

Introduction to Intel Cache



i n k t o m i

Intel Cache Course Overview



inktom i

- Technology: TCP/IP Overview
- Proxy Service: Overview
- Intel Cache: Overview
- Intel Cache: Installation
- Intel Cache: Features and Benefits
- Intel Cache: Reverse Proxy
- Intel Cache: Processing Request
- Intel Cache: Architecture

Intel Cache Course Overview



inktom i

- Intel Cache: Configuration
- Intel Cache: Monitoring
- Intel Cache: Miscellaneous

Technology: TCP/IP Overview



i n k t o m i

Chapter Overview



inktom

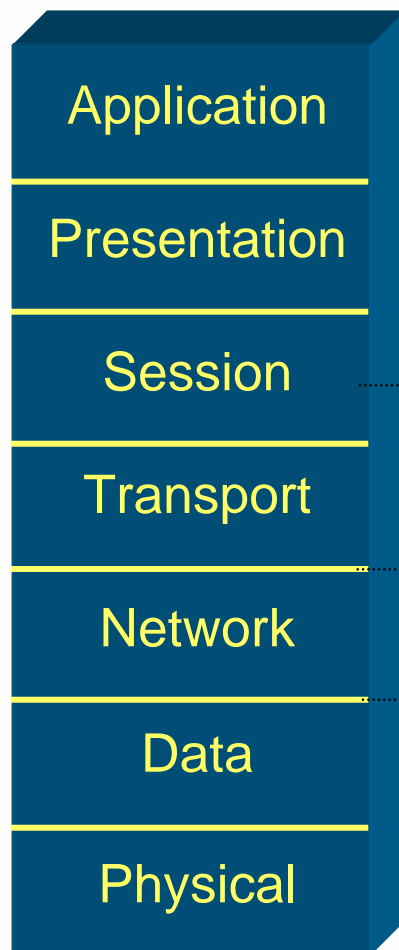
- **Internet Protocol Suite**
- **Layer Addressing**
- **Layer Switches**
- **Monitoring a HTTP Request**

Comparing OSI & IP Suite

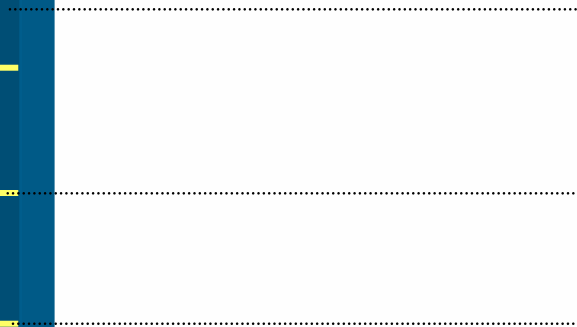
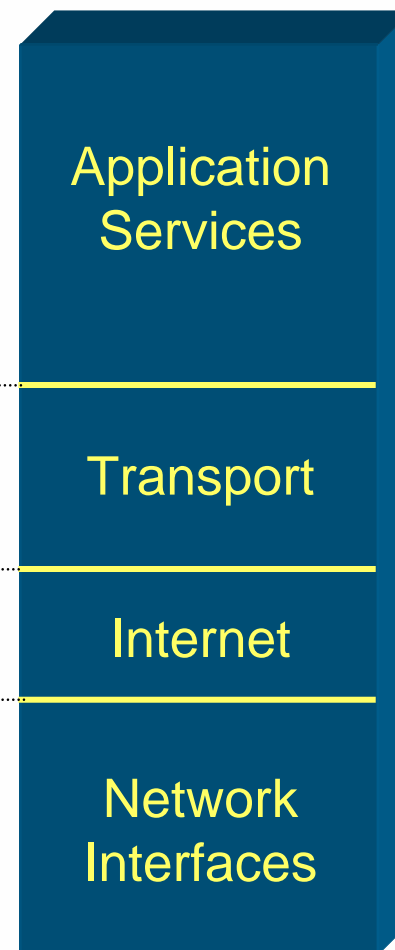


inktom i

Open Systems Interconnection



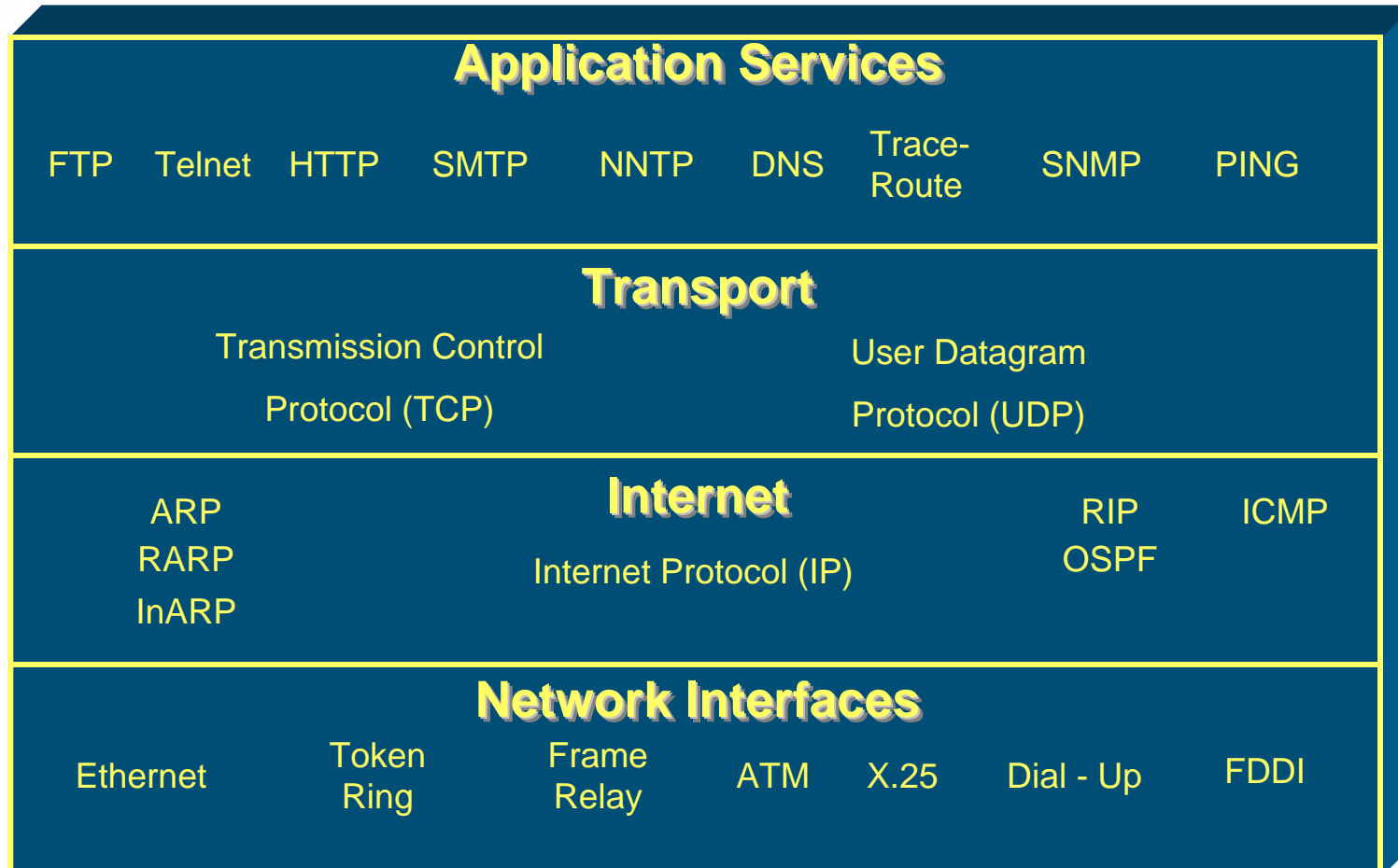
Internet Protocol Suite



Internet Protocol Suite

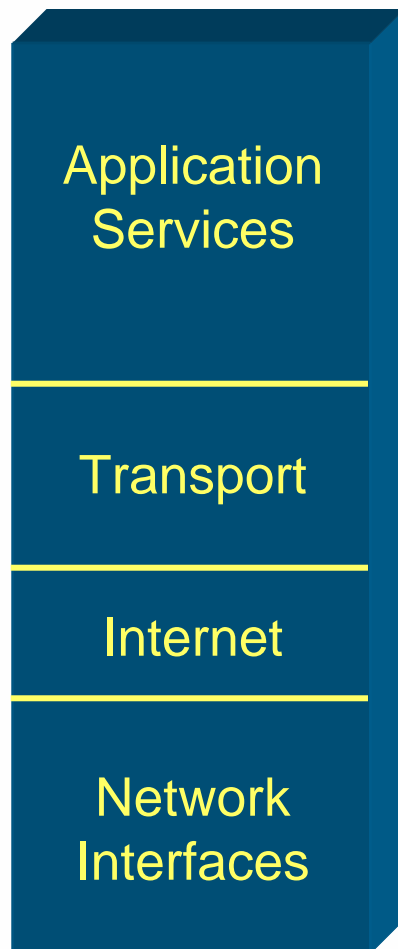


inktom i





Internet Protocol Suite



There is always a Source and Destination Address

→ Port Address

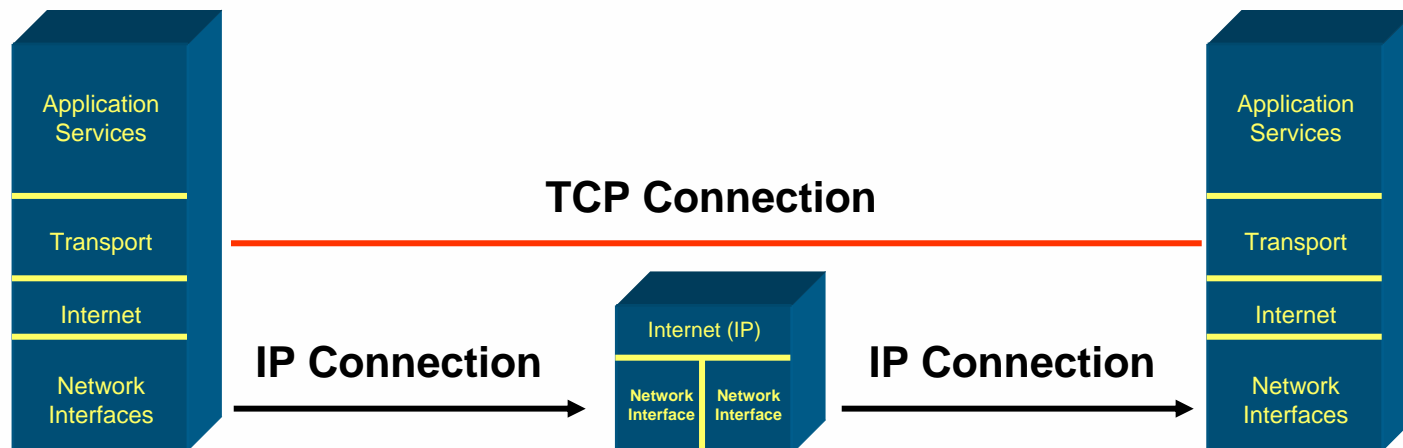
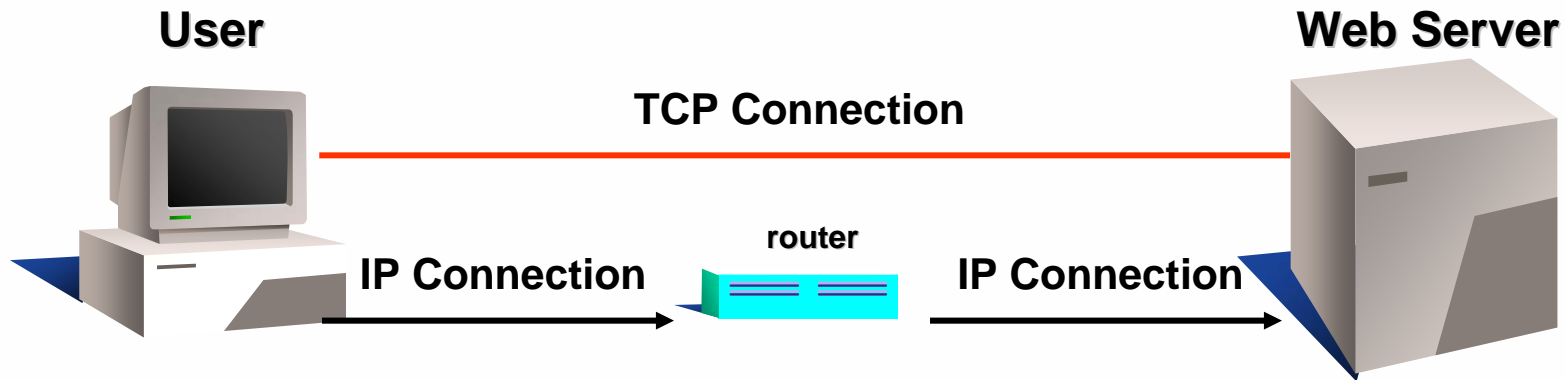
→ IP Address

→ Ethernet/Token Ring = MAC
Frame Relay = DLCI
ATM = NSAPA

Requesting A Web Page



inktom i

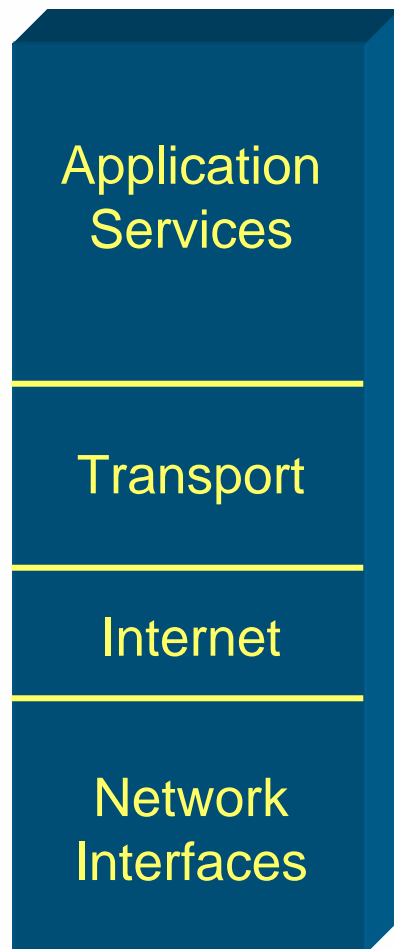


Building A Request

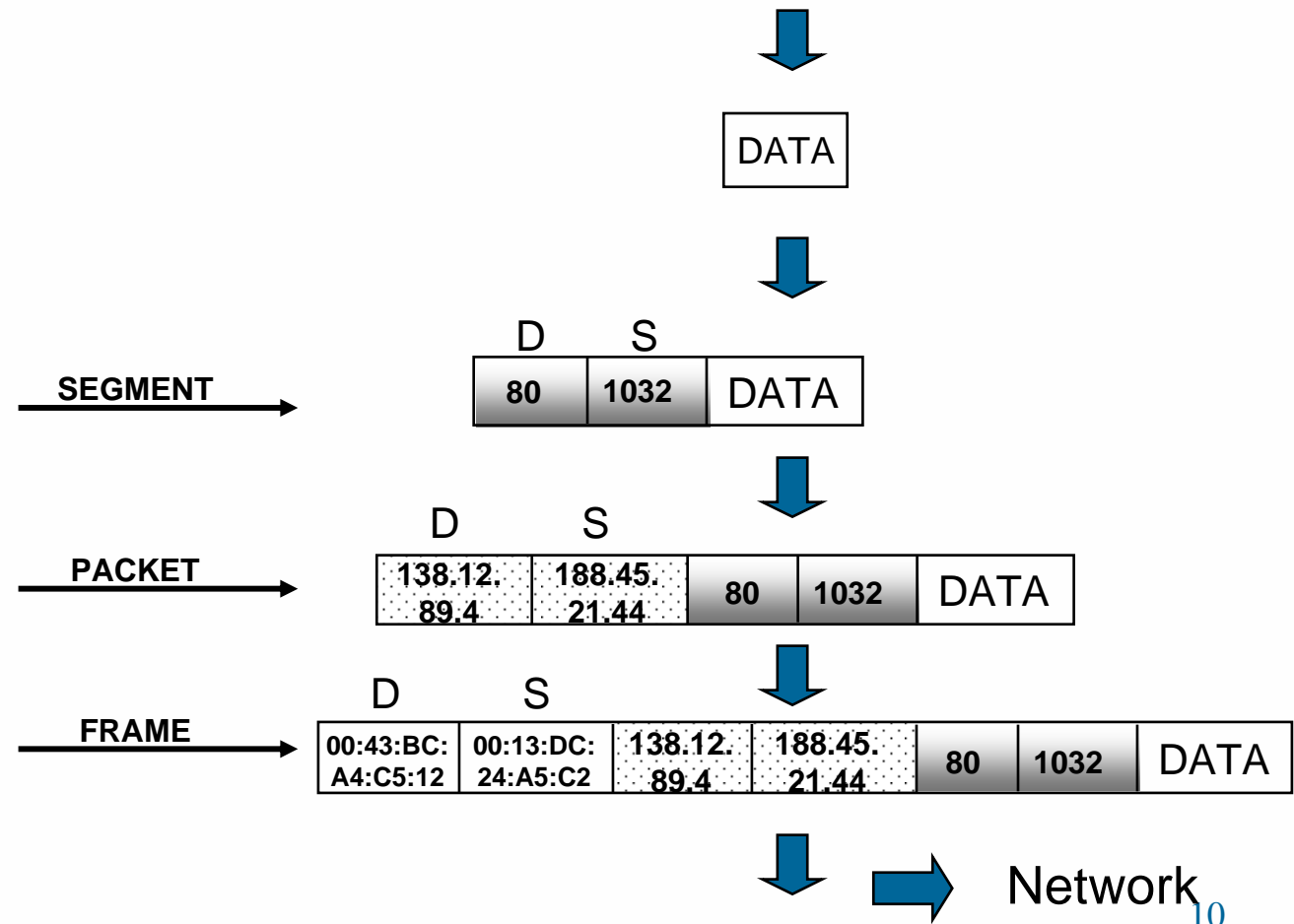


inktom i

Internet Protocol Suite



User's (HTTP) Request

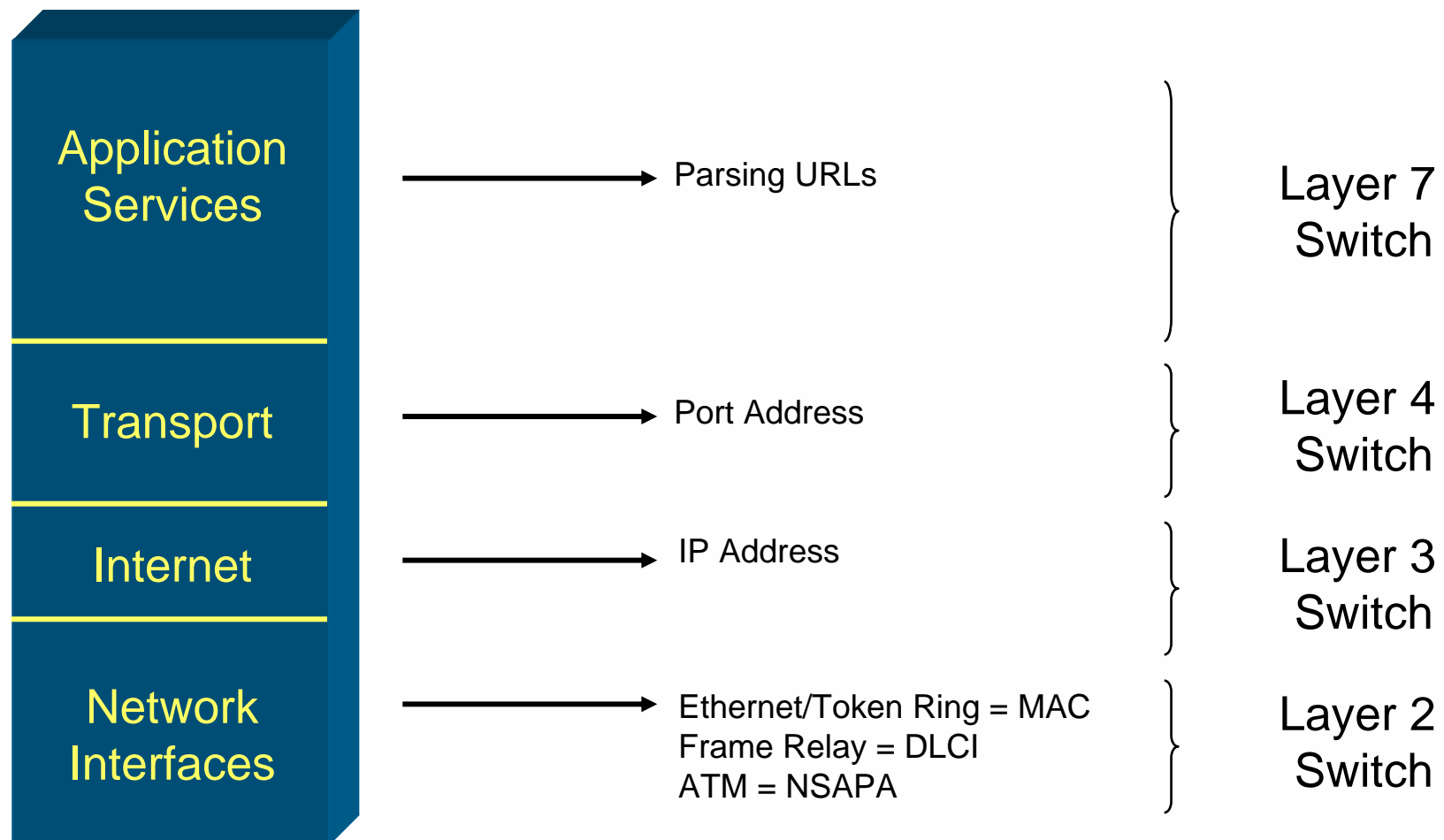


Layer Switches



inktom i

Internet Protocol Suite

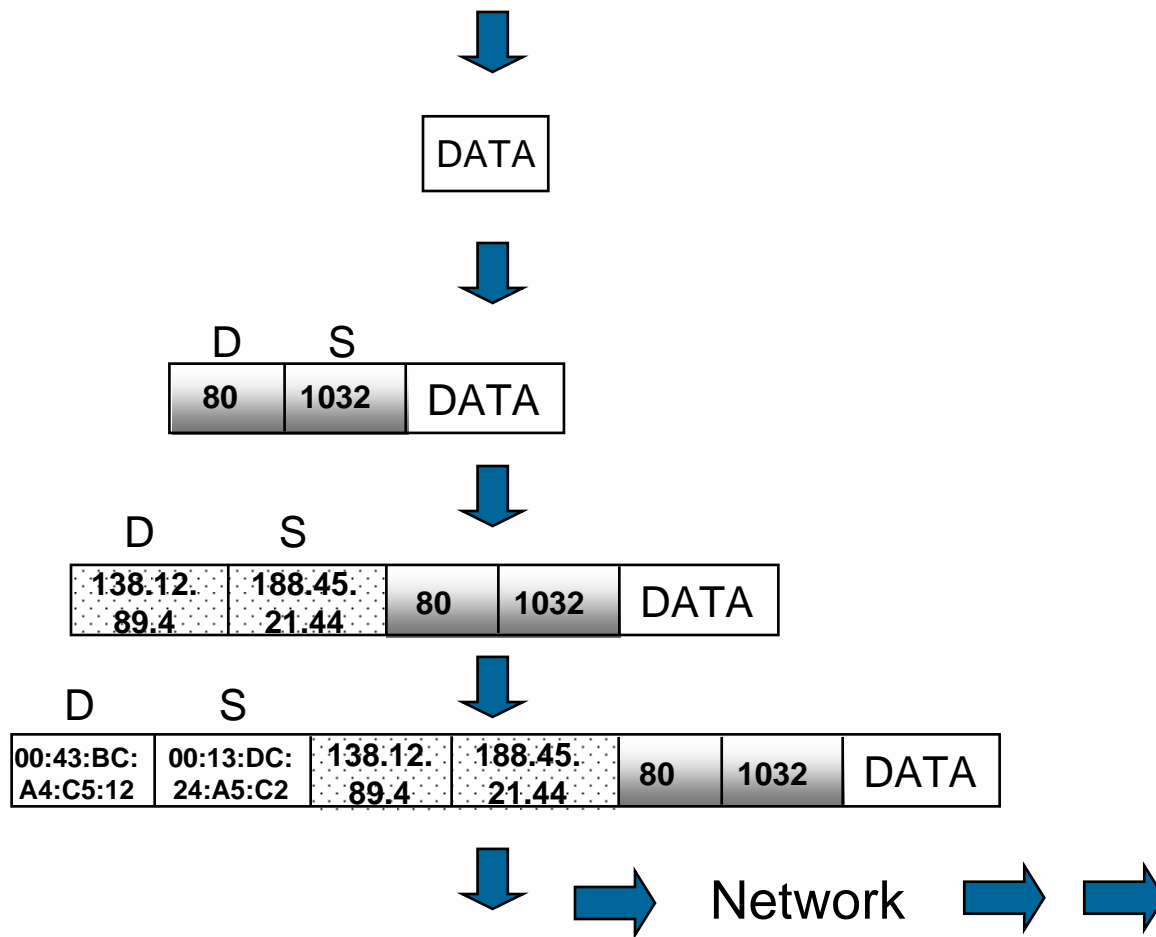


Switching

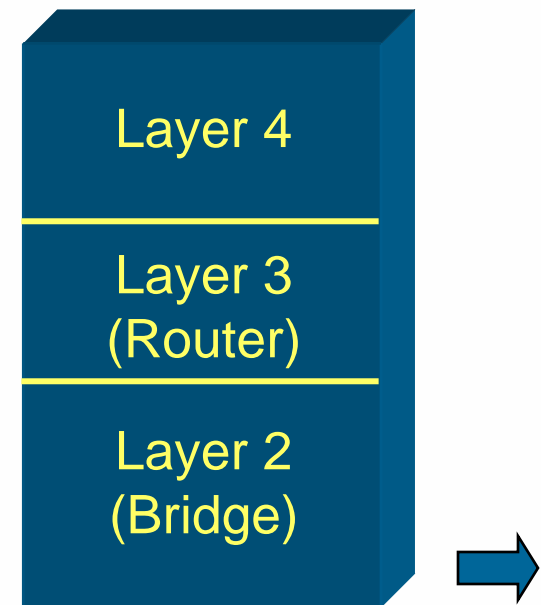


inkomi

User's Request



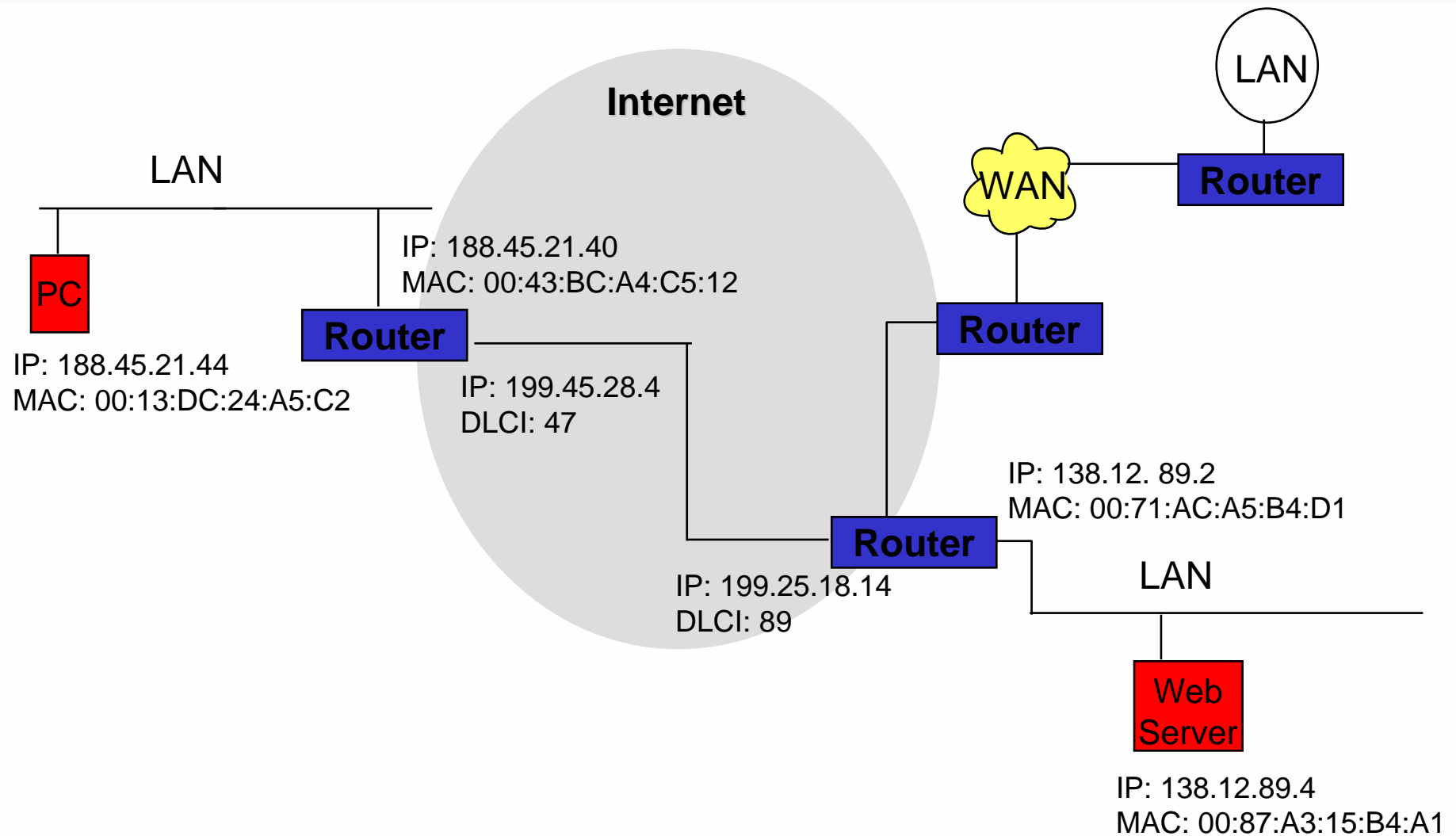
Switch



Routing A Request



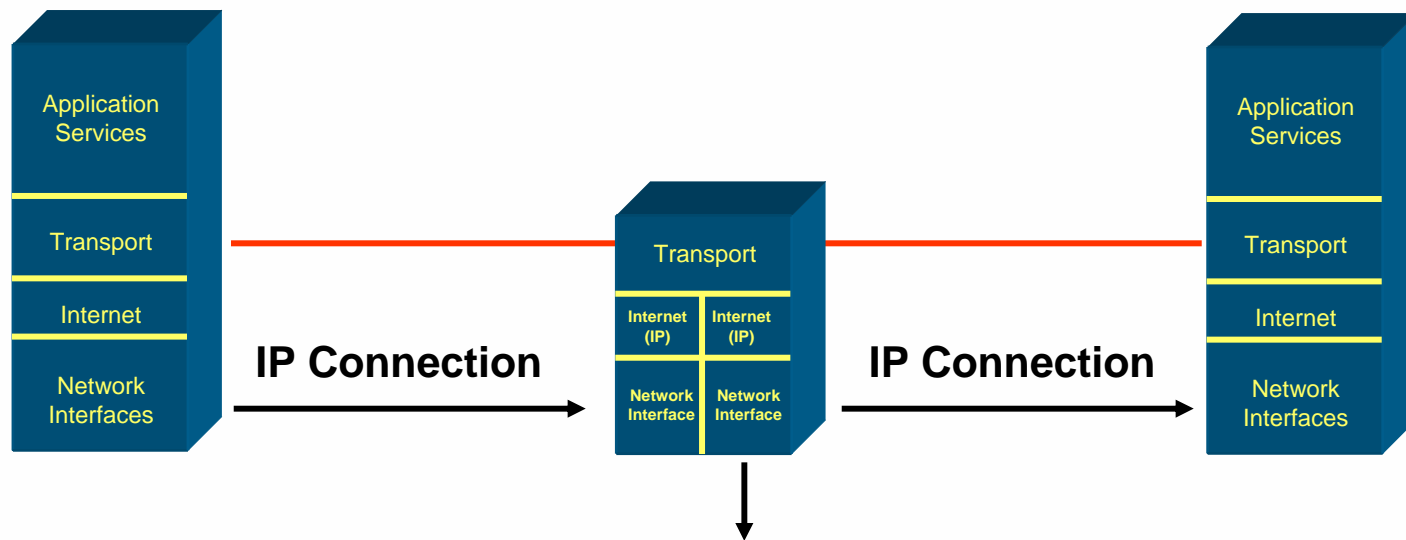
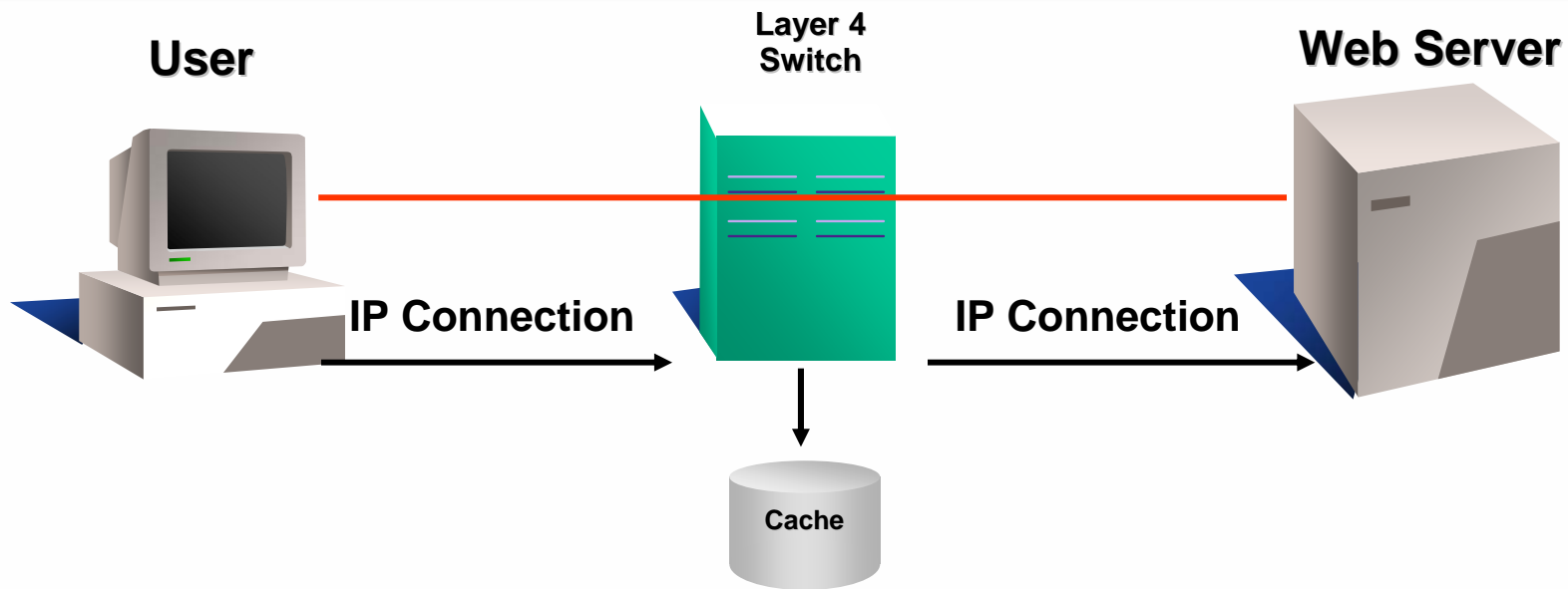
inktom i



