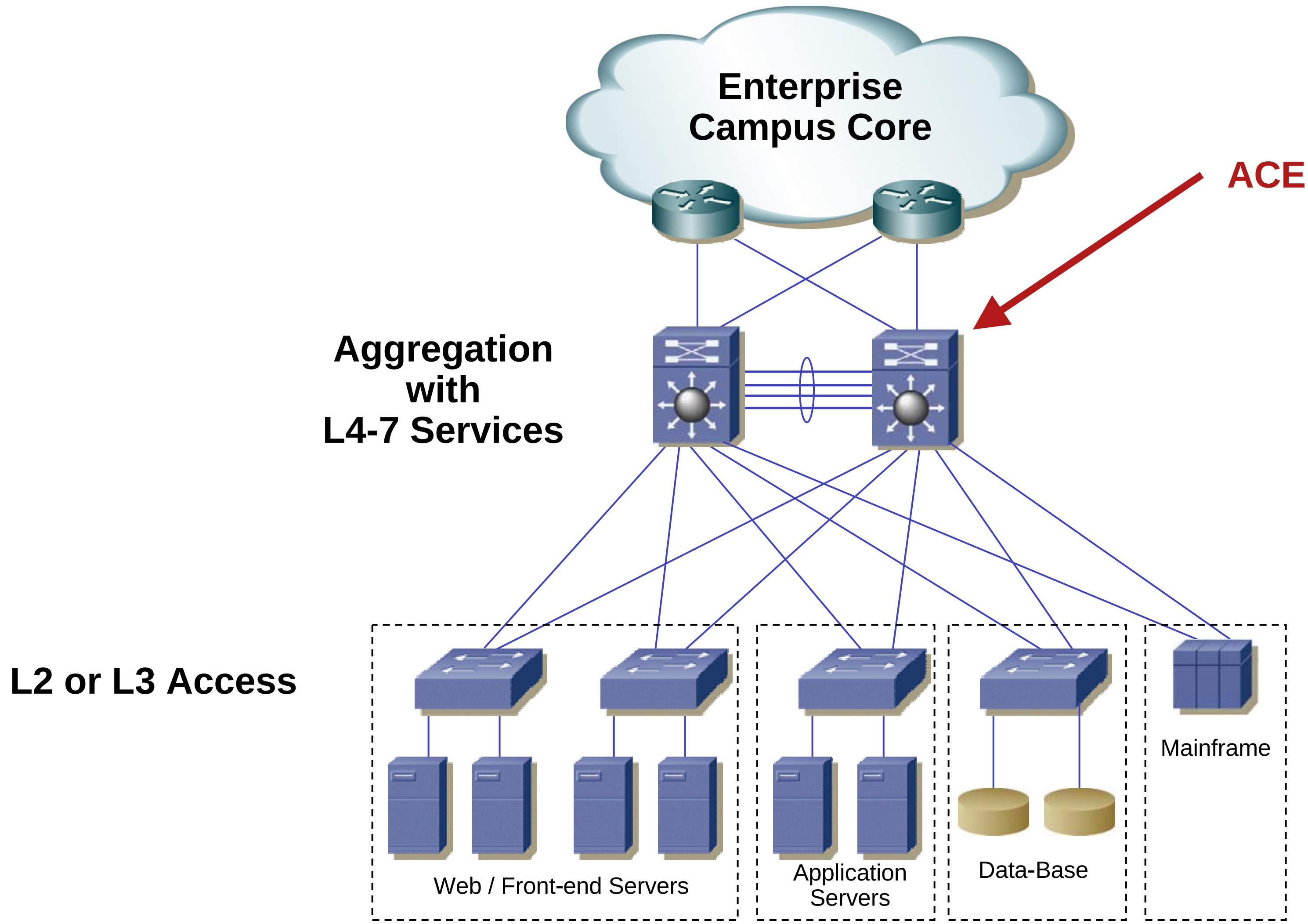


ACE Design Considerations



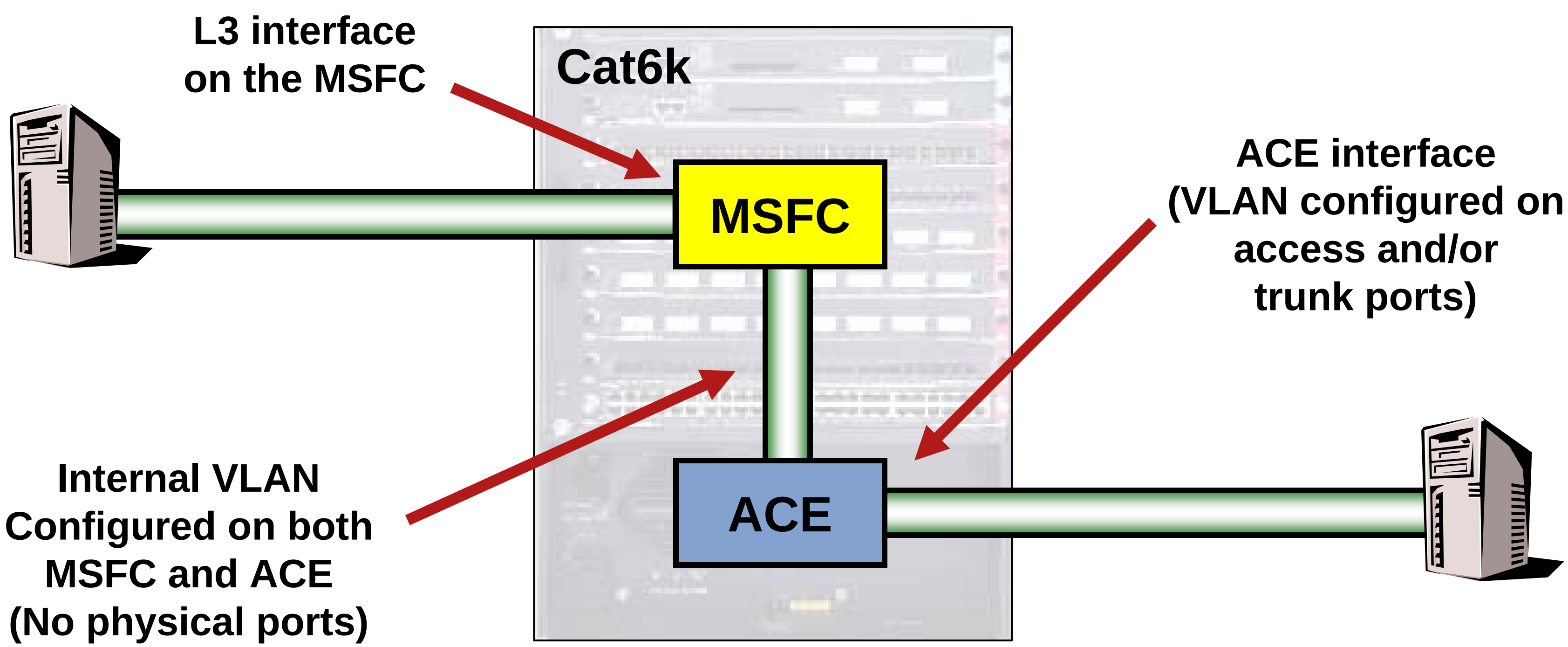
Providing the highest reliability for Data-Center Applications

Typical Data Center Design with ACE

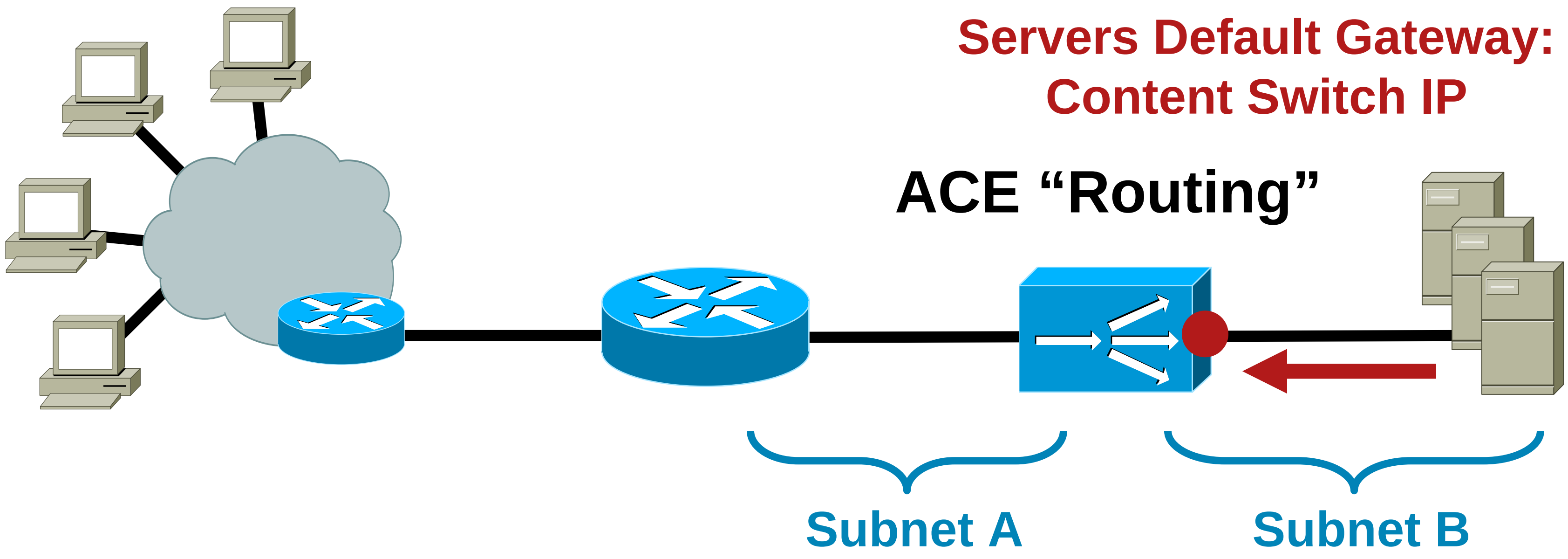


ACE Design Guidelines

- Always "expand" the Catalyst 6500 and represent the MSFC and services modules
- From a network perspective, ACE is a distinct entity

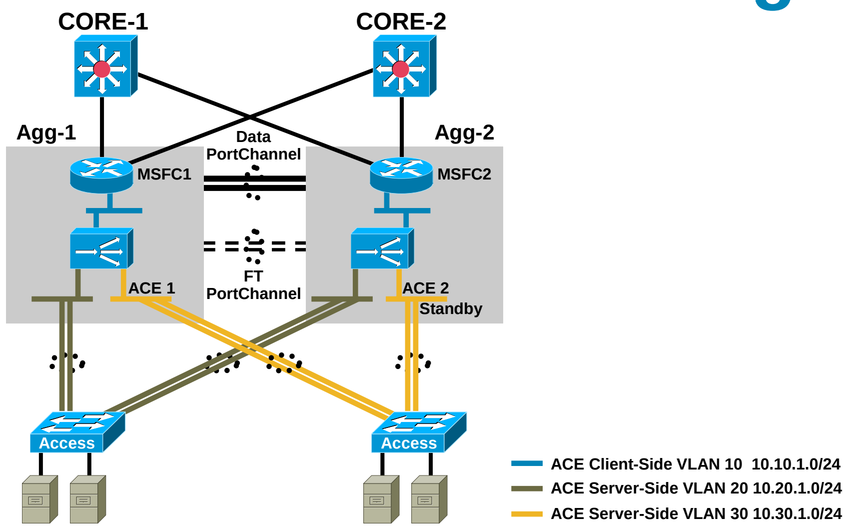


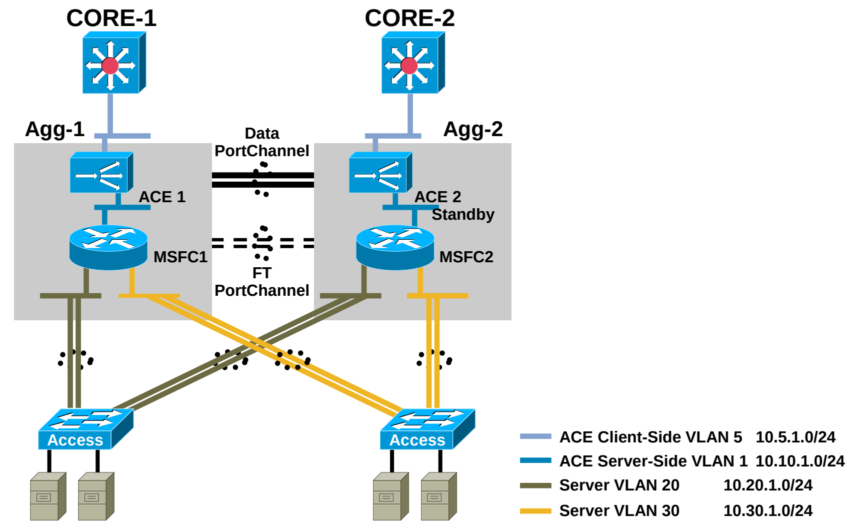
Design Considerations
ACE Router Mode



- Servers in dedicated IP subnet
- VIPs usually in different, routable subnet from servers
- Requires at least two IP subnets
- Easy to deploy with many server IP subnets

Content Switching Design Approaches
Routed Mode: Design





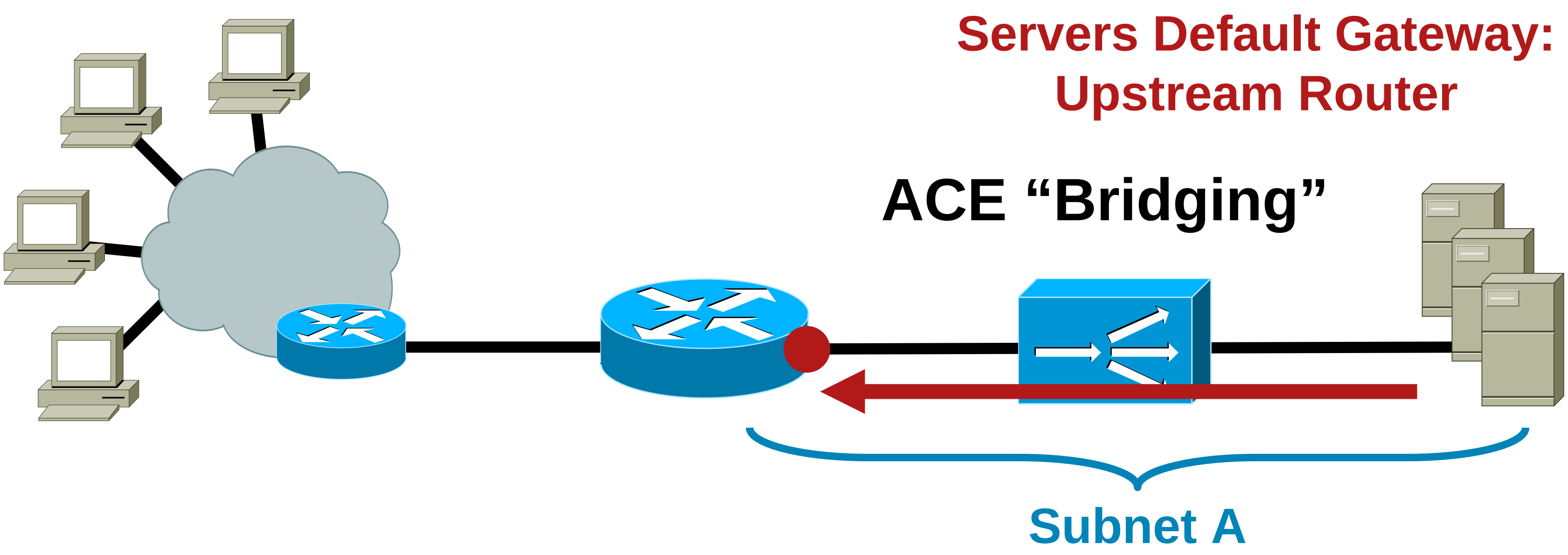
(2A) Routed Mode Design with MSFC on Client Side

- Servers default gateway is the alias IP on the ACE
- Extra configurations needed for:
  - Direct access to servers
  - Non-load balanced server initiated sessions
- ACE's default gateway is the HSRP group IP address on the MSFC
- RHI possible
- Load balancer inline of all traffic

(2B) Routed Mode Design with MSFC on Server Side

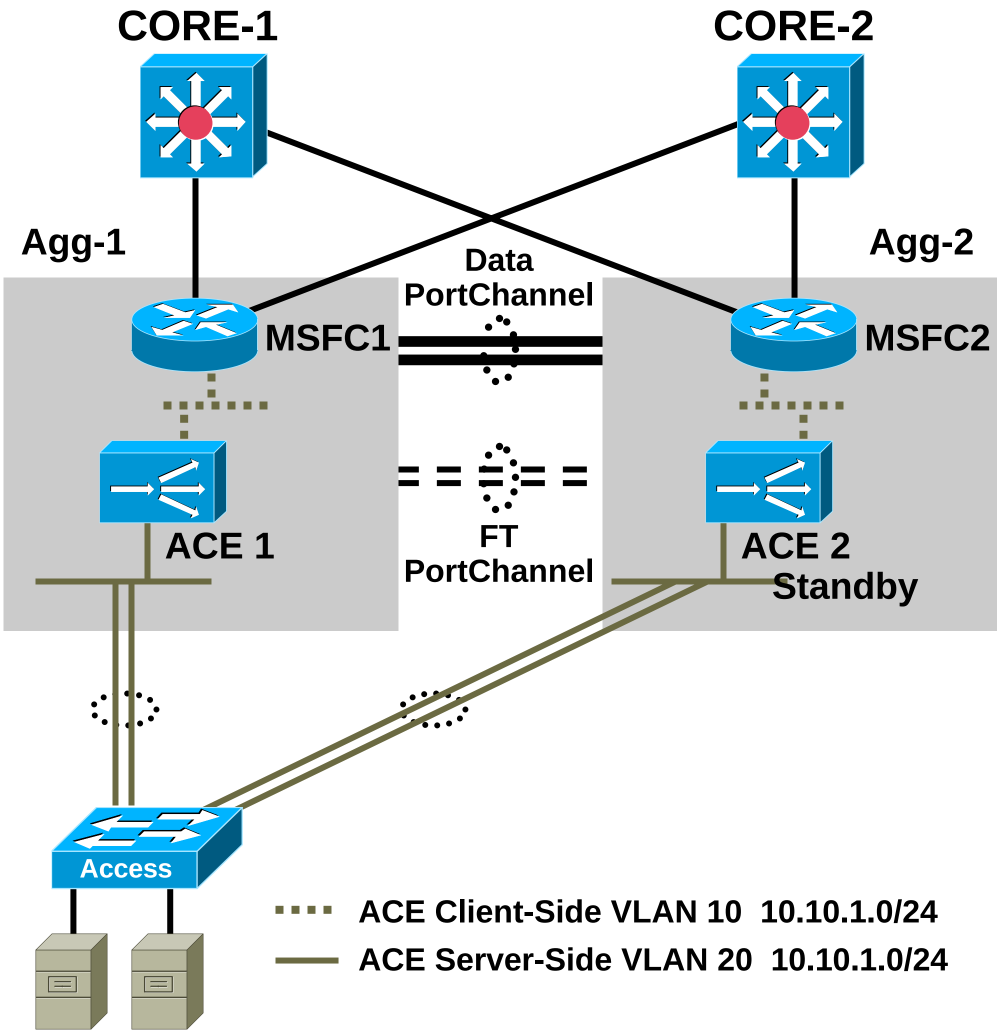
- Servers default gateway is the HSRP group IP address on the MSFC
- Extra configurations needed for:
  - Direct access to servers
  - Non-load balanced server initiated sessions
- SM's default gateway is the core router
- RHI not possible
- Server to server communication bypasses the load balancer

Design Considerations
ACE Bridge Mode



- Servers in routable IP subnet
- VIP's can be in the same or different subnet
- Requires one IP subnets for each farm

Content Switching Design Approaches
Bridged Mode: Design



(1) Bridged Mode Design Considerations

- Servers default gateway is the HSRP group IP address on the MSFC
- Broadcast/multicast/route update traffic bridges through
- No extra configurations for:
  - Direct access to servers
  - Server initiated sessions
- RHI possible
- Load balancer inline of all traffic

Content Switching Design Approaches
Bridged Mode: BPDU Forwarding

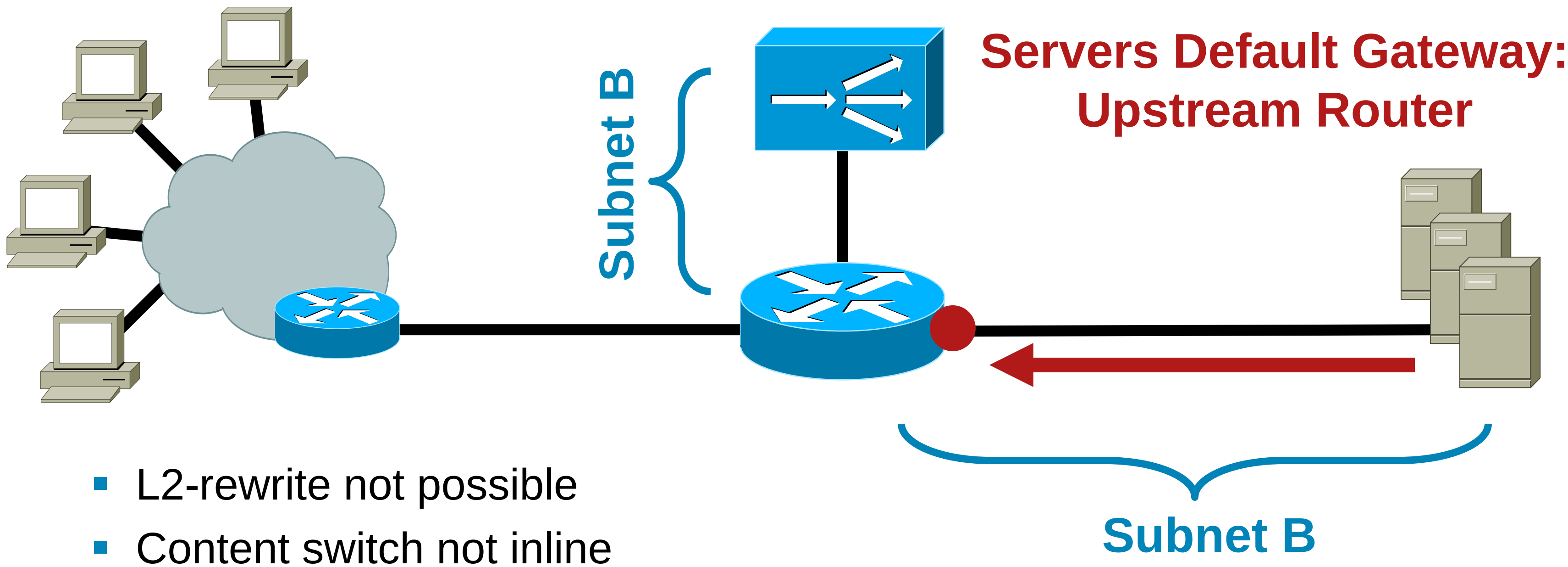
ACE Configuration to Allow BPDUs

```
!
access-list bpduallow ethertype permit bpdu
!
interface vlan 10
 bridge-group 10
 access-group input bpduallow
 no shutdown
!
interface vlan 20
 bridge-group 10
 access-group input bpduallow
 no shutdown
!
```

Similarly to the FWSM, ACE can let BPDU’s through and can rewrite their payload, correctly handling STP merged domains

Protects against accidental loops in case of FT heartbeat cable or VLAN disconnected

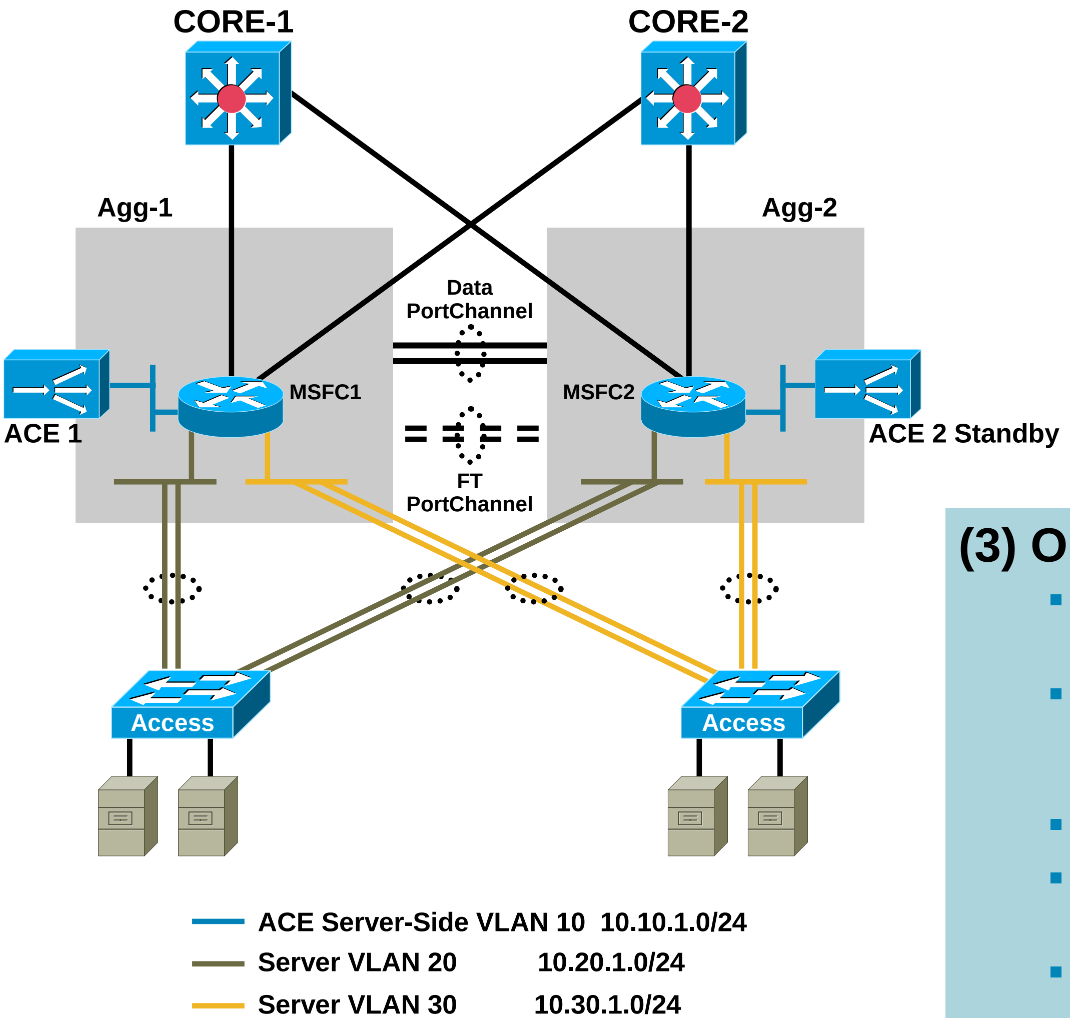
Design Considerations
L3 One-Arm Mode: Overview



- L2-rewrite not possible
- Content switch not inline
  - Does not see unnecessary traffic
- Requires PBR, server default gateway pointing to load balancer or client source NAT
  - The return traffic is needed!
- Not as common as bridge or routed mode due to problems with forcing traffic back to ACE in return direction

PBR—Policy Based Routing, NAT—Network Address Translation

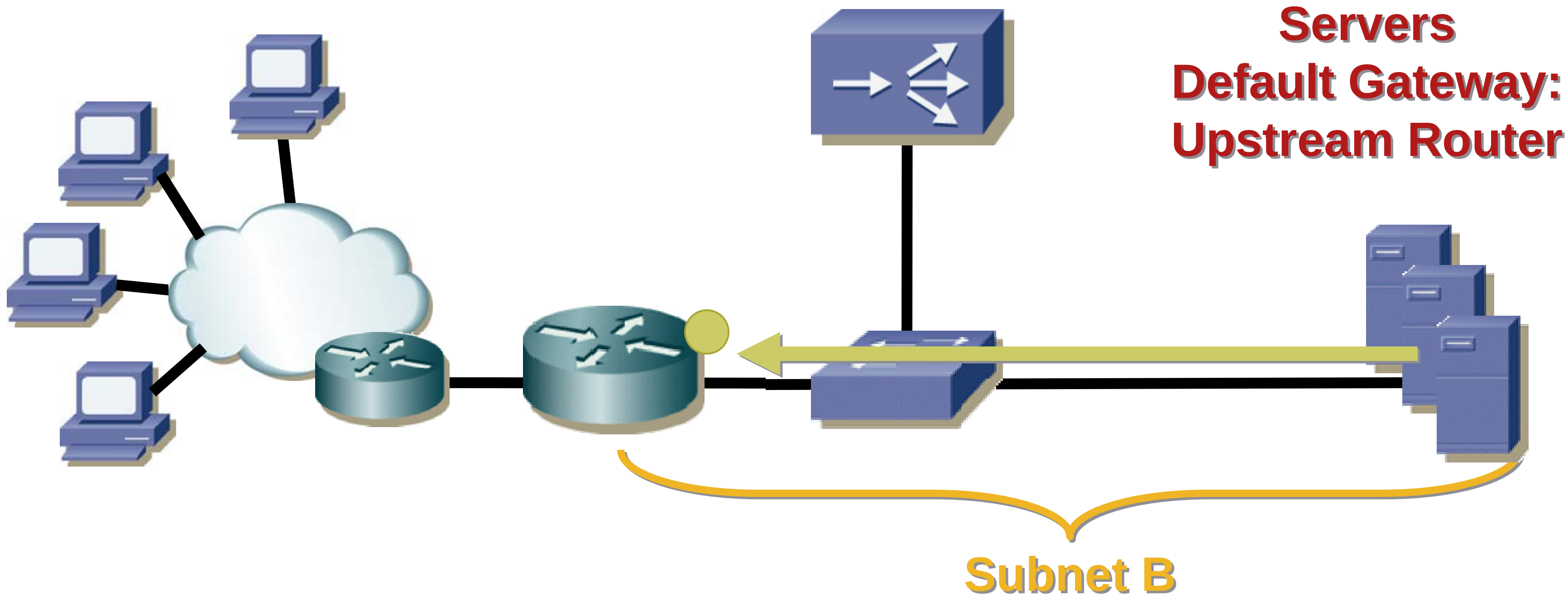
Content Switching Design Approaches
L3 One-Armed Mode: Design



(3) One-Armed Design Considerations

- Servers default gateway is the HSRP group IP address on the MSFC
- No extra configurations for:
  - Direct access to servers
  - Server initiated sessions
- RHI possible
- CSM/ACE inline for only server load balanced traffic
- Policy based routing or source NAT can be used for server return traffic redirection to the load balancer

L2 One-Arm Mode
Return Traffic Bypassing ACE

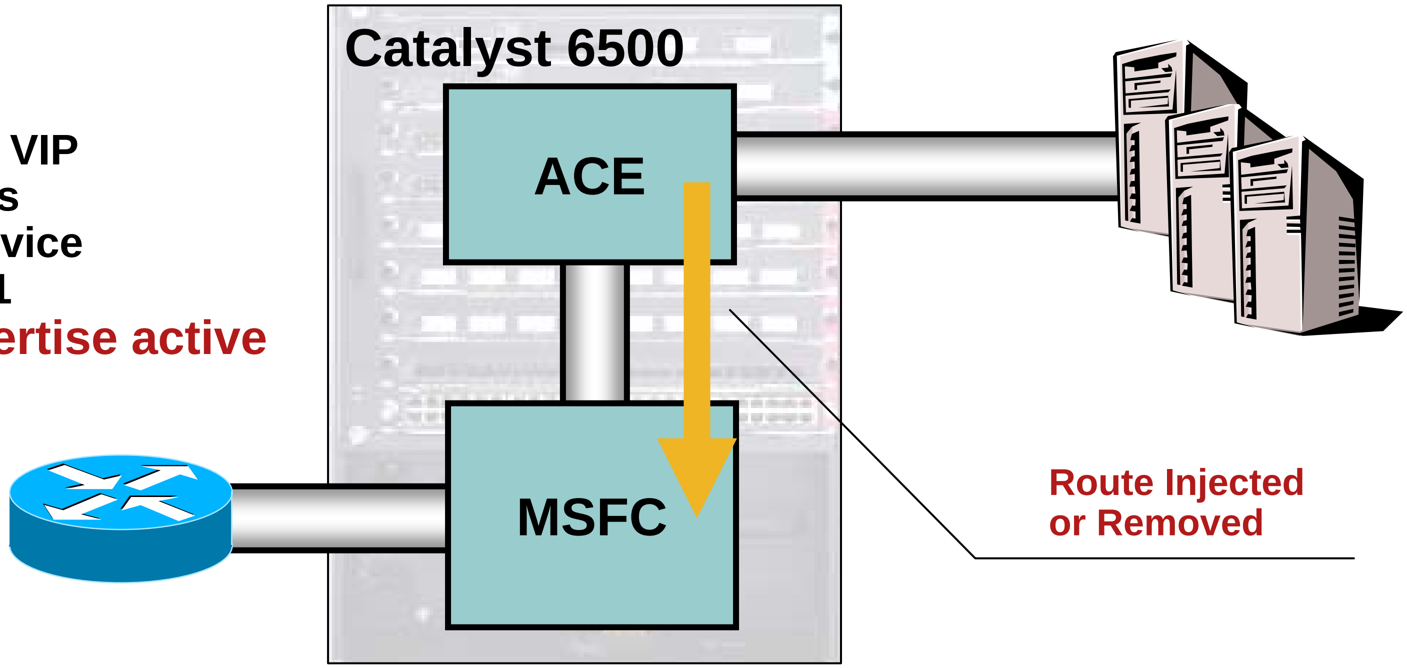


- Bypass for return traffic: high throughput!
- Requires MAC rewrite, L2 adjacency
- Servers need identical loopback addresses (one per VIP)
- TCP termination not possible: no L7 features!
- Load balancer blind to return traffic (inband, accounting)

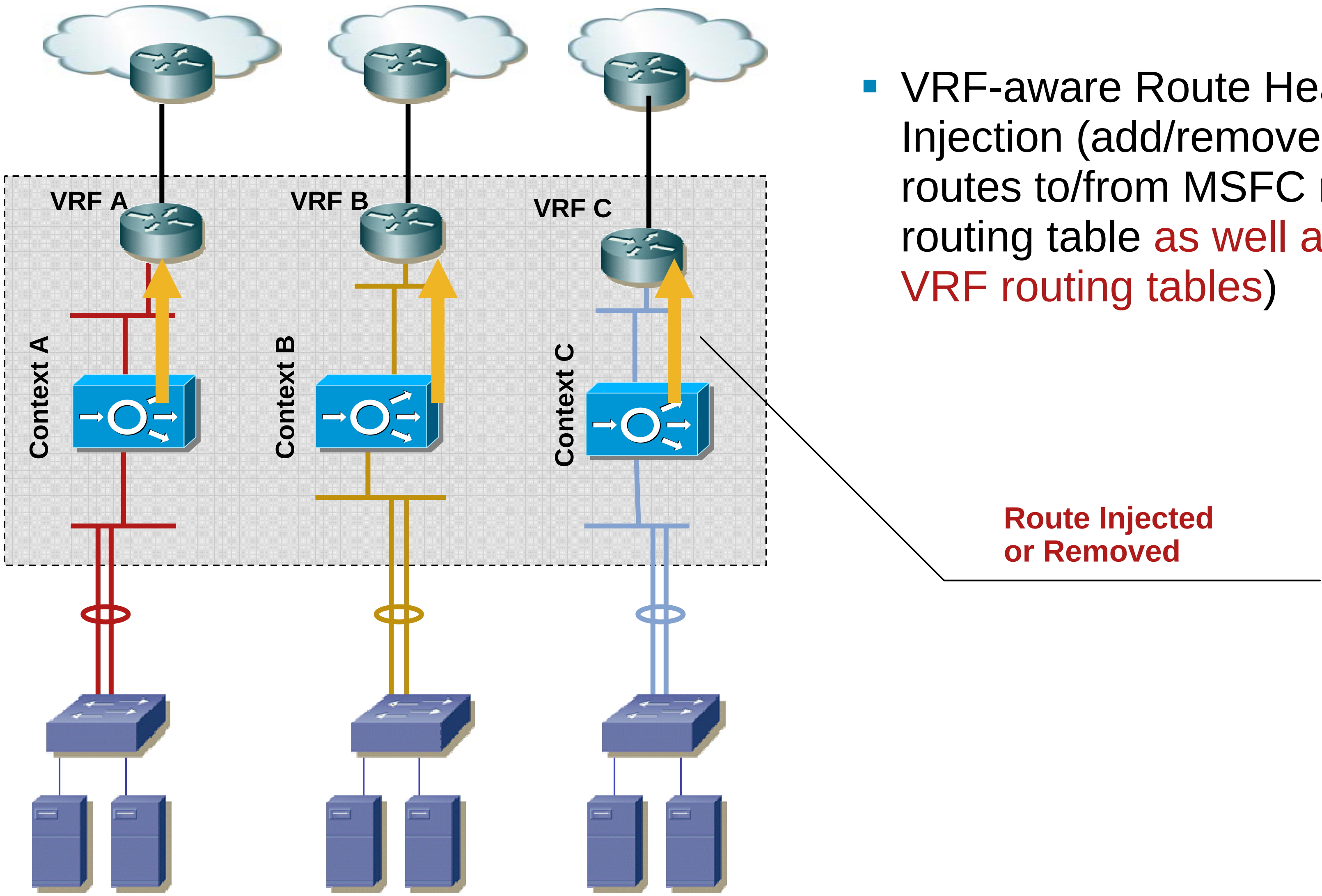
Catalyst 6500 Integration Features
Route Health Injection

- ACE can be configured to “inject” static routes in the MSFC routing table, with configurable metric
- The ACE injects or remove the route based on the health of the back-end servers (checked with L3-7 probes)

policy-map multi-match VIP
class virtual-ip-address
loadbalance vip inservice
loadbalance policy P1
loadbalance vip advertise active



Catalyst 6500 Integration Features
RHI is VRF-aware



- VRF-aware Route Health Injection (add/remove routes to/from MSFC main routing table as well as VRF routing tables)