Layer 4 Connection



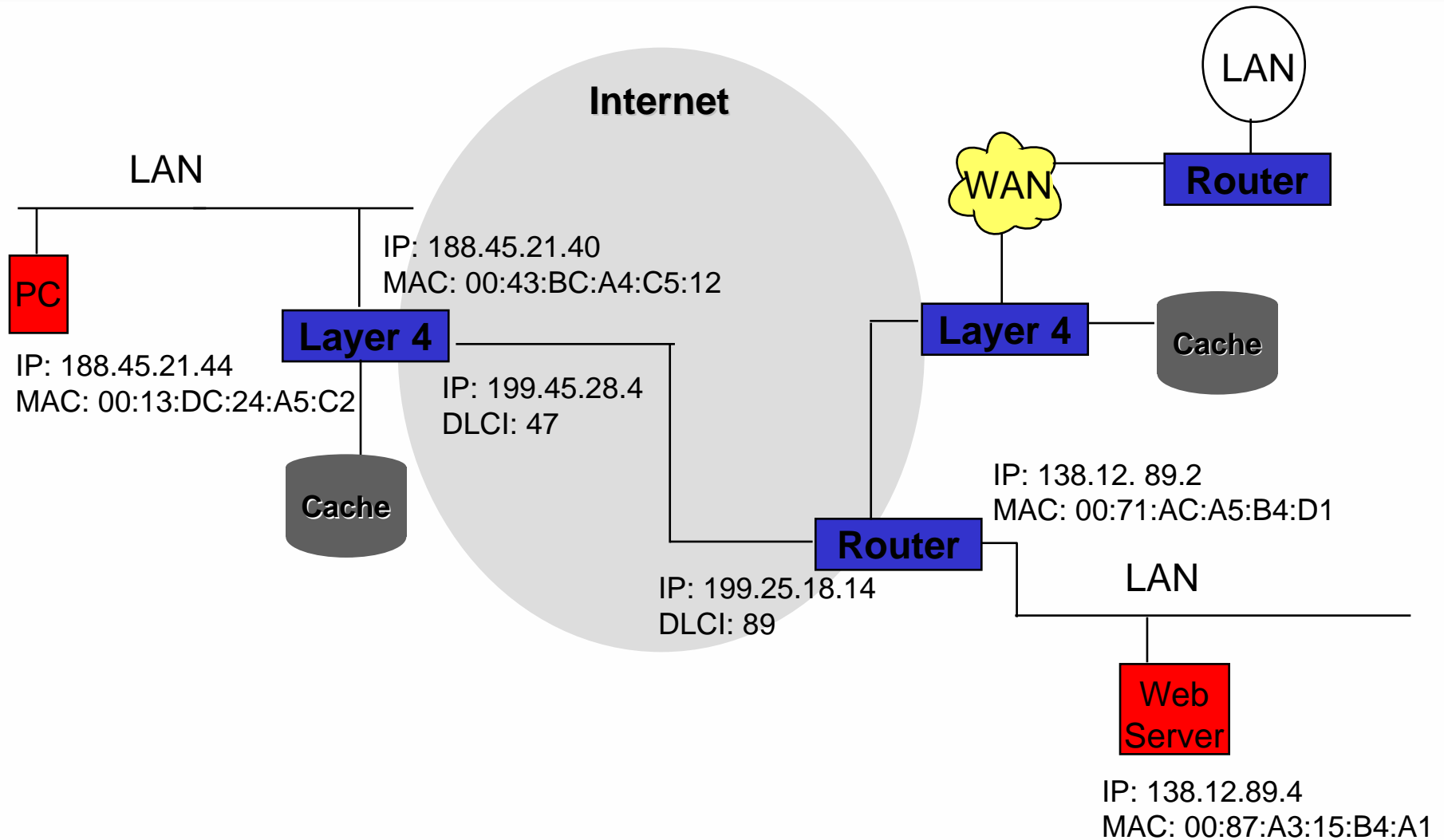
inktom i



Routing A Request



inktom i



Review Questions:



inkomi

- What is the benefit of the TCP/IP pairing?
- Why has TCP/IP become the protocol of choice?

Proxy Service: Overview



i n k t o m i

Chapter Overview



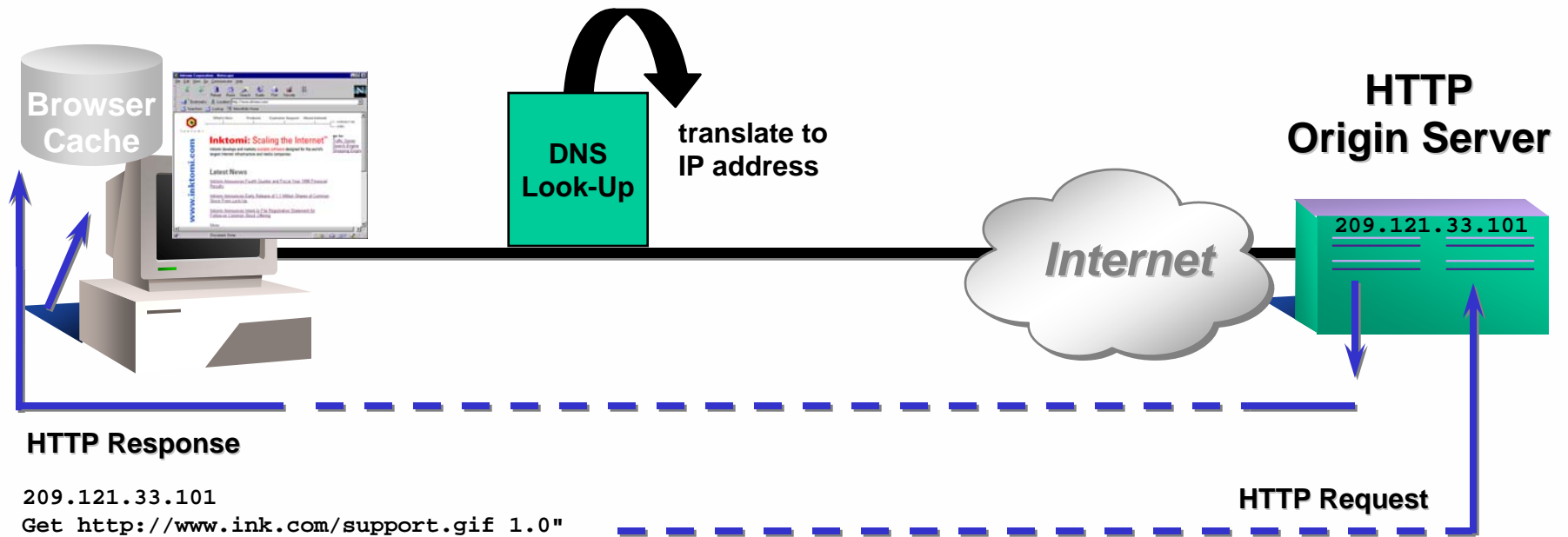
inktom

- **Forward Proxy**
- **Transparency**
- **Reverse Proxy**
- **Content Distribution**
- **Media Caching**
- **Practice Lab**

The Direct Connection



inktom i



Drawbacks of Direct Connections



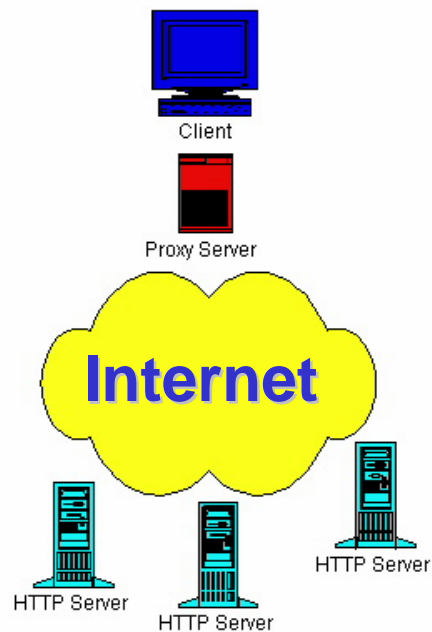
inktom i

- **Each request is served by the origin web server generating redundant traffic over the Internet**
- **Each request utilizes exorbitant amounts of resources**
- **Each request can create unpredictable spikes in traffic**
- **Each request demands a level of performance**

Proxy Cache: Forward & Reverse

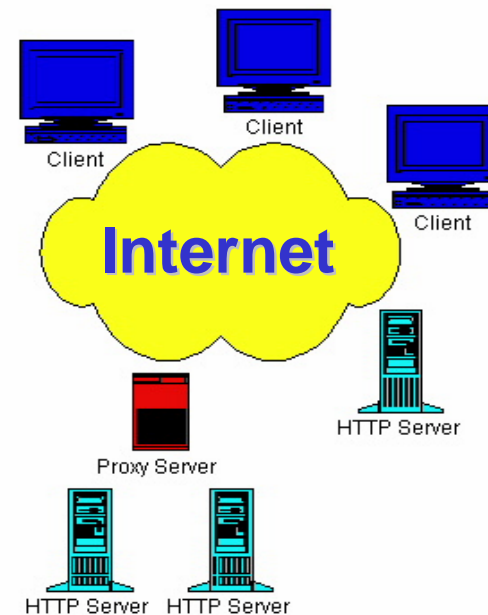
Forward Proxy Cache

Cache server acts as a proxy for client requests.



Reverse Proxy Cache

Cache server acts as a proxy for the origin server.



Benefits of Forward Proxy



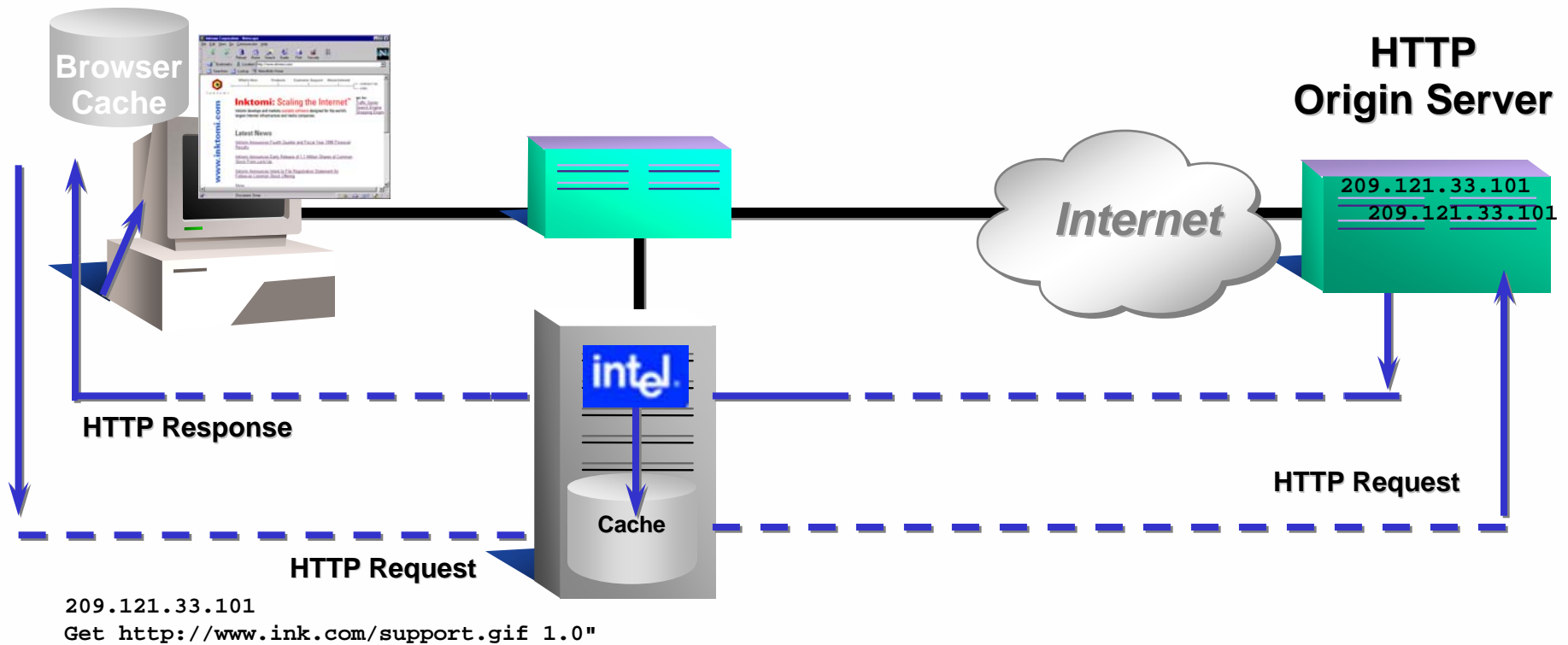
inktom

Original requests are served by the origin server and redundant requests are served by the cache!

Origin Server Responds



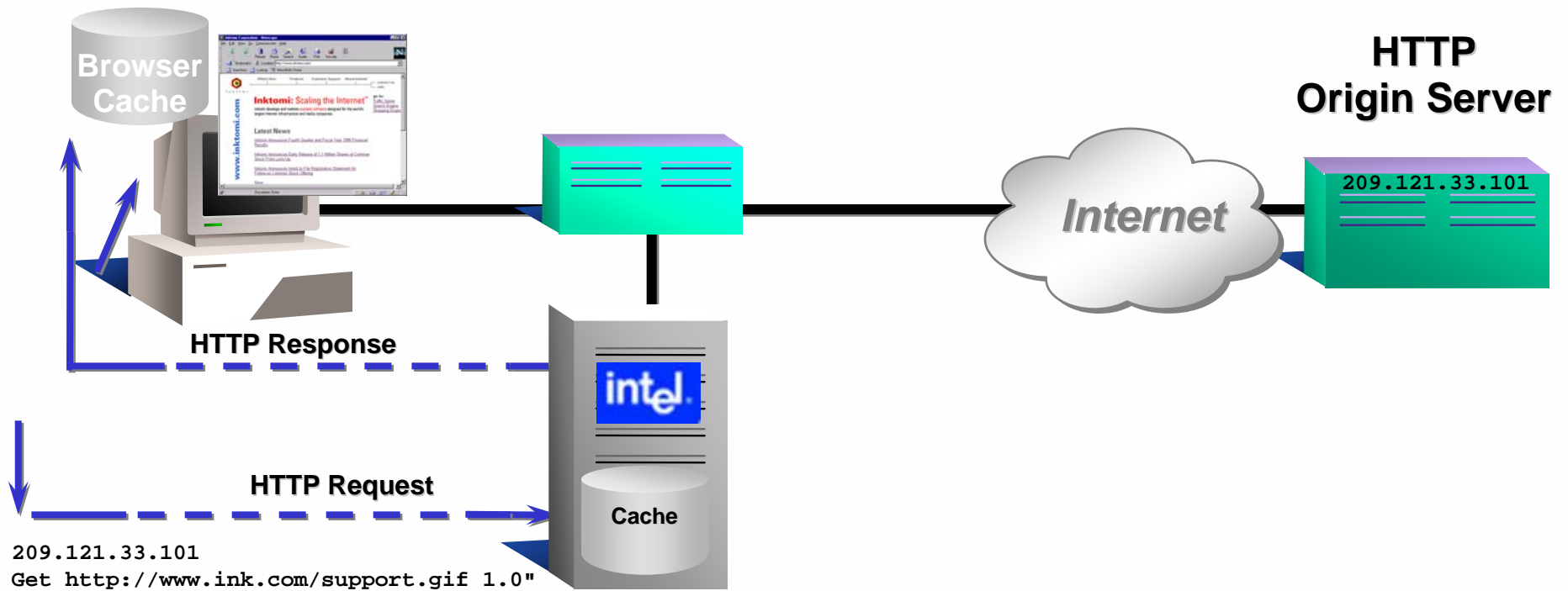
inktom i



Intel Cache Responds



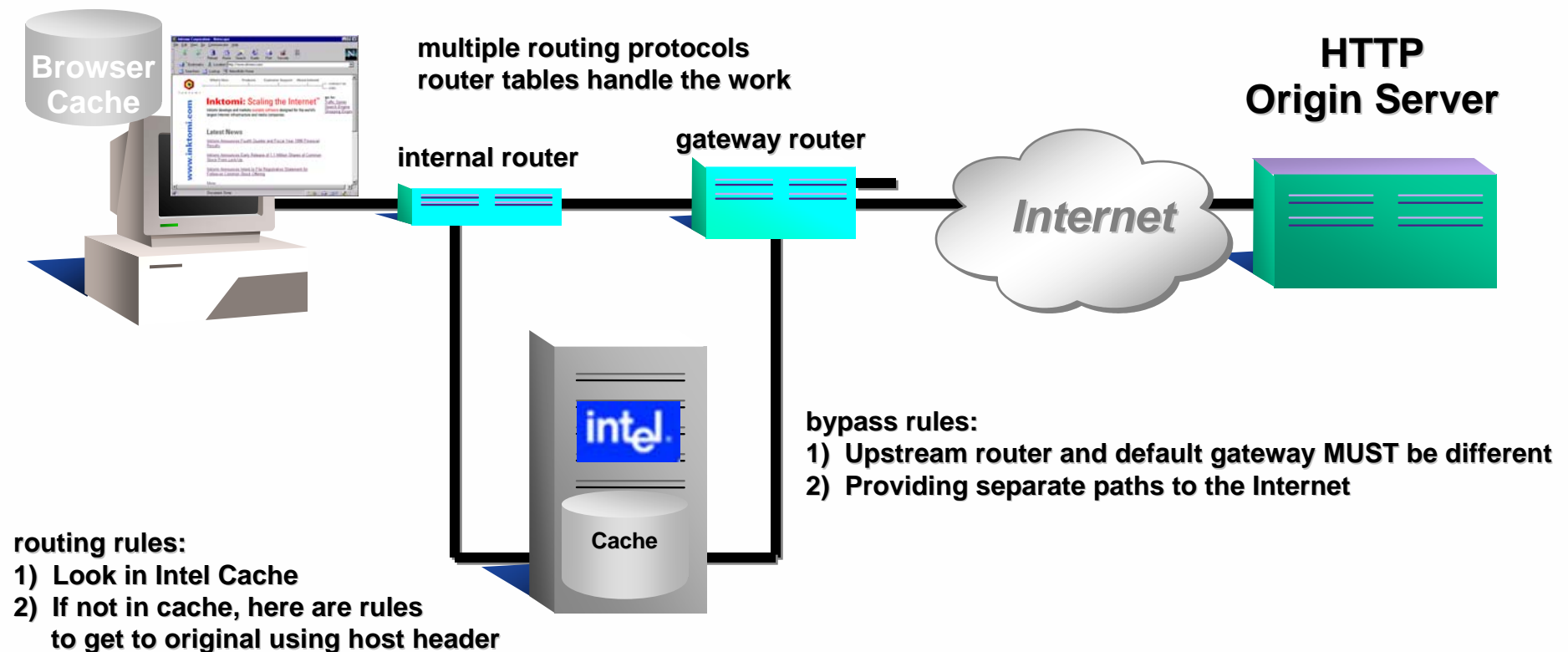
inktom i



Redirecting Requests via Router



inktom i

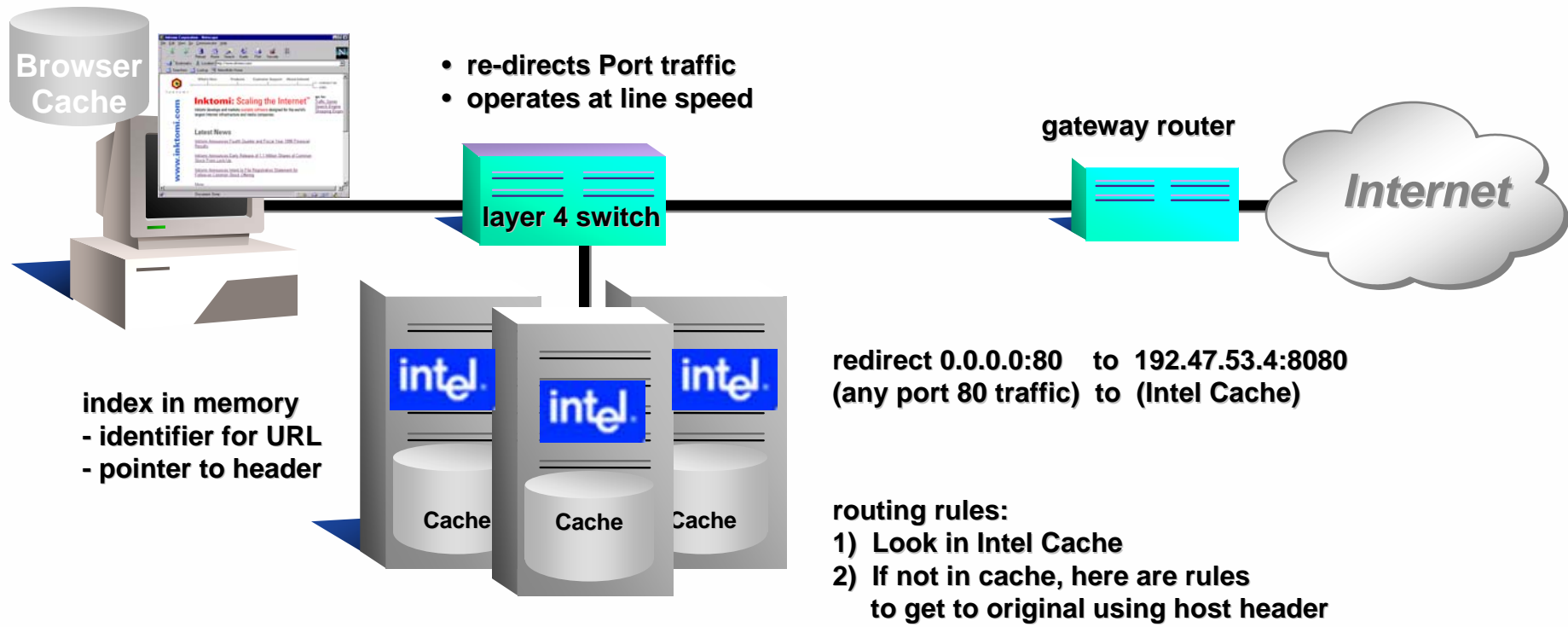


An environment with an interconnection device is optimal for transparency

Redirecting Requests via Layer 4 Switch



inktom i



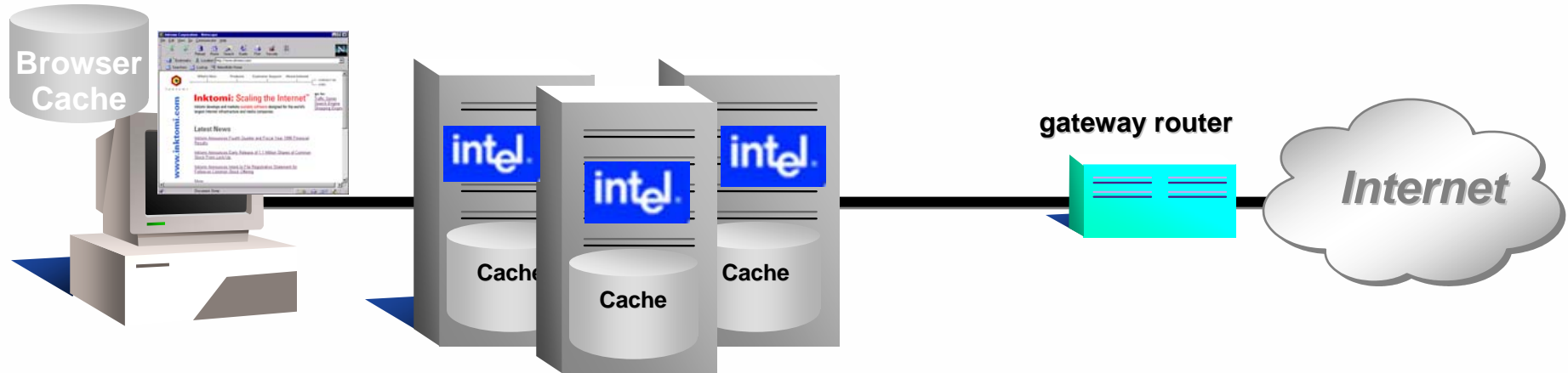
objects stored on disk
- header with full meta data
- URL, last modified date, time to live, etc.

An environment with an interconnection device is optimal for transparency

Software Based Proxy



inktom i



Cache must do additional work to compensate for non-existent interconnection device

This environment may introduce the potential for problems

Cache must load additional software required to handle port redirection which adds latency to the caching process

This environment is not optimal!



- **Manual Proxy Configuration**
 - Client browser is manually configured to send particular types of traffic to the Cache
- **Automatic Proxy Configurations**
 - Client browser is provided with a URL that will access a “pac” file which will automatically configure the browser to send particular types of traffic to the Cache
- **Transparency**
 - Client browser is set for a direct connection (default setting in browser) to the origin server and is unaware that the Cache exists
- **WPAD (Web Proxy Auto Discovery)**
 - Client browser will automatically discover the presence of the Cache without manual intervention



- **Transparency makes it possible to automatically route user traffic directly to the Intel Cache**
 - Redirects web requests transparently through cache
 - Respects sites having no control over user browsers or their settings
 - Can be implemented as a hardware or software solution
 - Hardware switch is best
 - Software solution requires the use of external software packages

Benefits of Forward Caching



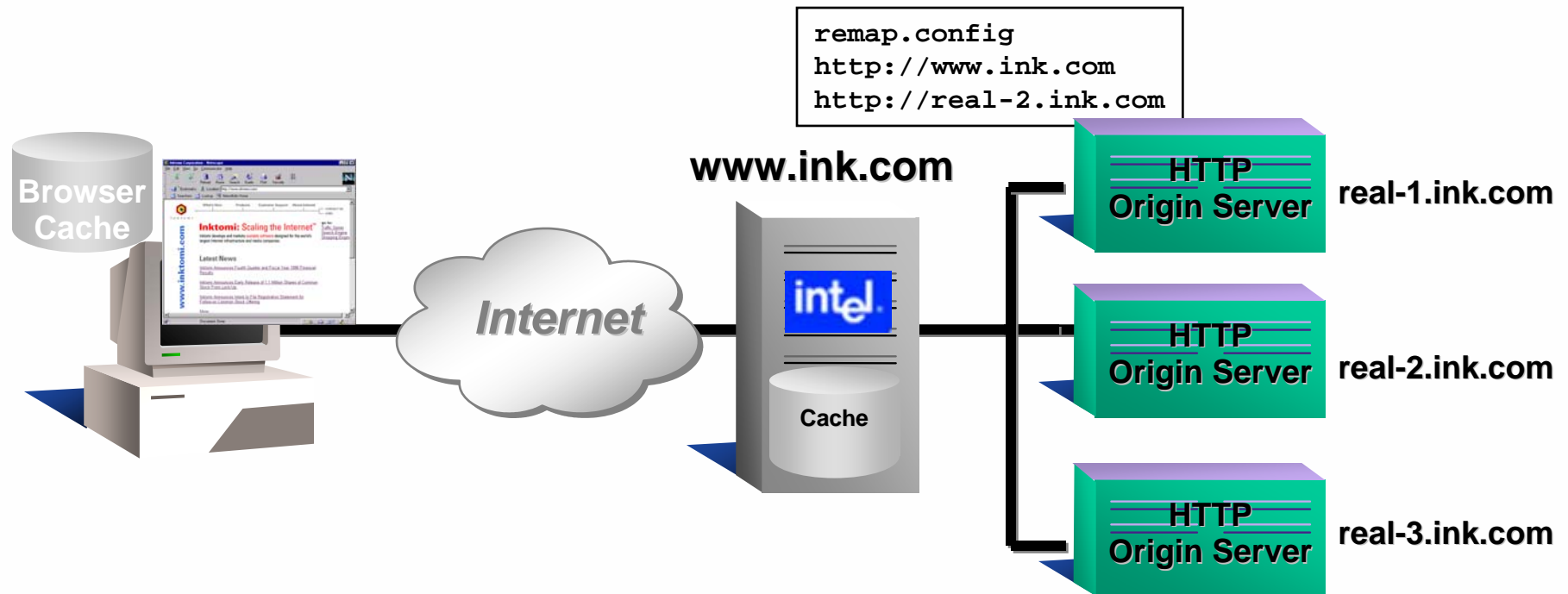
inktom i

- **Alleviates network congestion by significantly reducing the transmission of redundant Internet data**
 - lower bandwidth costs
 - more controlled network build-out
 - reduced load on critical network routers and origin Web servers
- **Increased quality of service for the end user by storing frequently requested information close to users**
- **Reduce wasteful resource consumption**
- **Opportunity for optimization and cost savings**

Reverse Proxy



inktom i



Intel Cache impersonates your web server to assume load and respond to requests that would normally be destined for your web servers

Reverse Proxy Environments



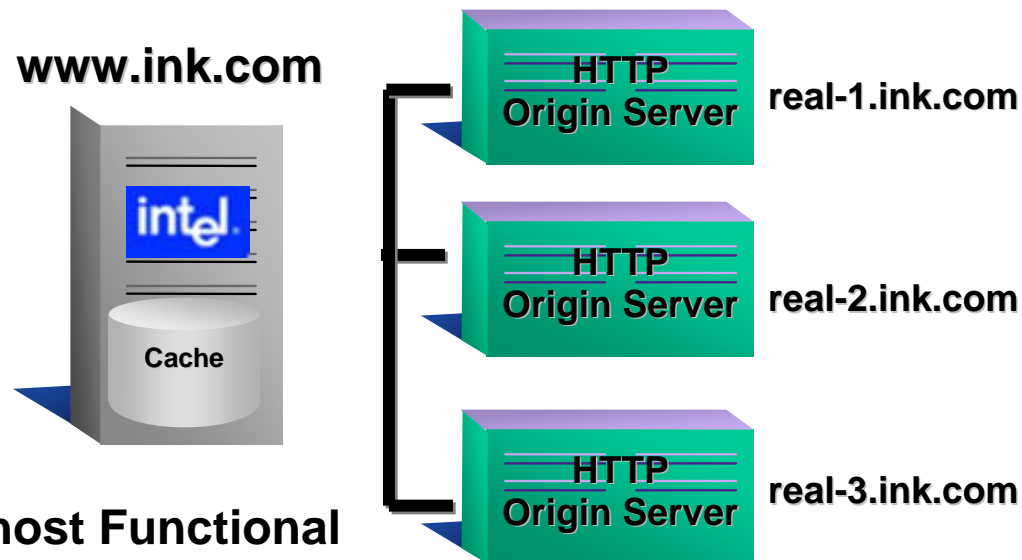
inktom i

Option 1: Multi-host Round Robin

- 1 name with 3 IP addresses
- All have same content
- Optimal for load balancing

Option 2: Multi-host Load Director

- 1 name with 3 IP addresses
- Specify which machine responds
- All have same content
- Optimal for regional web servers



Option 3: Multi-host Functional

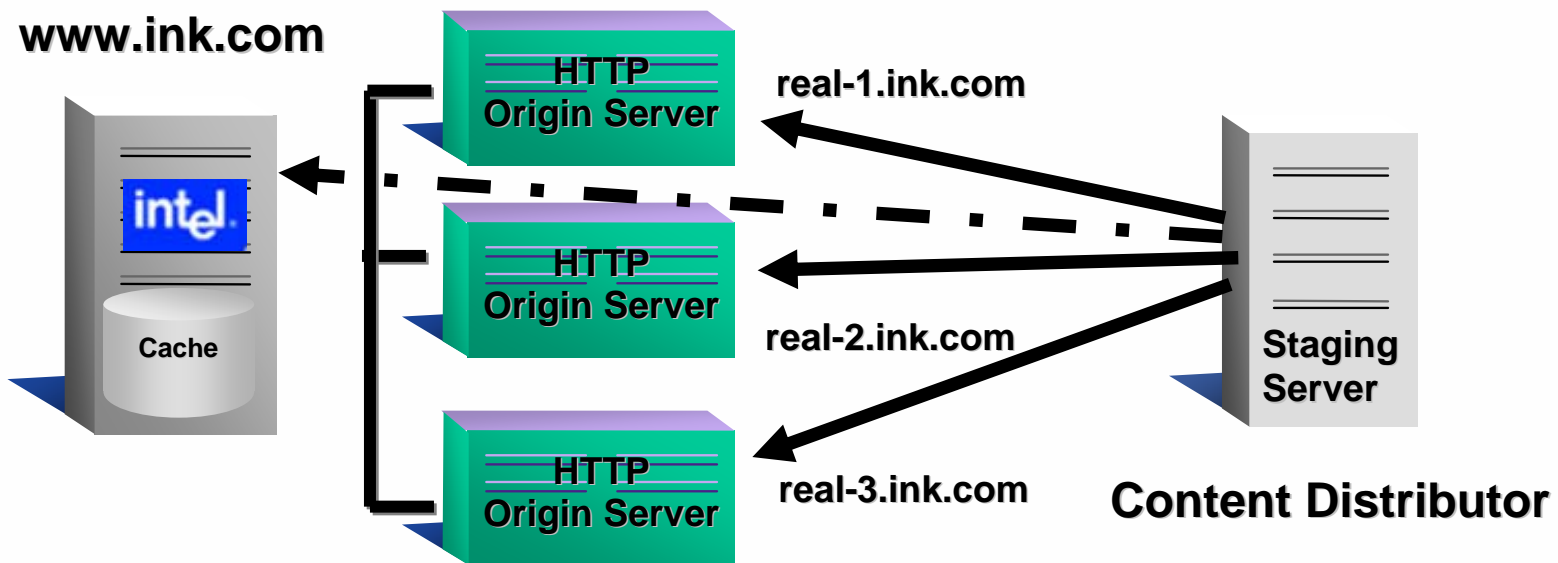
- 3 names with 3 IP addresses
- Specify which machine responds
- All have different content
- Optimal for intranets - how you do business
- support.ink.com, sales.ink.com...

Reverse Proxy & Content Distribution



inktom i

Content synchronization across origin servers and the Intel Cache can be achieved by using the Inktomi Content Distributor to replicate content



As content changes are made to the origin servers, deletes are sent to the Intel Cache

Benefit of Reverse Proxy



- **Enhanced End-User Experience**

End-users demand the highest levels of quality

- **Scalability**

Provision for event-driven and unpredictable spikes in traffic

- **Security**

E-commerce makes this feature more critical

- **Management and Reporting**

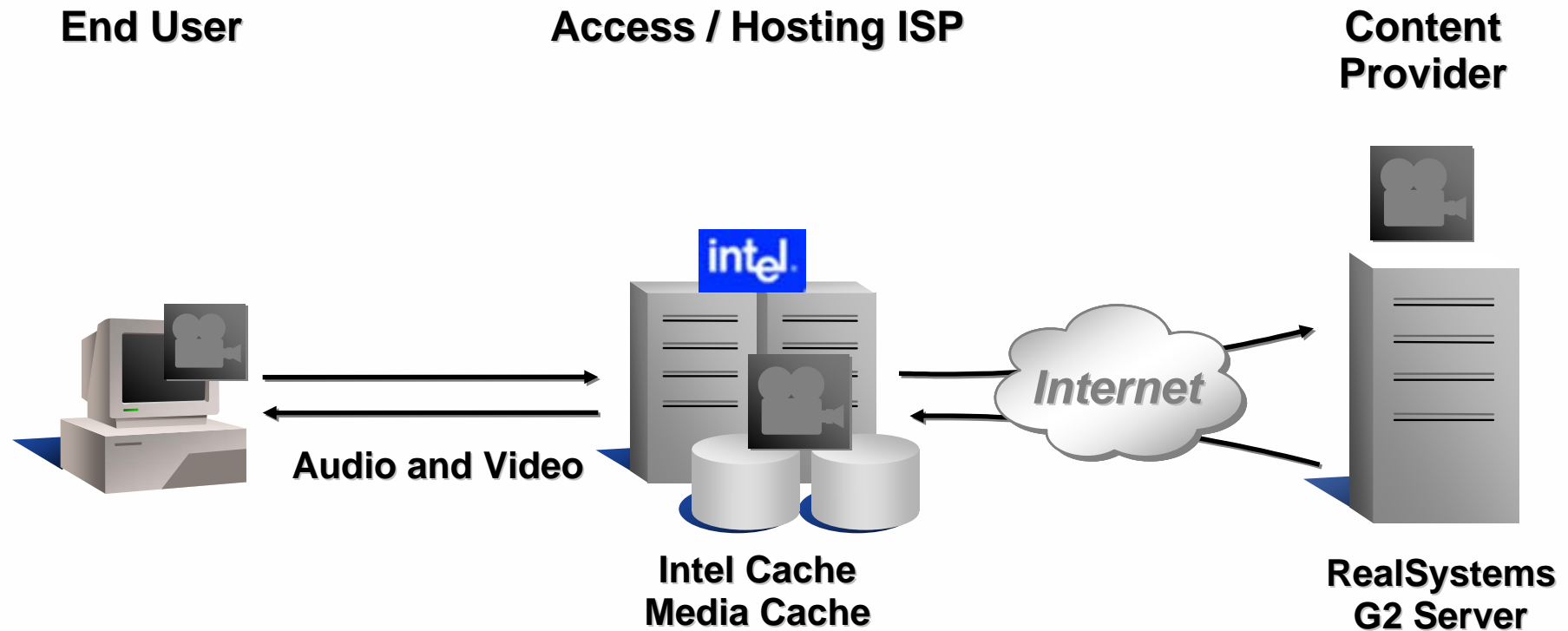
Logging information; user and usage data that will help increase profits

- **Fault Tolerance and Disaster Recovery**

On Demand with Media Caching



inktom i

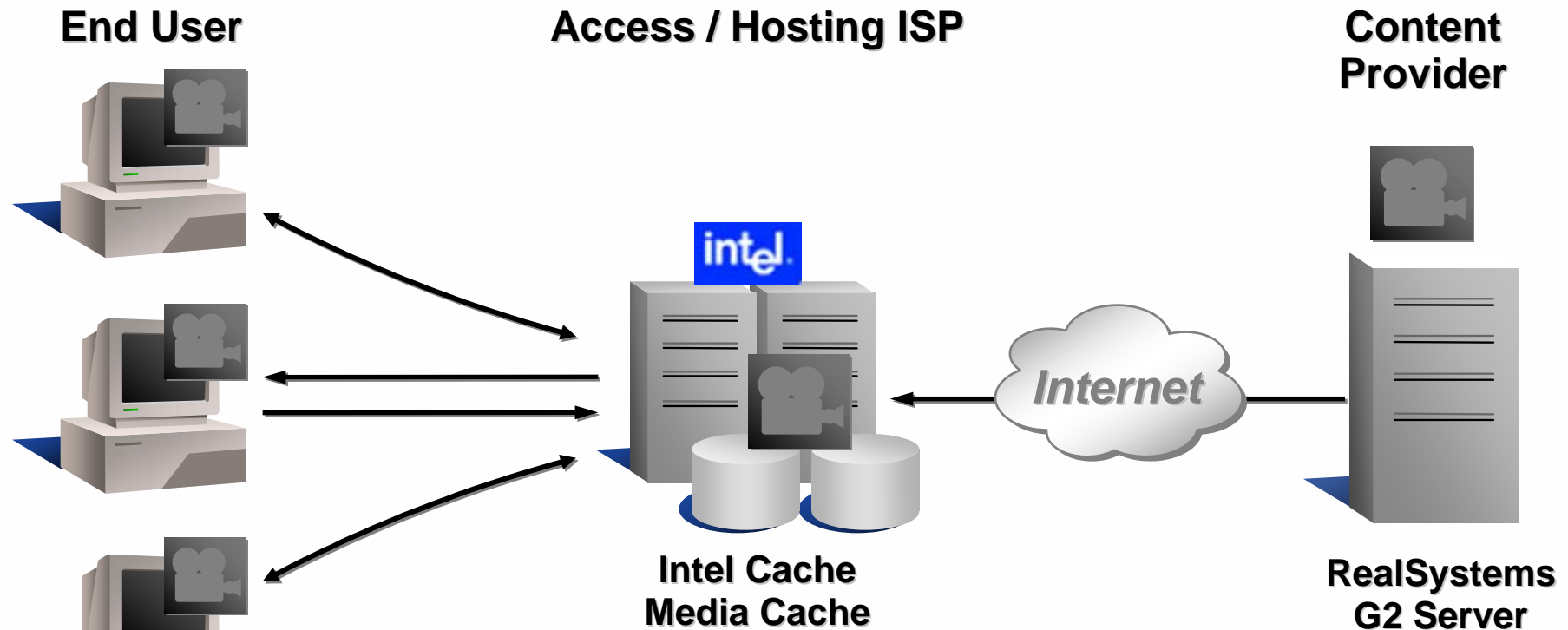


**First Request Served Directly
from RealSystems G2 Server**

On Demand with Media Caching



inktom i



Subsequent Requests Served Directly from Cache

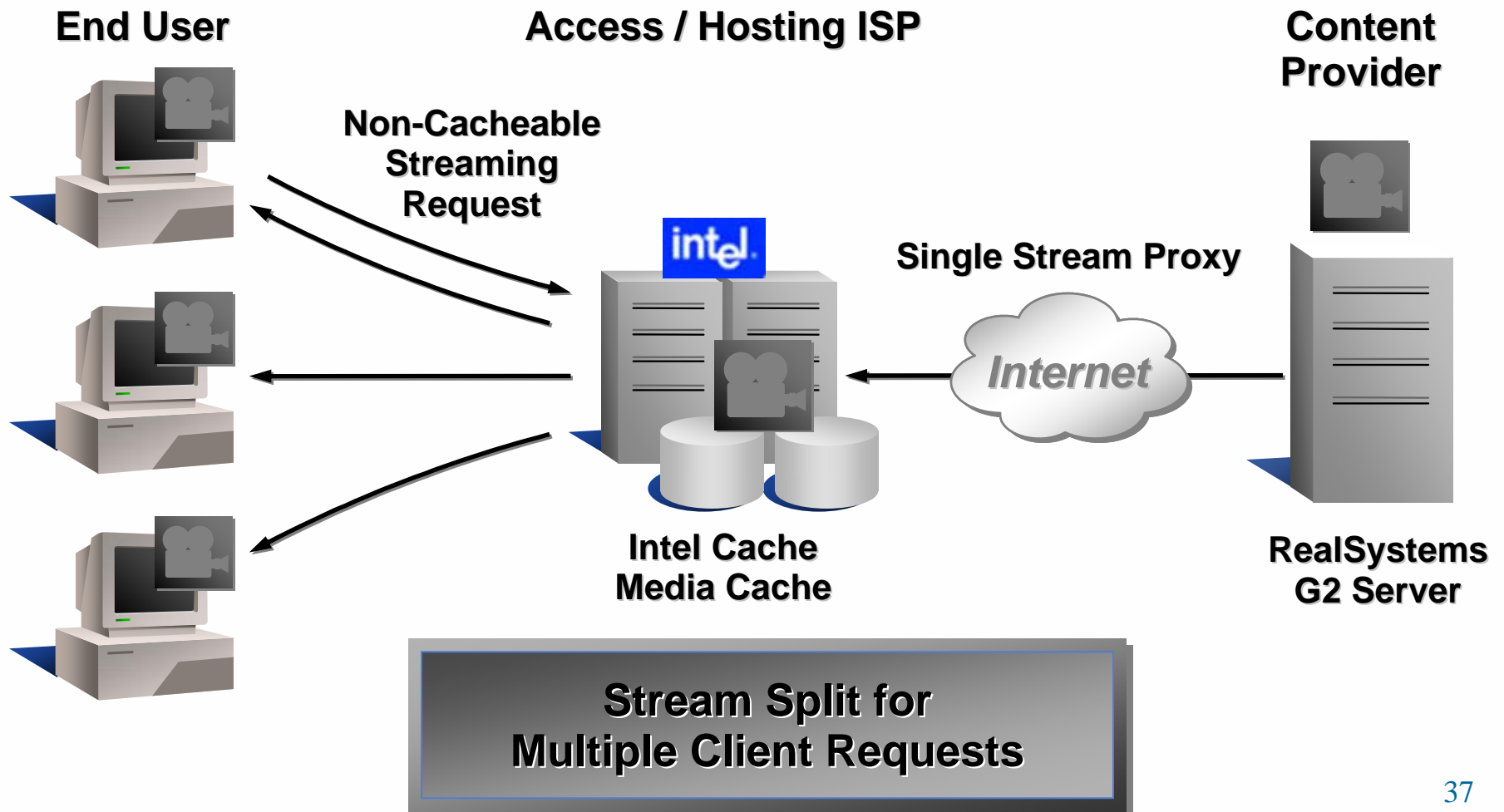
- ◆ Improving Response Time
- ◆ Reducing Bandwidth Requirements

Live Streams with Media Cache



inktom i

Intel Cache supports live streams with Media Cache Option



Benefit of Distributed Streaming Media

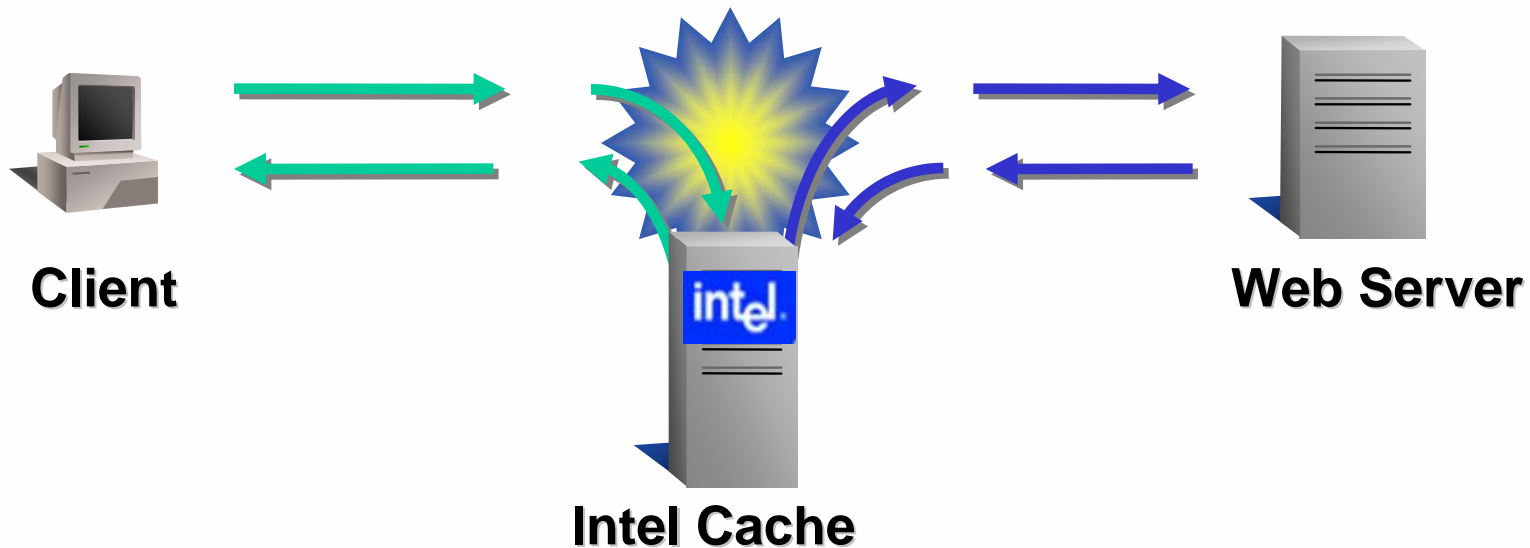


inktom i

- **Superior content delivery through dynamically mirrored content**
- **Decreased server and network loads**
- **Minimized latency**
- **Improved quality of streaming media**
 - Higher bit-rate video with Intel Cache
 - Enhanced image resolution
 - Enhanced audio reproduction
- **RTSP - Real Time Streaming Protocol**

Caching Summary

Intel Cache provides multiple solutions for intercepting and caching Internet traffic

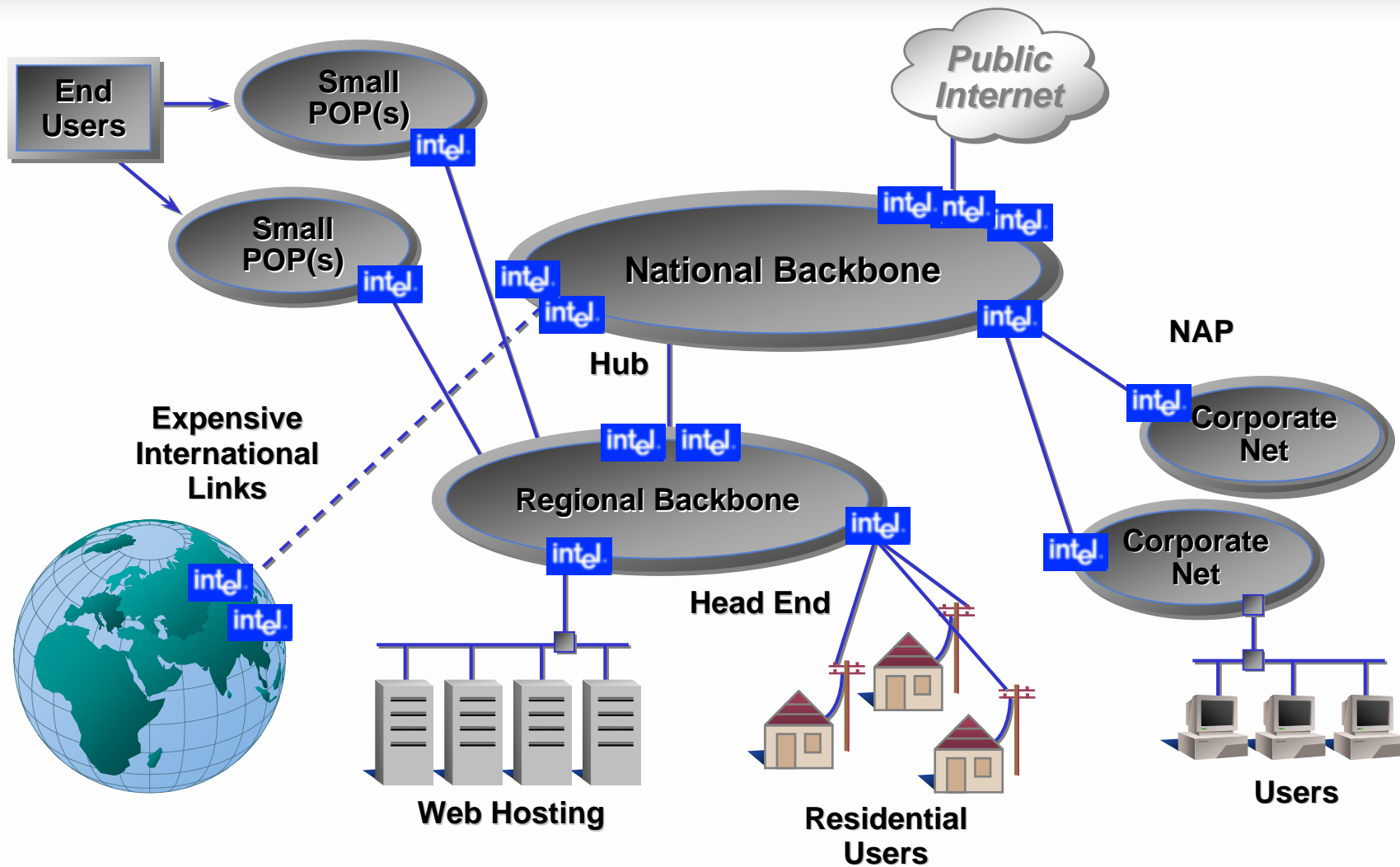


Creative configuration can address many unique problems

Meets Any Network Needs



inktom i



Review Questions:



inkomi

- What are the benefits of caching?
- What is the difference between forward and reverse proxy?
- What are the benefits of transparent caching?
- How can port traffic be identified in a network?



- **Please Complete Lab #1: Configuring the Netscape Browser**
 - Set a home page
 - Configure proxy settings
 - Configure disk cache and memory cache settings
 - Verify the new browser configuration settings
 - Answer review questions

Intel Cache: Overview



inktom i

Chapter Overview



inktom i

- **Intel Cache Releases**
- **Intel Cache Applications**
- **Extensions**

Where is it?

How do I get it?

Portal Services

Search
Engine

Directory
Engine

Shopping
Engine

Network Products

Streaming
Media

Filtering

Virus
Protection

Transformation

Analysis and
Reporting

...

Intel Cache

Content
Delivery
Suite

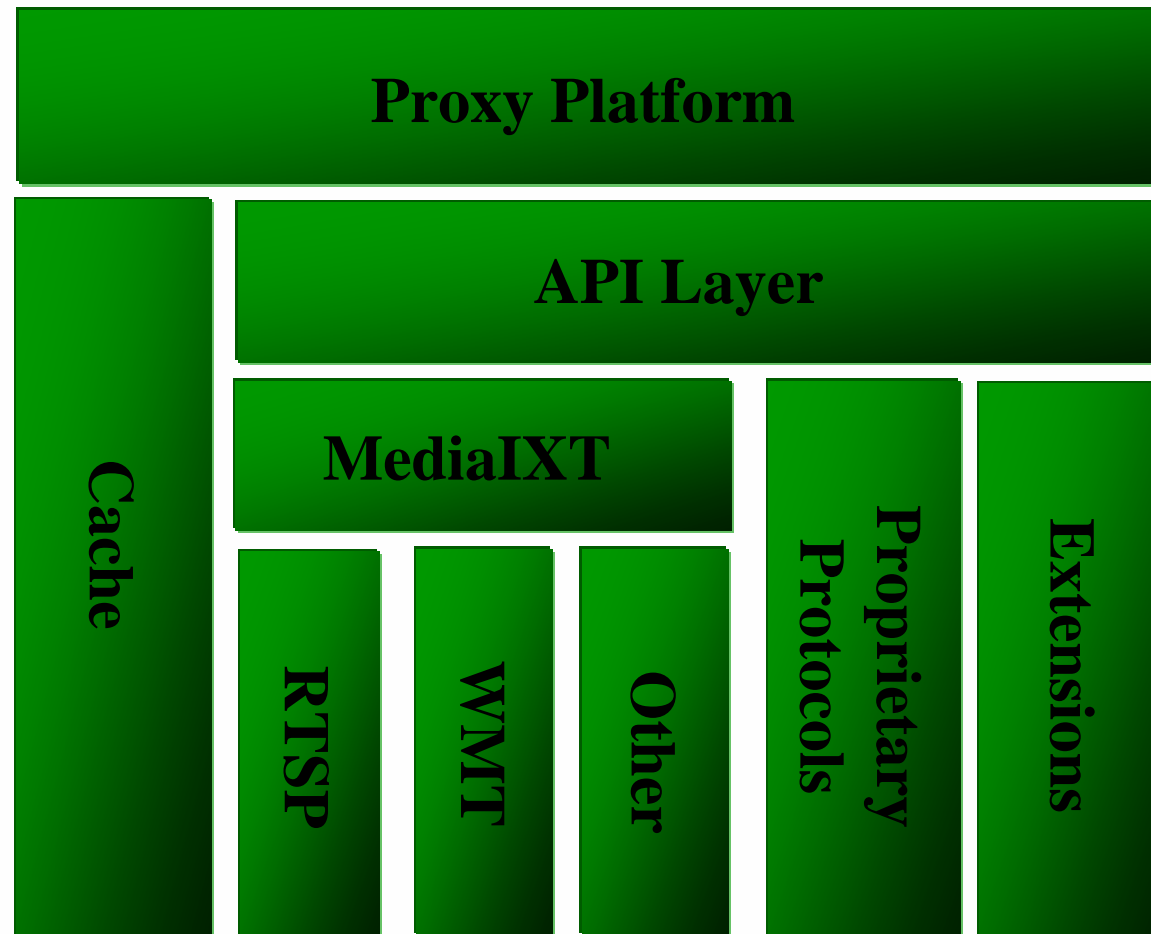
Core Technology

1. Coupled Cluster
2. DataFlow
3. Concept Induction

Intel Cache - An Extensible Platform for Rich Media



inktom i



Intel Cache Partners and Services



inktom i

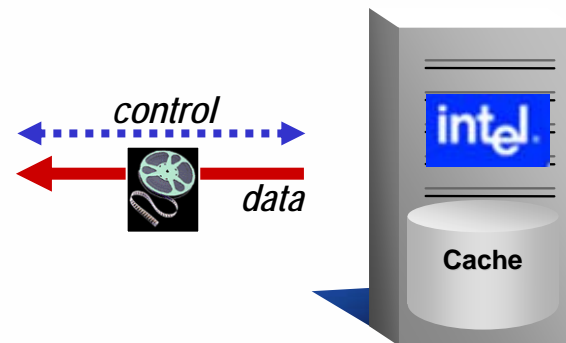
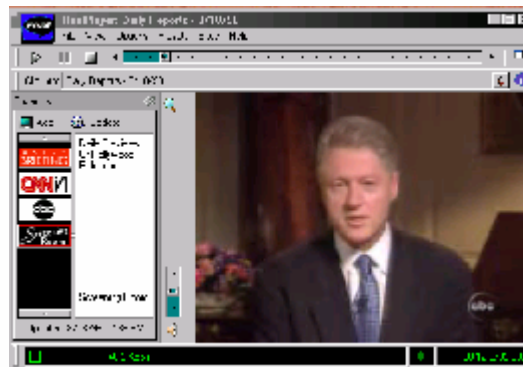
- **Media Delivery**
 - RealNetworks
- **Content Distribution**
 - WebSpective (Inktomi)
 - Sandpiper
- **Interactive Applications**
 - Arepa
- **Content Transformation**
 - Spyglass Prism
- **Filtering**
 - NetPartners WebSENSE

Extension - Media Delivery



inktom i

- **Intel Cache caches RealVideo and RealAudio**
 - Service providers can offer content-on-demand
 - Real-time streaming of live events
 - Value-added broadcast networks that provide better end-to-end bandwidth through managed replicas

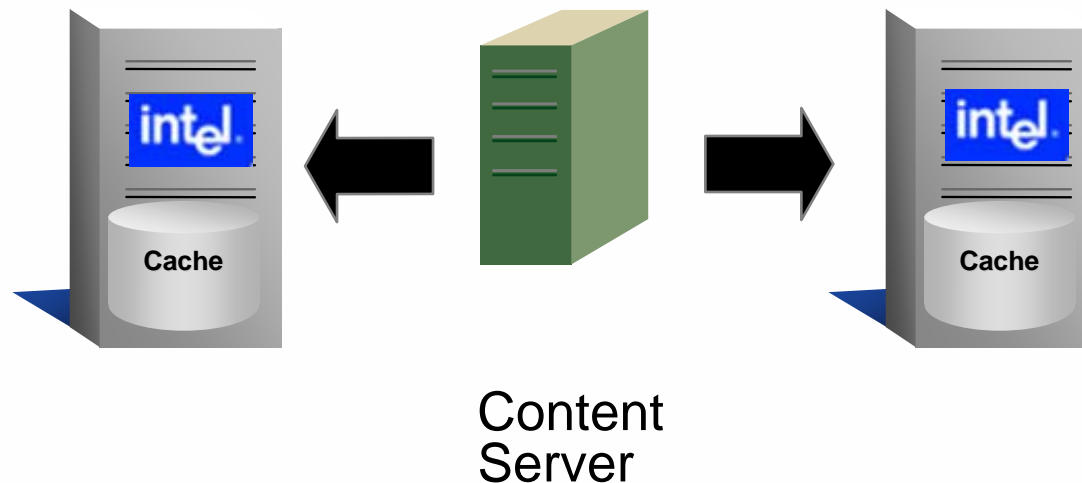


Extension - Content Distribution



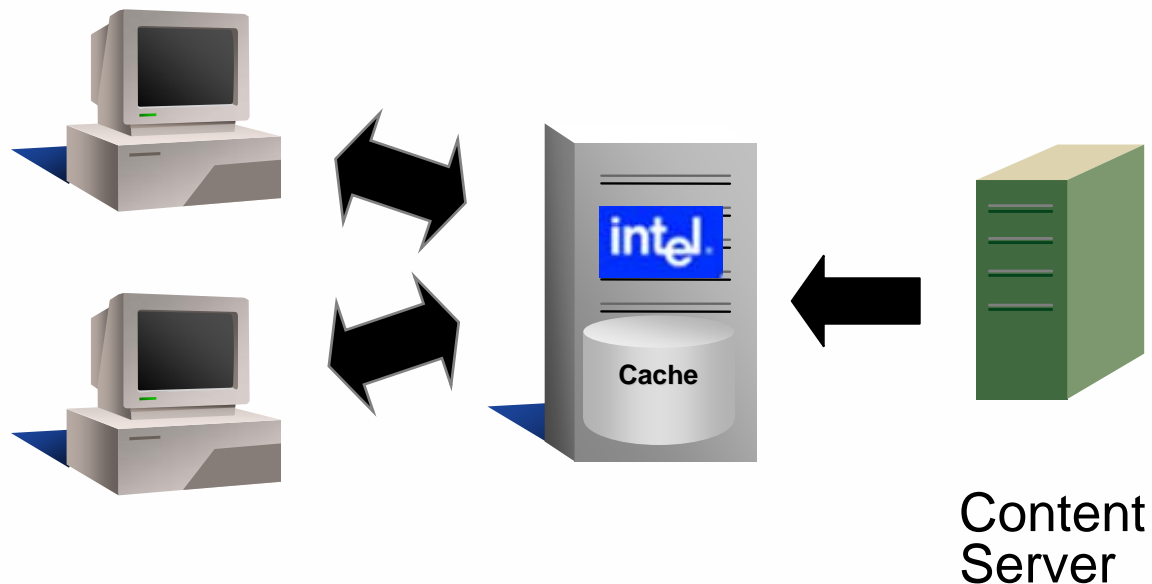
inktom

- **Intel Cache caches product or service, to guarantee availability and freshness**
 - Forced expiry
 - Central management with less “tag” hassle
 - Synchronized content updates



Extension - Interactive Applications

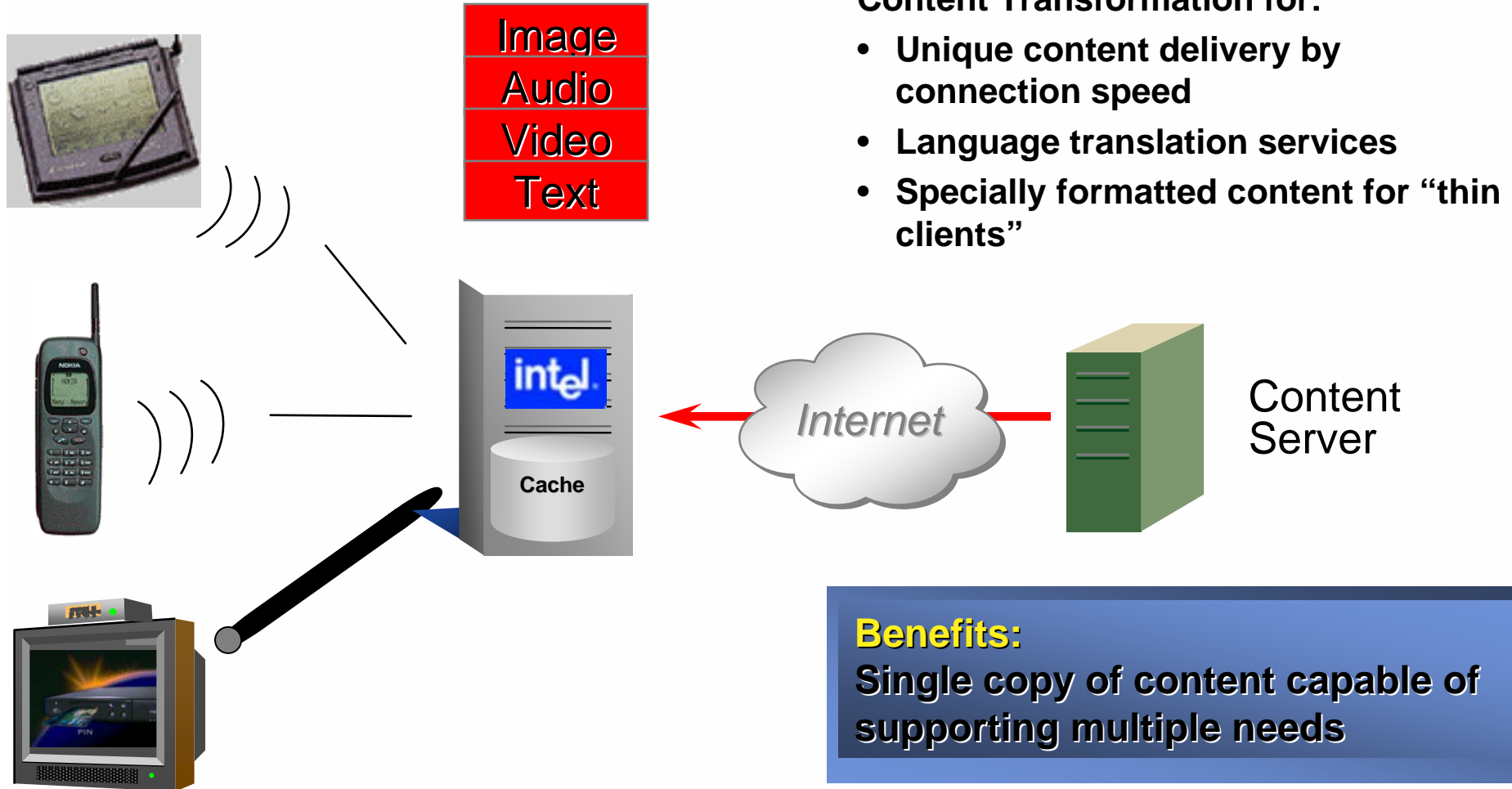
- **Intel Cache caches rental software**
 - Real-time, click and use, by -the-minute charging
 - Focused set of applications, broken up into small chunks
 - “Killer Ap” for broadband: think arcade games online...