Application Velocity System

Advanced HTTP-based Application Acceleration …

Web Application Acceleration & Web Firewall Solution

- Optimize at Layer-7
  - 2x-5x – response time improvements
  - 80% decrease in bandwidth requirements
  - 80% fewer server cycles
- Stop application hacking
  - Safely deploy applications
  - Secure mission critical data
  - Streamline operations

Secure, Fast & Reliable Applications

Cisco AVS 3120

Acceleration Features

Functional Areas	AVS Acceleration Features
Latency Reduction	▪ FlashForwarding* ▪ Browser TCP multiplexing* ▪ PDF download optimization ▪ Response redirection control*
Bandwidth Reduction	▪ GZIP Compression ▪ Delta encoding* ▪ Dynamic browser caching* ▪ Dynamic image optimization ▪ Flexible processing rules
Server Offload	▪ TCP Offload ▪ SSL Offload ▪ RAM Caching ▪ Dynamic caching* ▪ Load-based caching* ▪ Lazy request evaluation* ▪ Single sign-on optimizations ▪ XML merging/transformation

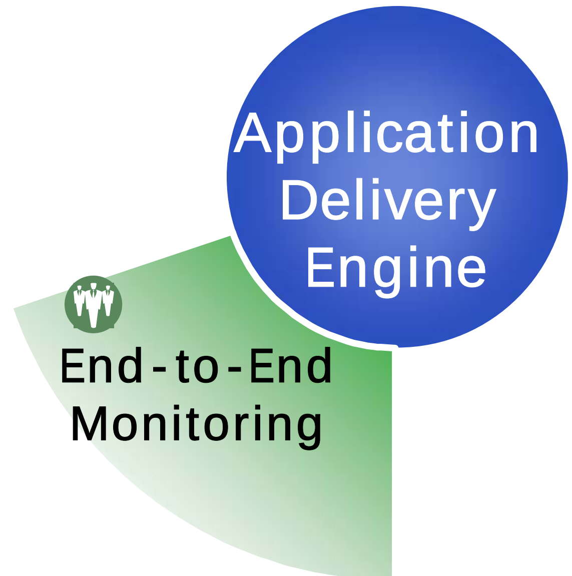
Application Acceleration Examples
Delta Encoding

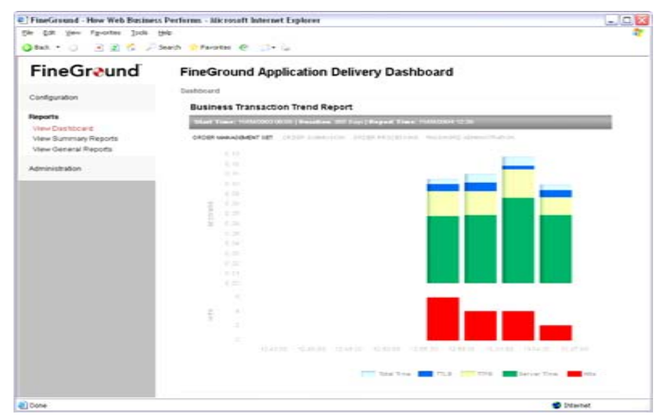
- HTML pages today are largely dynamically generated making it not cacheable
- Browser must download entire page each visit.
- Delta works by calculating and sending only the difference between two visits to a dynamic HTML page
- Benefits:
  - Reduced bandwidth usage
  - Reduced page download times
  - Works in combination with other optimizations

Application Monitoring

- End-user response time monitoring
  - Actual users and transactions
  - Business- and process-level aggregation
  - Full drill-down to page and location
- "Drop-in" deployment
  - No changes to application or desktop
  - Data center installation
- Delivery Dashboard and flexible reporting
  - Wizard-based transaction builder
  - Support for Enterprise Consoles (BMC, Tivoli, OpenView…)
- Benefits

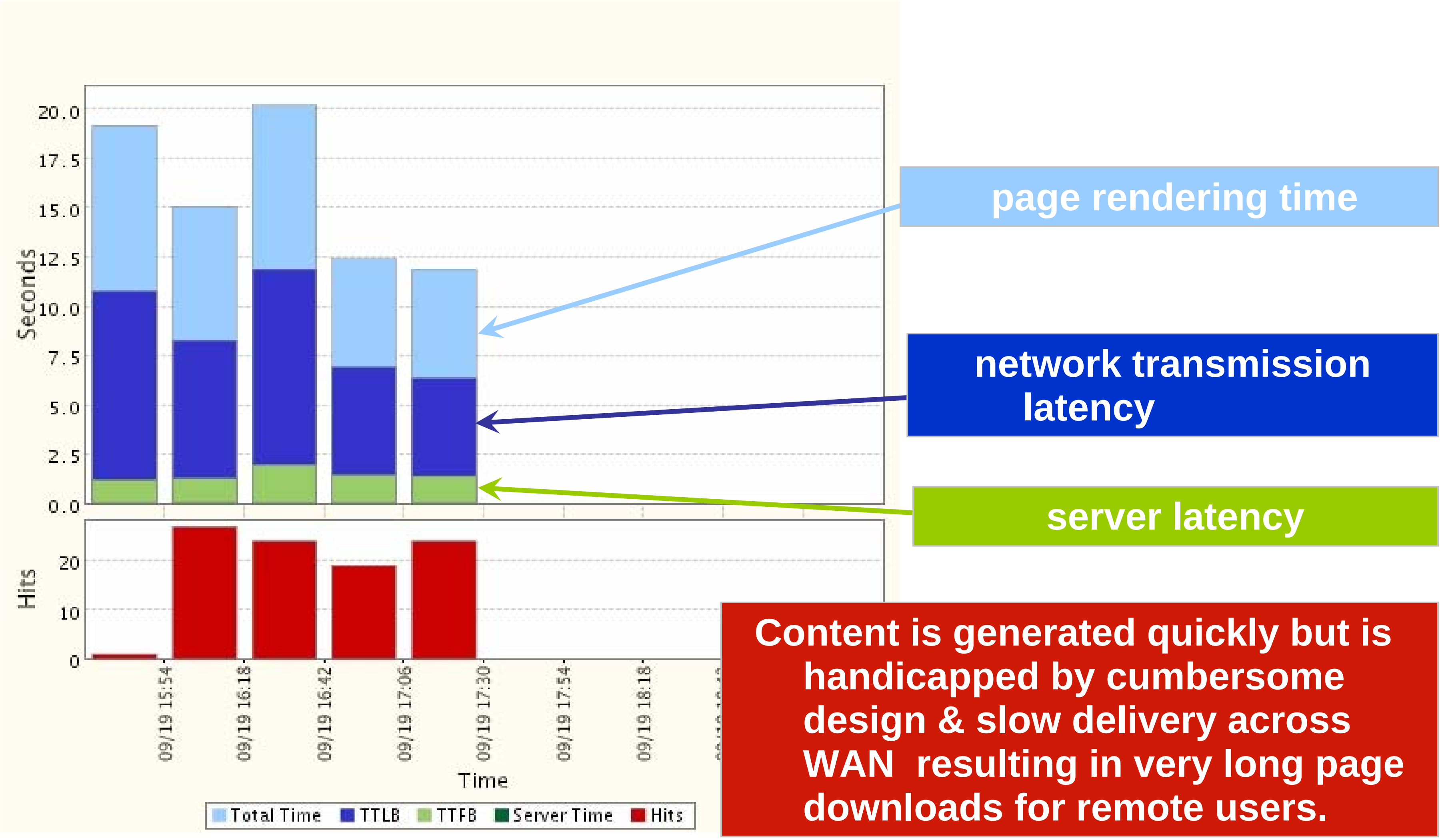
  End-user visibility
  First-line problem triage
  Reduce mean-time-to-repair





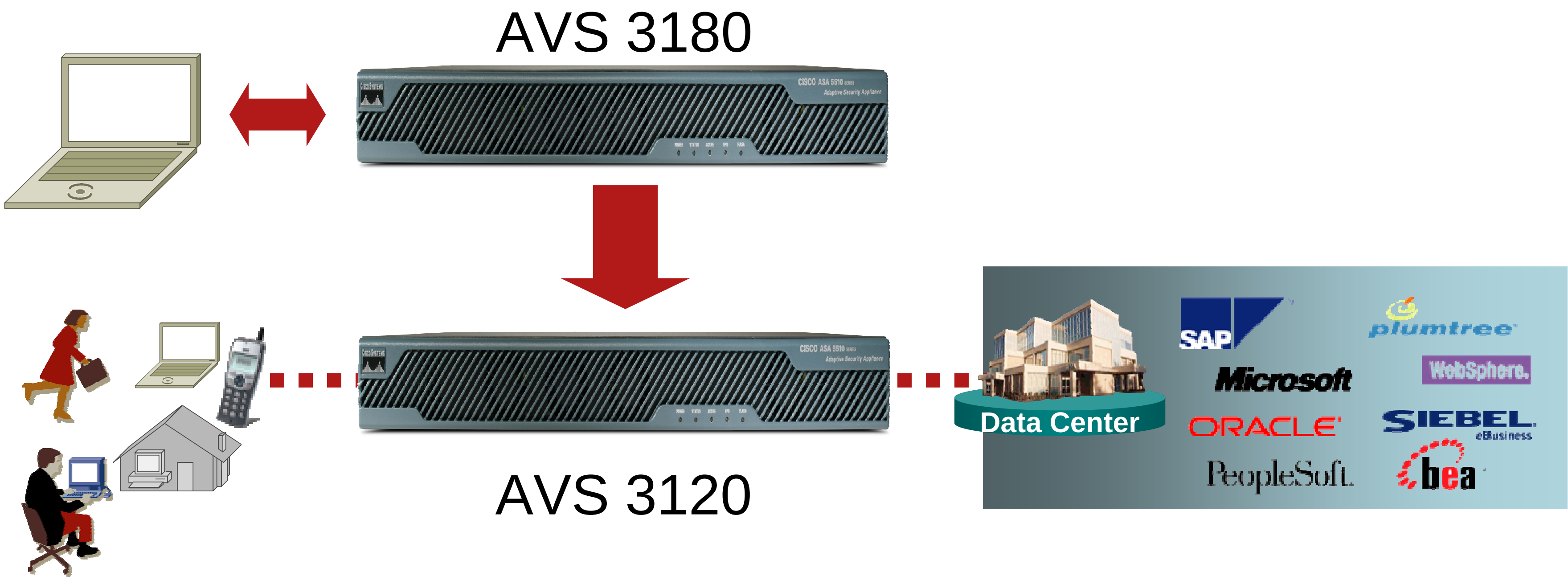
Performance Reporting – No acceleration

- AVS provides insight into application delivery bottlenecks:



Monitoring Requires AVS 3120 & 3180

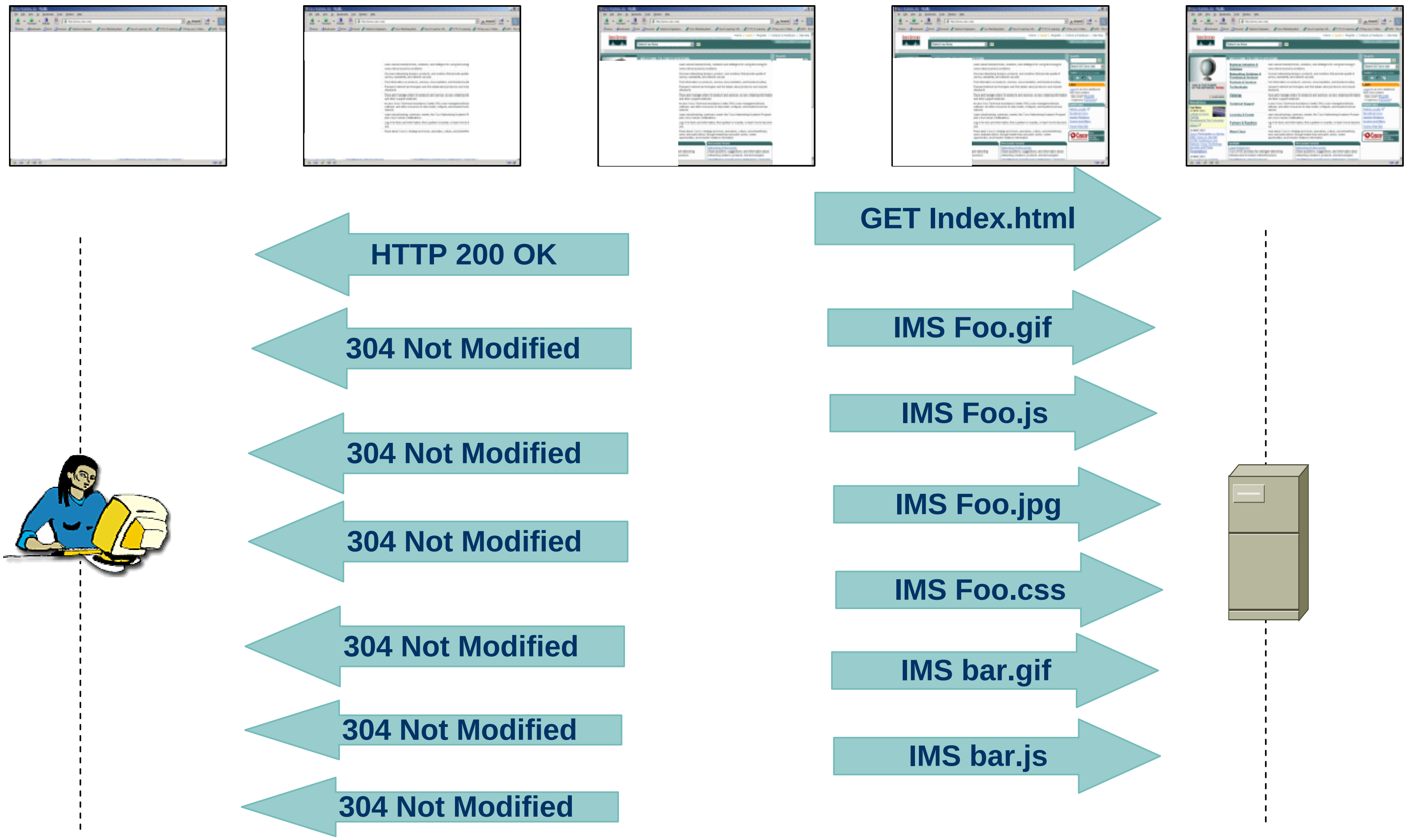
- AVS 3180 Polls the 3120 for Performance Data
- Browser into the 3180
- AVS 3180 also allows you to manage one or more AVS 3120



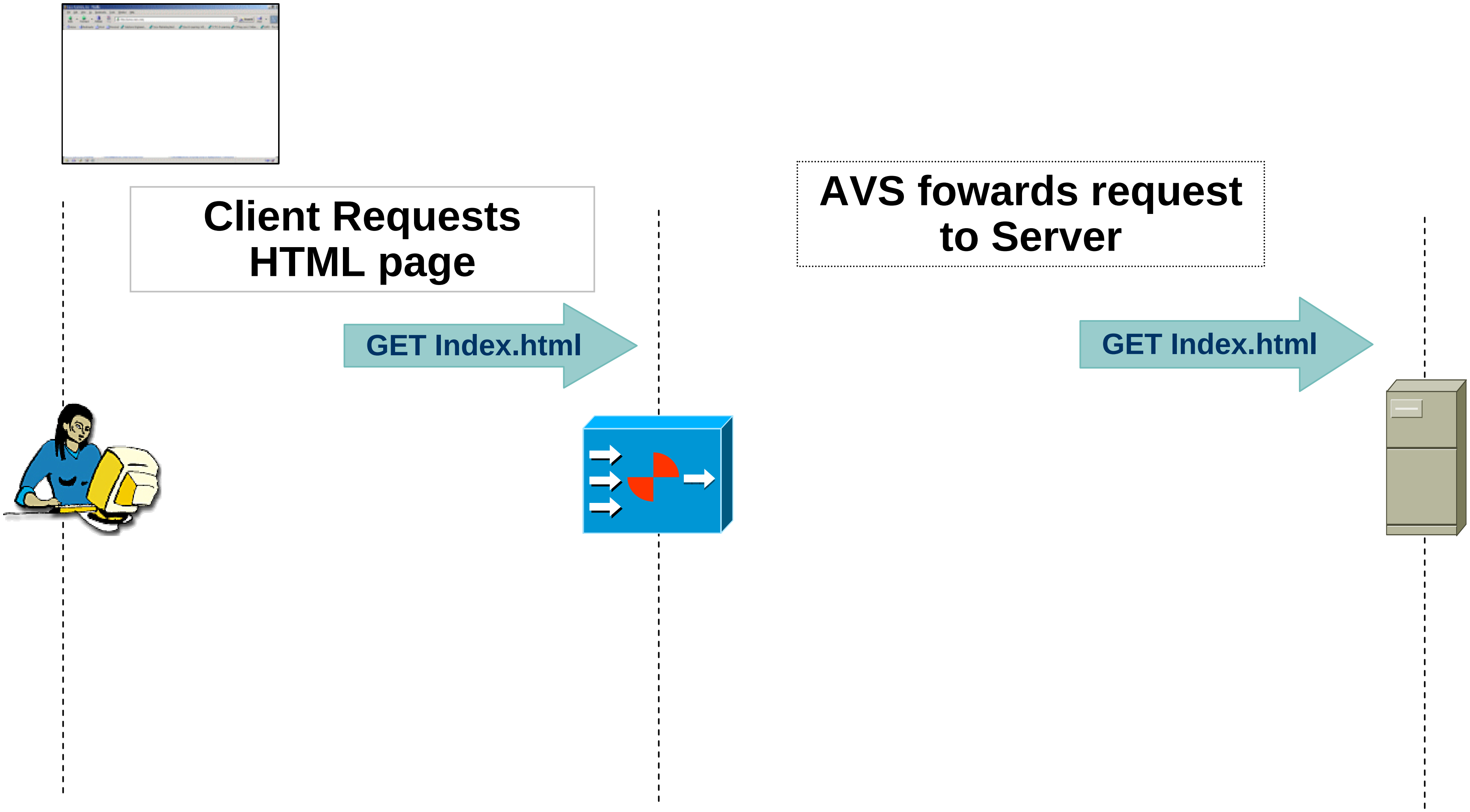
Application Acceleration Examples *FlashForward*

- Embedded objects referenced in HTML container pages are served with Expires: which sets expiry in the future.
- On 2nd visit Browser will not send GET for objects in cache if the current date & time is not greater than the object expiry date.
- This reduces the total number of HTTP requests for subsequent visits to the same page.
- Benefits:
  - Decreased page download time
  - Decreased network congestion
  - Decreased number of requests to origin server