Extension - Content Transformation



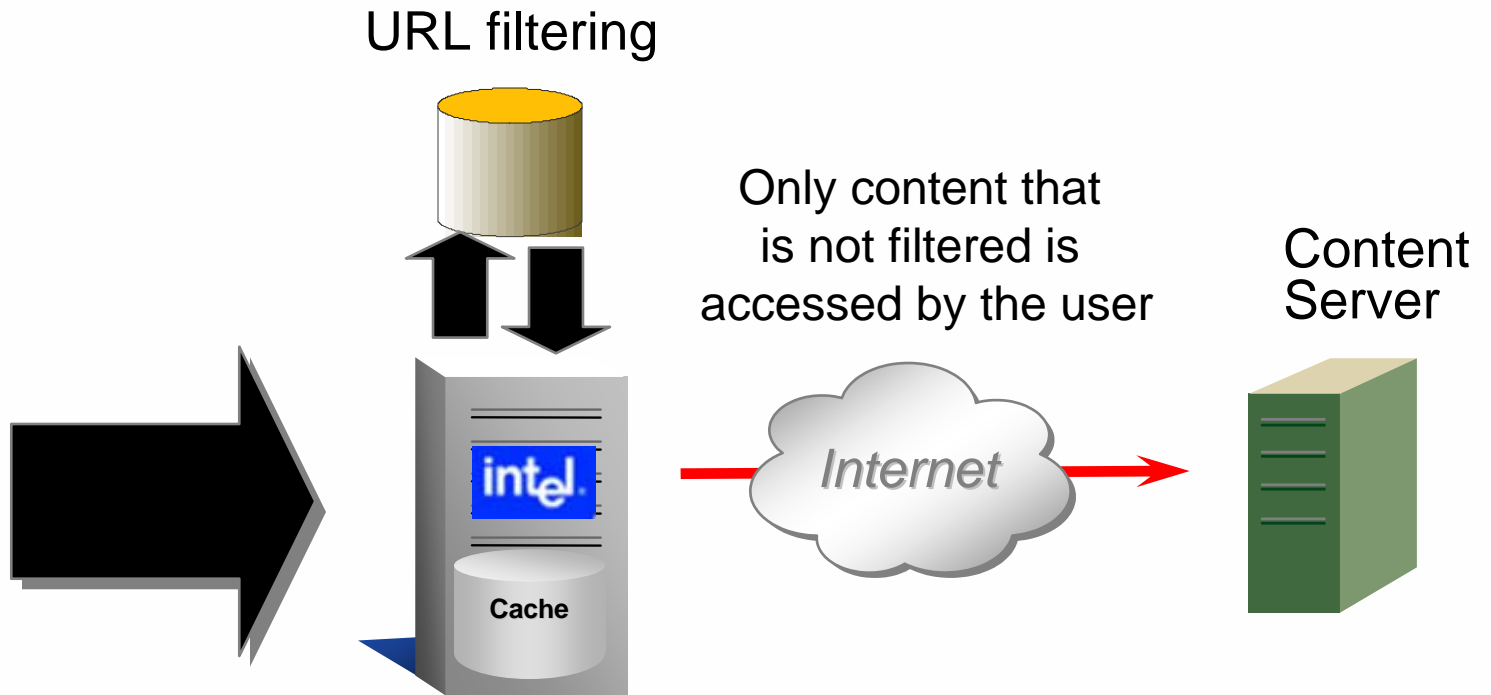
inktom i



Extension - Filtering



inktom i



Benefits:

- Incremental Revenue for Service Providers
- Restricts content to appropriate users

Review Questions:



inkomi

- What types of environment would extensions be beneficial?

Intel Cache: Installation



i n k t o m i

Chapter Overview



inkomi

- **Place holder for Installation module to be finished**

Intel Cache: Features and Benefits



i n k t o m i

Chapter Overview



inktom

- **Deployment Environments**
- **Protocols**
- **Security**
- **Monitoring Tools**
- **Browser User Interface**
- **Command Line Interface**

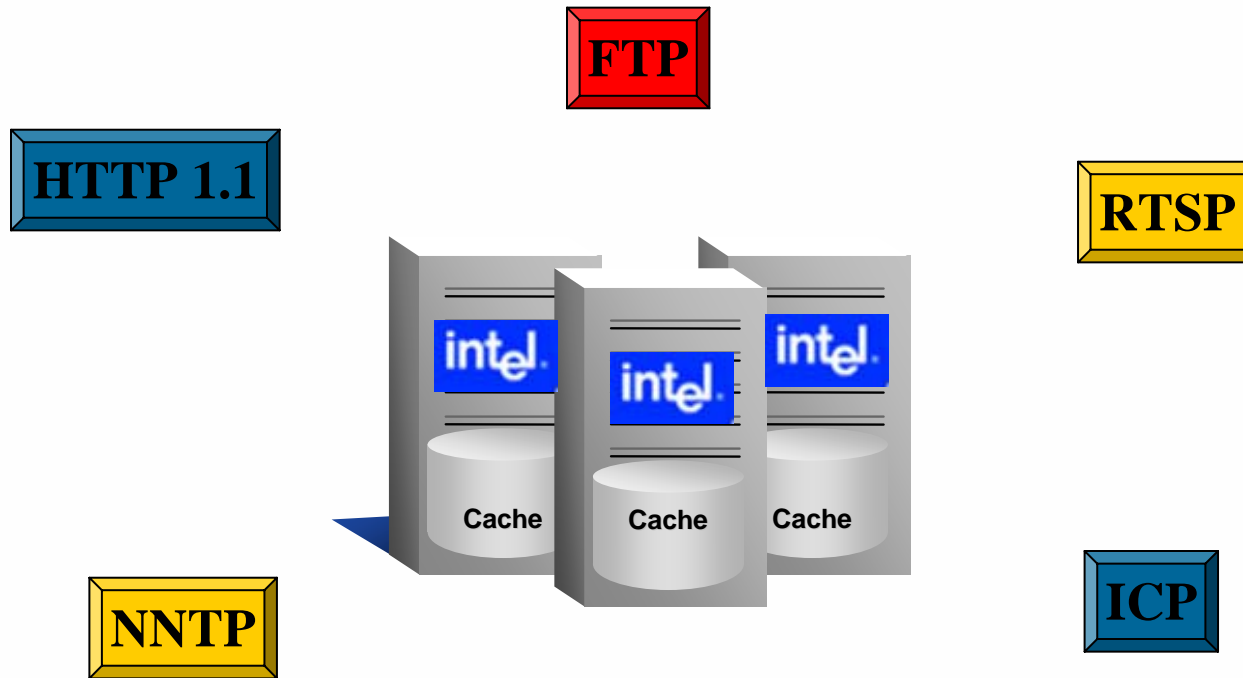


- **Transparent Caching Solutions (Forward Proxy)**
 - Layer 4 Switches
 - WCCP
 - WPAD
 - Policy-Based Routing
- **Reverse-Proxy Capability**
 - Before server farms
 - Edges of hosting network
 - Distant points

Protocol Support



inktom i



Manageability and Ease of Use



inktom i

- **Browser-Based Graphical UI**
 - Full cluster with a single point administrator interface
 - Extensive real-time statistics and graphical analysis
 - Configurable cluster-wide alerts and alarms
- **Command Line Based Configuration**
- **Installation and Logging Facilities**
- **Simple Network Management Protocol (SNMP)**
- **Configuration “snapshots” allow you to capture a set of configuration files**
 - In less than a minute you could restore an archived configuration
 - You can toggle between configurations for “what if” tuning

Configuration Management



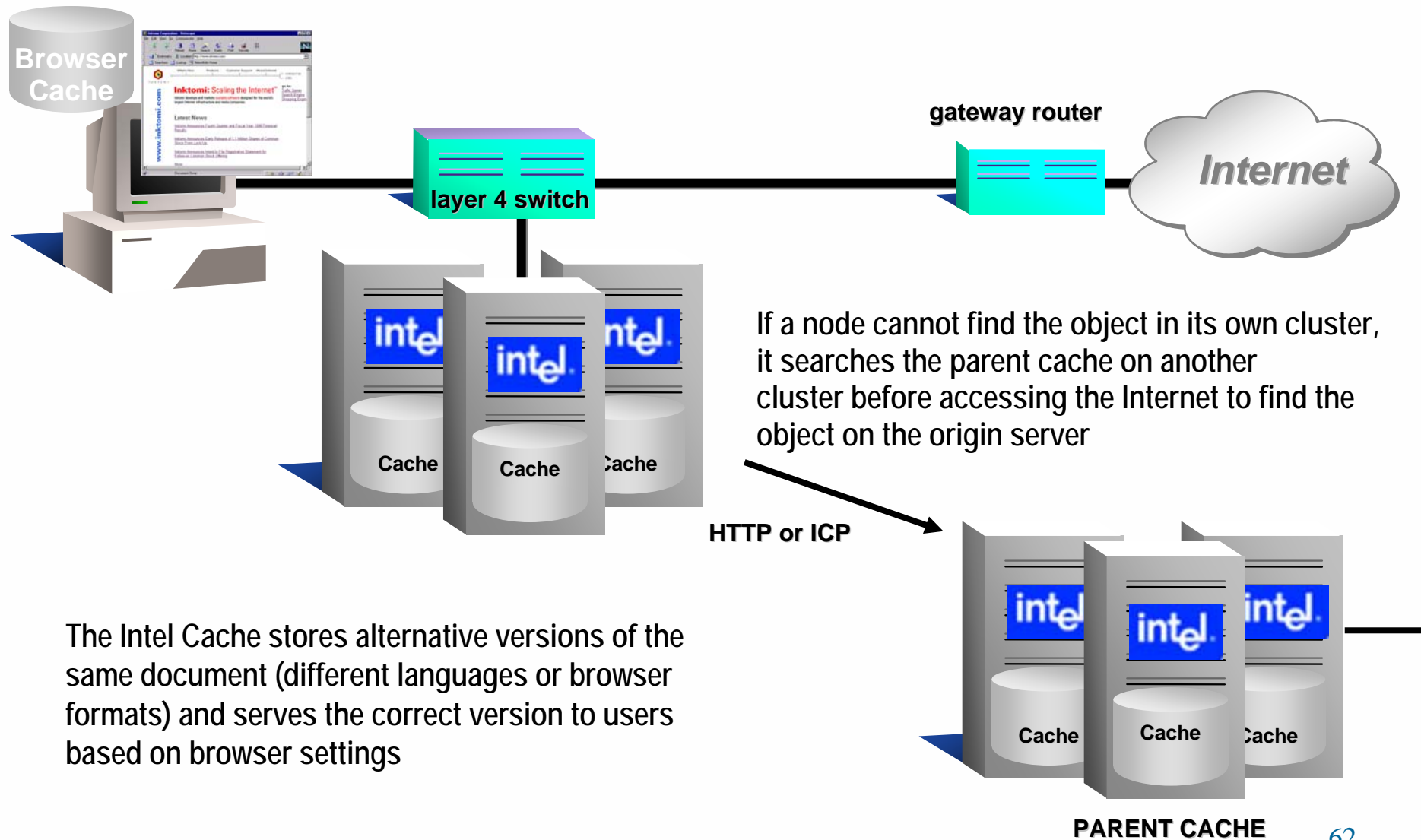
inktom i

- **Since Intel Cache is composed of “look-alike” nodes, you can**
 - Easily add one or more nodes
 - Bring nodes up and down for maintenance
 - Remove a node

Hierarchical Caching



inktom i

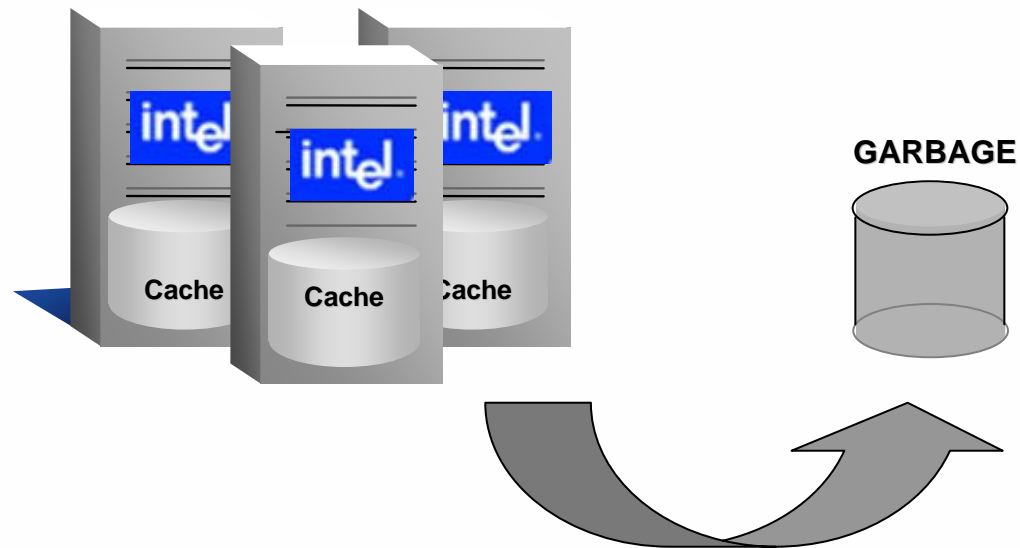


The Intel Cache stores alternative versions of the same document (different languages or browser formats) and serves the correct version to users based on browser settings

Maintaining Current Information



inktom i



Sophisticated Garbage Collectors and Auto-Deletion functions remove stale data

Flexible Logging



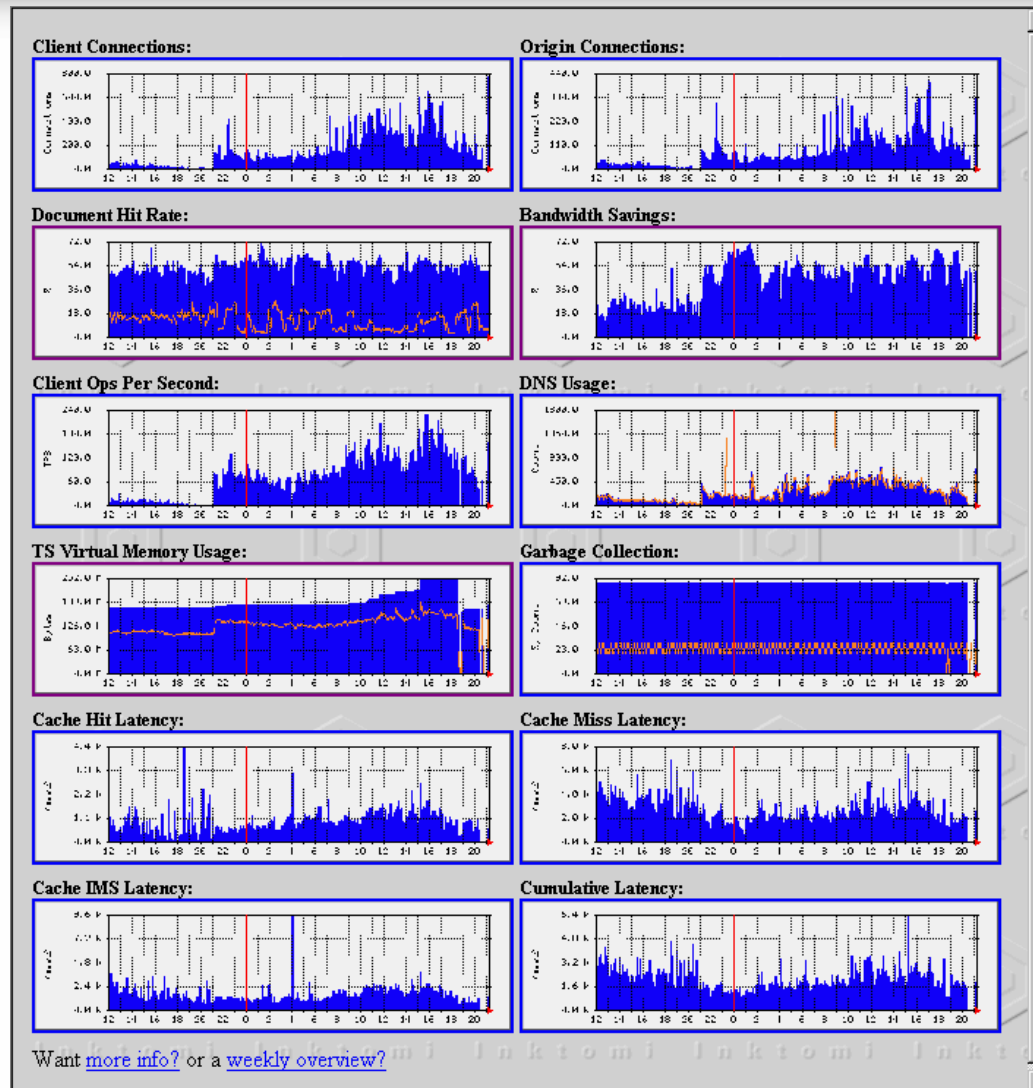
inktom i

- **Intel Cache provides a powerful logging system to meter and record network accesses:**
 - Provides information about:
 - Every user request handled
 - All Intel Cache Processes
 - All error conditions detected
 - Log size is limited to 200MB:
 - If the logging directory grows to over 190MB, logging is suspended
 - Users can specify that logs are automatically moved from the Intel Cache to an FTP server at regular time intervals
- **Intel Cache logs are in Squid format, which can be analyzed by several common tools**

MRTG - Multi Router Traffic Grapher



Inktomi



Each data point corresponds to the execution of a perl script, once every 5 minutes.

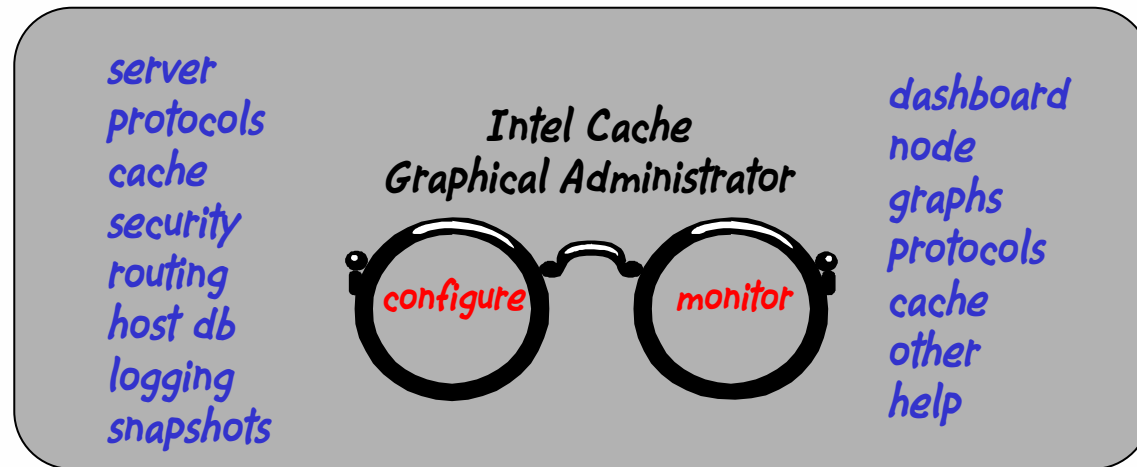


- **The Intel Cache secures access to the Traffic Manager**
 - Authentication on or off - provide ID and password
 - Guest ID allows Monitoring access of the Traffic Manager UI
- **The Intel Cache provides a tunnel to supports Client and Web Server communication using SSL**
 - Does not cache or examine encrypted data

Graphical Administration



Intel

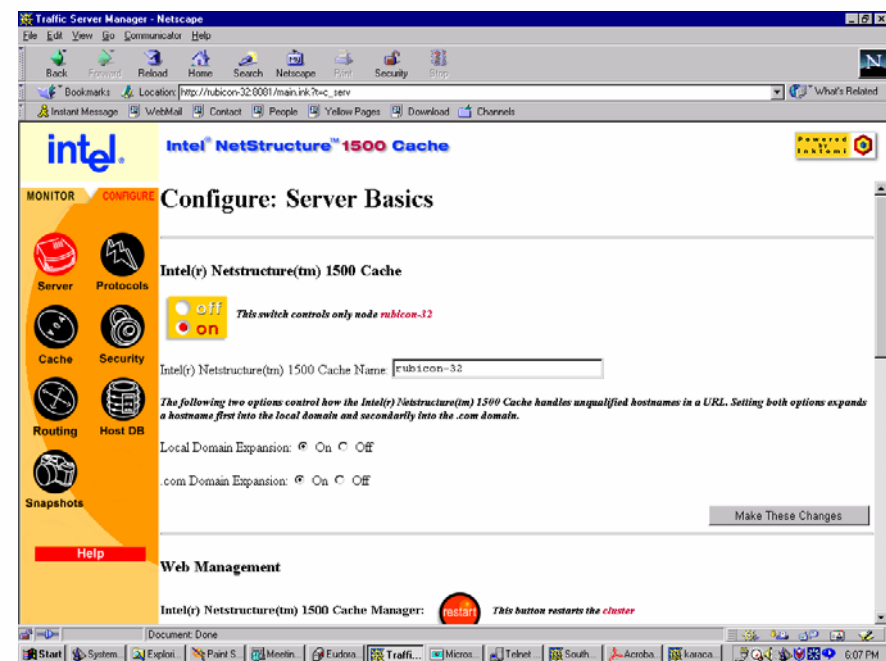
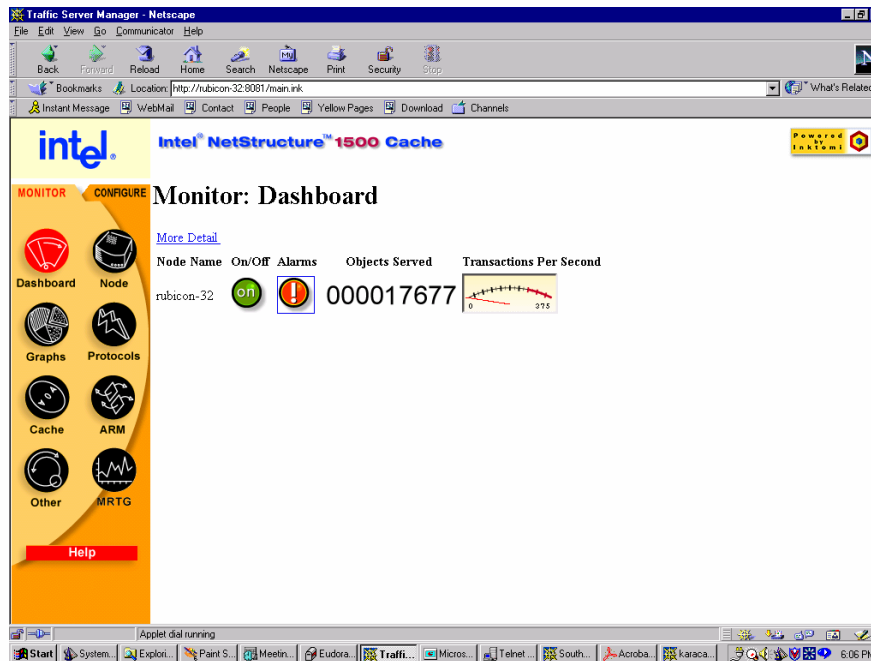


- Provides secure *single-point* administration
 - Configure, monitor and tune features and services
 - Encrypted remote administration
 - Efficient for clusters

Traffic Manager UI



The Intel Cache provides a series of pre-defined tools and utilities to manage your nodes and clusters



http://<host name>:<admin port>

Monitoring Performance



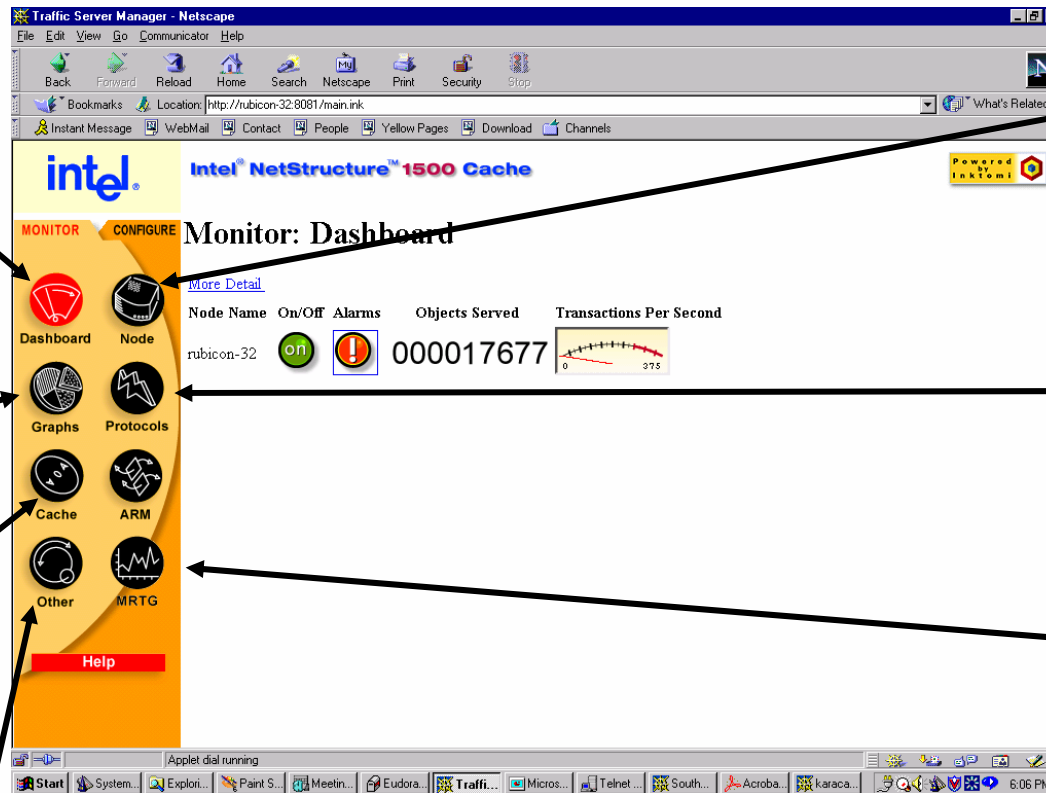
inktom i

Displays Nodes On and Off, alerts lights, depicts loads, and number of objects served

Node-specific performance graphs for cache results, garbage collection, transfer rates, and object size

Node-specific statistics for the cache: connections, transactions, objects reads and writes, and document hits and misses

Node-specific statistics for logging, DNS lookups, and SOCKS



Display statistics for: single node, links, document hit rate, cache %free, and display graphs for all cluster nodes

Cluster-wide statistics for HTTP, FTP, NNTP, and ICP

Multi Router Traffic Grapher graphs Intel Cache performance over a period of time

the Traffic Manager process monitors all Intel Cache activities and reports performance through a series of browser pages

Configuring the Intel Cache



inktom i

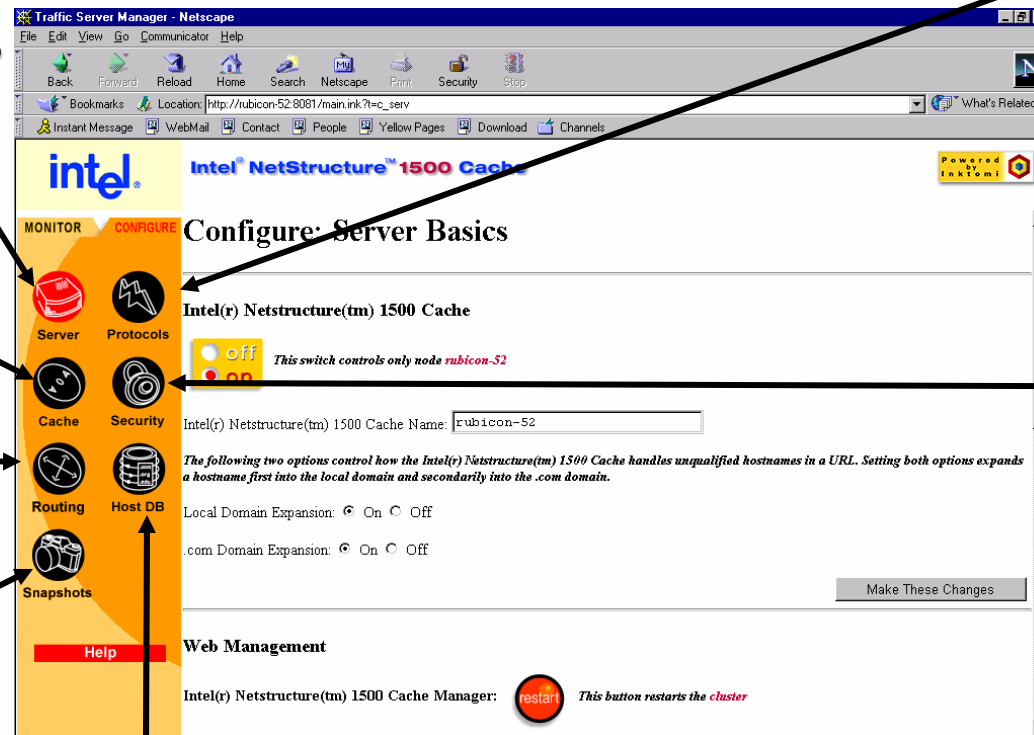
Turn Intel Cache On and Off, restart or reconfigure Traffic Manager, auto-configure browser, and enable SNMP

Configuring HTTP, NNTP, HTTPS, and FTP

Cache activation, document freshness, and document content

HTTP parent caching, ICP support, server acceleration, and transparency

Save and restore configurations



Controlling access to the UI, set admin and guest Ids and passwords

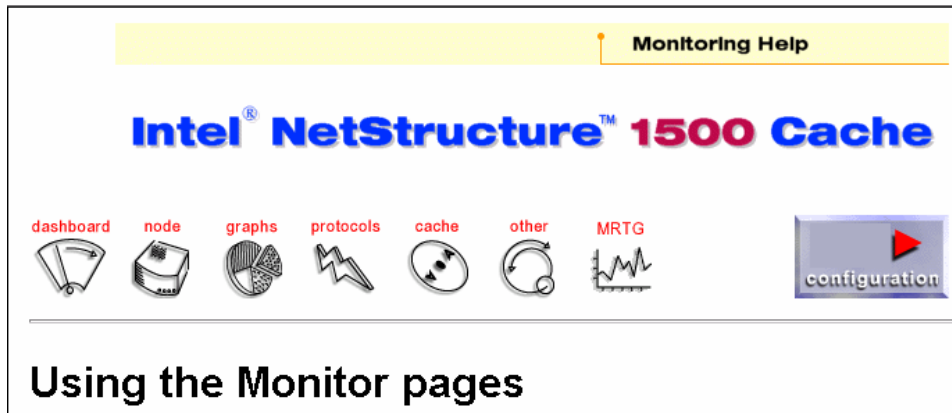
Host database options and DNS lookups

Default (recommended) configuration values are assigned during installation

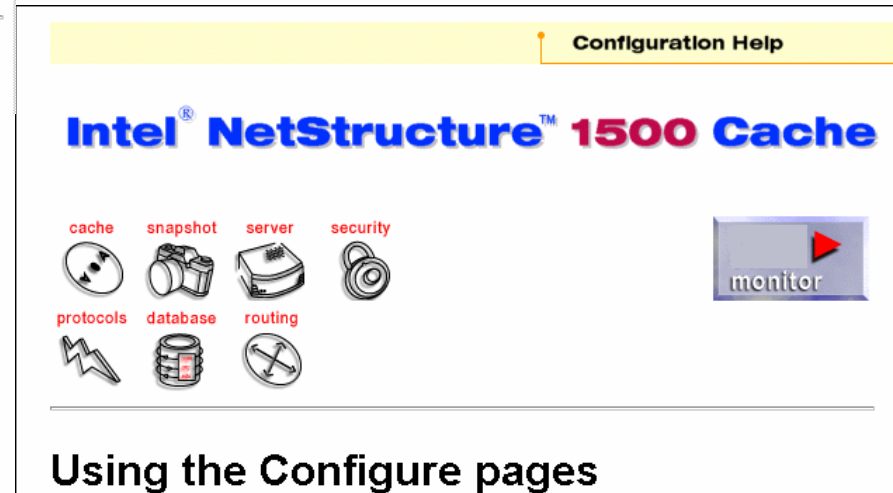
Traffic Manager Help



The Traffic Manager UI provides help for both modes,
monitor and configure



Using the Monitor pages



Using the Configure pages

Online HELP allows you to learn more about particular pages in the Traffic Manager

Command Line Interface



inktom i

- Performs many of the same tasks as the Traffic Manager UI
- There are some tasks which can only be done by the Traffic Manager UI
- There are some tasks which can only be done by the Command Line Interface

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----
Intel(r) NetStructure(tm) 1500 Cache
-----