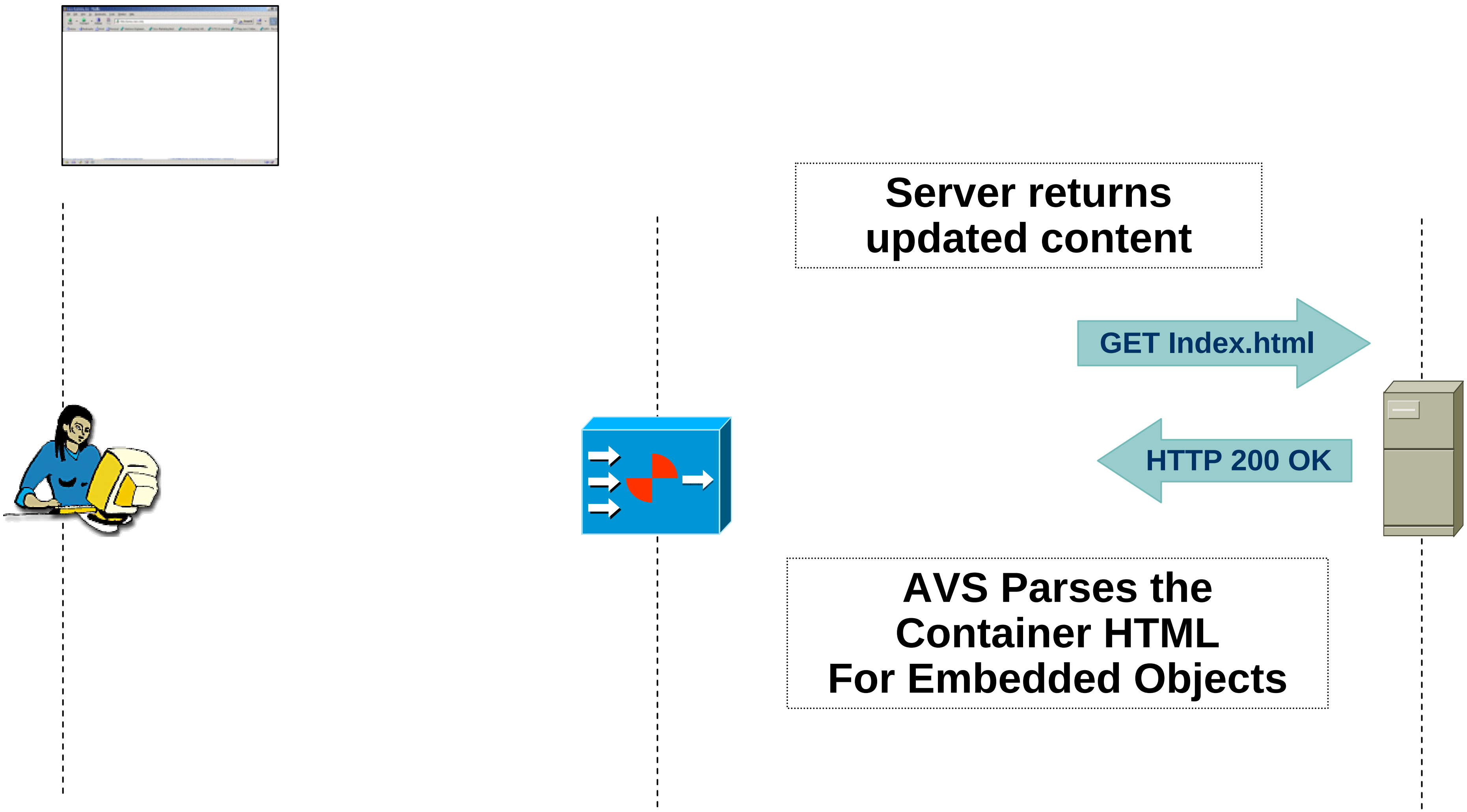
Browser Behavior Without FlashForward



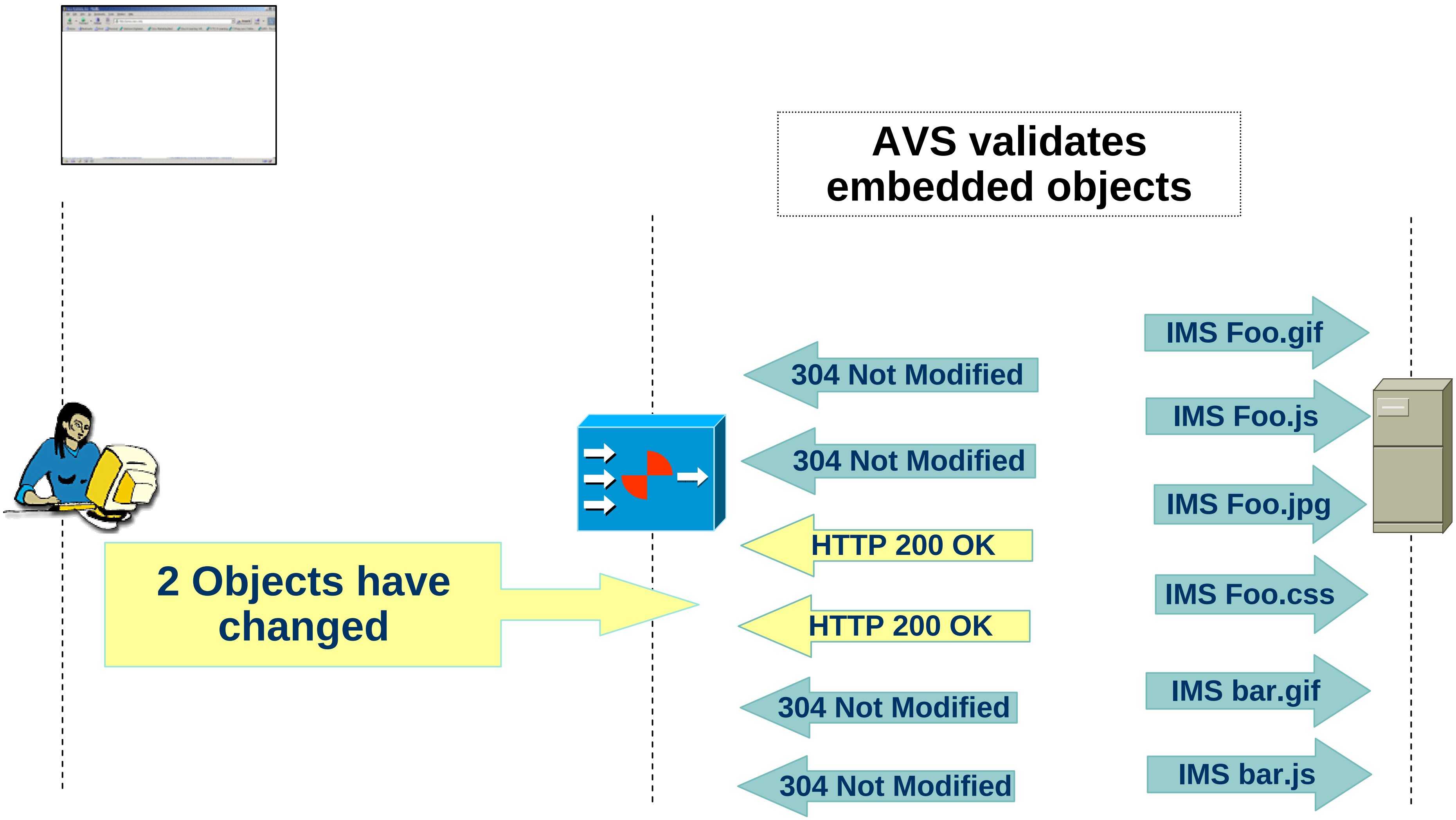
Browser Behavior With FlashForward



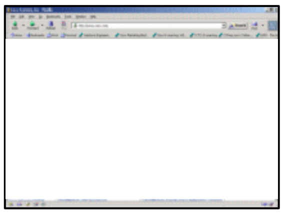
Browser Behavior With FlashForward

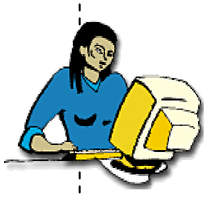


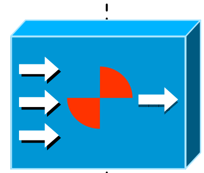
Browser Behavior With FlashForward



Browser Behavior With FlashForward







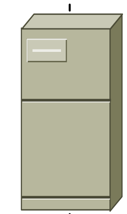
AVS Updates the References to Embedded Objects In HTML

Object Reference:

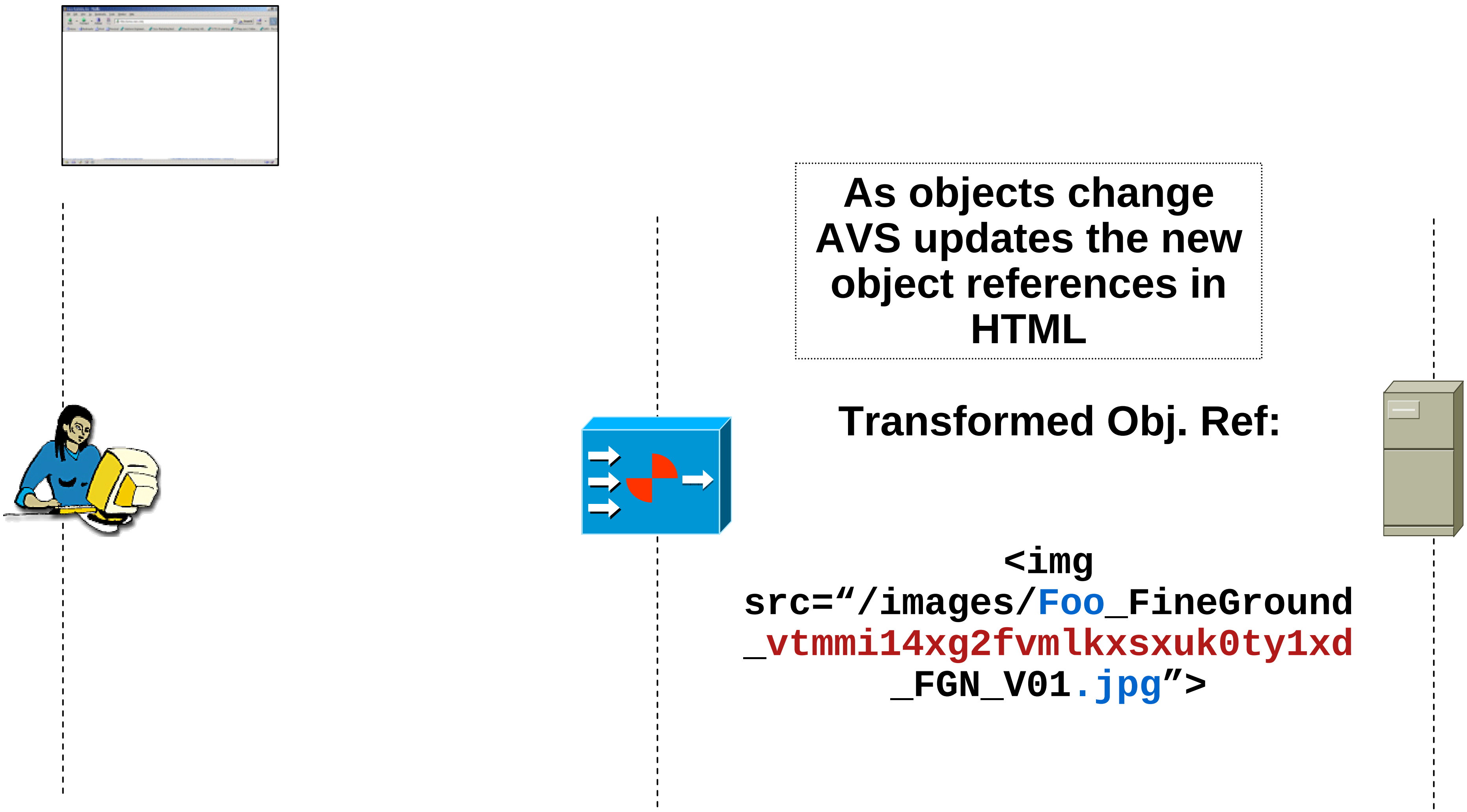
<img src="/images/Foo.jpg">

Transformed Obj. Ref:

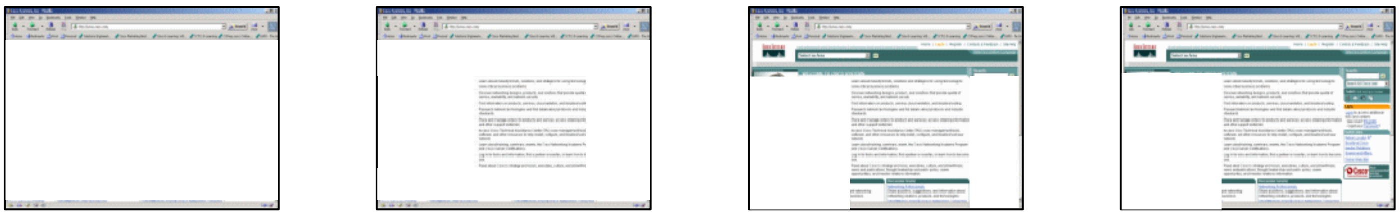
<img src="/images/Foo_FineGround_vtmmi14xg2fvmlkxsxuk0ty1xd_FGN_V01.jpg">



Browser Behavior With FlashForward

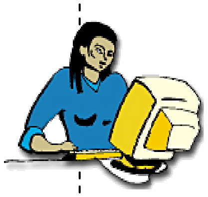


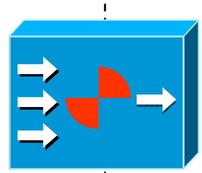
Browser Behavior With FlashForward

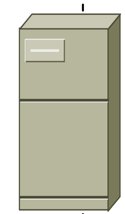


Client downloads HTML page

The requested HTML with updated object references is now sent to the client



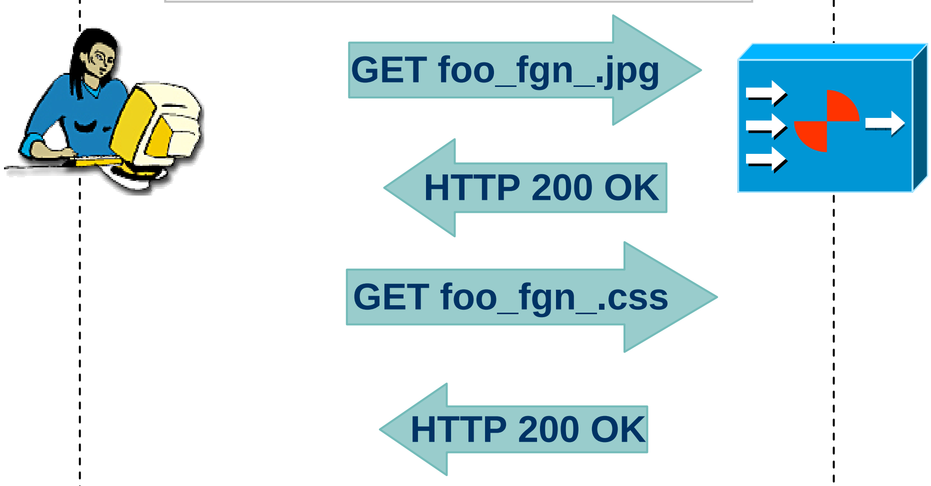




Client parses HTML and retrieves fresh embedded objects from browser cache

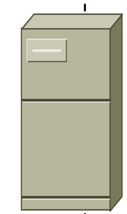
Browser Behavior With FlashForward

Client downloads new updated objects referenced in HTML Not found in cache.



Only 3 round trips across the WAN were required to build the updated page.

Round trip latency is avoided with each Flash Forwarded object cached in the browser's cache.



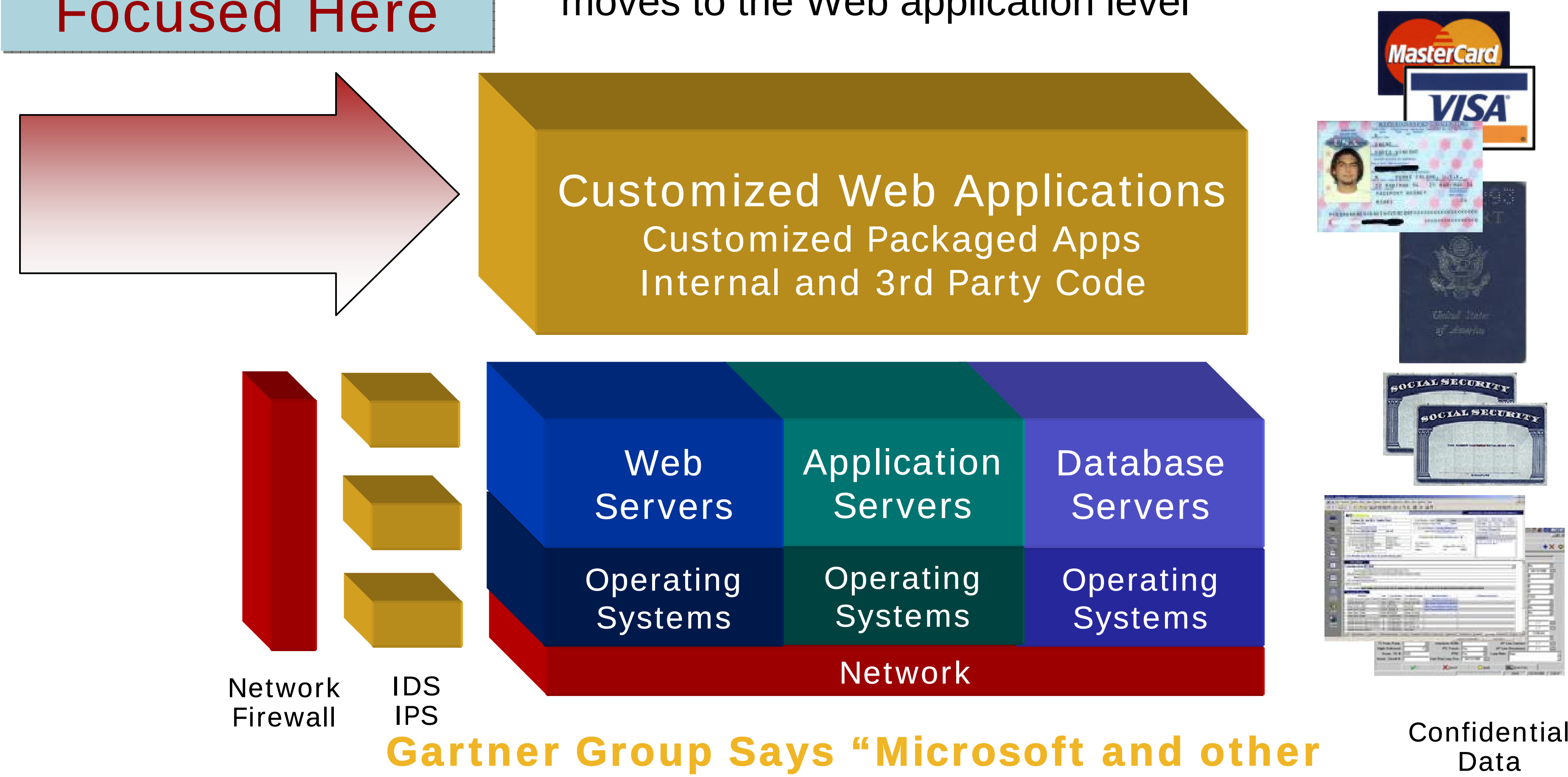
Application Velocity System

… and Web Application Security

No Patches or Signatures to Protect Custom Code

75% of Attacks Focused Here

The focus of security efforts now moves to the Web application level



Gartner Group Says “Microsoft and other vendors are building more secure platforms”

AVS 3120: Deployment Options from SW Version 6.0

Out of Band Monitoring Mode

Risk Free Deployment
Attack Visibility
Per Application Monitoring

Inline Attack Blocking Mode

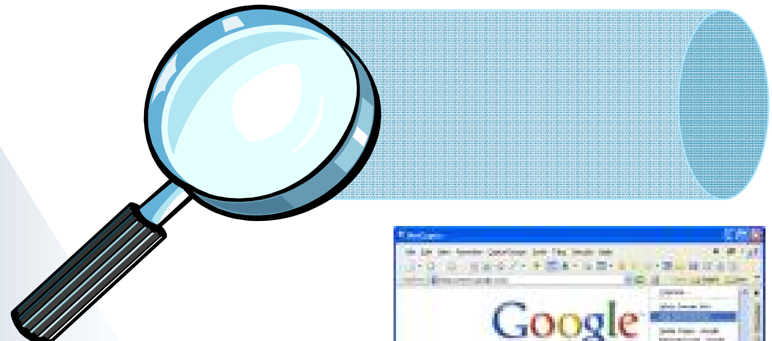
Transparent Bridge Deployment
App Security only
No Changes to Network Infrastructure
Block, Log, Rewrite, Redirect, Generate Error pages, URL tagging, Apply Per Rule Passive - Active

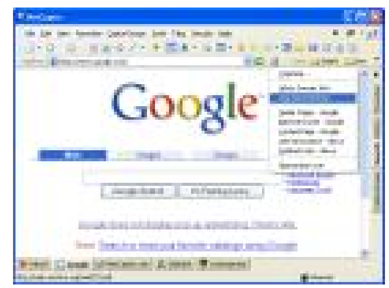
Gateway Mode

All Application Security features from inline mode, AVS Web Acceleration, Latency reduction, Clustering/Flow based load balancing, SSL termination

AVS Delivers Applications Securely

INSPECTS FOR:
SQL Injection
Cross-Site Scripting
Command Injection
Cookie/Session Poisoning
Application Reconnaissance
LDAP Injection
Buffer Overflows
Directory Traversals
Attack Obfuscation
Application Platform Exploits
Zero Day Attacks
Parameter Tampering
Data-theft

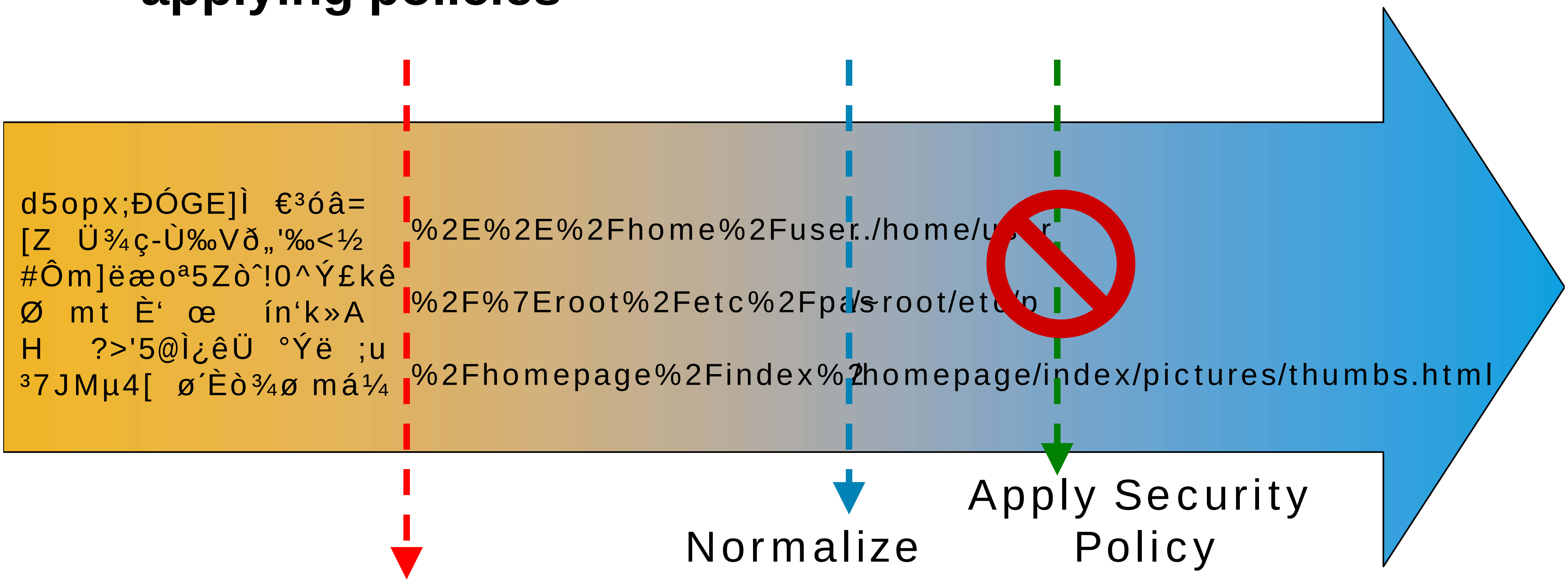




- Bi – Directional Deep Inspection and Rewrite capabilities
- Positive & Negative Security
- Protocol compliance and anomaly detection
- Transaction logging and report for application security forensics

AVS 6.0 Foundation – Full visibility

Normalization of all traffic to a canonical form before applying policies



Terminate and decrypt SSL

Stops attacks disguised by encrypting and encoding

Dynamic Form Learning

Enforces security policy *as defined by the application.*
Learns the following elements from responses

- **Hidden fields:**
  INPUT TYPE=hidden NAME=hidden1 VALUE=3
- **Maximum length of form fields:**
  INPUT TYPE="TEXT" MAXLENGTH="30"
- **Bounded value fields like radio buttons, lists:**
  INPUT TYPE="radio" value="01"
- **Query Strings :**
  href = “/query_refs.cgi?id=123&Uid=abc”

Cookie Encryption
Example : Server to Client

Clients

Web Server

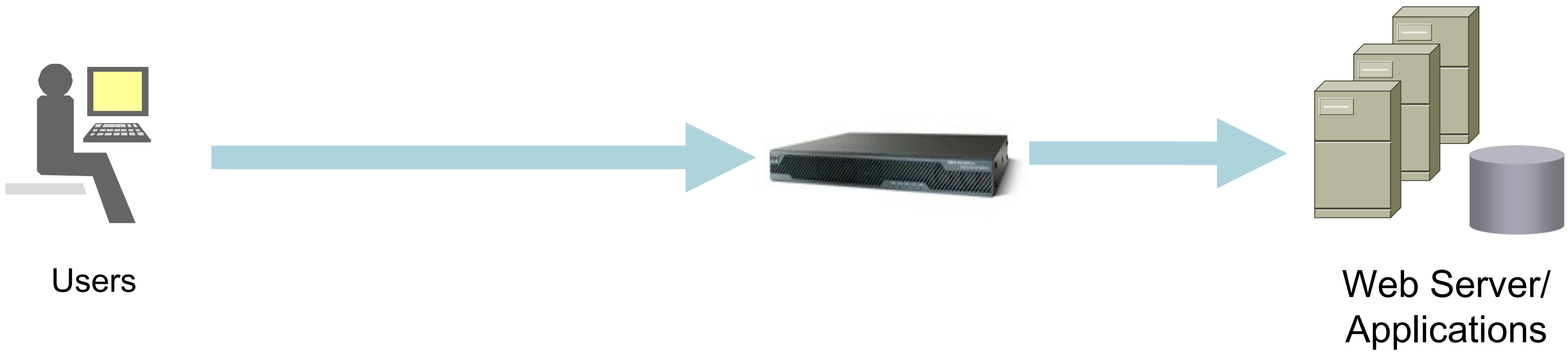
CP_EN-
e7a989b1f1b9e966e47d629eec63302d3571d1677b27fe1bebb
a48df648b2edc=
1-
0c49cd6655b1ffd32746970b5f21876c2c700e088b923d38d506
fea0e7c15d7a;
expires=Mon, 15-Dec-2006 1:03:00 GMT; path=/;
domain=.google.com; secure

Cookie after processing by AVS for encryption

sess1=1800;
expires=Mon, 15-Dec-2006
1:03:00 GMT;
path=/;
domain=.google.com;
secure

Cookie from Server

Cookie Encryption
Example : Client to Server



CP_EN-
e7a989b1f1b9e966e47d629eec63302d3571d16
77b27fe1bebba48df648b2edc=
1-
0c49cd6655b1ffd32746970b5f21876c2c700e08
8b923d38d506fea0e7c15d7a

Cookie from Client

sess1=1800;

Cookie to Server after being processd and decrypted by AVS

Web Cloaking

Users requesting Web pages

Web Applications

PROBLEM

Responses from the server could reveal software, their version and even topology

HTTP/1.1 200 OK
Date: Mon, 07 Jun 2004 14:31:03 GMT
Server: Apache/1.3.29 (Unix)
mod_perl/1.29
Connection: close

RFC too warns against revealing server identity

The AVS can rewrite or hide specific headers in the reply

Data Theft Protection

Credit Card
1234-5678-9012-3456

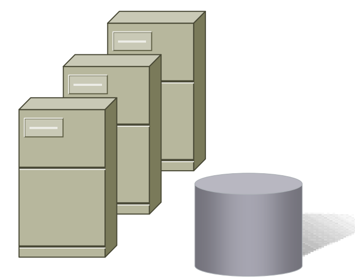
Social Security
123-45-6789

Driver's License
A123456

Employee ID
S-924600

Patient ID
134-AR-627

Users



Web
Applications

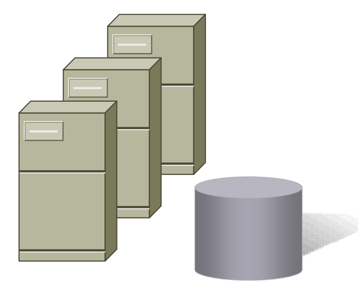
PROBLEM

Any web app that links to critical data
may expose that data to hackers

Data Theft Protection

MASK		Credit Card XXXX-XXXX-XXXX-3456
MASK		Social Security XXX-XX-XXXX
BLOCK		Driver's License A123456
MASK		Employee ID XXXX
BLOCK		Patient ID 134-AR-627

Users



Web Applications

AVS 3120

Helps ensure compliance

Wide Area Application Services

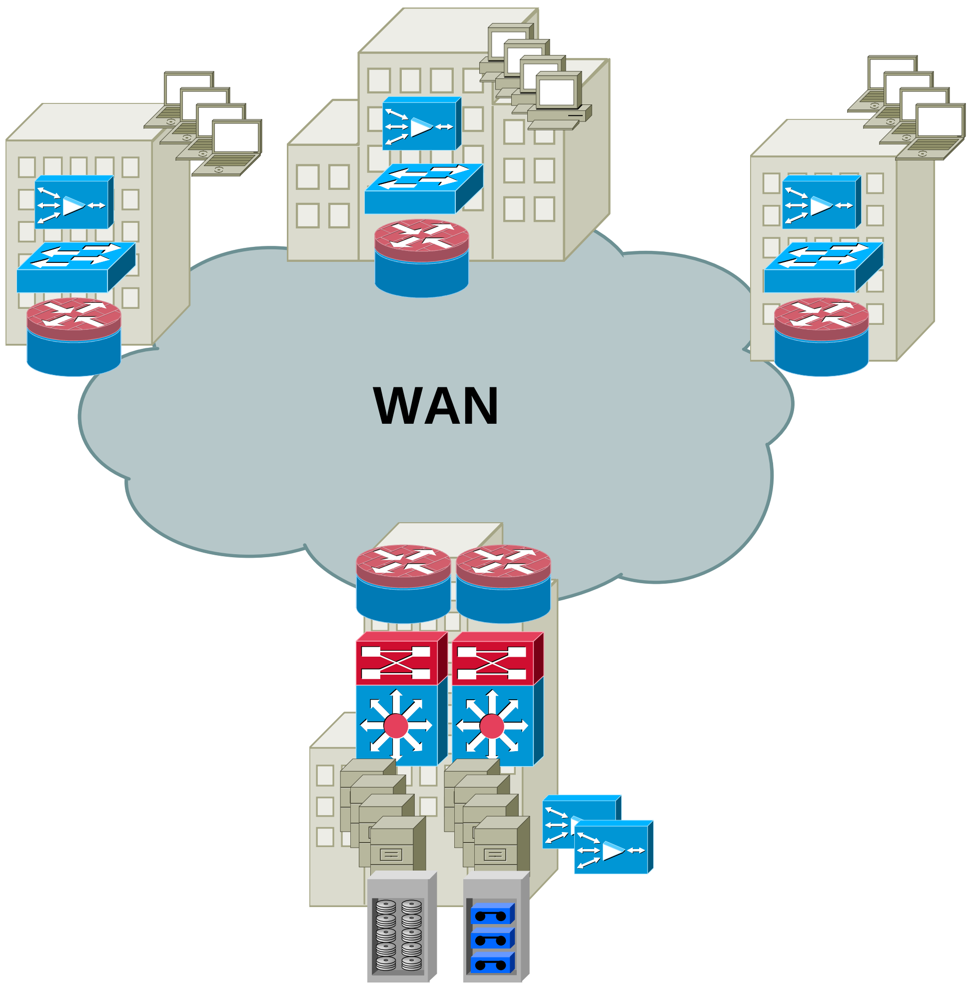
Overcoming WAN Limitations

Application Acceleration Overcomes the WAN

Source	Need	Technology
Latency	▪ Reduced number of network roundtrips from chatty application protocols	▪ Intelligent protocol proxies
Bandwidth Utilization	▪ Improve application response time on congested links by reducing the amount of data sent across the WAN	▪ Application caching ▪ Compression
Transport Throughput	▪ Improve network throughput (total # of data) by improving transport behavior	▪ TCP optimizations ▪ Adaptive congestion mgmt
Traffic Differentiation	▪ Identify application flows on the network to prioritize business-critical or latency-sensitive applications	▪ Quality of service, NBAR ▪ IP SLA, NetFlow
Administrative Traffic	▪ Replacement for services that branch office servers provide	▪ Centrally managed remote services interface

Cisco WAAS Enables Consolidation

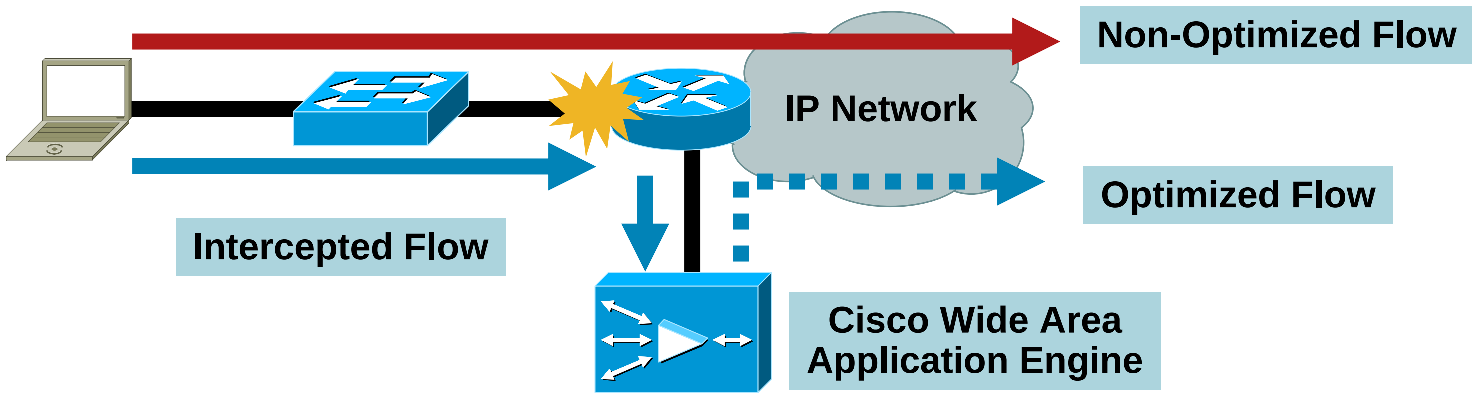
- Cisco Wide Area Application Services (WAAS)
  - Transparent integration
  - Robust optimizations
  - Auto discovery
- Infrastructure consolidation
  - Remote costly servers
  - Centralize data protection
  - Save WAN resources
- Application acceleration
  - Application adapters
  - Advanced compression
  - Throughput optimizations
  - Policy-based configuration



Network Interception

Network Attached Optimizations Rely on Devices Physically Attached to the Network at Strategic Locations

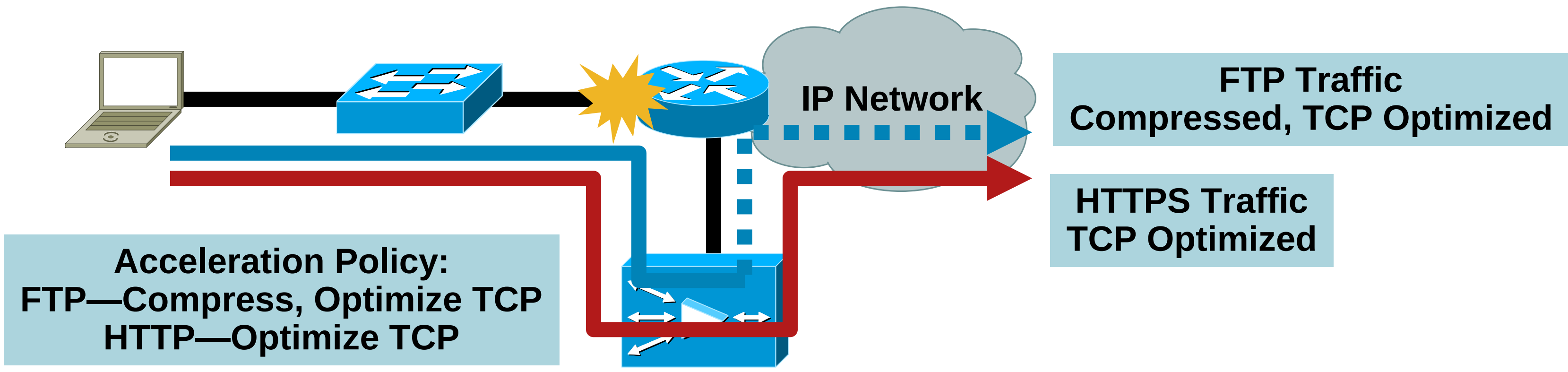
- Generally deployed at network entry/exit points
- Rely on network interception to supply flows to optimize



Flexible Acceleration Policies

Application Acceleration Must Provide Users with Flexible Configuration of Optimizations—Not All Flows Are Created Equal

- Low layer implementation to ensure high performance
- Default policies provided but able to be modified