-----root-----
|-setup  Initial Intel Cache Setup
|main   Main Intel Cache Controls
|config Intel Cache Configuration
|monitor View Statistics
|expert  Enter Expert Mode
|save   Save Config To Floppy
|load   Load Config From Floppy
-----

Initial Intel Cache Setup
```


Review Questions:



inkomi

- What protocols can be cached by Intel Cache?
- What two ways can one modify Intel Cache configuration?

Intel Cache Reverse Proxy



inktom i

Chapter Overview



inktom

- **Benefits of Reverse Proxy**
- **Deployment Environments**
- **Installation Tips**
- **Practice Lab**

Reverse Proxy



inktom i

- **Caches store the most frequently accessed data and are optimized to serve data quickly**
- **Reduce traffic load off the origin server**
- **Increased Quality of Service**
- **Guaranteed Service Levels**
- **Incremental Capacity and Surge Protection**
- **Mirror Sites**

Reverse Proxy Deployment



inktom i

- **In front of server farms**
 - servers to serve content requests
 - alternative to the need for complicated load-balancing solutions
- **At edges of hosting network**
 - decrease the number of router hops required to serve content
- **At distant points**
 - efficient alternative to replication at mirror sites

Web Server Redirects



inktom i

- Intel Cache uses reverse mappings to prevent redirects from origin servers to cause clients to bypass the Intel Cache
- There should be a reverse map rule for every map rule with the origin URL and the replacement URL reversed

```
map          / http://real.hopalong.com /
map          http://www.hopalong.com/
reverse_map http://real.hopalong.com/
```

*maps incoming requests
lacking a host: header*

Reverse Proxy Setup



inktom i

- **A Intel Cache and a Web Server**
- **DNS entries**
- **Remove the existing Intel Cache on the first system**
- **Install Intel Cache with Reverse Proxy**
- **Configure the Reverse Proxy remap rules through the Routing page.**
- **Verify Reverse Proxy mapping rules are working properly by browsing with both advertised URL and real origin server URL**

Review Questions:



inkomi

- What are the benefits of reverse proxy?
- What environment would you install reverse proxy?



- **Please Complete Lab #3: Reverse Proxy**
 - Remove the current Intel Cache installation
 - Install the Intel Cache
 - Start the Intel Cache
 - Configure Netscape browser
 - Verify Reverse Proxy configuration
 - Setup Reverse Proxy remap rules
 - Answer review questions



- **Please Complete Lab #3: Reverse Proxy**
 - Remove the current Intel Cache installation
 - Install the Intel Cache in a Forward Proxy
 - Start the Intel Cache
 - Configure Netscape browser (80 ⇔ 8080)
 - Populate the Intel Cache cache with Synthetic Data
 - Setup and view MRTG
- **Please See Lab 2 for Detailed Instructions**

Intel Cache: Processing Requests



inktom i

Chapter Overview



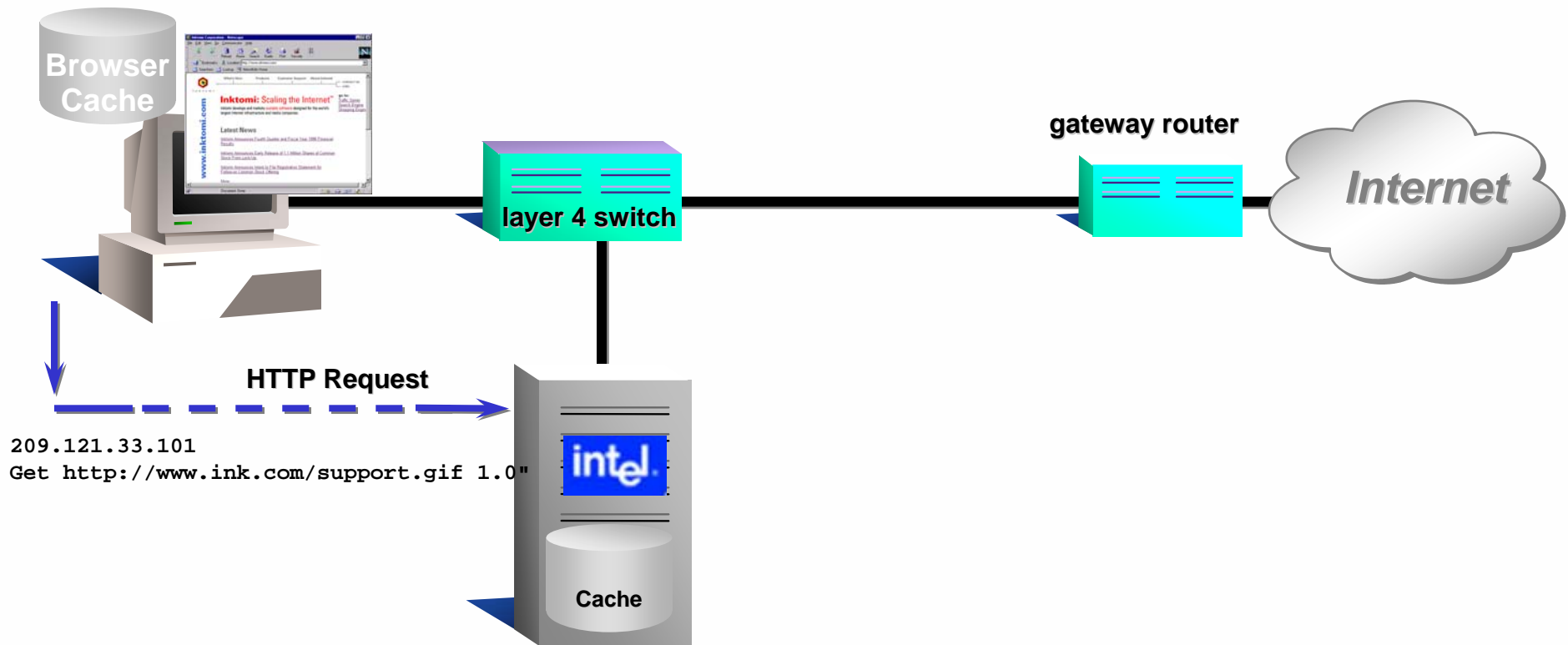
inkomi

- **Cache Hit**
- **Cache Miss**
- **Benefits of Forward Caching**

Client Request



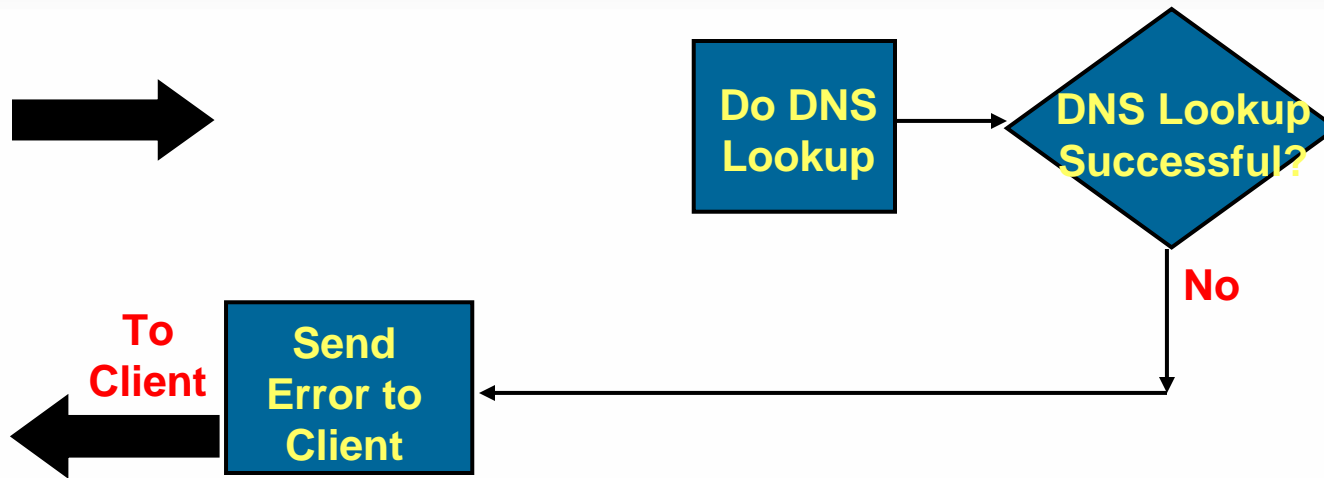
inktomi



Client Request



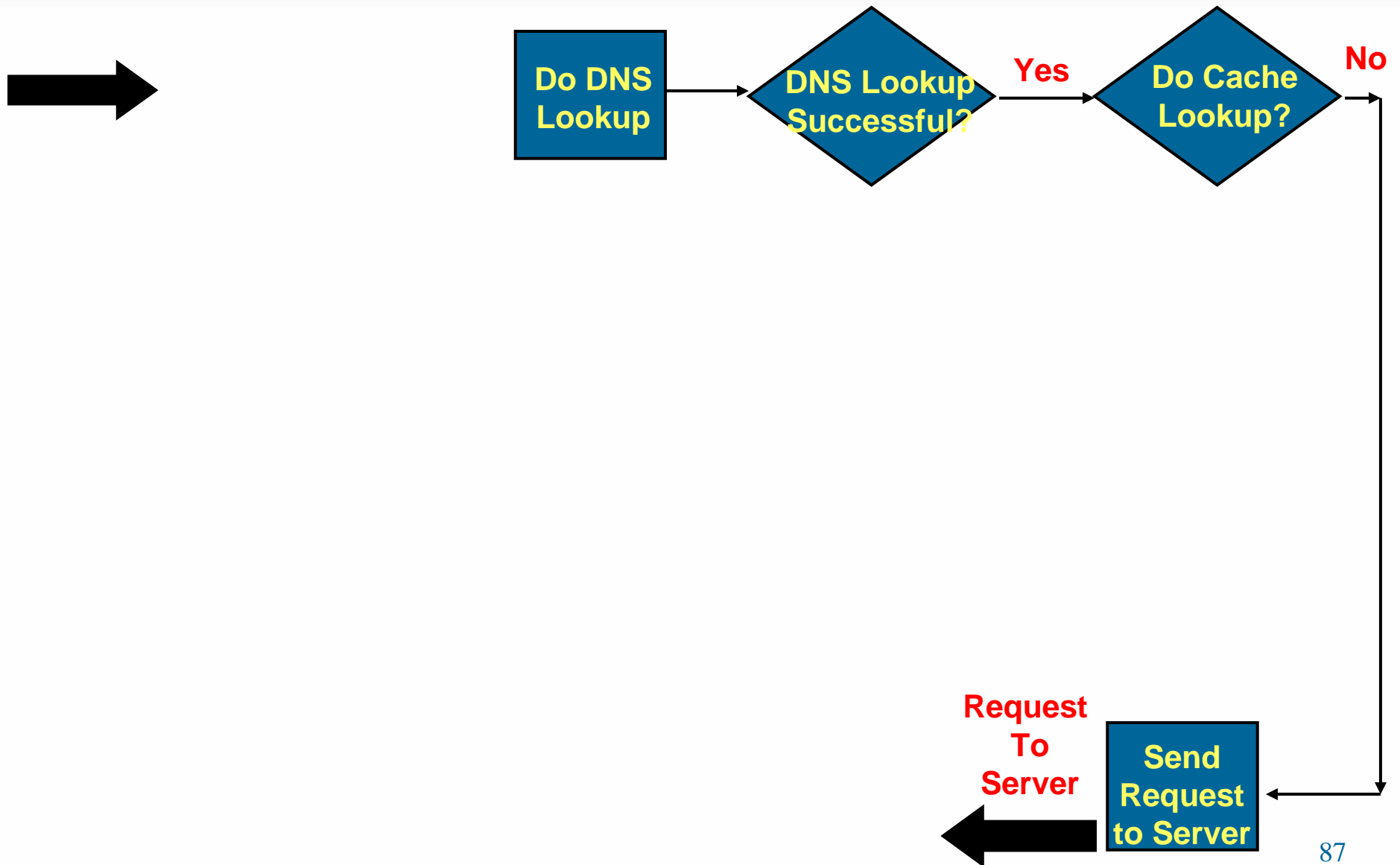
inktom i



Client Request



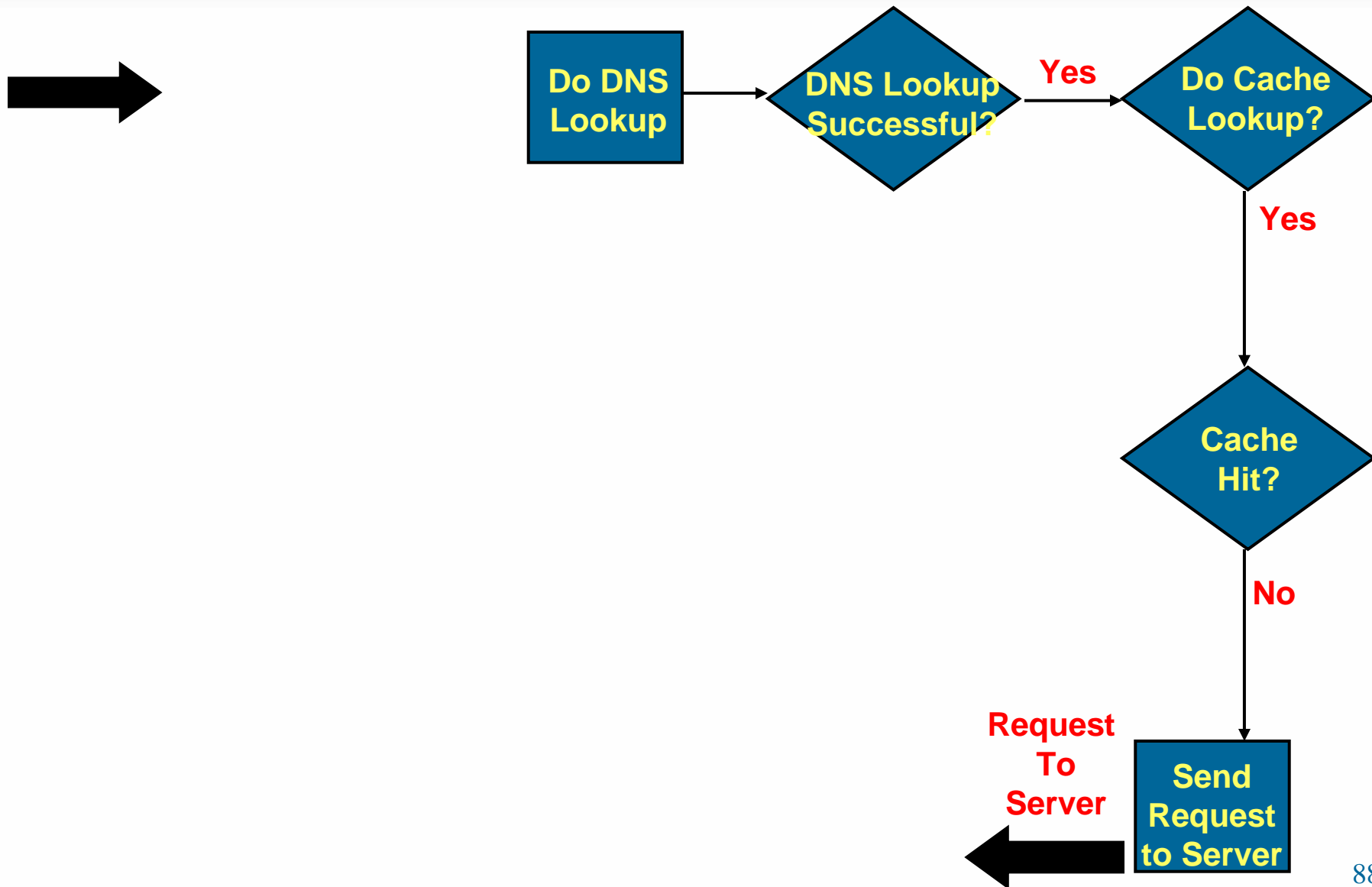
inkomi



Client Request



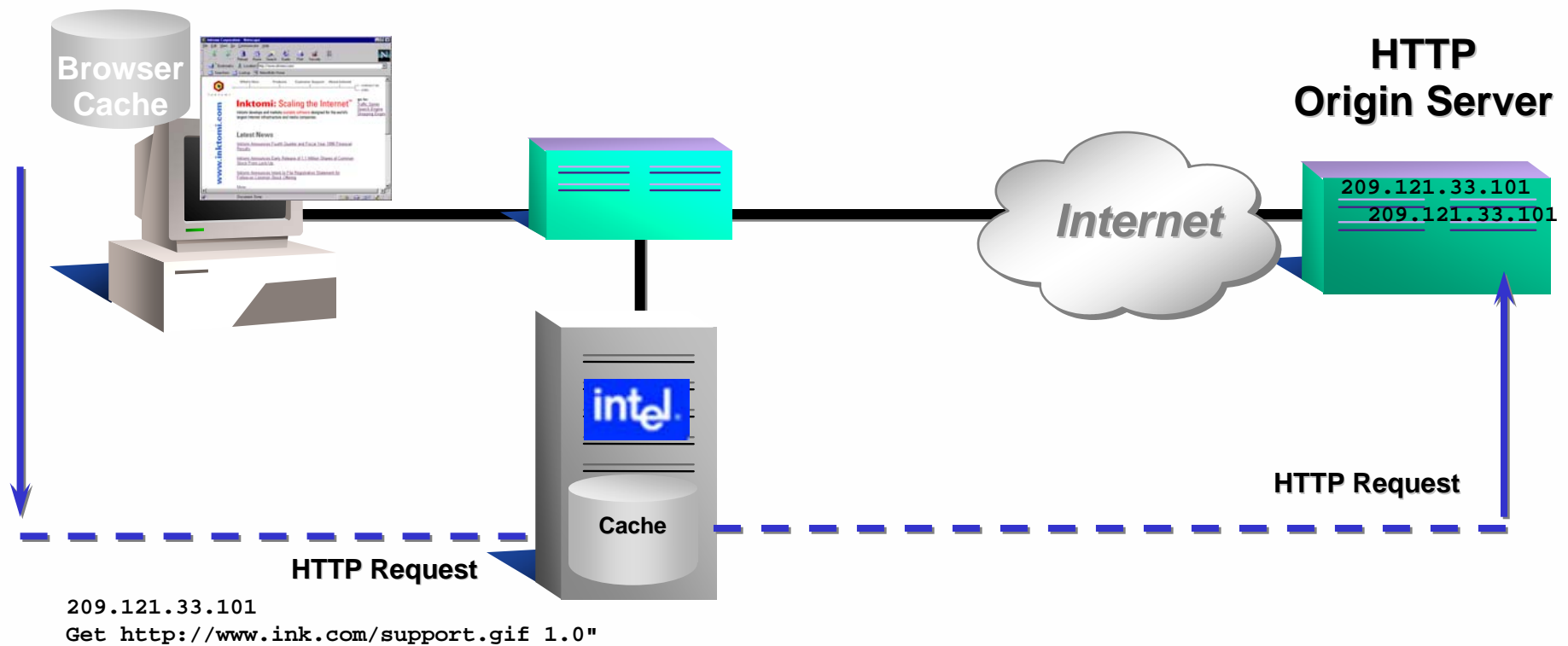
inkomi



Cache Miss



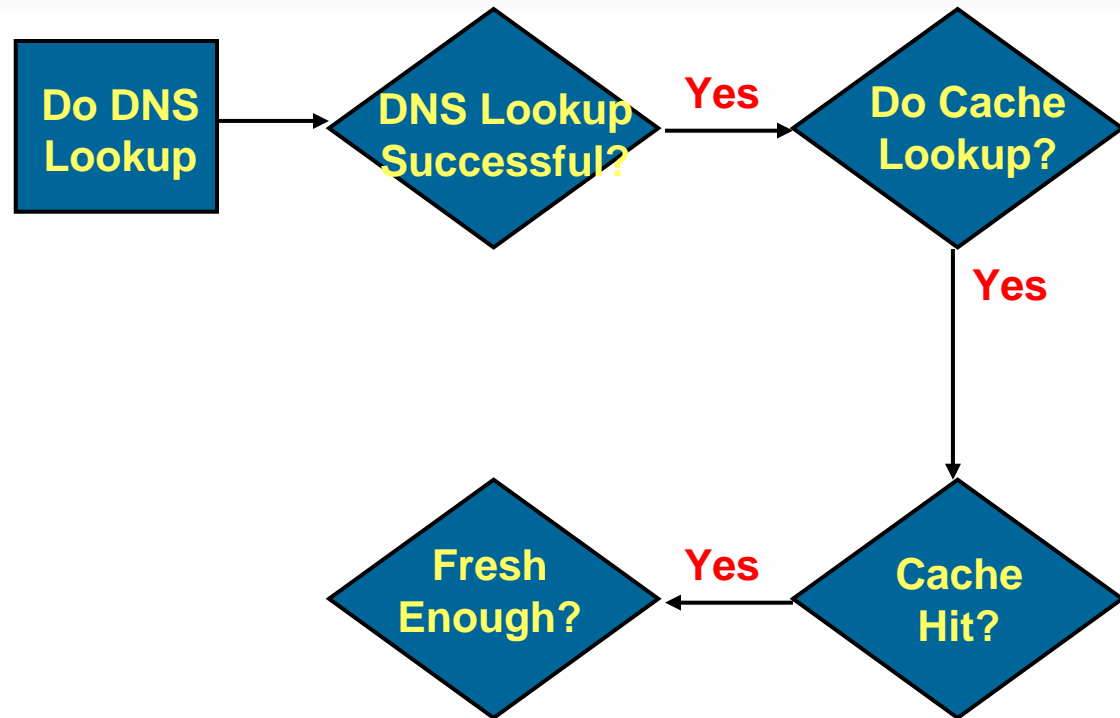
inktom i



Client Request



inktom i



Content Freshness



inktom

- **Keeping the Content Fresh is the Goal of all Parties involved in Object Content and Delivery**
 - Designers that maintain the content
 - Origin Server that stores and delivers the content
 - Intel Cache that stores and deliver the content
 - Client Browser that generates the request

Maintaining Freshness via Designer



inktom

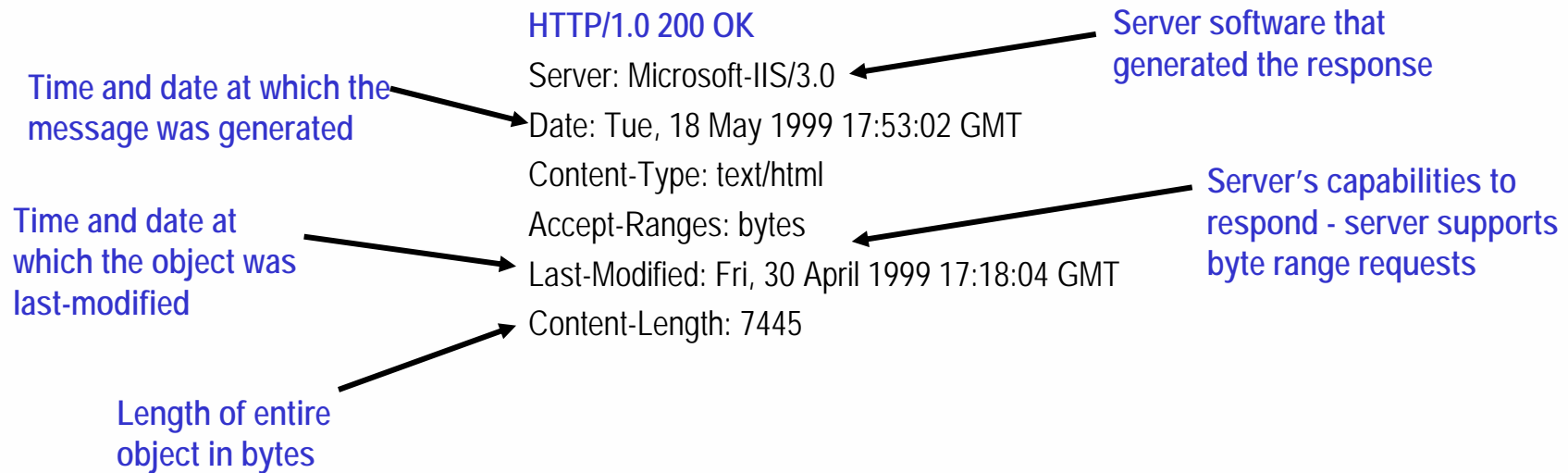
- **Meta-Tags can be coded in the <Head> section of HTML to describe the object's attributes**
 - Easy to implement
 - Usually only honored by browsers

Maintaining Freshness via Origin Server



inktom i

Intel Cache utilizes header information to determine content freshness



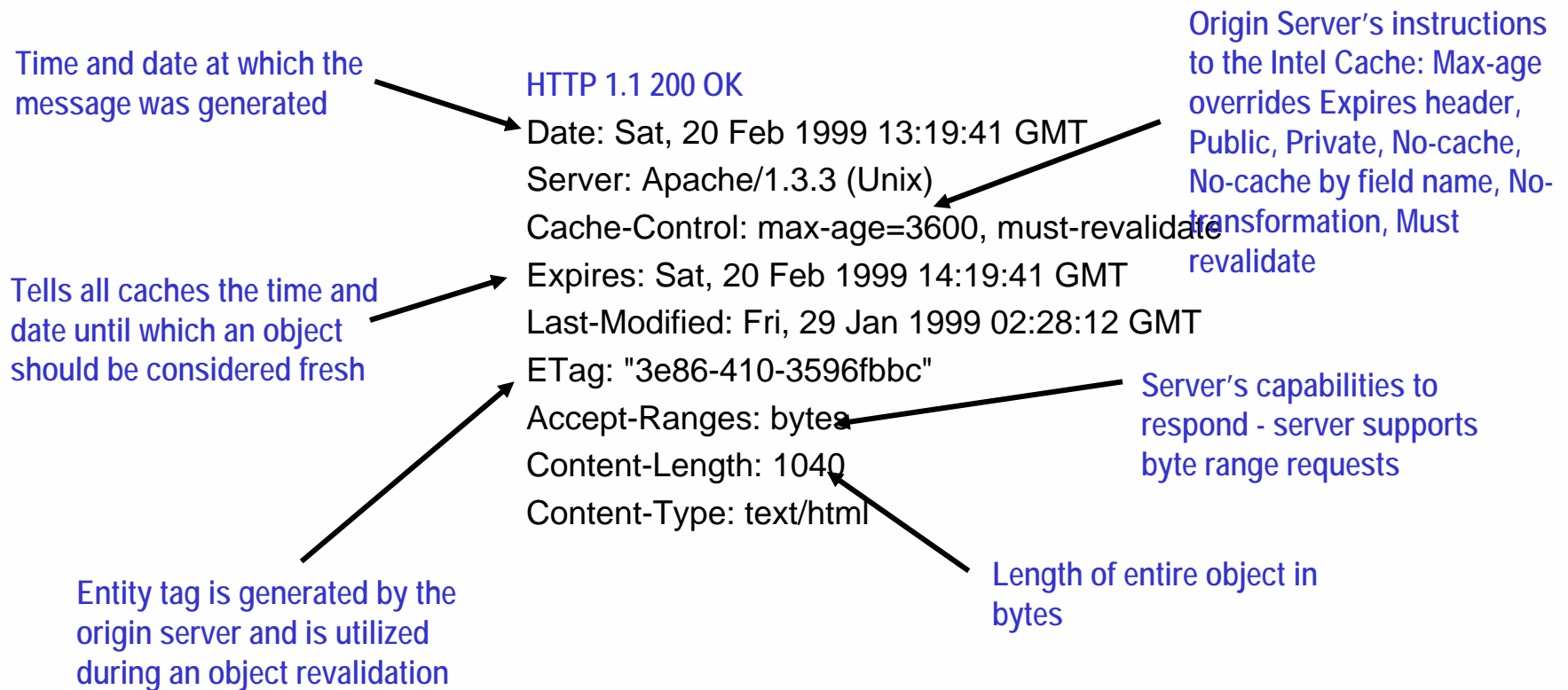
$$(\text{Last-Modified} - \text{Date}) * 10\% = \text{Cache Freshness Limit}$$

Maintaining Freshness via Origin Server



inktom

Intel Cache utilizes header information to determine content freshness



HTTP Request also contains Cache-Control. This allows the client to monitor the source of a requested object. No-cache, No-store, Max-age, Max-stale, Min-fresh, Only-if-cached

Maintaining Freshness via Intel Cache



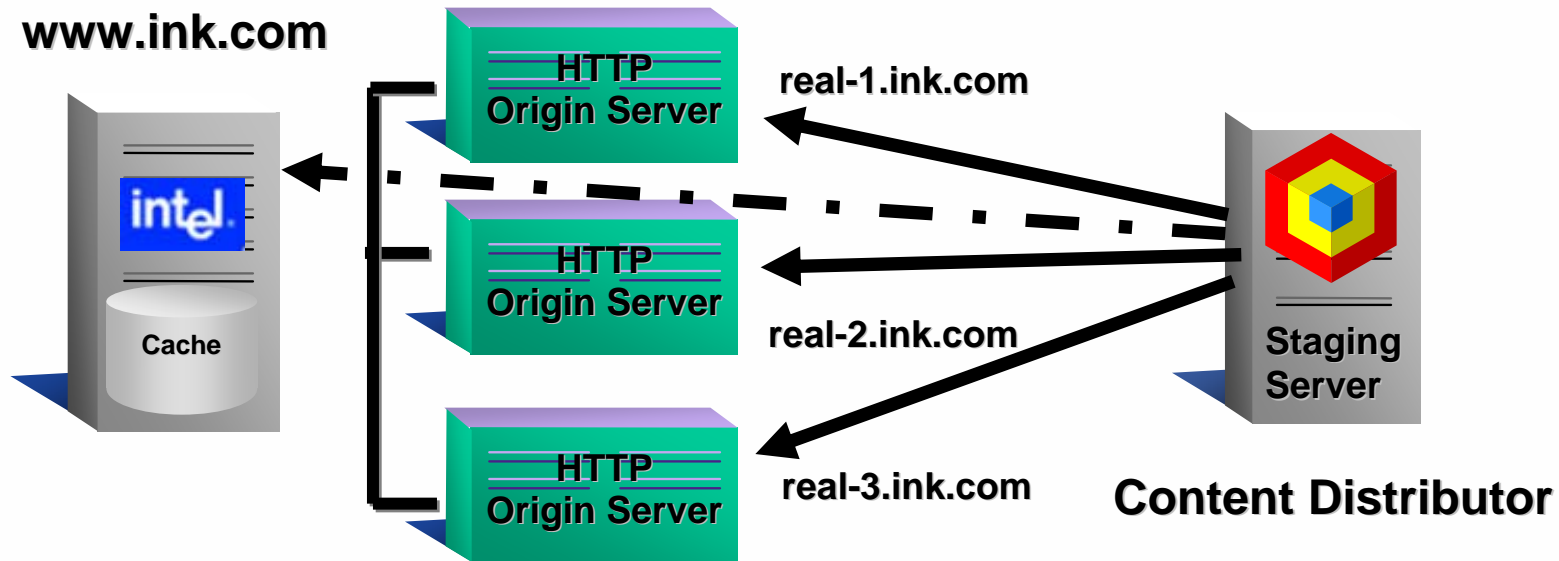
- **Configuring the Intel Cache to Selectively Cache and Properly Maintain Objects**
 - Customized to ask the origin server to verify the object's freshness (when the object expires, has no expiration date, always, or never)
 - Preventive caching with objects that contain "?" or "/cgi-bin" or end in "asp" and cookies
 - FTP objects have no time stamp or date information and will be removed from the cache as specified by the Intel Cache Administrator (15 min - 2 weeks)
 - News (NNTP) articles are refreshed each time Intel Cache polls parent news servers for changes in group list, article overview lists, and article updates (updating, retrieving, and accepting)

Maintaining Freshness via Content Distribution



inktom i

In reverse proxy mode, content can be synchronized using the Content Distributor.



1. Content is updated at the staging server.
2. Content is replicated to origin servers.
3. After update, Intel Cache is told to delete stale content.



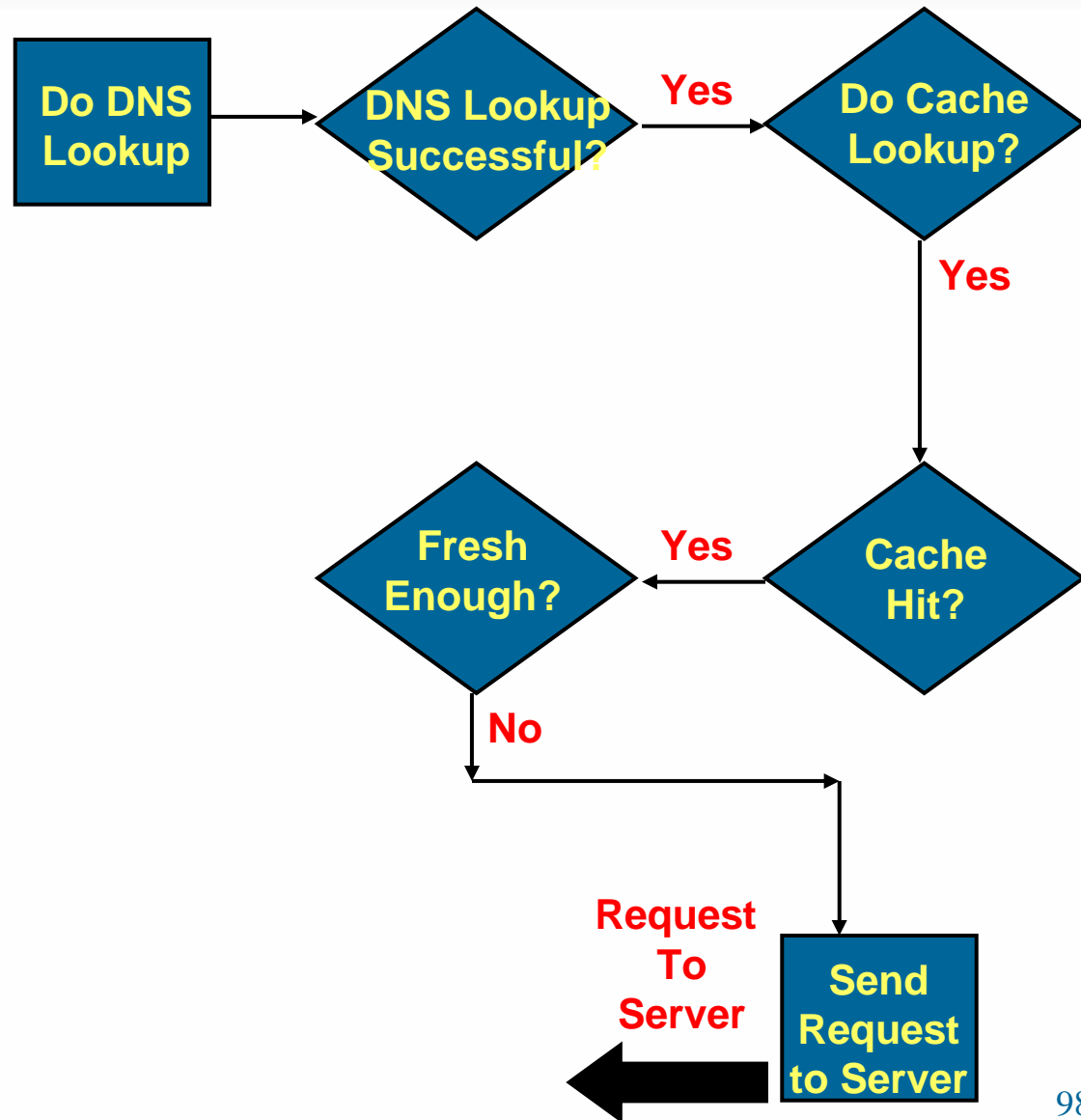
Users always receive fresh content

- **Netscape Navigator headers (Navigator version 3 and later):**
 - Reload -> sends an If-Modified-Since header (IMS).
 - Shift-Reload -> sends a pragma: no-cache or Cache-Control: no-cache header
- **Microsoft Internet Explorer headers (MSIE version 4 and later):**
 - Refresh -> sends an If-Modified-Since header (IMS).
 - Shift-Refresh -> same as just Refresh
 - Ctrl-Refresh -> sends a pragma: no-cache or Cache-Control: no-cache header
- **When Intel Cache receives a client request with a no-cache header, Intel Cache will always request the document from the origin server, unless it has been configured to ignore client no-cache requests.**

Client Request



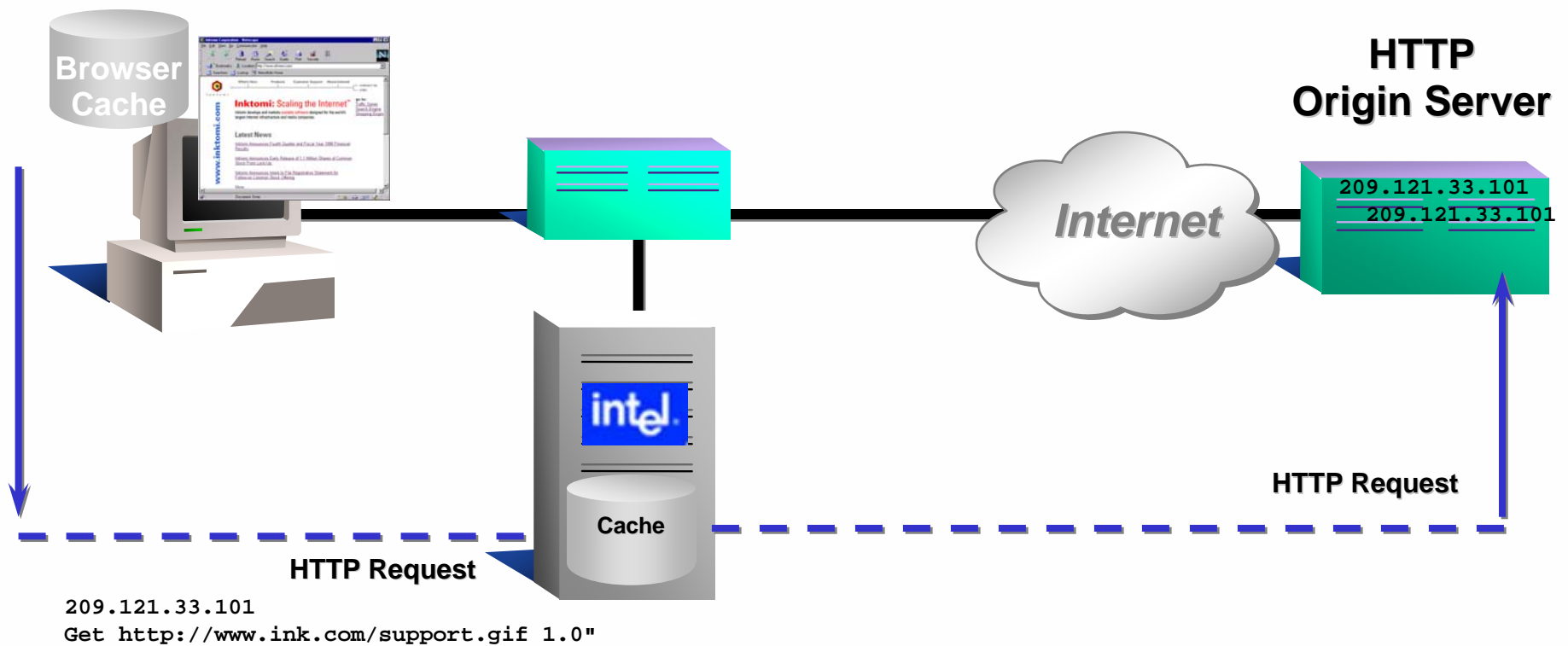
inktom i



Cache Refresh



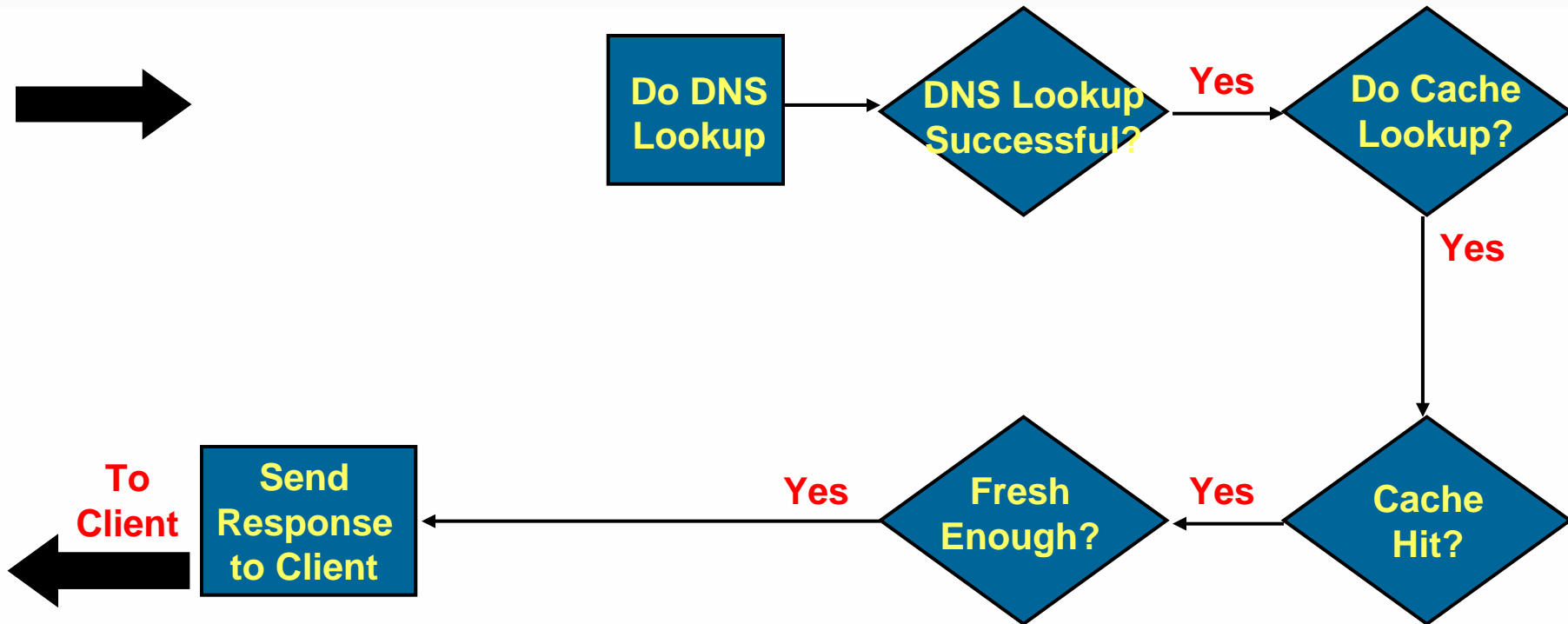
inktom i



Client Request



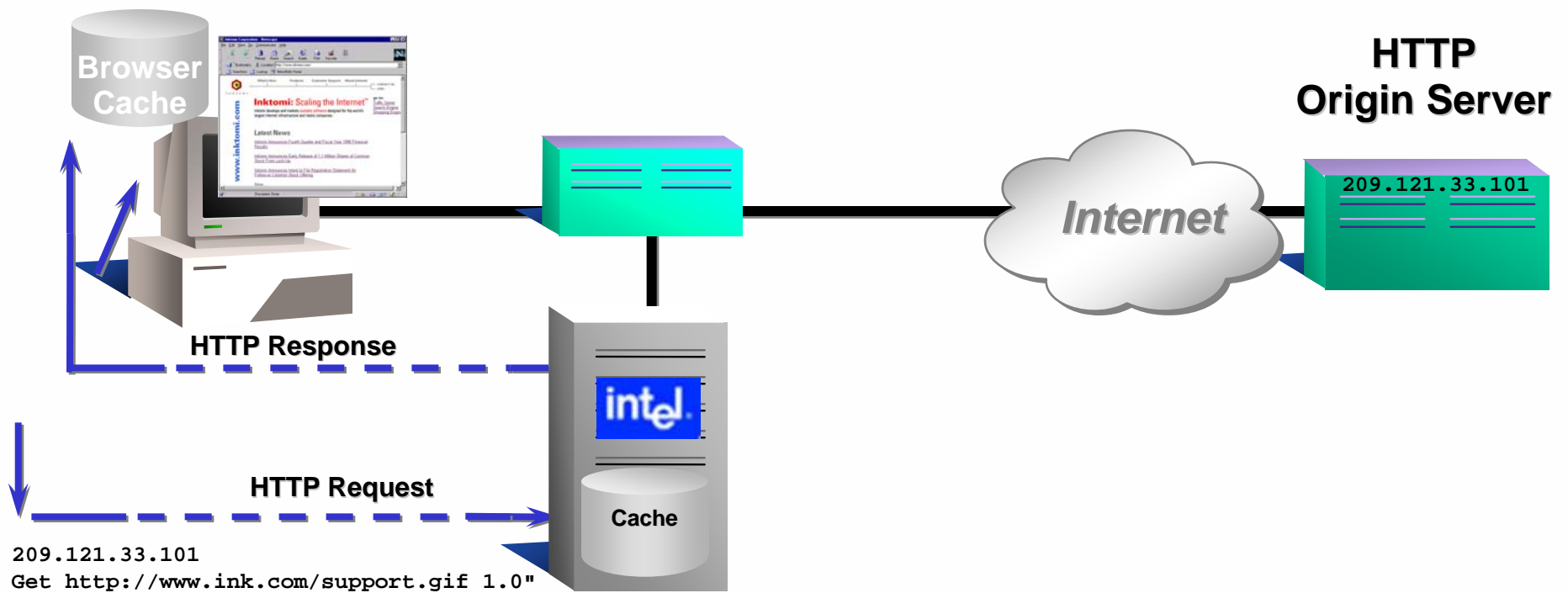
inktom i



Cache Hit



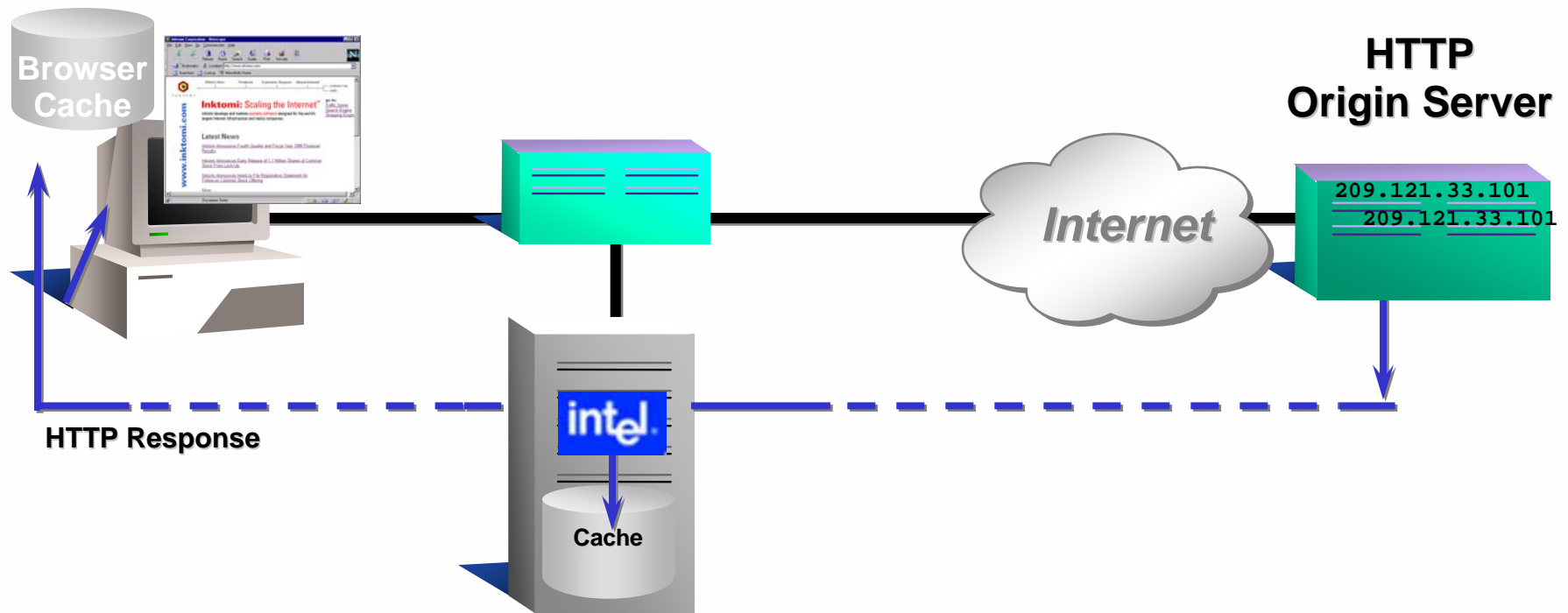
inktom i



Origin Server Response



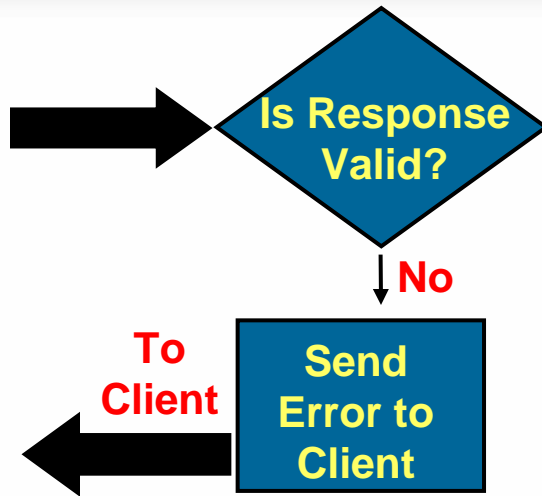
inktom i



Response from Server



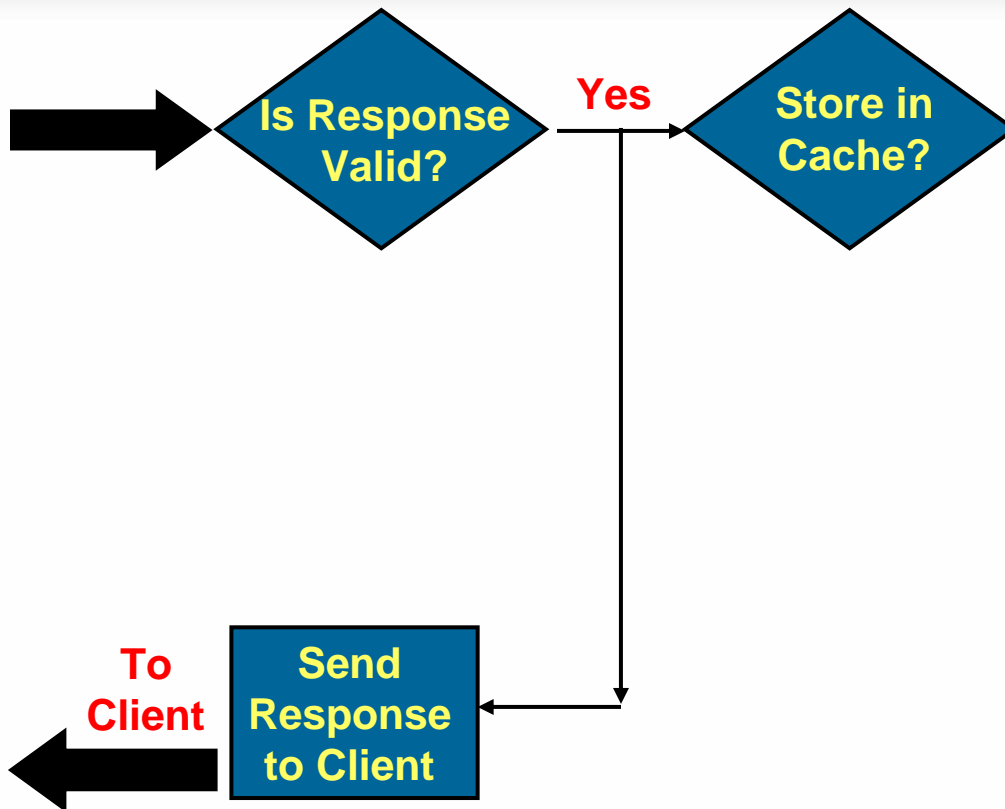
inktom i



Response from Server



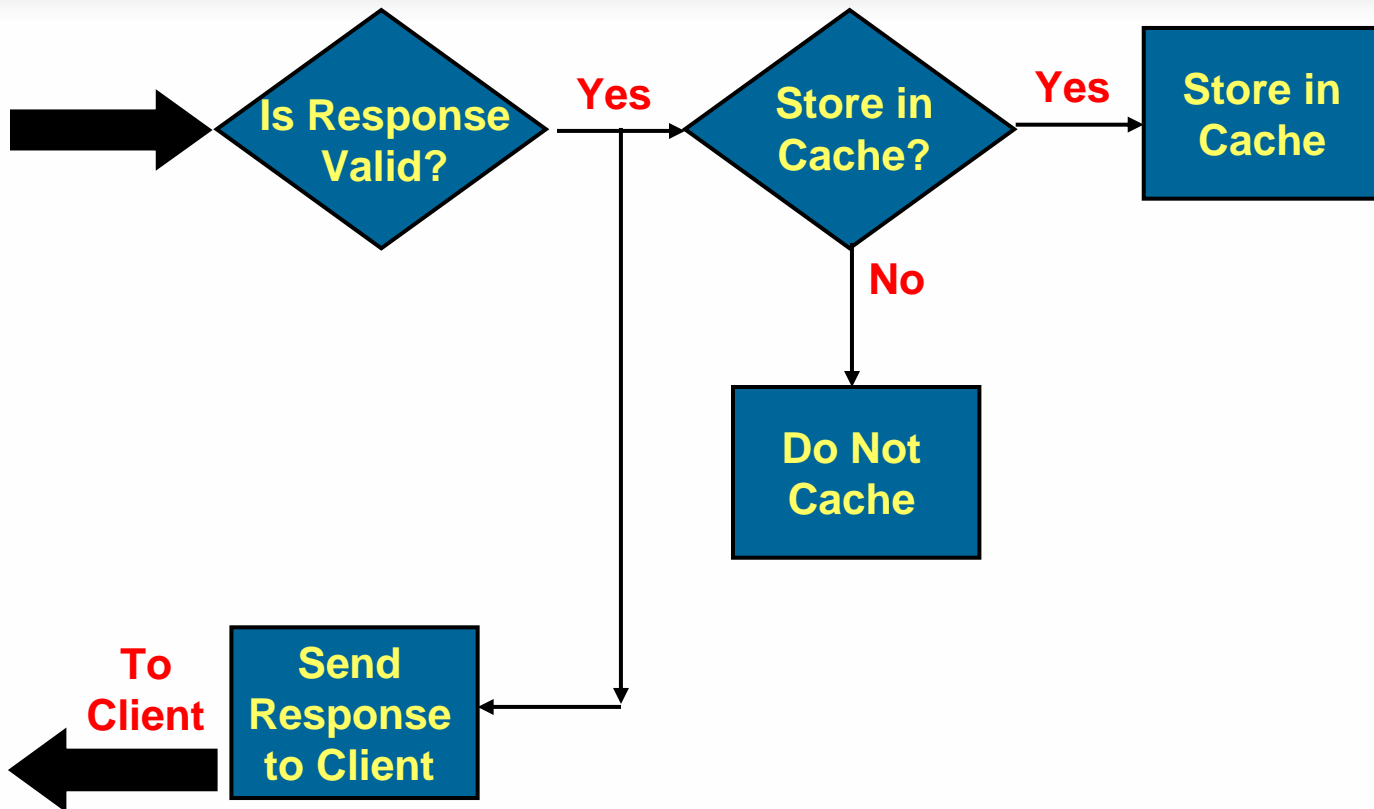
inktom i



Response from Server



inkomi



Benefits of Forward Caching



inktom i

- **Alleviates network congestion by significantly reducing the transmission of redundant Internet data**
 - lower bandwidth costs
 - more controlled network build-out
 - reduced load on critical network routers and origin Web servers
- **Increased quality of service for the end user by storing frequently requested information close to users**
- **Reduce wasteful resource consumption**
- **Opportunity for optimization and cost savings**

Review Questions:



inkomi

- What are the benefits of installing a Intel Cache?
- What is the difference between a Hit and Miss?

Intel Cache: Architecture



i n k t o m i

Chapter Overview



inktom

- **Intel Cache Innovations**
- **Intel Cache Processes**
- **Data Flow Architecture**



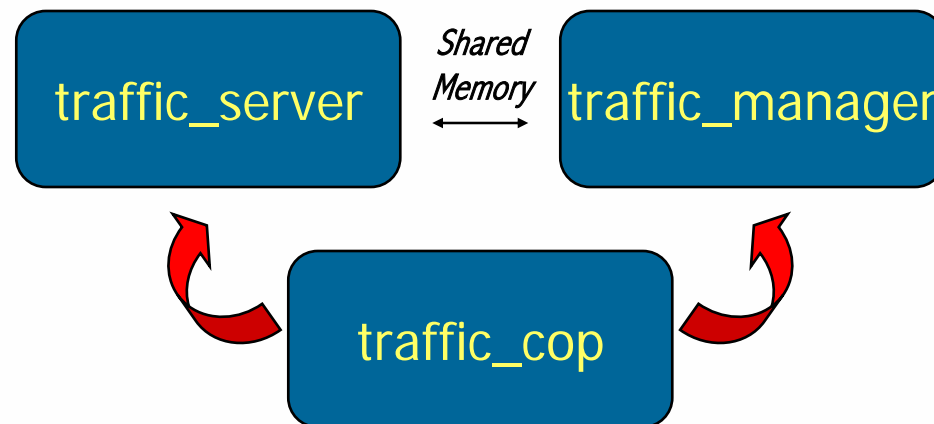
- **Completely modular architecture designed like an OS**
 - High-performance and portable to fastest hardware
 - Native streaming and transformation
 - Reads from origin and writes to client/cache simultaneously
 - Converts or compresses on the fly to match browser features
 - Peer configuration & monitoring via multicast
 - Custom object store
 - Centralized administration

Intel Cache Processes



inktom i

- Three-process, multi-threaded design per node
- Shared memory for communications with separate address space provide highest performance with safeguards that prevent a crash from taking both down



Performs most of the work...

- *Incoming & Outgoing Requests*
- *Caching Services*

Handles administration, clustering and fail-over UI changes through here

Very small - watches the Intel Cache Traffic Server and Traffic Manager to make sure they are running. Restarts as necessary.

The Traffic_Server Process



inktom i

- **First in the trinity of cooperating processes**
- **This is the cache processing engine**
- **Responsibilities:**
 - Accept connections
 - Process protocol requests
 - Serve all documents (derived from cache or origin server)
 - Collect statistics (for traffic_manager to present)

The Traffic_Manager Process



inktom i

- **This is the command and control facility**
- **Responsibilities:**
 - Stops, starts and restarts the traffic_server process
 - Monitors the proper functioning and configuration of the traffic_server
 - Provides graphical Web administration
 - Collect and present statistics
 - Provides cluster administration
 - Virtual IP failover
 - Manages proxy auto-configuration port
 - Maintains a queue of connections in the event of a server restart

The Traffic_Cop Process



- This is the health monitor for both traffic_server and traffic_manager processes
- Responsibilities:
 - Heartbeat tests (fetches synthetic.txt)
 - Occurs every 10 seconds
 - Heartbeat is logged to Intel Cache's access log
`http://127.0.0.1:8083/synthetic.txt`
 - A crontab process ensures that the traffic_cop is running
 - Runs every five minutes, logged to `syslog`
 - In the event of failure, automatically restarts failed processes

```
Oct 25 03:30:00 wolverine traffic_cop[1166]: Cop Starting - Version:
traffic_cop 2.0.0e - (build # 92219 on Oct 22 1998 at 19:19:47)
Oct 25 03:30:00 wolverine traffic_cop[1166]: Periodic heartbeat
successful, another cop still on duty
```

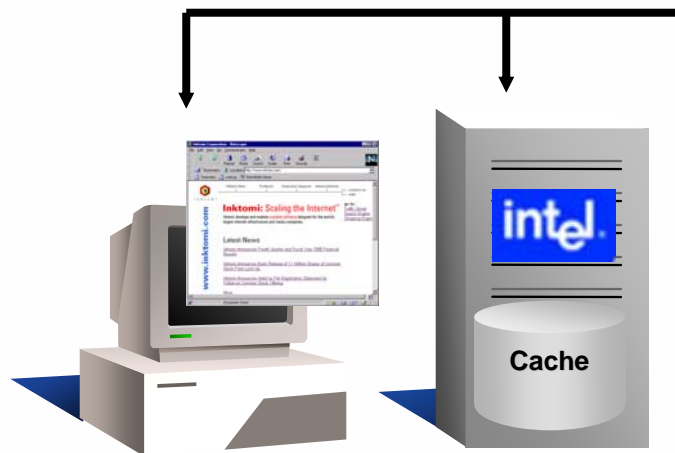
DataFlow Architecture



inktom i

- **Designed for Performance**

- Streaming dataflow engine rapidly transfers data to and from disk and network connections
 - Adapts to network and disk performance
 - Minimizes use of system resources

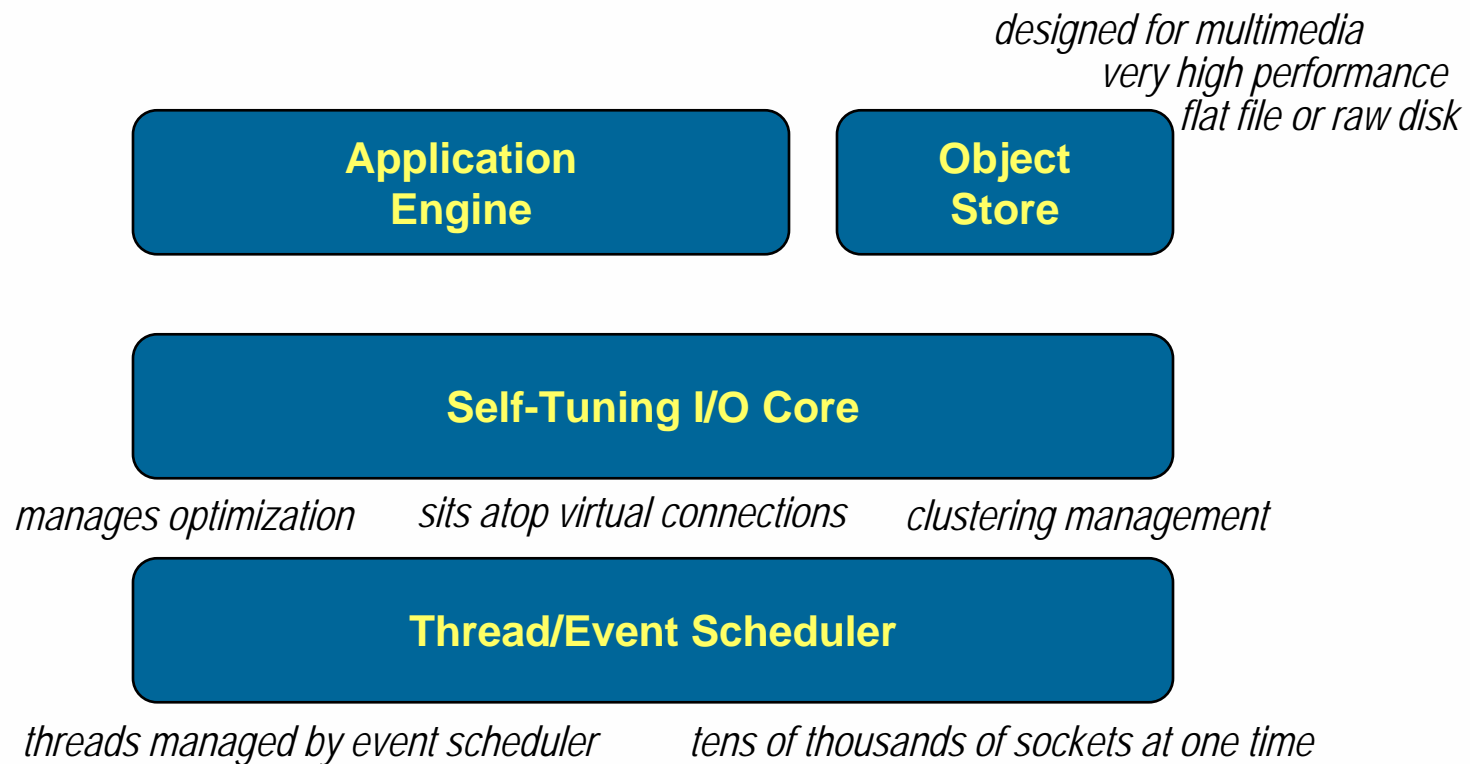


**Pipeline streams data objects
from web hosts to users
while it caches them**

Intel Cache Architecture



- **Modular, layered system**
- **Uniform API for network, disk, cluster and cache**

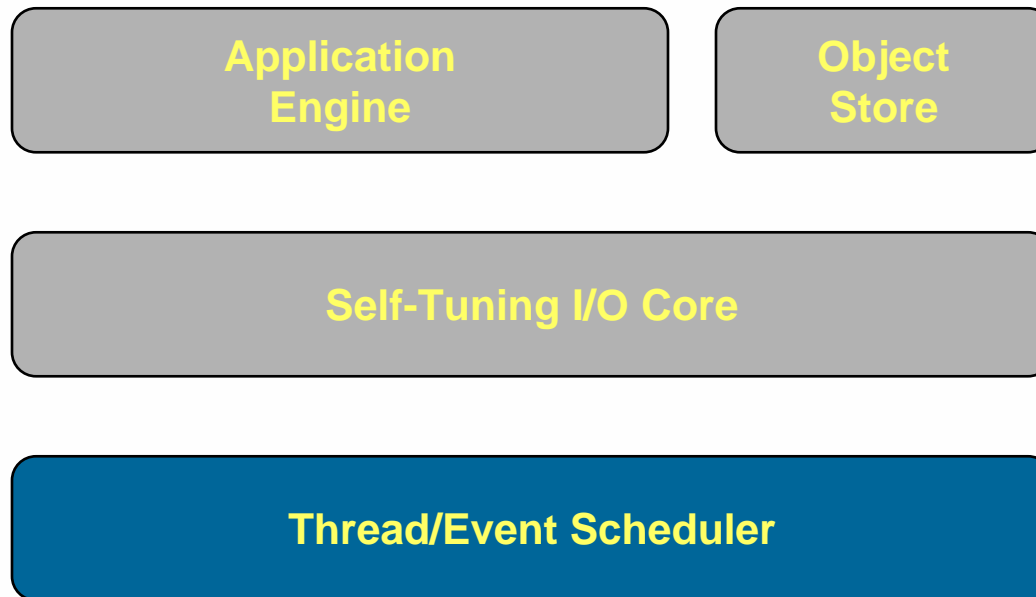


Thread/Event Scheduler



inktom i

- **Threads managed by event scheduler**
 - multithreaded, timestamped event queue scheduler
 - threads activate C++ objects (called continuations) at scheduled times, passing event codes



threads managed by event scheduler

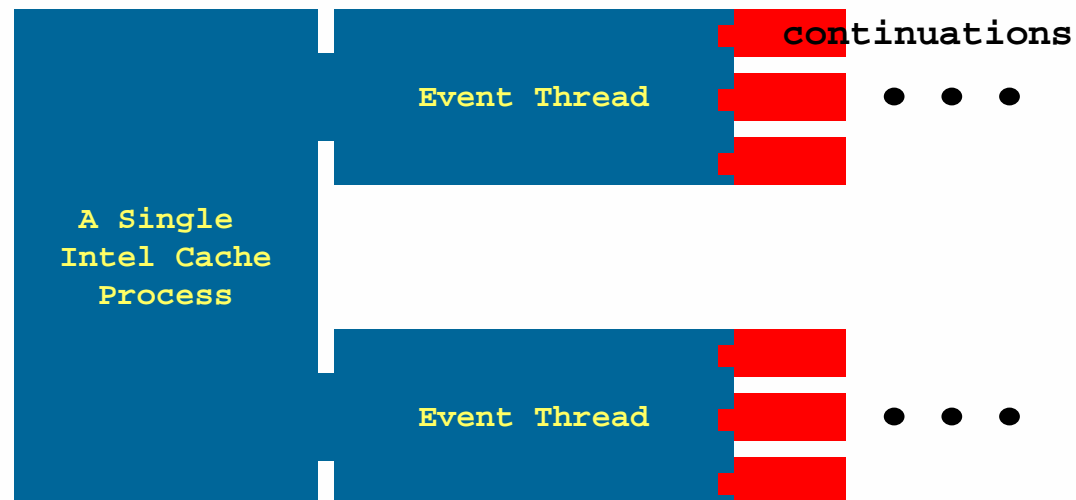
tens of thousands of sockets at one time

Continuations



Intel

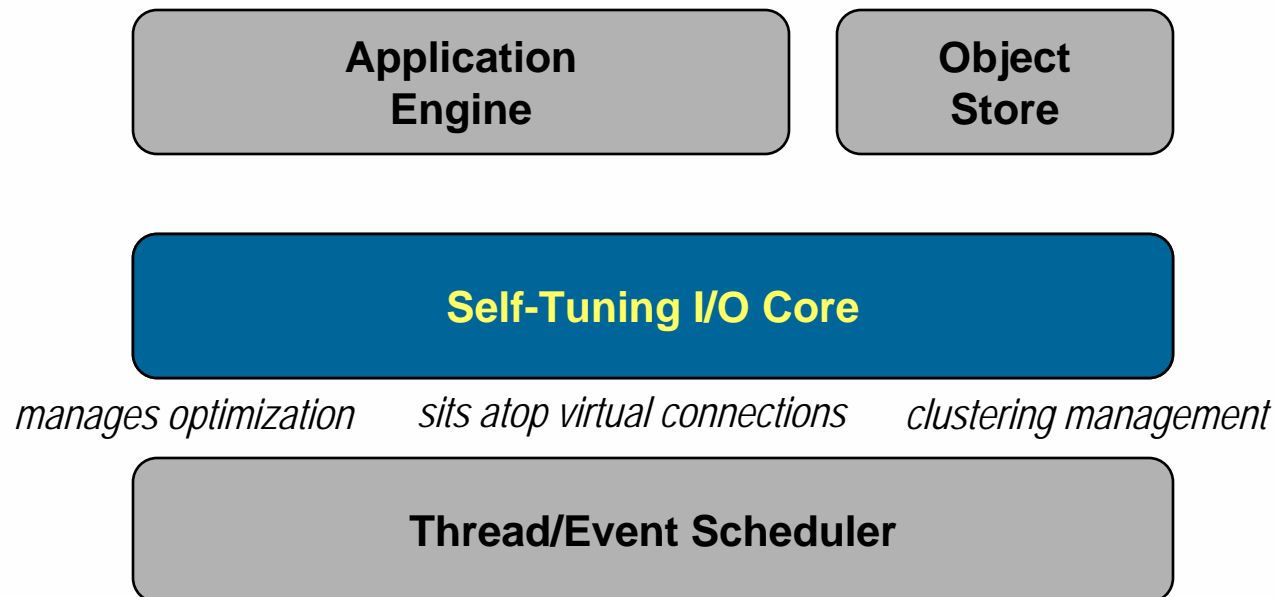
- Continuations are very small C++ objects that capture state, activation functions and currency controls
- Thousands of continuations running simultaneously through advanced multiplexing techniques
- Each performing a limited task as a part of an event



Self-Tuning I/O Core



- **No monolithic select loop for I/O**
- **Self-tuning I/O schedulers**
 - Adapt to dynamics, and open congestion windows
 - The I/O core does the optimization, not protocol engine
 - Same API for network, disk, cluster, cache, etc!

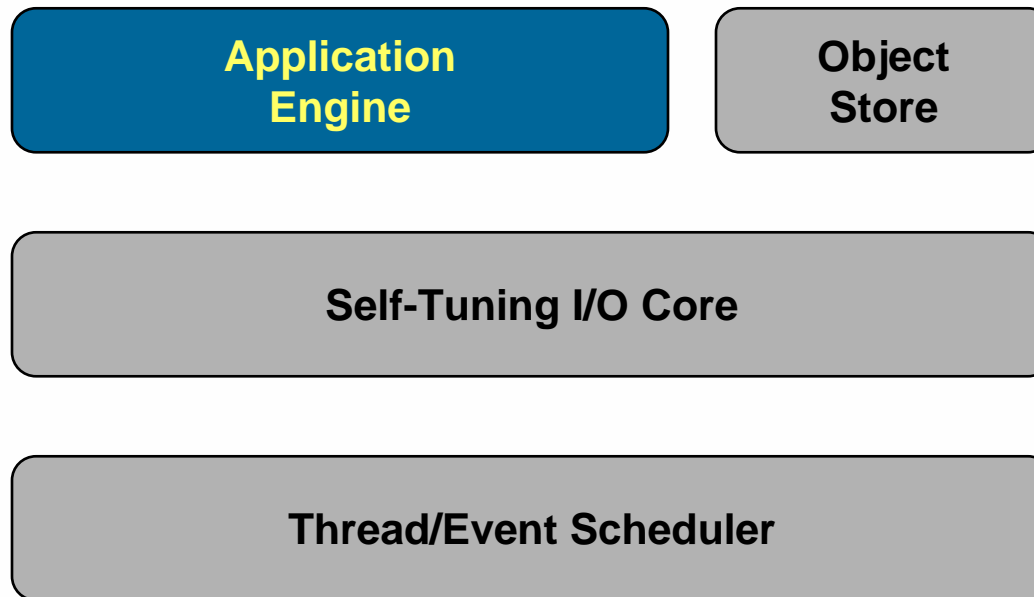


Application Engine



inktom i

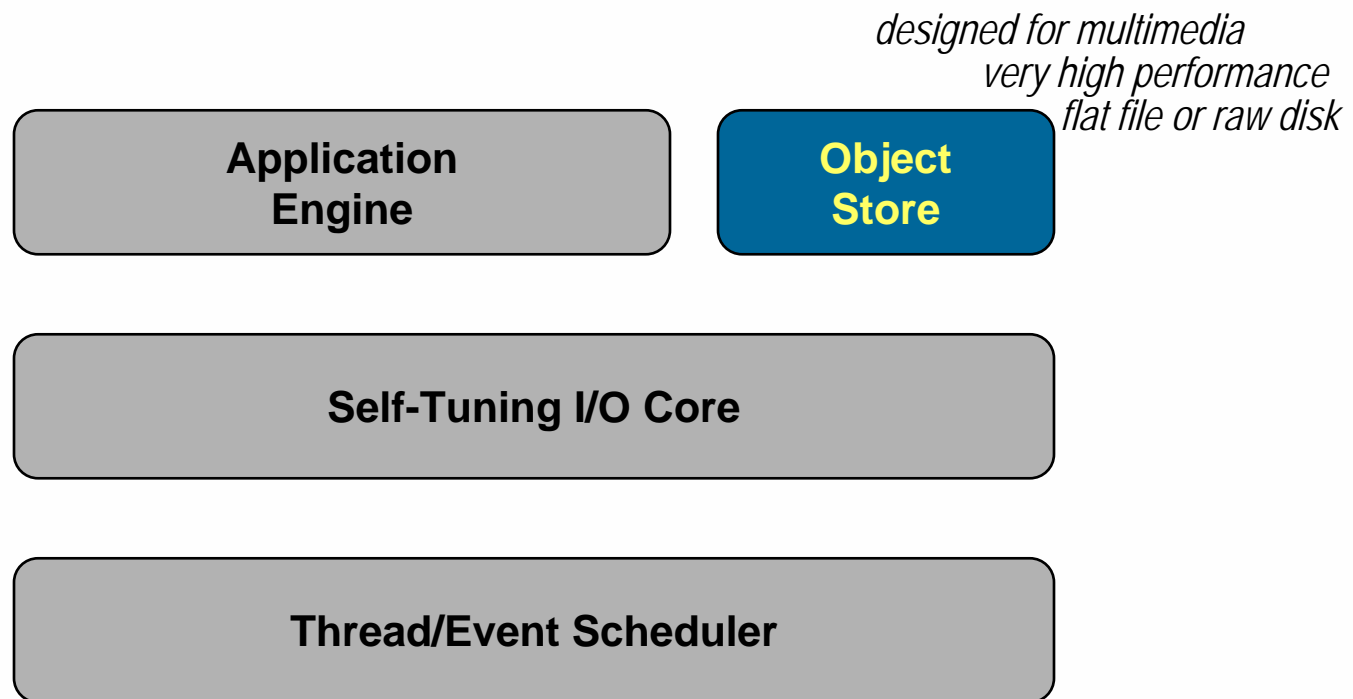
- Asynchronous state machines
- One state machine per transaction
- Many thousands can be active simultaneously



Object Store



inktom



Object Database



inktom i

- **Objects are stored in raw disk space**
 - Seldom fragmented regardless of size
 - Indexes are stored separately from objects and cached in memory to reduce index search time
 - High performance by storing objects contiguously on disk
- **Most read or write requires only a single movement of the disk head**
- **Replicating frequently requested objects to each node in the cluster squeezes more performance out of the cluster**

DNS/Host Information Database



inktom i

- **Includes a fast, asynchronous DNS resolver to streamline conversion of host names to IP addresses**
- **All nodes in the cluster share this synchronized database, minimizing latency and needless duplicate look-ups**
- **DNS database automatically redistributes the workload when the cluster configuration changes**
- **DNS bindings are cached in a distributed host database**
 - The database stores information about hosts on the Internet
 - DNS data for converting host names to IP addresses
 - HTTP version (1.1, 1.0, or 0.9)
 - Common to achieve 90%+ hit rates
 - Short time to live

Transparent to the User



Intel

- **Data goes directly to the user while caching is underway**
- **Users never notice the Intel Cache or its caches**
 - Simple browser options activate the Intel Cache
 - Transparency can be set on the server side for automatic browser configuration
 - Users specify standard web addresses
 - The Intel Cache searches its own caches first, and accesses the Internet only when needed
 - Every user is supported by each of the nodes without having to be aware of data location

Review Questions:



inkomi

- What are three processes in Intel Cache?
- What are the benefits of the Data Flow Architecture?

Intel Cache: Configuration and Monitoring Interfaces



i n k t o m i

Chapter Overview



inktom

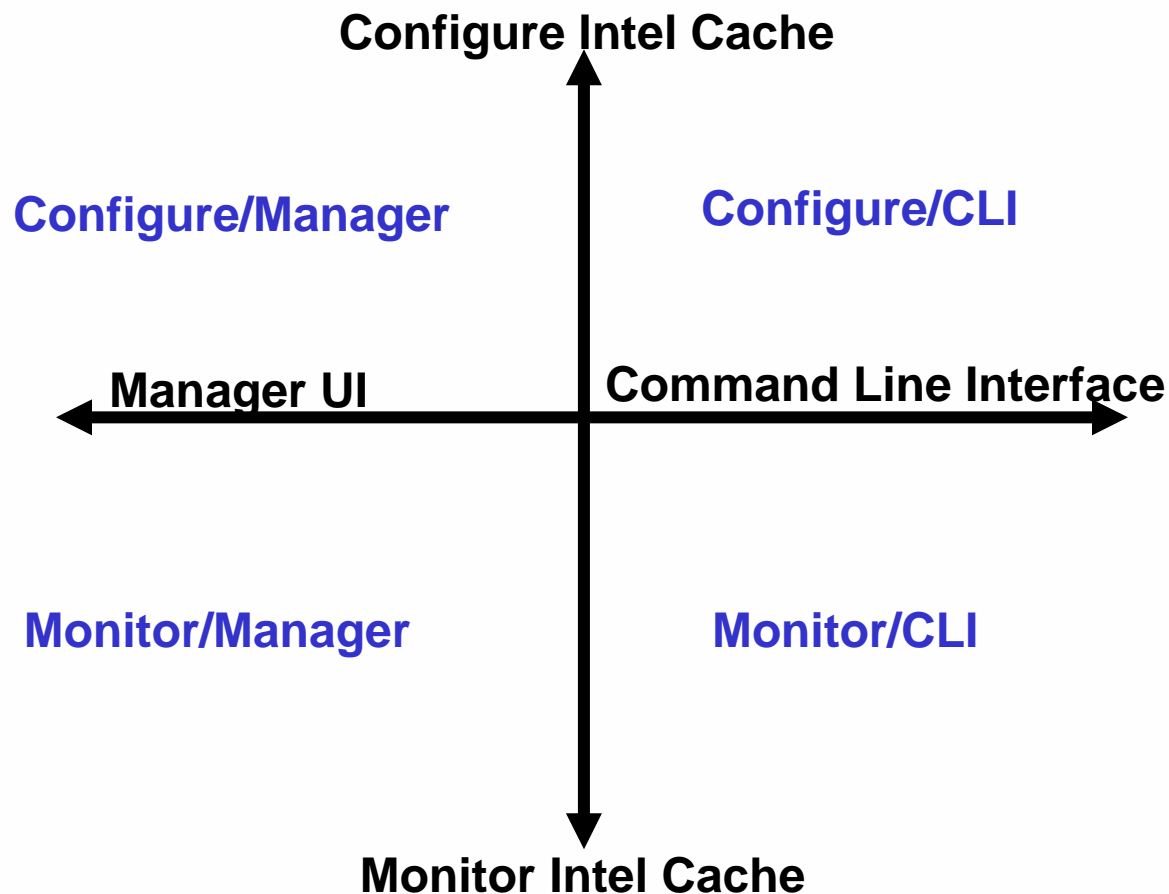
- **Explain different interfaces and their functions**
- **Manager user interface usage and navigation**
- **Command line interface usage and navigation**

Interfaces and Usage



inktom i

- **Two interfaces to the Intel Cache: Manager UI and Command Line Interface**
- **Two uses for interfaces: Configuration and Monitoring**



Manager User Interface



inktom i

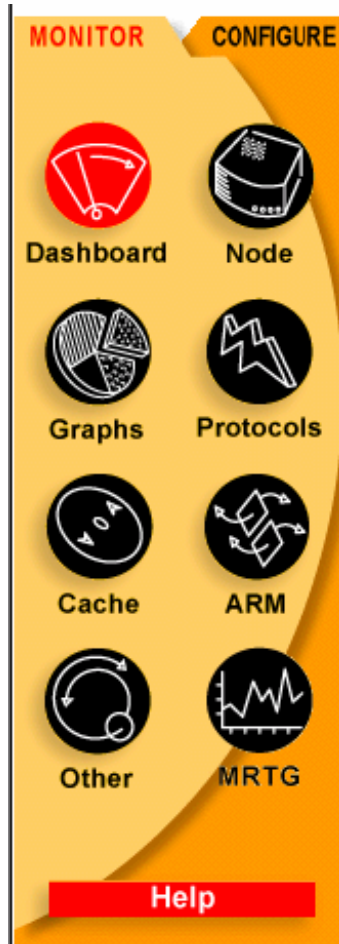
- Enter URL of Intel Cache and proxy port (usually 8081)

The screenshot shows a Netscape browser window titled "Traffic Server Manager - Netscape". The address bar contains the URL "http://rubicon-32:8081/main.ink?t=c_serv". The main content area displays the "Configure: Server Basics" page for the "Intel(r) Netstructure(tm) 1500 Cache". The page features a left-hand navigation menu with icons for "Server", "Protocols", "Cache", "Security", "Routing", "Host DB", and "Snapshots". The "Cache" section is currently selected. The main configuration area includes a toggle switch for "off" (selected) and "on", with a note: "This switch controls only node rubicon-32". Below this is a text input field for "Intel(r) Netstructure(tm) 1500 Cache Name:" containing the value "rubicon-32". A descriptive note states: "The following two options control how the Intel(r) Netstructure(tm) 1500 Cache handles unqualified hostnames in a URL. Setting both options expands a hostname first into the local domain and secondarily into the .com domain." Below this are two radio button options: "Local Domain Expansion:" with "On" selected and "Off" unselected; and ".com Domain Expansion:" with "On" selected and "Off" unselected. A "Make These Changes" button is located at the bottom right of the configuration section. The "Web Management" section at the bottom includes a "restart" button with a note: "This button restarts the cluster". The browser's status bar shows "Document: Done" and the system tray at the bottom right displays the time "7:17 PM".

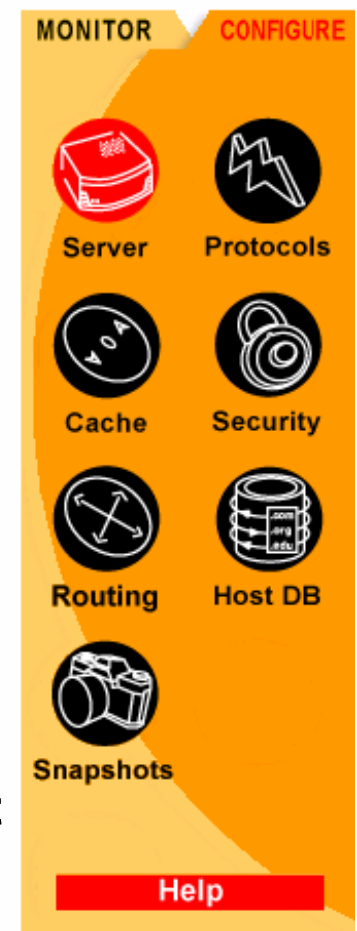
Manager UI Usage



inktom i



- 2 menus: Monitor and Configure
- Select menu by clicking on title (Monitor or Configure)
- **Red** text indicates which menu is selected
- Select menu items by clicking on them
- **Red** icon indicates which menu item is selected
- *Help* icon opens a new browser window with a help page for the currently selected menu item
- Browser must allow Java and Javascript for some pages



Manager UI Help Pages



inktom i

- Separate overall help pages for Monitoring and Configuration
- Click menu icons on help pages to move to a help page for that menu item

Configuration Help

Intel® NetStructure™ 1500 Cache

cache snapshot server security
protocols database routing

monitor

Using the Configure pages

Monitoring Help

Intel® NetStructure™ 1500 Cache

dashboard node graphs protocols cache other MRTG

configuration

Using the Monitor pages

- Get to overall help pages by clicking either <Mode> Help in upper right corner or by clicking labeled arrow boxes

Command Line Interface



Intel

- telnet or otherwise establish a serial interface connection to the Intel Cache
- Set your terminal for vt100 emulation

Cursor indicating the menu item that will be selected if Enter is hit

Brief description of selected menu item

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----
|                               -----root-----                               |
| -setup  Initial Intel Cache Setup                               |
|  main   Main Intel Cache Controls                               |
| config  Intel Cache Configuration                               |
| monitor View Statistics                                         |
| expert  Enter Expert Mode                                       |
| save    Save Config To Floppy                                   |
| load    Load Config From Floppy                                |
-----
|                               Initial Intel Cache Setup                               |
```