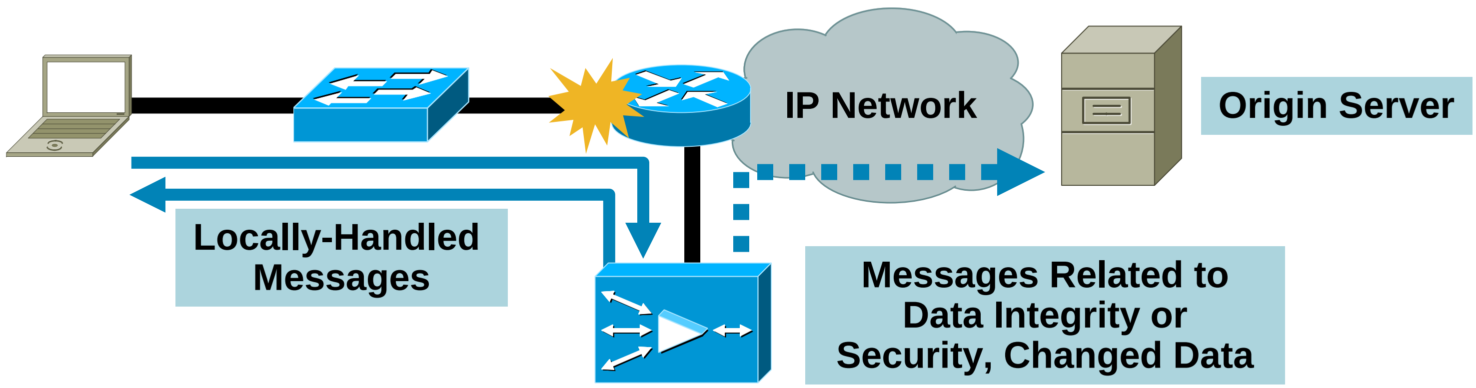


Application Latency

- Application latency is defined as the amount of response time increase caused by the exchange of application-layer message
- Applications can be considered “chatty” when their protocols require the exchange of many messages
- Common examples of chatty applications include
  - Common Internet File System (CIFS) file sharing
  - Transactional applications using Hypertext Transport Protocol (HTTP)

Mitigating Application-Layer Latency

- An application proxy-cache is defined as a trusted entity that can safely handle operations on behalf of another
- Application proxy-caching allows an intermediary device (local to the user) to handle some workload (where safe) as if it were the origin server

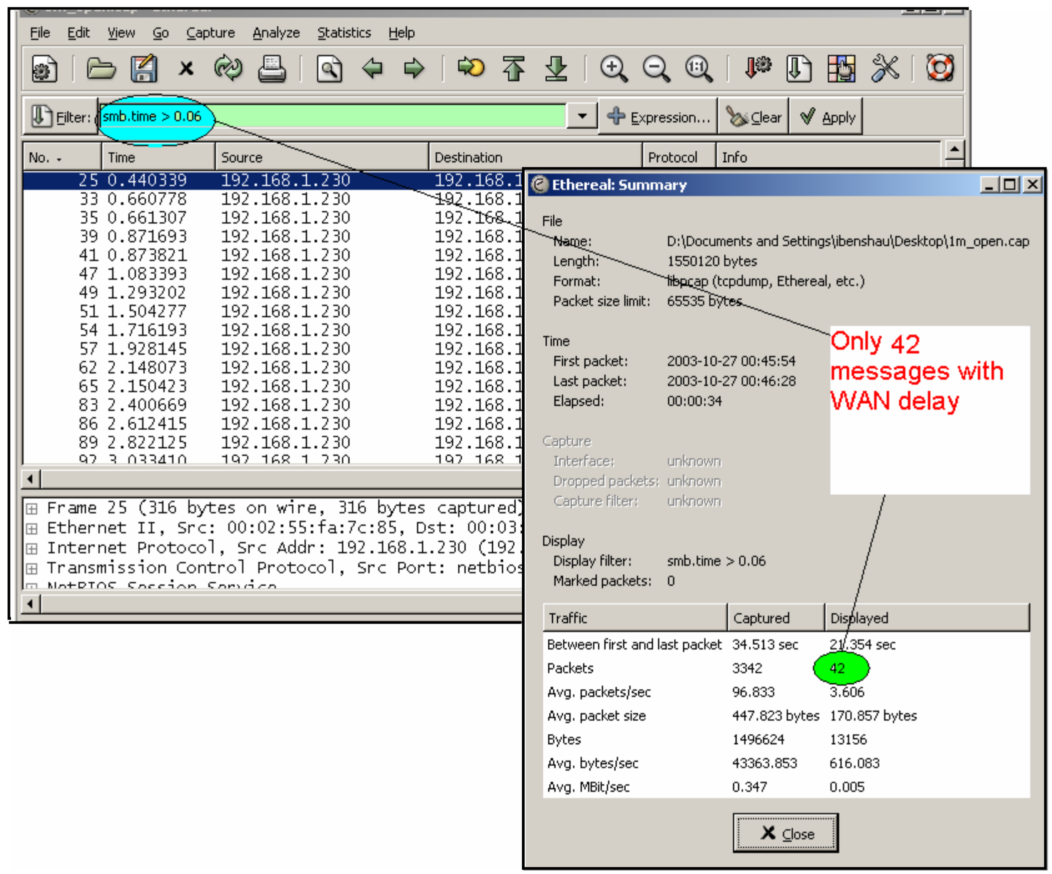


Mitigating Application-Layer Latency

- Proxy-caches perform the following functions
  - Handling of non-critical messages locally
  - Suppress (where safe) unnecessary protocol messaging
  - Forward (synchronously) messages that relate to integrity
  - Delivery of cached, validated data locally (no stale data)
- Which provide the following results
  - Minimize message exchange over the WAN
  - Eliminate redundant data transfer over the WAN
  - Ensure that only accurate, valid data is served
  - Safely offload origin servers and mitigate WAN upgrades

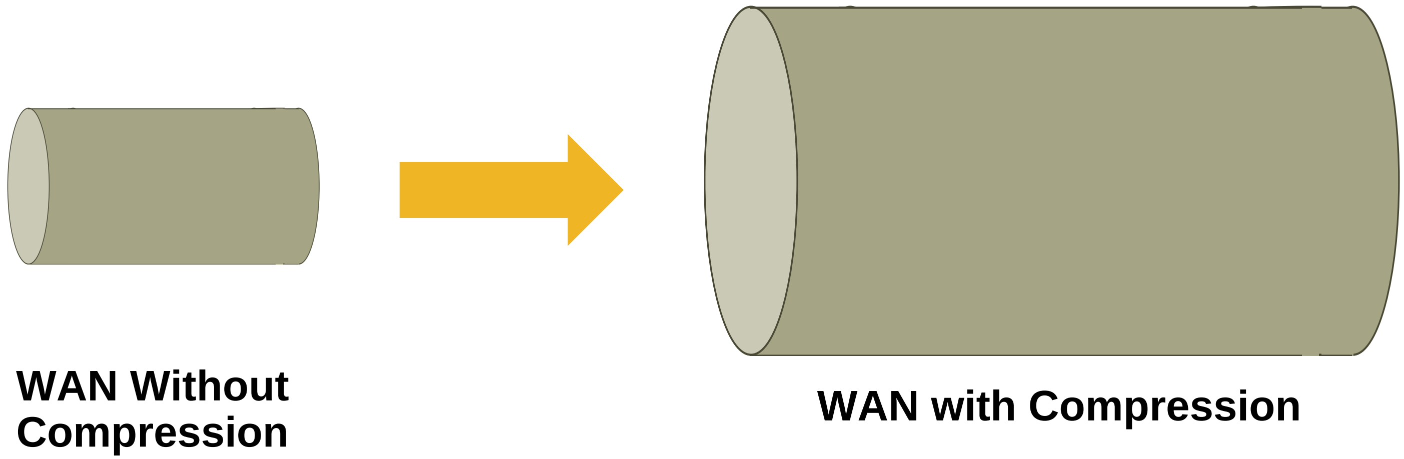
Impact of Application Proxy-Caching

- Application proxy-caching eliminates the majority of messaging from the WAN
- Safely responds to or otherwise handles application message exchanges
- Synchronously passes messages critical to user authenticity, data integrity, and collaboration
  - User authentication
  - User authorization
  - File and record locking
  - File validation
  - Changed data



The Need for Compression

- Advanced compression technologies allow customers to virtually increase WAN capacity
- Allows customers to leverage existing WAN capacity and may mitigate the need for a costly bandwidth upgrade



The Need for Compression

- Some data sets are not good candidates for compression unless adaptation is first performed

  Previously-compressed data—no additional compression provided by computational compression, good candidate for data suppression

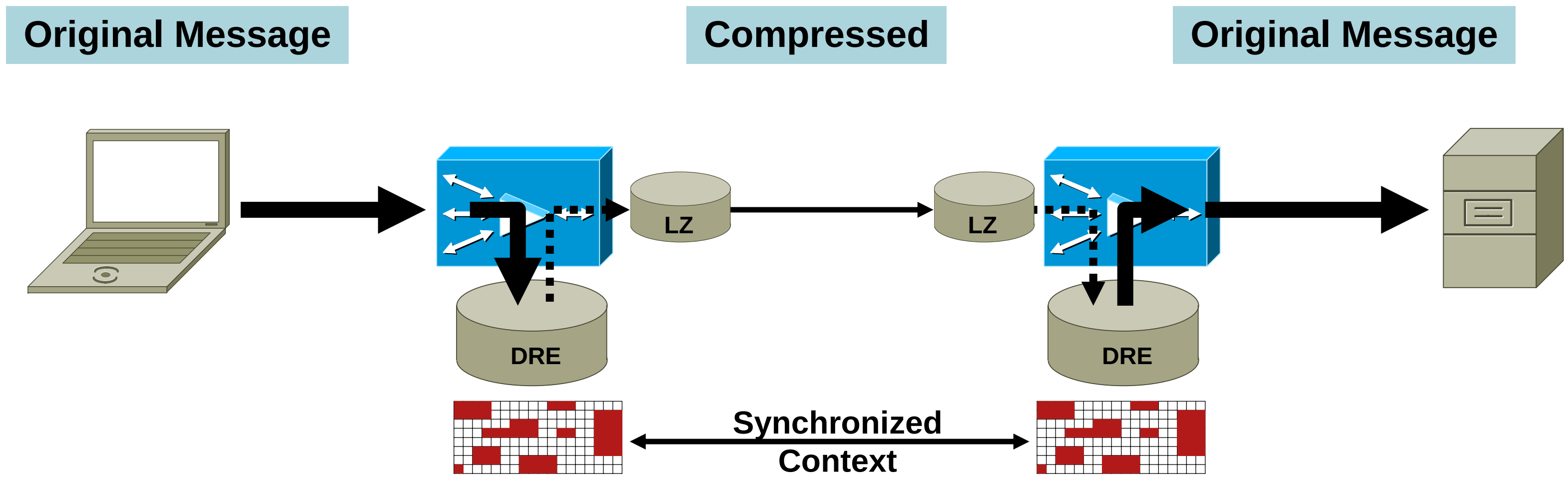
  Previously-encrypted data—minimal additional compression provided by computation compression,
  good candidate for data suppression if not using session-based encryption (i.e., non-repeatable data)

- Such adaptation could include local termination of encryption, apply compression, then re-encrypt

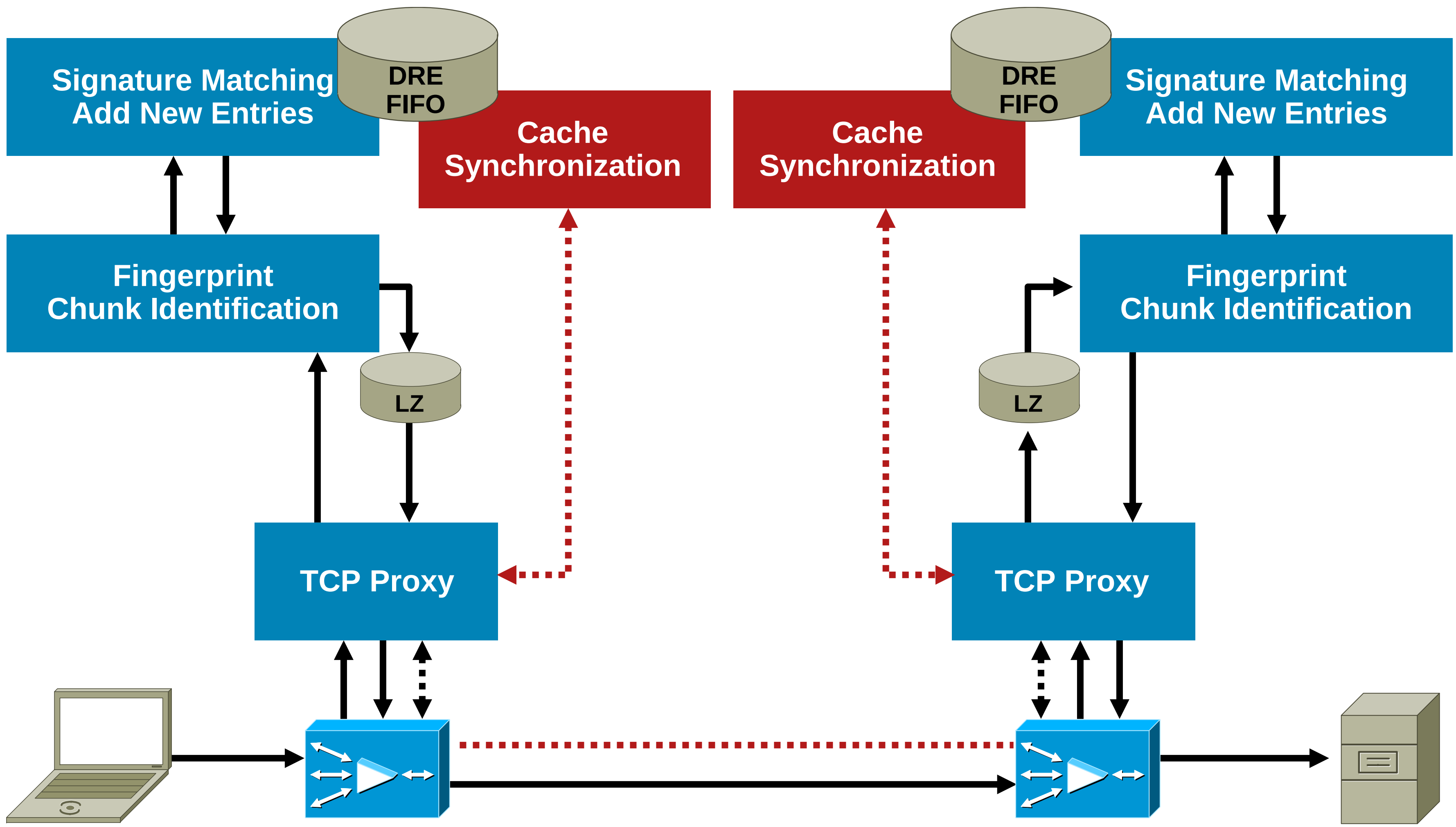
Advanced Compression Overview

Two Forms of Compression (Together) Enable Significant Savings of WAN Bandwidth

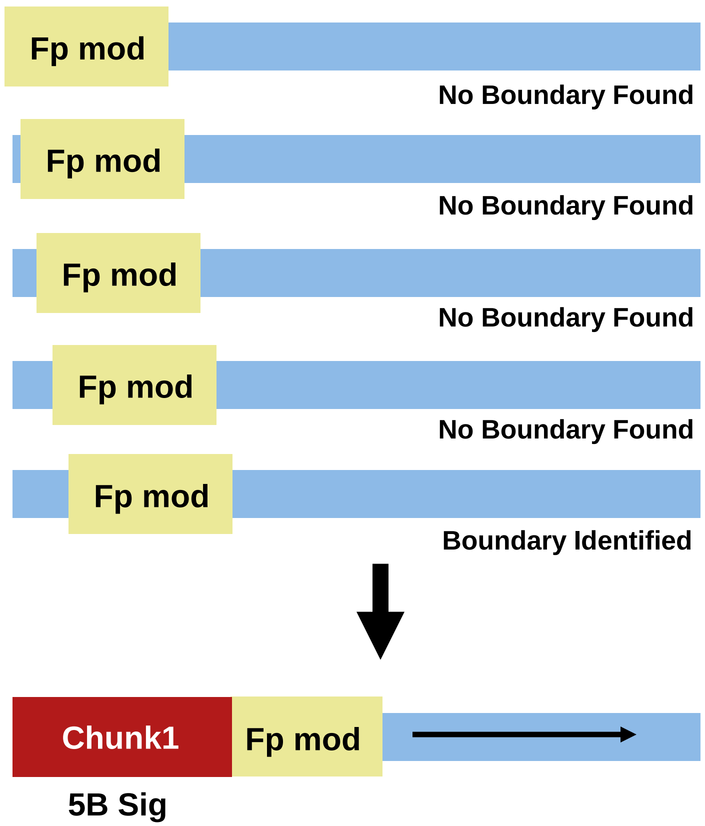
- Data suppression (DRE): store chunks of TCP traffic patterns in loosely-synchronized contexts to suppress transmission of redundant chunks
- Standards-based compression: i.e., Lempel-Ziv, deflate



Advanced Compression Block Diagram



DRE Encoding—Chunk ID

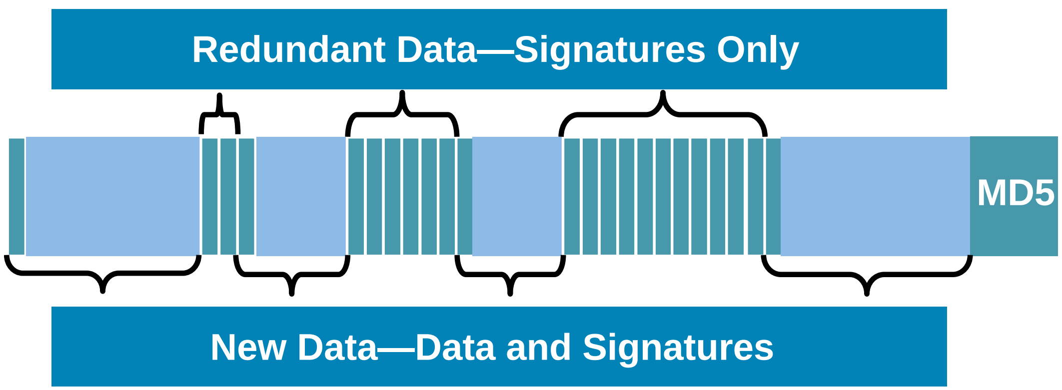


- DRE analyzes incoming data streams using a sliding window to identify “chunks”
- Each chunk assigned a 6-byte signature
- Single-pass used to identify chunks at multiple levels
  - Basic chunks
  - Chunk aggregation (nesting)
- After chunks are identified, DRE will begin pattern matching
  - First look for largest chunks
  - Look for smaller chunks if necessary

DRE Encoding—Resultant Message

DRE Sender, Cont.