Navigating the Command Line Interface



inktom i

- Move within a menu: Up and Down Arrow Keys
- Select a menu item: Enter
- Return to previous menu: Ctrl-x
- Save information field in a form window Enter
- Accept data confirmation box after entering information Ctrl-x
- Exit a form window Ctrl-x

Current location and path in menu hierarchy

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----
root->monitor->cache
|Cache Bytes used      134742016
|Cache Size           50002919424
|Cache Lookups Completed      0
|Cache Lookups Failed        0
|Cache Reads Completed        0
|Cache Reads failed      27378
|Cache Writes Completed      0
|Cache Writes failed        0
|Cache Updates Completed     0
|Cache Updates Failed       0
|Cache Removes Completed     0
|Cache Removes Failed       0
-----
View Cache Statistics
```


Intel Cache: Setup



inktom i

CLI: Setup->IP Parameters



Intel

- Network Interface Card *must* be configured first time you connect to the Intel Cache
- Afterwards, this menu changes card's network settings

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----
Intel(r) NetStructure(tm) 1500 Cache
-----

+-----+
|               Configure Hostname and IP address               |
+-----+
| Current IP Address  209.131.52.32                             |
| New IP Address      █ _____                             |
| Current Hostname    rubicon-32                               |
| New hostname       _____                             |
| Current netmask     255.255.255.0                             |
| New Netmask        _____                             |
+-----+

Configure IP parameters
```


CLI: Setup->DNS



inktomi

- Change the DNS name and address

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----

      +-----+
      |                               DNS Entry Form                               |
      |-----|
      | Current DNS Address 209.131.48.11 |
      | Current domainname  inktomi.com  |
      | New DNS Address      █           |
      | New domainname      _____ |
      |-----|
      +-----+

Set DNS Server
```

CLI: Setup->Gateway



inktom i

- Change the Gateway address used by the Intel Cache

The screenshot shows a Telnet window titled "Telnet - rubicon-32" with a menu bar containing "Connect", "Edit", "Terminal", and "Help". The main content area displays "Intel(r) NetStructure(tm) 1500 Cache" and a "Gateway Entry Form". The form contains the following text:

```
Gateway Entry Form
Current Gateway      209.131.52.1
New Gateway         █
```

At the bottom of the window, there is a black bar with the text "Set Router" in white.

CLI: Setup->View Settings



inktom i

- View the current Setup settings

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----+-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----+-----

+-----root->setup->view-----+
| Current IP          209.131.52.32      |
| Current Hostname    rubicon-32        |
| Current DNS         209.131.48.11     |
| Current Gateway     209.131.52.1     |
| Time Zone           United States, Pacific |
| Date/Time           Sat Nov 6 18:11:14 1999 |
+-----+-----+

View Settings
```

Intel Cache: Manager UI Server / CLI Main Menu Configuration



i n k t o m i

Chapter Overview



inkomi

- **Explore Configuration Options**
- **Practice Lab**

Manager UI: Server Basics



inktom i

Turn Intel Cache On and Off, enable domain name expansion, enable load shedding, restart or reconfigure Traffic Manager, configure VIPs, auto-configure browser, and enable SNMP

Configure: Server Basics

Intel(r) Netstructure(tm) 1500 Cache

off *This switch controls only node rubicon-32*
 on

Intel(r) Netstructure(tm) 1500 Cache Name:

The following two options control how the Intel(r) Netstructure(tm) 1500 Cache handles unqualified hostnames in a URL. Setting both options expands a hostname first into the local domain and secondarily into the .com domain.

Local Domain Expansion: On Off

.com Domain Expansion: On Off

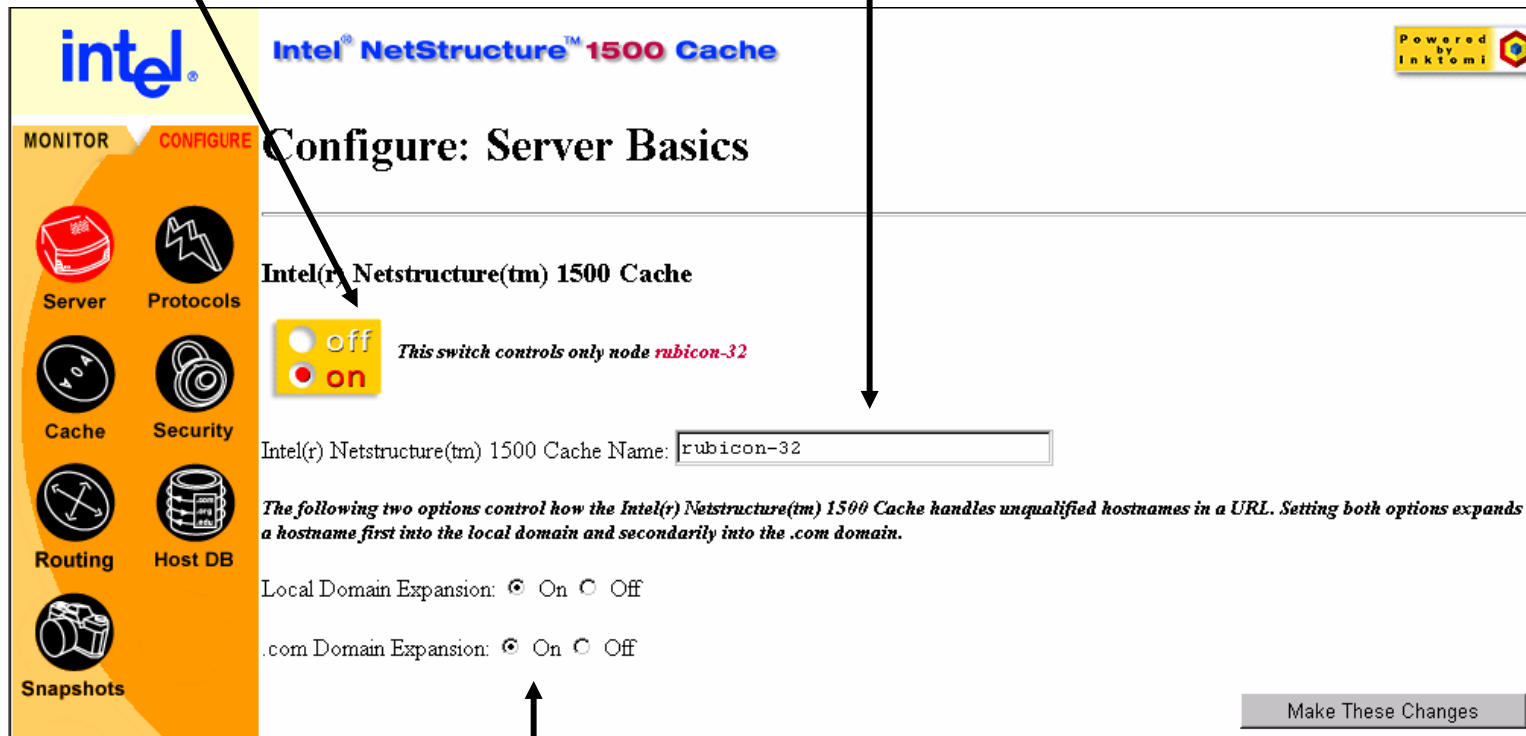
Web Management

Intel(r) Netstructure(tm) 1500 Cache Manager: *This button restarts the cluster*

Manager UI: Server Basics Details

Shutting down the server stops all caching and proxying services on a specific node

Cache name is the proxy name



Configure: Server Basics

Intel(r) Netstructure(tm) 1500 Cache

off *This switch controls only node rubicon-32*

Intel(r) Netstructure(tm) 1500 Cache Name:

The following two options control how the Intel(r) Netstructure(tm) 1500 Cache handles unqualified hostnames in a URL. Setting both options expands a hostname first into the local domain and secondarily into the .com domain.

Local Domain Expansion: On Off

.com Domain Expansion: On Off

Make These Changes

Turn on auto-expansion to have the Traffic Server automatically preface host names with www. and suffix them with .com

CLI: Main->Status



inktom i

- Check current Intel Cache status

```
Telnet - rubicon-32
Connect Edit Terminal Help
+-----+
| Intel(r) NetStructure(tm) 1500 Cache |
+-----+

+-----+
| Intel Cache Status |
+-----+
| Intel Cache Server      UP |
| Intel Cache Manager     UP |
+-----+

Check Status
```

CLI: Main->Start



inktom i

- Start the Intel Cache

```
Telnet - rubicon-52
Connect Edit Terminal Help
-----+-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----+-----

+-----+-----root->main-----+
| status   Check Status           |
| start    Run Intel Cache        |
| stop     Stop Intel Cache       |
| versions Change/View Versions  |
| clear    Clear Statistics       |
| reset    Reset To Factory Settings |
| replicate Replicate Primary Disk |
| passwd   Change 'admin' Password |
+-----+-----+

Run Intel Cache
```

CLI: Main->Stop



inktom i

- Stop the Intel Cache

```
Telnet - rubicon-52
Connect Edit Terminal Help
+-----+
| Intel(r) NetStructure(tm) 1500 Cache |
+-----+

+-----root->main-----+
| status   Check Status      |
| start    Run Intel Cache   |
| stop     Stop Intel Cache  |
| versions Change/View Versions |
| clear    Clear Statistics   |
| reset    Reset To Factory Settings |
| replicate Replicate Primary Disk |
| passwd   Change 'admin' Password |
+-----+

Stop Intel Cache
```


CLI: Main->Versions



inktom i

- Multiple versions of Intel Cache software can be installed; new versions do not delete or replace old versions automatically

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----+-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----+-----

+-----root->main->version-----+
| -View      View Currently Installed Versions      |
| Install    Install a New Version Using TFTP      |
| Switch     Run Another Version                   |
| Delete     Delete a version                       |
| Current    View the current version               |
+-----+-----+

View Currently Installed Versions
```

CLI: Main->Versions



inktom i

- Versions are configured only through the CLI; there is no Manager UI for versions
- Use this menu to switch between or delete versions

```
Telnet - rubicon-32
Connect Edit Terminal Help
+-----+
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
+-----+

+-----root->main->version-----+
| -View      View Currently Installed Versions      |
| Install    Install a New Version Using TFTP      |
| Switch     Run Another Version                   |
| Delete     Delete a version                       |
| Current    View the current version               |
+-----+

View Currently Installed Versions
```

CLI: Main->Reset



inktom i

- *Deletes the current installation!*
- Returns you to the setup menu for reinstallation

```
Telnet - rubicon-52
Connect Edit Terminal Help
-----+-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----+-----

+-----root->main-----+
| status   Check Status   |
| start    Run Intel Cache|
| stop     Stop Intel Cache|
| versions Change/View Versions|
| clear    Clear Statistics|
| -reset   Reset To Factory Settings|
| replicate Replicate Primary Disk|
| passwd   Change 'admin' Password|
+-----+

+-----+
| Are You Sure (y/n) |
+-----+

Reset To Factory Settings
```

CLI: Main->Replicate



inktom i

- Replicates primary disk to secondary disk
- May take several minutes

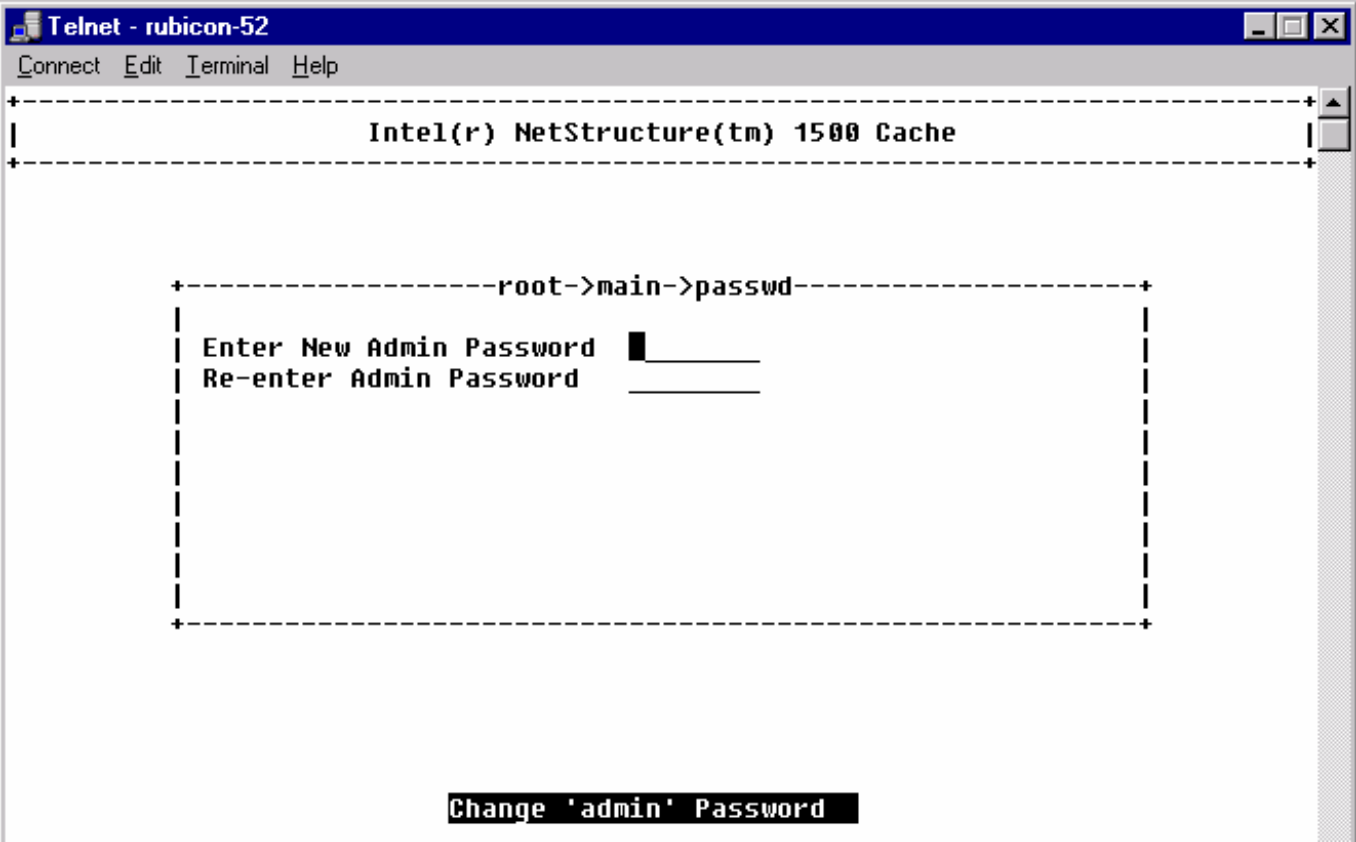
```
Telnet - rubicon-52
Connect Edit Terminal Help
+-----+
| Intel(r) NetStructure(tm) 1500 Cache |
+-----+