- A fully encoded message will contain:
  - Signatures only for previously-seen patterns
  - Signatures, data for non-redundant patterns (update adjacent WAE)
  - 16-byte MD5 hash of original message to verify integrity after rebuild
- Message is passed to LZ compression (based on policy) and to TCP proxy to return to the network



DRE Decoder

DRE Decoder

- Uncompress LZ packet and read data (if negotiated)
- Examine message to identify signatures and chunks
    - For signatures sent alone, replace with chunk from context
    - For signatures sent with accompanying data, update DRE context, remove the signature
    - ACK/NACK used to notify peer of success/failure
- Once fully analyzed and rebuilt, MD5 calculation performed for verification
    - MD5 match: message rebuilt with integrity, send to destination
    - MD5 mismatch: message rebuilt using incorrect data; have encoder resend

DRE Synchronization

- Upon connection establishment, DRE peers will compare FIFO clock information from each other's respective databases
- This includes "head" and "tail" of database timestamps

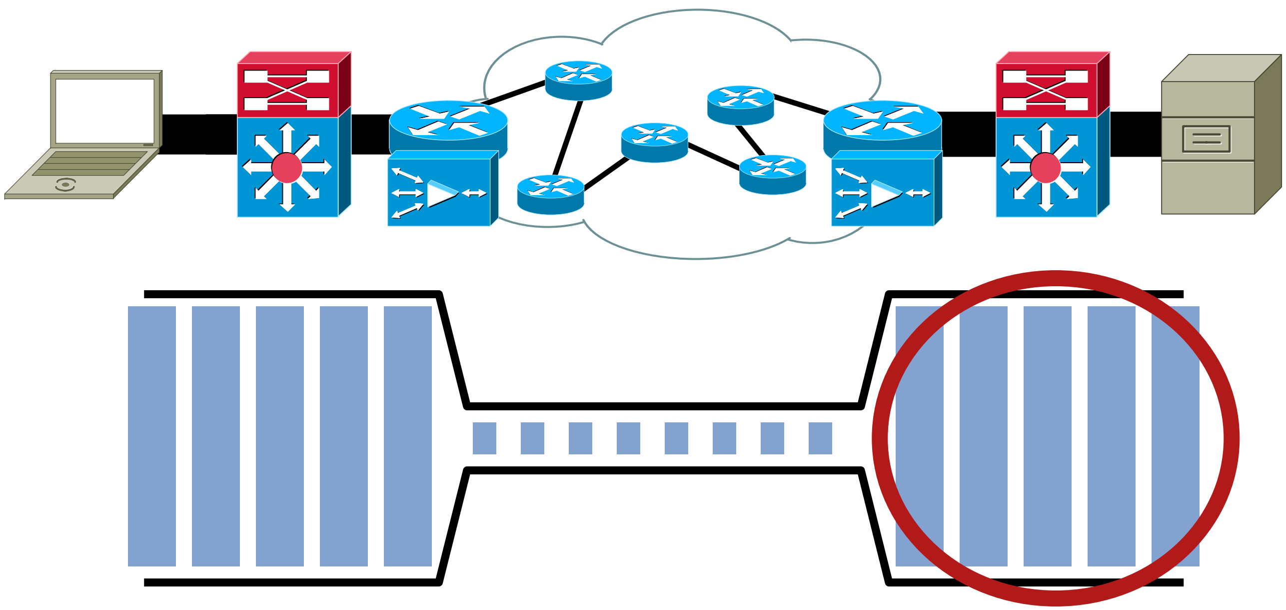
    Head: oldest entry contained in the FIFO database, first to be evicted if additional capacity is needed

    Tail: newest entry contained in the FIFO database, last to be evicted

- FIFO clock timestamps are not relative to actual system time, rather they are relative to the connection time itself

Impact of Advanced Compression

- Advanced compression can significantly minimize the amount of data that traverses the WAN
- Flows are safely rebuilt in their entirety at the distant end, allowing large amounts of application data to traverse the network



Transport Challenge

- Common TCP implementations on client and server operating systems can be bottlenecks to application performance
  - Inability to fill-the-pipe, i.e., utilize available bandwidth
  - Inefficient recovery from packet loss, retransmission
  - Bandwidth starvation for short-lived connections
- Cisco WAAS Transport Flow Optimization (TFO) utilizes industry-standard TCP optimizations to remove these application performance barriers

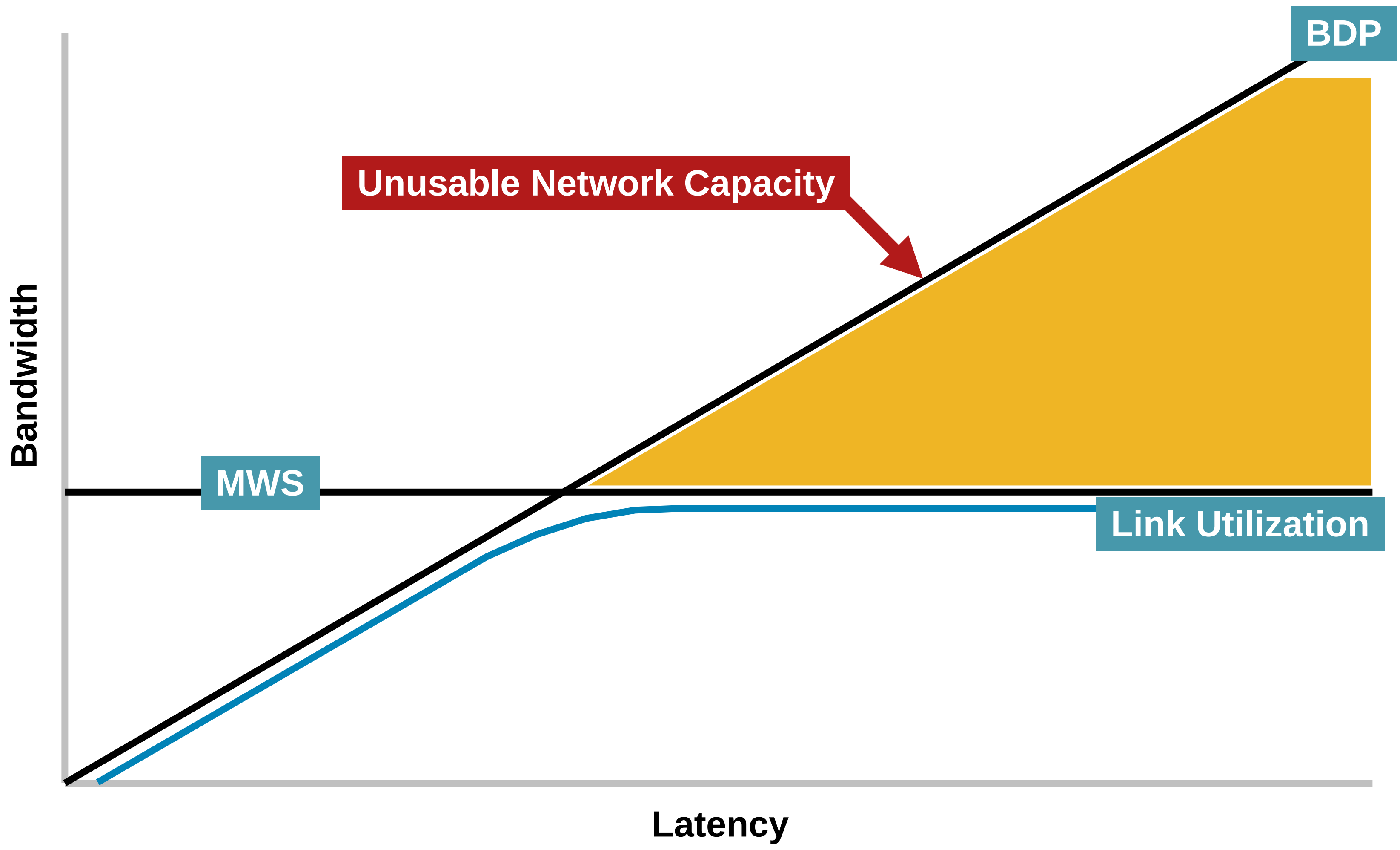
TCP Maximum Window Size (MWS)

- MWS (maximum window size) determines the maximum amount of data that can be in transit and unacknowledged at any given time
- BDP (bandwidth delay product) defines the amount of data that can be contained within a network at any given time

  If MWS > BDP, then application may not be throughput bound (i.e., application can "fill the pipe")

  If BDP > MWS, then application will not be able to fully utilize the network capacity (i.e., application can not "fill the pipe")
- Does not account for application-layer (L7) latency such as found with protocol-specific messaging

Link Utilization and MWS, BDP



TCP Window Scaling (RFC 1323)

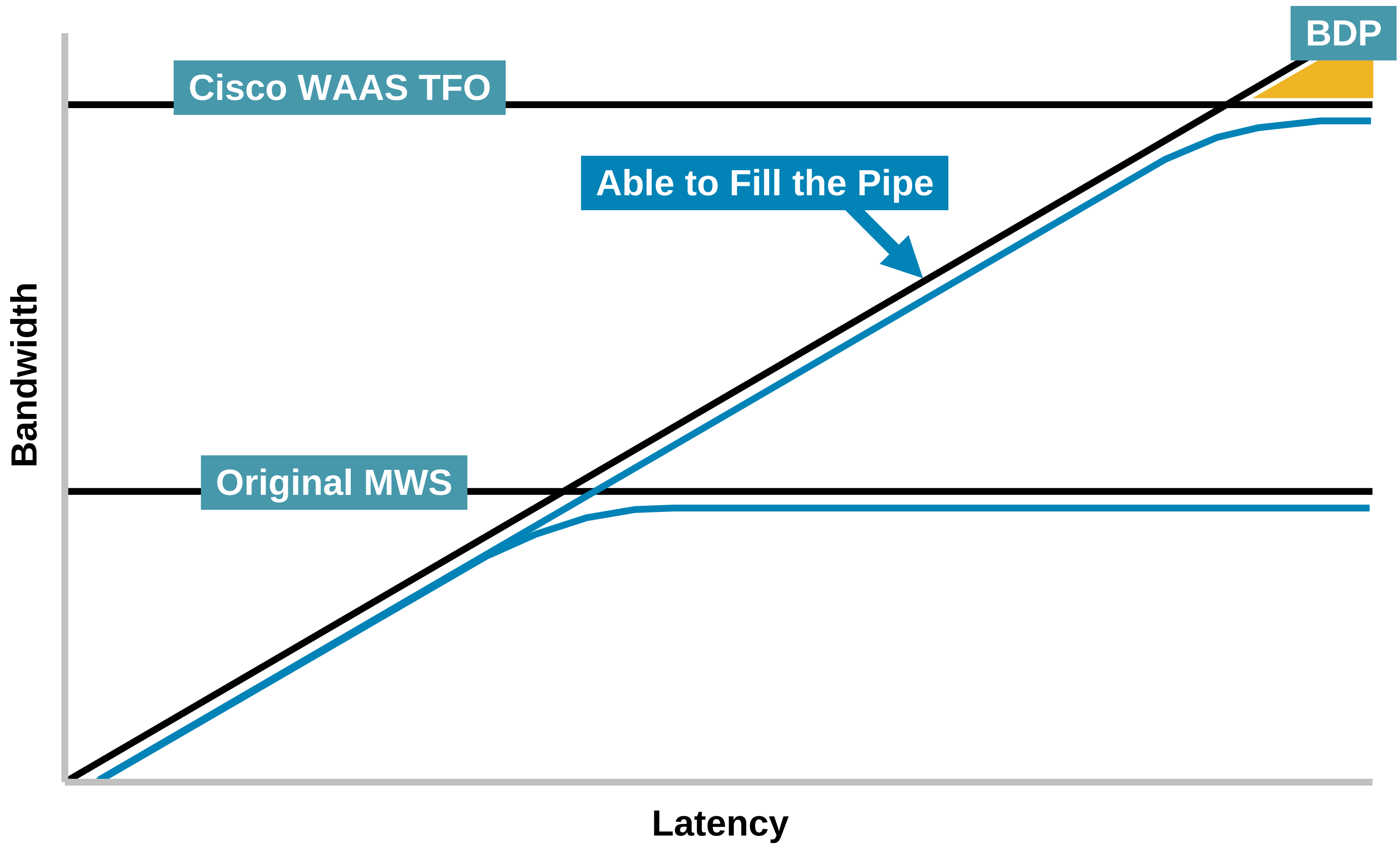
- RFC 1323—TCP Performance Extensions—defines the use of a TCP option to scale the TCP window beyond the standard 16-bit limitation (64KB)
- Window scaling applies a binary shift by the decimal value supplied in the data field

    A window scale value of 0010 (2) would shift the requested window size to the left by 2 bits

    1000 0000 0000 0000 (64KB) would become

    1000 0000 0000 0000 00 (256KB)
- Cisco WAAS provides window scaling up to 2MB per optimized TCP connection

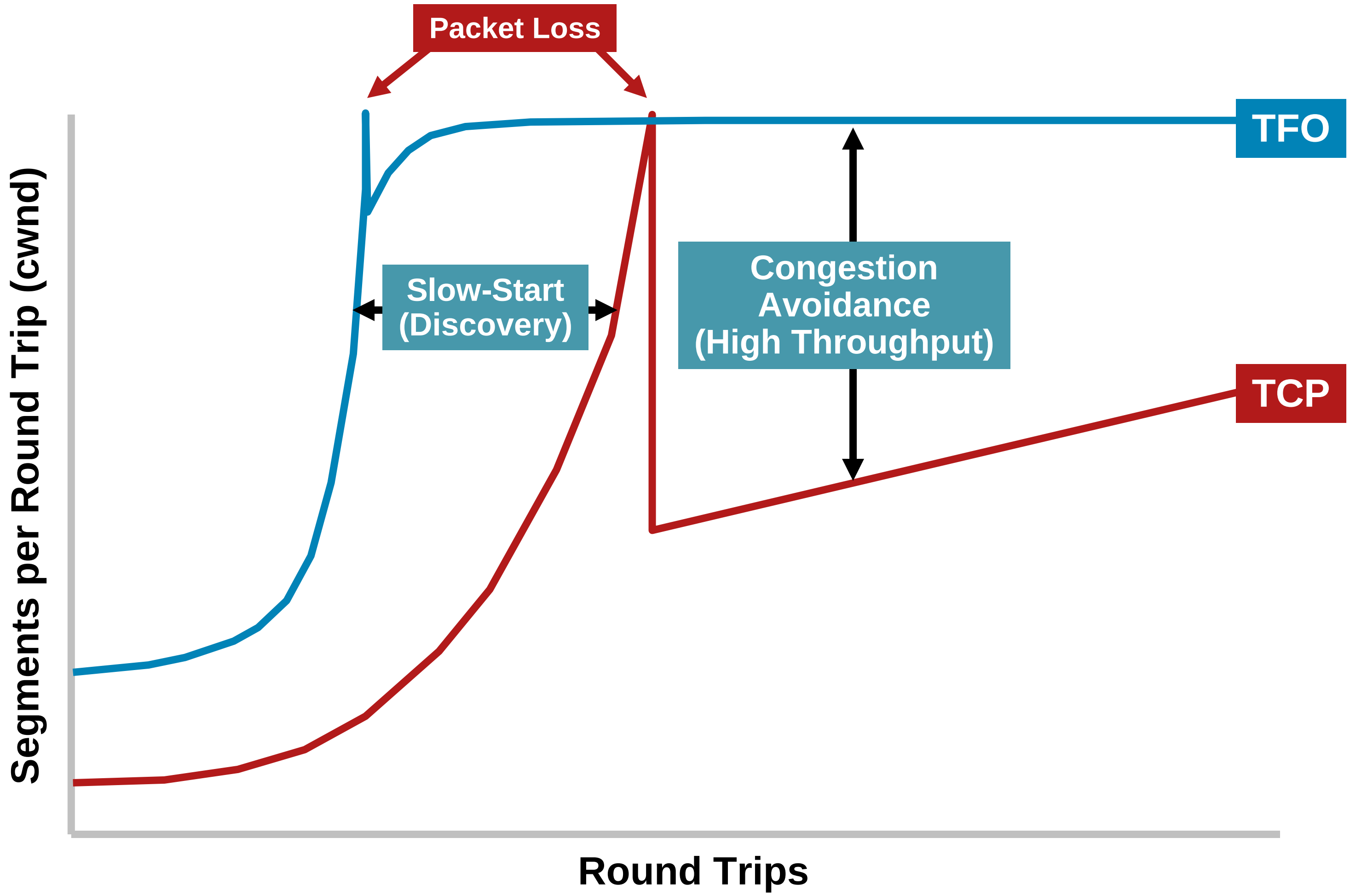
Link Utilization After Window Scaling



Large Initial Windows

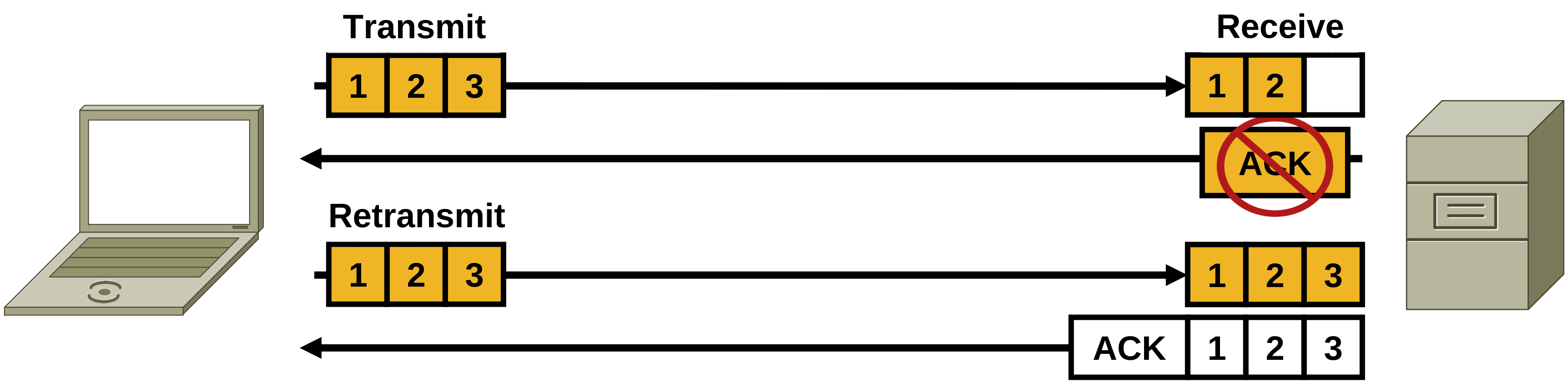
- While 80% of network traffic is typically associated with long-lived connections (elephants), approximately 80% of network connections are short-lived (mice)
- Short-lived connections transmit smaller numbers of packets and are torn down before ever leaving the slow-start phase of TCP
- Cisco WAAS Large Initial Windows, based on RFC3390, increases initial window size to expedite entry into congestion avoidance mode for high throughput

Large Initial Windows



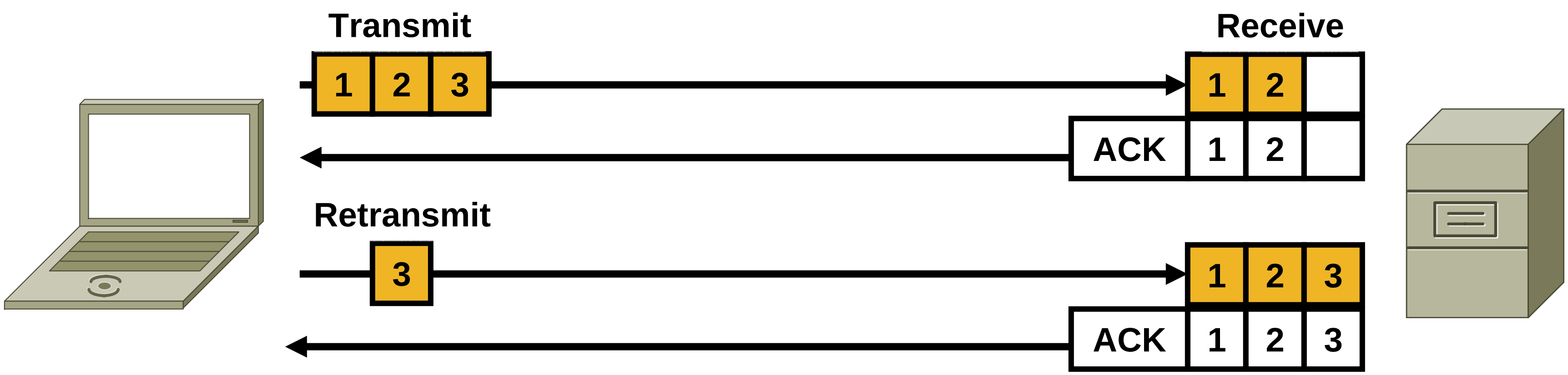
Selective Acknowledgement

- Standard TCP implementations acknowledge receipt of data by acknowledging the entire window has been received
- Loss of a packet causes retransmission of the entire TCP window, causing performance degradation as the window becomes larger



Selective Acknowledgement (Cont.)

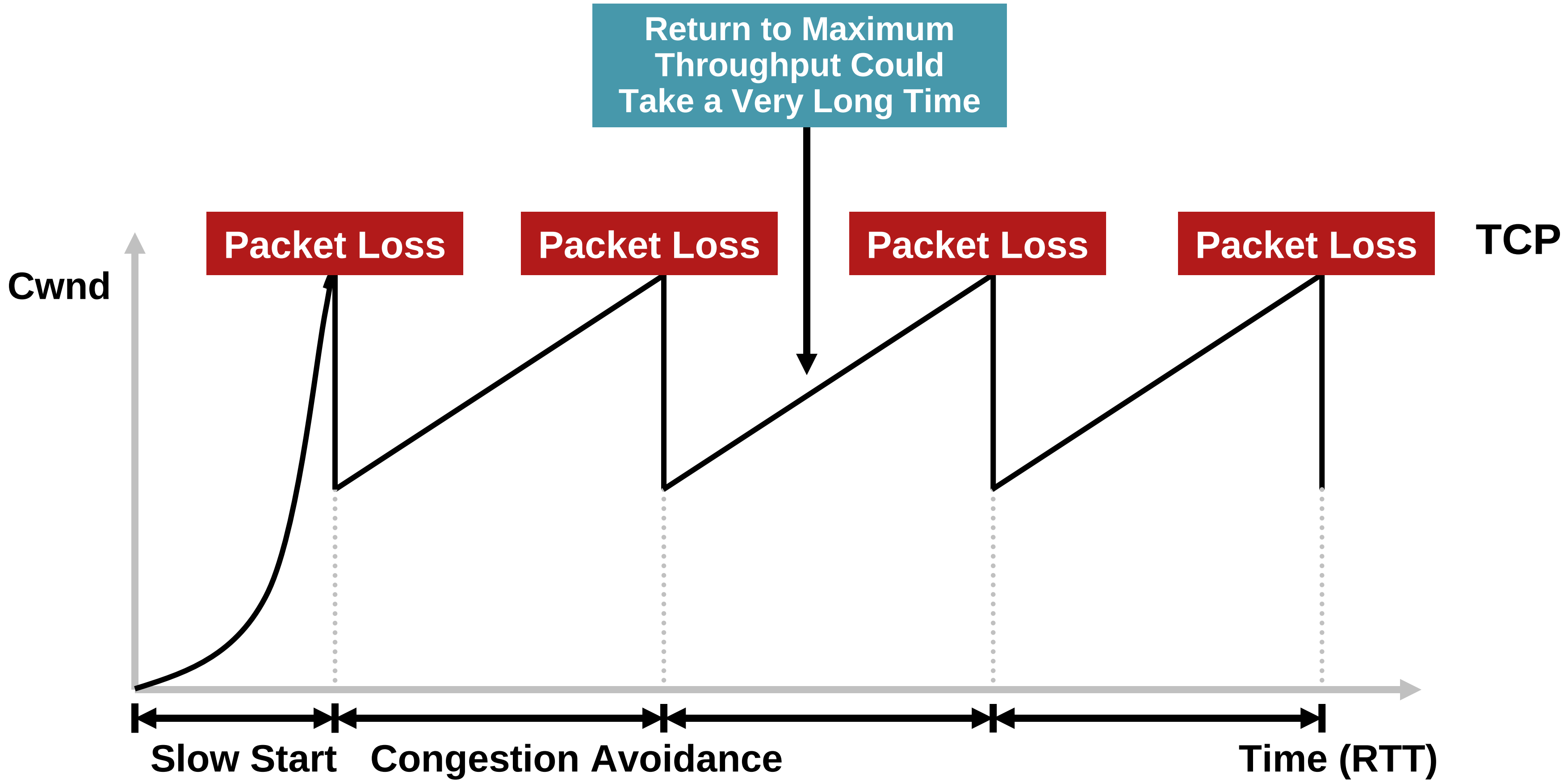
- Selective acknowledgement improves acknowledgement of transmitted data, improve delivery of missing segments, and unnecessary minimize retransmission



Standard TCP Congestion Avoidance

- Standard TCP implementations employ an exponential slow start to increase throughput to the slow start threshold
- From the slow start threshold, the congestion window is increased linearly by one packet per round-trip until packet loss is encountered
- Upon encountering packet loss, the congestion window is cut in half to return to a throughput level safe given the congested environment
- The net result is “saw-tooth” throughput, and return to maximum throughput can take hours for long-lived connections and LFNs

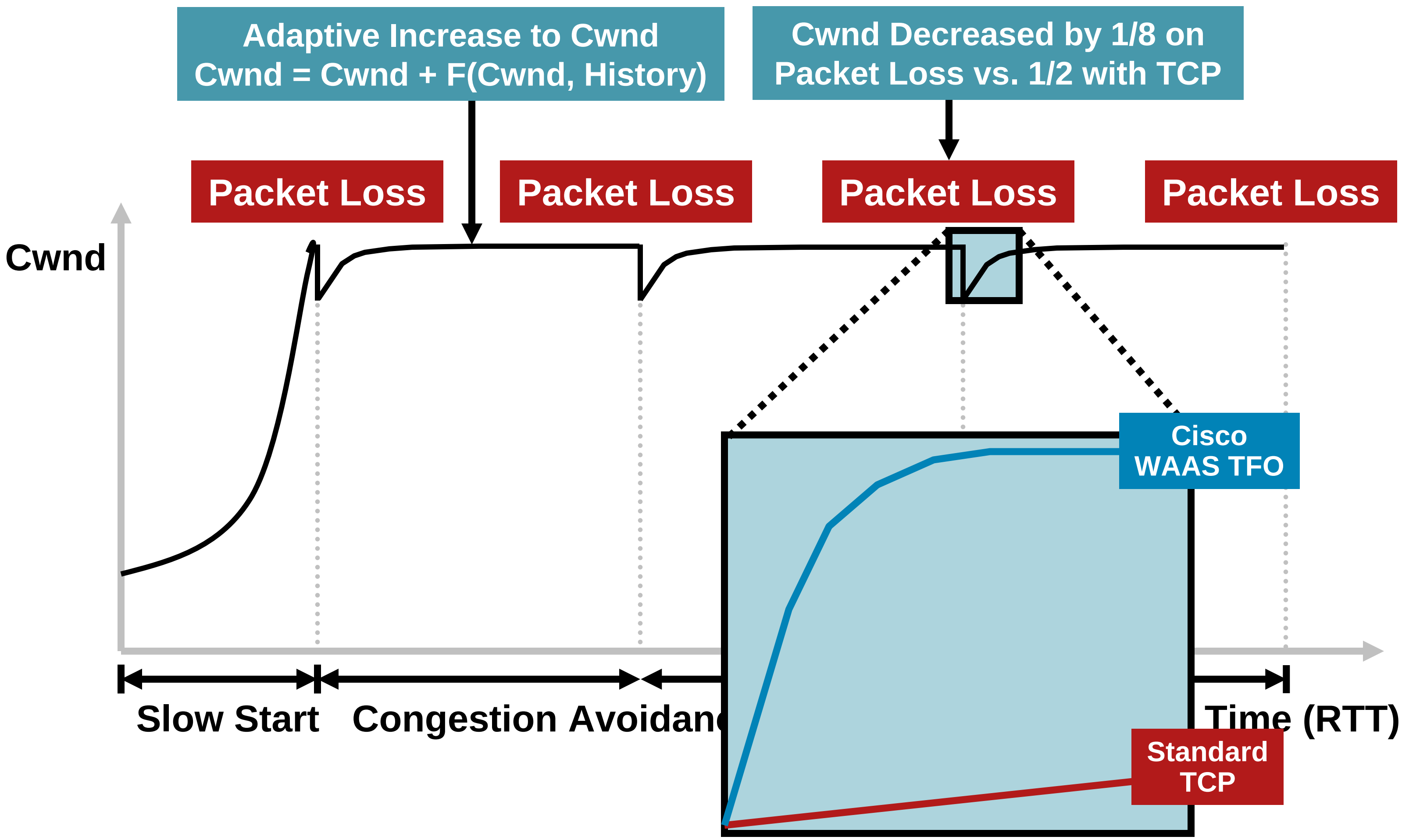
“Saw-tooth” TCP Throughput



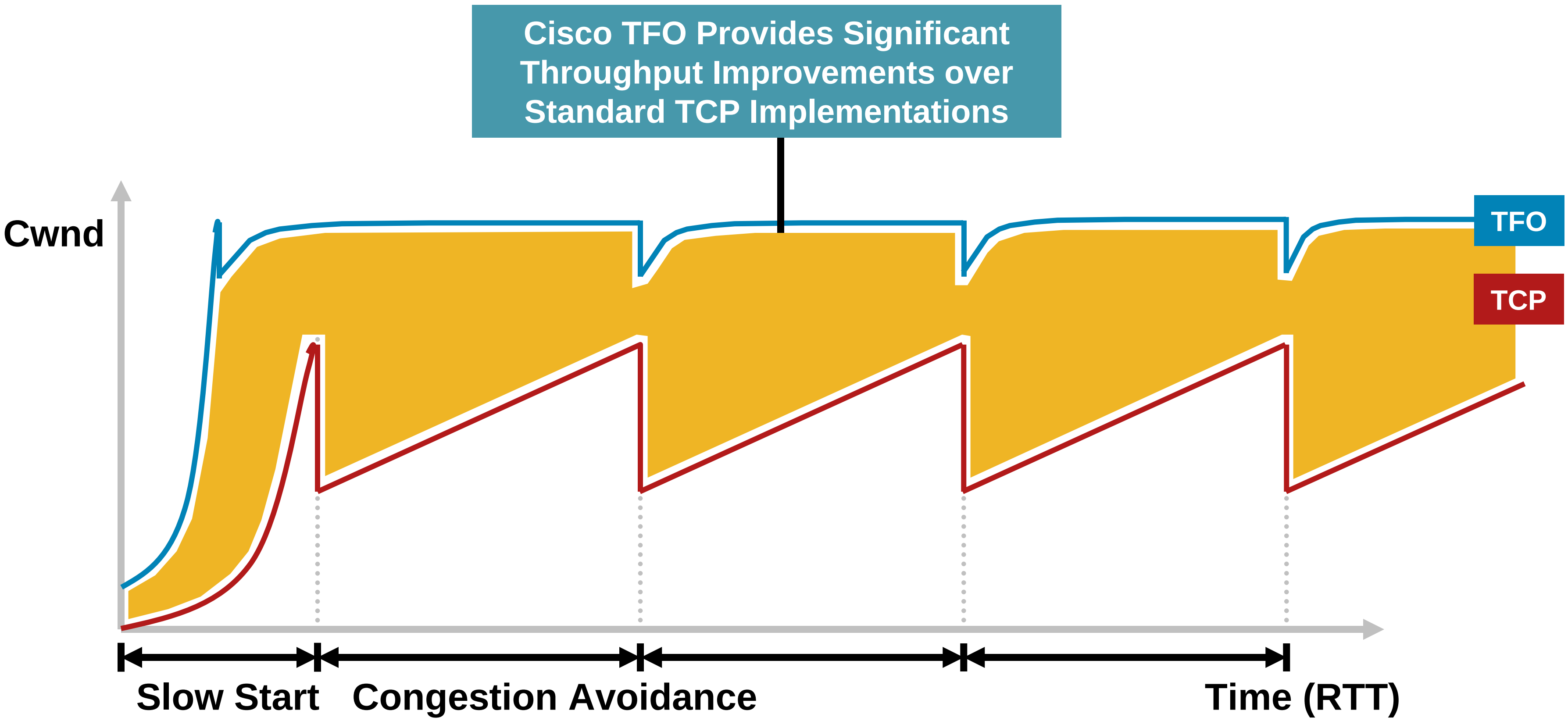
Binary Increase Congestion (BIC)

- Binary Increase Congestion (BIC) congestion avoidance system is used to improve throughput in lossy environments
- Uses a binary search to adaptively increase the congestion window, resulting in a stable and timely return to higher levels of throughput
- Decreases congestion window only by 1/8 (rather than 1/2 as compared to TCP) when packet loss is encountered, mitigating the majority of the performance penalty

WAAS Throughput and Congestion Avoidance

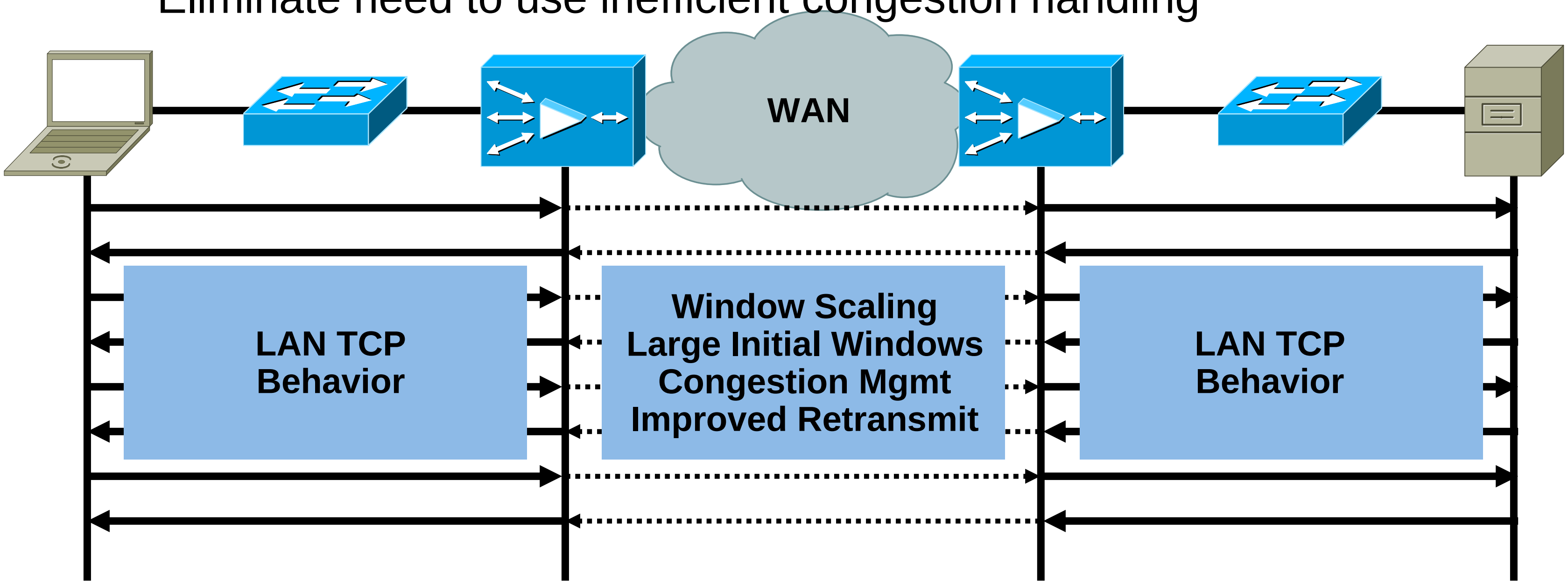


Comparing TCP and TFO



Impact of Transport Flow Optimizations

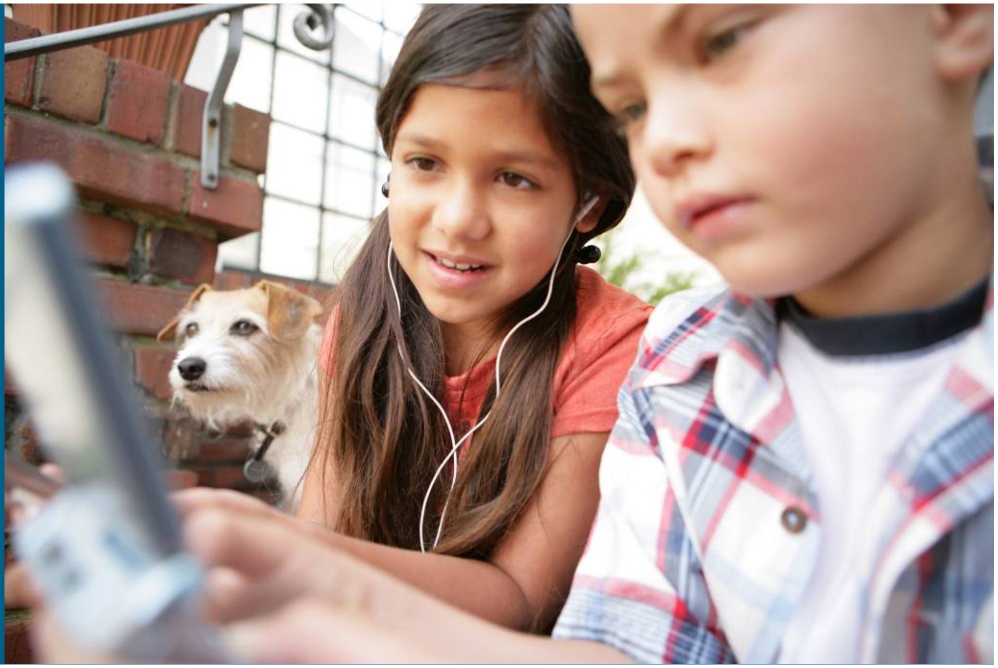
- TFO overcomes TCP performance bottlenecks
- Shields nodes connections from WAN conditions
  - Clients experience fast acknowledgement
  - Minimize perceived packet loss
  - Eliminate need to use inefficient congestion handling



Summary

- The network provides the foundation necessary for ensuring high performance access to centralized applications and other content
- Traffic differentiation provides the visibility necessary to configure the network to respond according to business priority and ensure responsiveness
- TCP can be optimized to overcome performance challenges associated with packet loss, maximum throughput and inefficiency
- Advanced compression can be leveraged to more efficiently utilize available WAN capacity by mitigating redundant segment transmissions, thereby saving on bandwidth
- Application-specific acceleration is required to properly improve performance for application protocols that require many synchronous and serial operations

Q and A



Any Questions ?