+-----root->main-----+
| status   Check Status      |
| start    Run Intel Cache   |
| stop     Stop Intel Cache  |
| versions Change/View Versions |
| clear    Clear Statistics   |
| reset    Reset To Factory Settings |
|  replicate Replicate Primary Disk |
| passwd   Change 'admin' Password |
+-----+

Replicate Primary Disk
```

CLI: Main->Passwd

- Set the admin password for the Manager UI



```
Telnet - rubicon-52
Connect Edit Terminal Help
+-----+
| Intel(r) NetStructure(tm) 1500 Cache |
+-----+

+-----root->main->passwd-----+
| Enter New Admin Password █ |
| Re-enter Admin Password   |
+-----+

Change 'admin' Password
```


Manager UI: Server Web Management



- Restart is used to effect changes to port numbers and virtual IP address numbers
 - Takes about 15 seconds,
 - Disables all caching and proxying across the entire cluster
- Refresh rate governs how often displays of graphics and statistics will be updated for you to monitor performance

intel® Intel® NetStructure™ 1500 Cache Powered by Inktomi

MONITOR CONFIGURE

Server Protocols

Cache Security

Routing Host DB

Snapshots

Web Management

Intel(r) Netstructure(tm) 1500 Cache Manager: **restart** *This button restarts the cluster*

Refresh rate in Monitor mode: 30 Seconds

Make These Changes

Virtual IP Addressing

Without Virtual IP addressing, nodes can not cover one another's failures.

Virtual IP (takes effect at restart): On Off

[Edit virtual IP addresses](#)

Make These Changes

Manager UI: Server Virtual IP Addressing



- Virtual IP addresses are additional IP addresses not oriented to any particular machine, but assigned dynamically within the cluster
- You can set up a DNS round robin so client requests will rotate among available nodes
- In the event a node fails, a peer node can take over the failed node's virtual interface
- If Virtual IP is OFF -- server nodes cannot cover each other's failures

intel Intel® NetStructure™ 1500 Cache Powered by Inktomi

MONITOR CONFIGURE

Server Protocols

Cache Security

Routing Host DB

Snapshots

Web Management

Intel(r) Netstructure(tm) 1500 Cache Manager: **restart** *This button restarts the cluster*

Refresh rate in Monitor mode: 30 Seconds

Make These Changes

Virtual IP Addressing

! *Without Virtual IP addressing, nodes can not cover one another's failures.*

Virtual IP (takes effect at restart): On Off

[Edit virtual IP addresses](#)

Make These Changes

Incorrect IP Addresses can effectively disable your system

Manager UI: Server: Browser Auto-Configuration



inktom i

- If you are not using transparency options, you may specify a preference to use the proxy server through an auto-configuration file
 - If none exists, it will be created
 - If the server detects an auto-configuration file you will have options to view, replace or delete the existing file

intel Intel® NetStructure™ 1500 Cache Powered by Inktomi

MONITOR **CONFIGURE**

Auto-Configuration of browsers

[Auto-configuration file](#)

Load Shedding

HTTP hit transaction time - low watermark (l): msec

HTTP hit transaction time - high watermark (h): msec

A percentage of the new client connections will be bypassed when the average HTTP hit transaction time is above (l), with all the new client connections being bypassed when the average HTTP hit transaction time crosses (h)

[Make These Changes](#)

SNMP

SNMP Agent: On Off

[Make These Changes](#)

Setting Browser Auto-Configuration




- Users will need to set their browsers to connect to your Intel Cache auto-configuration file as appropriate for each browser

The screenshot shows the Intel NetStructure 1500 Cache configuration interface. At the top left is the Intel logo, and at the top right is the text "Intel® NetStructure™ 1500 Cache" and a "Powered by Inktomi" logo. Below the Intel logo is a navigation menu with two tabs: "MONITOR" and "CONFIGURE". The "CONFIGURE" tab is active. The menu contains several icons with labels: "Server", "Protocols", "Cache", "Security", "Routing", "Host DB", and "Snapshots". At the bottom of the menu is a red "Help" button. The main content area is titled "Configure: Browser Auto-Configuration" and contains the text "There is no autoconfiguration file. [Create One](#)". Below this text is a green arrow icon pointing left, followed by the text "[Configure: Server Basics](#)".







Setting Browser Auto-Configuration



Intel® NetStructure™ 1500 CachePowered by Inktomi

MONITOR CONFIGURE

Configure: Create Auto-Configuration File

-  Server
-  Protocols
-  Cache
-  Security
-  Routing
-  Host DB
-  Snapshots

Bypass the Proxy Server for:

Hosts with non-qualified domain name

For hosts in the following domains: (whitespace separated list)

For the following hosts: (whitespace separated list)

Failover Options

Internal Cluster Failover

Failover to Secondary Proxy

Hostname: Port:

Go Direct as Last Resort

Help

Manager UI: Server: Load Shedding



inktom i

- The Intel Cache can restrict the number of network connections it will accept to prevent system overload if a traffic bottleneck develops
- Non-transparency, Intel Cache uses Throttles
- Transparency, Intel Cache utilizes ARM to perform load shedding

intel Intel® NetStructure™ 1500 Cache Powered by Inktomi

MONITOR **CONFIGURE**

Server **Protocols**

Cache **Security**

Routing **Host DB**

Snapshots

Help

Auto-Configuration of browsers

[Auto-configuration file](#)

Load Shedding

HTTP hit transaction time - low watermark (l): msec

HTTP hit transaction time - high watermark (h): msec

A percentage of the new client connections will be bypassed when the average HTTP hit transaction time is above (l), withall the new client connections being bypassed when the average HTTP hit transaction time crosses (h)

SNMP

SNMP Agent: On Off

Manager UI: Server: Configure SNMP



- Intel Cache supports SNMP
 - View performance information about the Intel Cache
 - Warning messages (SNMP traps) to SNMP monitoring stations
- Two Management Information Bases (MIBs)
 - Master Agent is MIB-2 (standard MIB)
 - Intel Cache MIB (contains node-specific and cluster-wide information)

intel Intel® NetStructure™ 1500 Cache Powered by Inktomi

MONITOR **CONFIGURE**

Server Protocols

Cache Security

Routing Host DB

Snapshots

Help

Auto-Configuration of browsers

[Auto-configuration file](#)

Load Shedding

HTTP hit transaction time - low watermark (l): msec

HTTP hit transaction time - high watermark (h): msec

A percentage of the new client connections will be bypassed when the average HTTP hit transaction time is above (l), with all the new client connections being bypassed when the average HTTP hit transaction time crosses (h)

SNMP

SNMP Agent: On Off

**Intel Cache:
Manager UI Protocols/ CLI
Config->Protocols**



i n k t o m i

Chapter Overview



inkomi

- **Explore Configuration Options**
- **Practice Lab**

CLI: config menu



inktom i

- Similar menu items to buttons in Manager UI Configure menu

```
Telnet - rubicon-32
Connect Edit Terminal Help
+-----+
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
+-----+

+-----root->config-----+
| -server   General Intel Cache Controls |
| protocols Protocols (HTTP/NNTP etc)  |
| cache     Configure Cache              |
| security  Configure Security           |
| routing   Configure Routing            |
| arm       Adaptive Redirection Module  |
| hostdb    Host Database Lookup         |
| logging   Configure Logging           |
+-----+

General Intel Cache Controls
```

Manager UI: Protocols



inktom i

Configuring HTTP, NNTP,
HTTPS, and FTP

The screenshot shows the Intel NetStructure 1500 Cache Manager UI. The top left features the Intel logo and the product name "Intel® NetStructure™ 1500 Cache". The top right has a "Powered by Inktomi" logo. A navigation sidebar on the left is divided into "MONITOR" and "CONFIGURE" sections. The "CONFIGURE" section is highlighted in orange and contains icons for "Server", "Cache", "Routing", "Snapshots", "Protocols", "Security", and "Host DB". The "Protocols" icon is selected and highlighted in red. A red arrow points from the text "Configuring HTTP, NNTP, HTTPS, and FTP" to this icon. Below the sidebar is a red "Help" button. The main content area is titled "Configure: Protocols" and is currently showing the "HTTP" configuration page. The page includes a description: "Keep-alive time-outs set how long idle keep-alive connections remain open." and several input fields: "Keep-Alive Timeout Inbound: 30 seconds", "Keep-Alive Timeout Outbound: 10 seconds", "Inactivity Timeout Inbound: 120 seconds", "Inactivity Timeout Outbound: 120 seconds", "Activity Timeout Inbound: 7200 seconds", and "Activity Timeout Outbound: 7200 seconds". A note at the bottom states: "Remove HTTP headers to increase the privacy of your site and users." followed by the text "Remove the following common headers:".

Manager UI: Protocols HTTP Details



- Tune HTTP and FTP timeouts and set user privacy features
 - Keep-alive timeouts (hold a connection open for a subsequent request)
 - Inactivity timeouts (hold connections open if a transaction stalls)
 - Inbound (connections to users)
 - Outbound (connections to servers)

The screenshot shows the 'Configure: Protocols' page in the Intel NetStructure 1500 Cache management interface. The interface includes a navigation sidebar on the left with options for Server, Protocols, Cache, Security, Routing, Host DB, and Snapshots. The main content area is titled 'HTTP' and contains several configuration fields for timeouts and activity limits. The 'Keep-alive' section includes 'Keep-Alive Timeout Inbound' (30 seconds) and 'Keep-Alive Timeout Outbound' (10 seconds). The 'Inactivity timeouts' section includes 'Inactivity Timeout Inbound' (120 seconds) and 'Inactivity Timeout Outbound' (120 seconds). The 'Activity timeouts' section includes 'Activity Timeout Inbound' (7200 seconds) and 'Activity Timeout Outbound' (7200 seconds). There is also a section for 'Remove HTTP headers to increase the privacy of your site and users' with a dropdown menu for selecting common headers to remove.

intel. Intel® NetStructure™ 1500 Cache Powered by Inktomi

MONITOR CONFIGURE **Configure: Protocols**

Server Protocols

Cache Security

Routing Host DB

Snapshots

Help

HTTP

Keep-alive time-outs set how long idle keep-alive connections remain open.

Keep-Alive Timeout Inbound: seconds

Keep-Alive Timeout Outbound: seconds

Inactivity timeouts set how long the Intel(r) Netstructure(tm) 1500 Cache waits to abort stalled transactions.

Inactivity Timeout Inbound: seconds

Inactivity Timeout Outbound: seconds

Activity timeouts limit the duration of transactions.

Activity Timeout Inbound: seconds

Activity Timeout Outbound: seconds

Remove HTTP headers to increase the privacy of your site and users.

Remove the following common headers:

Manager UI: Protocols HTTP Privacy



inktom i

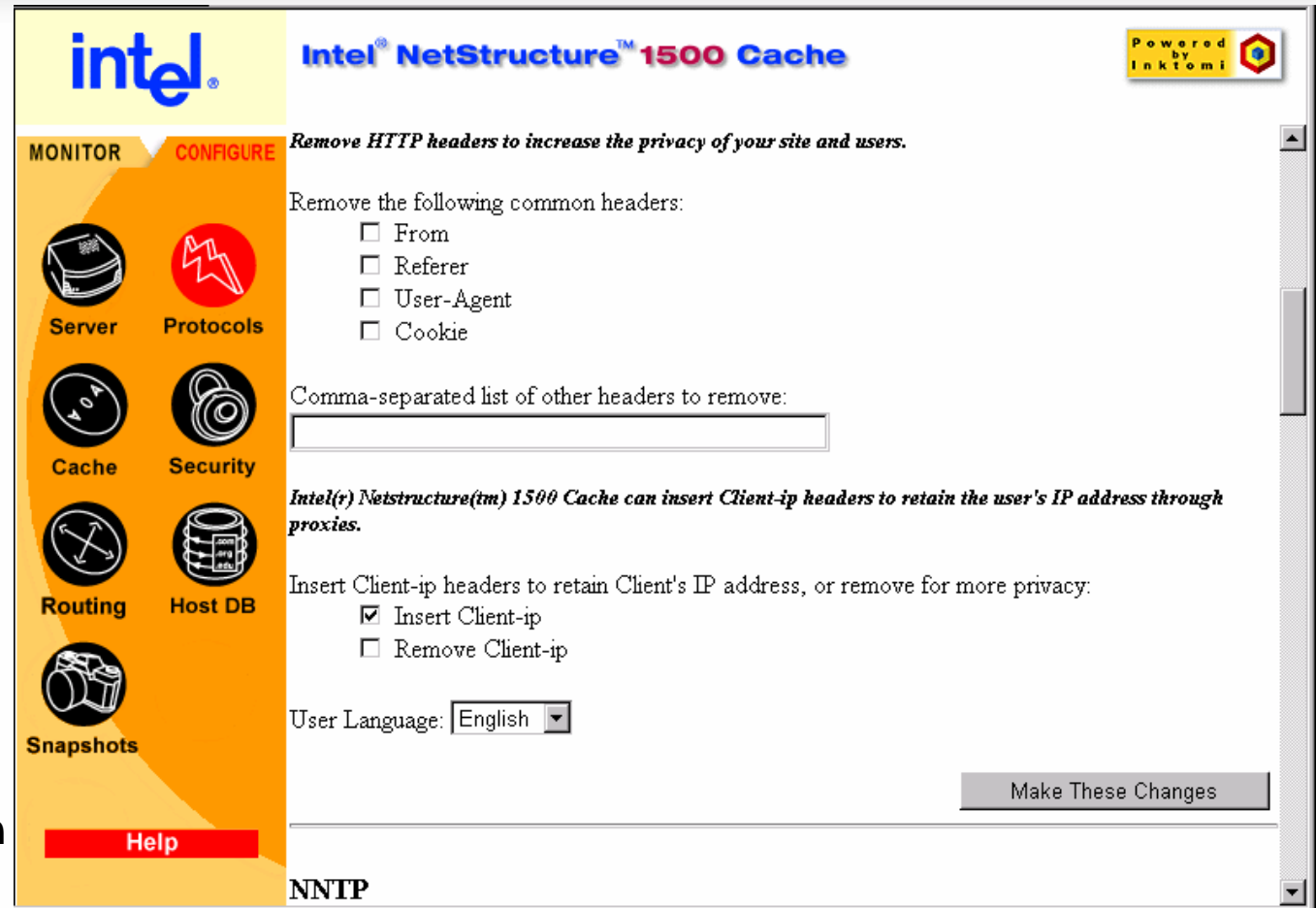
- Remove these headers to protect the privacy of your site:
 - The from header (user's email address)
 - The referred header (the link followed by the user)
 - The browser making the request
 - The cookie field (which often identifies the user)
 - Customized header removal

The screenshot shows the Intel NetStructure 1500 Cache Manager interface. The top navigation bar includes the Intel logo, the product name "Intel NetStructure™ 1500 Cache", and a "Powered by Inktomi" badge. The main content area is divided into two tabs: "MONITOR" and "CONFIGURE". The "CONFIGURE" tab is active, showing a sidebar with icons for "Server", "Cache", "Routing", "Snapshots", "Protocols", "Security", and "Host DB". The "Protocols" icon is highlighted in red. The main panel under "CONFIGURE" contains the following text and controls:

- Remove HTTP headers to increase the privacy of your site and users.**
- Remove the following common headers:
 - From
 - Referer
 - User-Agent
 - Cookie
- Comma-separated list of other headers to remove:
- Intel(r) Netstructure(tm) 1500 Cache can insert Client-ip headers to retain the user's IP address through proxies.*
- Insert Client-ip headers to retain Client's IP address, or remove for more privacy:
 - Insert Client-ip
 - Remove Client-ip
- User Language:
-
- NNTP

Retaining HTTP Headers

- Client IP headers can be inserted or deleted by Intel Cache
 - Retain Client IP header because the Origin Server accepts or denies based on source IP
 - Remove Client IP header to conceal client's identity
- Messages from Intel Cache to user are displayed by default in English



The screenshot shows the Intel NetStructure 1500 Cache Manager interface. The top left features the Intel logo and the product name "Intel® NetStructure™ 1500 Cache". A "Powered by Inktomi" logo is in the top right. The interface is divided into two main sections: "MONITOR" and "CONFIGURE". The "CONFIGURE" section is active, showing options to "Remove HTTP headers to increase the privacy of your site and users." Below this, there are checkboxes for "From", "Referer", "User-Agent", and "Cookie". A text input field is provided for a "Comma-separated list of other headers to remove:". Below that, a note states: "Intel(r) Netstructure(tm) 1500 Cache can insert Client-ip headers to retain the user's IP address through proxies." There are checkboxes for "Insert Client-ip" (checked) and "Remove Client-ip". A "User Language" dropdown menu is set to "English". A "Make These Changes" button is at the bottom right. A "Help" button is visible in the "MONITOR" section. The "NNTP" protocol is indicated at the bottom.

Client-ip headers and User Language can only be configured through the Manager UI

CLI: config->protocols



inktom i

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----root-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----root-----
| server      General Intel Cache Controls |
| protocols   Protocols (HTTP/NNTP etc)   |
| cache       Configure Cache              |
| security    Configure Security           |
| routing     Configure Routing            |
| arm         Adaptive Redirection Module  |
| hostdb      Host Database Lookup         |
| logging     Configure Logging            |
-----root-----

Protocols (HTTP/NNTP etc)
```

CLI: config->protocols menu



inktom i

- Differs from Manager UI configure protocols:
 - Does not have https item (ssl is closest item)
 - ssl, filter, remap items not in Manager UI

```
Telnet - rubicon-32
Connect Edit Terminal Help
+-----+
| Intel(r) NetStructure(tm) 1500 Cache |
+-----+

+-----root->config->protocols-----+
| http  Configure HTTP (WEB) |
| nntp  Configure NNTP (NEWS) |
| ssl   Secure Sockets Layer |
| ftp   File Transfer Protocol |
| filter Filter Rules         |
| remap Remap Rules          |
+-----+

Configure HTTP (WEB)
```

CLI: config->protocols->http



inktom

- Options to view or change HTTP settings and headers

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----

+-----root->config->protocols->http-----+
| view      View Current HTTP Settings      |
| remove    Remove HTTP Headers             |
+-----+

View Current HTTP Settings
```


CLI: config->protocols->http->remove



inktom i

- Same header options as Manager UI
- Currently no way to restore headers

```
Telnet - rubicon-52
Connect Edit Terminal Help
-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----

+-----root->config->protocols->http->remove-----+
| From          Remove From Header                    |
| Referer       Remove Referer Header                 |
| User-agent    Remove User-agent Header              |
| Cookie       Remove Cookie Header                  |
+-----+

Remove From Header
```

Manager UI: Protocols NNTP



inktom i

- Enable/Disable Intel Cache to cache and serve news articles by turning NNTP server on or off
- Caution: you must click the Restart button to activate this change
- This page allows you to configure basic NNTP options to servers

The screenshot shows the Intel NetStructure 1500 Cache Manager interface. At the top, there is the Intel logo and the text "Intel® NetStructure™ 1500 Cache". In the top right corner, it says "Powered by Inktomi". The interface is divided into two main sections: "MONITOR" and "CONFIGURE". The "CONFIGURE" section is active, showing the "NNTP" configuration page. On the left side of the "CONFIGURE" section, there is a vertical menu with icons for "Server", "Protocols", "Cache", "Security", "Routing", "Host DB", and "Snapshots". The "Protocols" icon is highlighted. Below the menu is a red "Help" button. The main content area of the "NNTP" page includes:

- "NNTP Server:" with a dropdown menu set to "Off".
- "NNTP Server Port:" with a text input field containing "119".
- "Connect Message (posting allowed):" with a text input field containing "Intel NNTP server ready. posting ok".
- "Connect Message (posting not allowed):" with a text input field containing "Intel NNTP server ready. no posting".
- "NNTP Options:" with a list of checkboxes:
 - Posting
 - Access Control
 - NNTP V2 Authentication
 - Run Local Authentication Server
 - Clustering
 - Allow Feeds
 - Access Logs
 - Background Posting
 - Obey Cancel Control Messages
 - Obey NewGroups Control Messages
 - Obey RmGroups Control Messages

Configuring NNTP Polling



inktom i

- Configure other NNTP options, like inactivity timeout, polling, and authentication server

The screenshot displays the configuration page for the Intel NetStructure 1500 Cache. The interface is divided into a left sidebar and a main content area. The sidebar contains navigation options: MONITOR and CONFIGURE (selected), with sub-options for Server, Protocols, Cache, Security, Routing, Host DB, and Snapshots, along with a Help button. The main content area is titled 'Intel NetStructure™ 1500 Cache' and features a 'Powered by Inktomi' logo. It lists several configuration parameters for NNTP polling:

- Inactivity Timeout:** 600 seconds. Description: *Inactivity timeout sets how long idle connections remain open. A 3 minute minimum is recommended.*
- Check for New Groups Every:** 66400 seconds. Description: *The lists of groups on parent NNTP servers are checked periodically for new groups. They need not be checked frequently as the list changes slowly.*
- Check for Cancelled Articles Every:** 3600 seconds. Description: *If the Intel(r) Netstructure(tm) 1500 Cache is not set to obey cancel control messages, it can actively poll groups to detect cancelled articles. This should not be done too frequently as it involves communication with the parent NNTP server.*
- Check Parent NNTP Server Every:** 300 seconds. Description: *Poll the parent NNTP Server to see if new articles have appeared this often.*
- Check Cluster Every:** 60 seconds. Description: *Poll the other Intel(r) Netstructure(tm) 1500 Caches in the cluster see if new articles have appeared this often.*

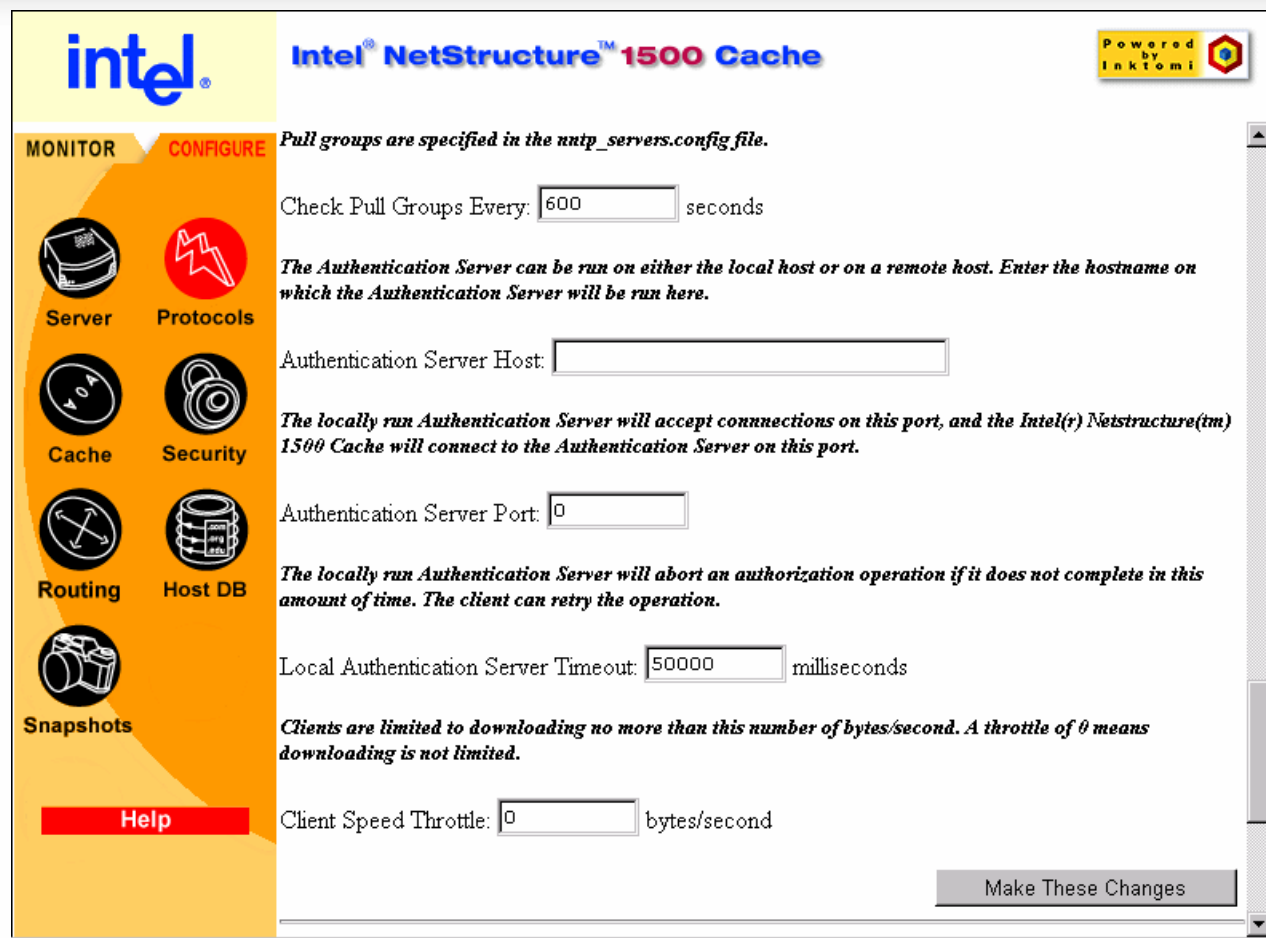
At the bottom, it notes: *Pull groups are specified in the nntp_servers.config file.*

Configuring NNTP Polling



inktom i

- Configure other NNTP options, like inactivity timeout, polling, and authentication server host, port, and timeout



The screenshot shows the configuration interface for Intel NetStructure 1500 Cache. The interface is divided into a left sidebar and a main configuration area. The sidebar has a yellow background and contains the Intel logo at the top. Below the logo are two tabs: 'MONITOR' and 'CONFIGURE', with 'CONFIGURE' being the active tab. The sidebar contains several icons representing different configuration areas: Server, Protocols, Cache, Security, Routing, Host DB, and Snapshots. A red 'Help' button is located at the bottom of the sidebar. The main configuration area has a white background and contains the following settings:

- Check Pull Groups Every:** seconds. Below this is a note: "Pull groups are specified in the nntp_servers.config file."
- Authentication Server Host:** . Below this is a note: "The Authentication Server can be run on either the local host or on a remote host. Enter the hostname on which the Authentication Server will be run here."
- Authentication Server Port:** . Below this is a note: "The locally run Authentication Server will accept connections on this port, and the Intel(r) Netstructure(tm) 1500 Cache will connect to the Authentication Server on this port."
- Local Authentication Server Timeout:** milliseconds. Below this is a note: "The locally run Authentication Server will abort an authorization operation if it does not complete in this amount of time. The client can retry the operation."
- Client Speed Throttle:** bytes/second. Below this is a note: "Clients are limited to downloading no more than this number of bytes/second. A throttle of 0 means downloading is not limited."

At the bottom right of the main configuration area, there is a button labeled "Make These Changes".

CLI: config->protocols->nntp->view

- Show current NNTP configuration settings

```
Telnet - rubicon-32
Connect Edit Terminal Help
+-----+
| Intel(r) NetStructure(tm) 1500 Cache |
+-----+

+-----root->config->protocols->nntp->view-----+
|NNTP Server on(1)/off(0)      0
|Server Port                   119
|Feeds                         Enabled
|Access Control                Disabled
|Inactivity Timeout (secs)    600
|Check For New groups (secs)  86400
+-----+

View Current NNTP Settings
```

CLI: config->protocols->nntp->servers



inktom i

- **NNTP server rules specify newsgroups, parent NNTP servers, types of NNTP activity, and network interfaces used to contact parent NNTP servers**

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----

+-----root->config->protocols->nntp->servers-----+
| add rules      Add NNTP Rules                       |
| delete rules   Delete NNTP Rules                    |
| view rules     View NNTP Rules                      |
+-----+

Add NNTP Rules
```


NNTP Server Rules



inktom

- **NNTP server rules are of the format**
hostname group-wildmat priority interface
where *priority* and *interface* are optional
- **Allowed values for hostname are:**
 - host name
 - host name:port
 - IP address
 - IP address:port
 - **.block (used to block access to specific news groups)**

NNTP Server Rules



inktom i

- NNTP server rules are of the format
hostname group-wildmat priority interface
where *priority* and *interface* are optional
- `group-wildmat` values must be a comma-separated list of group names and list files
 - `*` is a wildcard
 - List file values are subscriptions, distributions, and `distrib.pats`
 - `!` is used to indicate groups that are *not* included
 - List is processed in reverse order, so most specific restrictions should appear at the end of the list

NNTP Server Rules



Intel

- **NNTP server rules are of the format**
hostname group-wildmat priority interface
where *priority* and *interface* are optional
- **priority values tell the Intel Cache how to treat the specified host and newsgroups:**
 - **<no priority tag>** caches articles on demand
 - **feed** Cache receives news feeds as the parent NNTP server receives feeds
 - **push** Cache can both receive news feeds and cache articles
 - **post** Articles to be posted to newsgroups are sent to the specified server

NNTP Server Rules



Intel

- **priority** values tell the Intel Cache how to treat the specified host and newsgroups:
 - **pull** Cache actively pulls (caches) all articles at a frequency specified in the Manager UI
 - **pullover** Cache actively polls the overview database for the newsgroups but retrieves news articles on demand (pull and pullover must be preceded by a “cache on demand” line)
 - **dynamic** Cache decides, based on usage patterns, if a group is pull, pullover, or demand retrieval-based
 - **<positive integer>** Cache retrieves articles on demand according to the specified priority (defaults to 0). Multiple servers with the same priority are accessed round-robin

NNTP Server Rules



inktom i

- **NNTP server rules are of the format**
hostname group-wildmat priority interface
where *priority* and *interface* are optional
- **interface values are the network interface the Intel Cache uses to contact the parent NNTP server**

NNTP Server Rule Samples



inktom i

- **Block all requests from rec. groups except rec.soccer**
`.block !rec.soccer.rec.*`
- **Set the port associated with the hostname**
`news.webhost.com:999 *`
- **Associate an interface and priority with an IP address**
`news.webhost.com * 0 10.3.3.2`
- **Establish priorities for hostnames**
`news.webhost.com * 0`
`news.backup.com * 1`
- **Define a pull group**
`comp.webhost.com alt.*`
`comp.webhost.com alt.bicycles pull`

CLI: config->protocols->nntp->servers->view rules



inktom i

- **view rules** starts with a format description and rule examples

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----+-----
|                               |
| Intel(r) NetStructure(tm) 1500 Cache |
|                               |
-----+-----

1:#
2:# NNTP configuration
3:#
4:# $Id: nntp_servers.config,v 1.8 1999/06/23 00:16:28 elam Exp $
5:#
6:# The format of this file is a sequence of lines:
7:#
8:#   hostname <group-wildmat> (<priority>) (<interface>)
9:#
10:# which describe the upstream servers for particular groups.
11:#
12:# hostname is a hostname or IP and option port. The special token
13:# ".block" means block access to these newsgroups.
14:#
15:#   Examples:
16:#       localhost:120
17:#       qqqqq.com
18:#       10.100.34.1:9999
19:#       10.2.2.1
20:#       .block
```


NNTP Access Rules



inktom i

- **NNTP access rules are of the format**
client-group access=value authenticator users passwords
where *authenticator*, *users*, and *passwords* are optional
- **client-group values must be:**
 - **IP range** `ip=0.0.0.0-255.255.255.255`
 - **Domain** `domain=intel.com`
 - **Hostname** `hostname=myhost.mydomain.com`

NNTP Access Rules



inktom i

- **NNTP access rules are of the format**
client-group access=value authenticator users passwords
where *authenticator*, *users*, and *passwords* are optional
- **access values, along with authenticator, user, password status, are:**

– allow	not required	not required	not required
– deny	not required	not required	not required
– basic	not required	required	optional
– generic	optional	not required	not required
– custom	required	optional	optional

(only allowed entry for user and password is required for custom)

NNTP Access Rule Example



inktom i

- **NNTP access rules are of the format**
client-group access=value authenticator users passwords
where *authenticator*, *users*, and *passwords* are optional
- `ip=127.0.0.1 access="custom" authenticator="hb" user=required pass=required`

CLI: config->protocols->nntp->access->view rules



inktom i

- **view rules** starts with rule format and examples

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----+-----
|                               Intel(r) NetStructure(tm) 1500 Cache                               |
-----+-----

0:#
1:#
2:# nntp_access.config
3:#
4:# This file is not consulted unless nntp.access_control is enabled.
5:#
6:# There are three ways of specifying groups of clients:
7:# by IP range, domain and hostname.
8:#
9:# Examples:
10:#
11:# ip = 0.0.0.0-255.255.255.255
12:# ip = 127.0.0.1
13:# domain = inktomi.com
14:# hostname = myhost.mydomain.com
15:#
16:# For each group of clients an access directive can be given. The directive
17:# are "allow", "deny", "basic", "generic", and "custom".
18:#
19:# For 'basic', the "user" option is required and "pass" is optional.
20:#
```

CLI: config->protocols->nntp->access->view rules



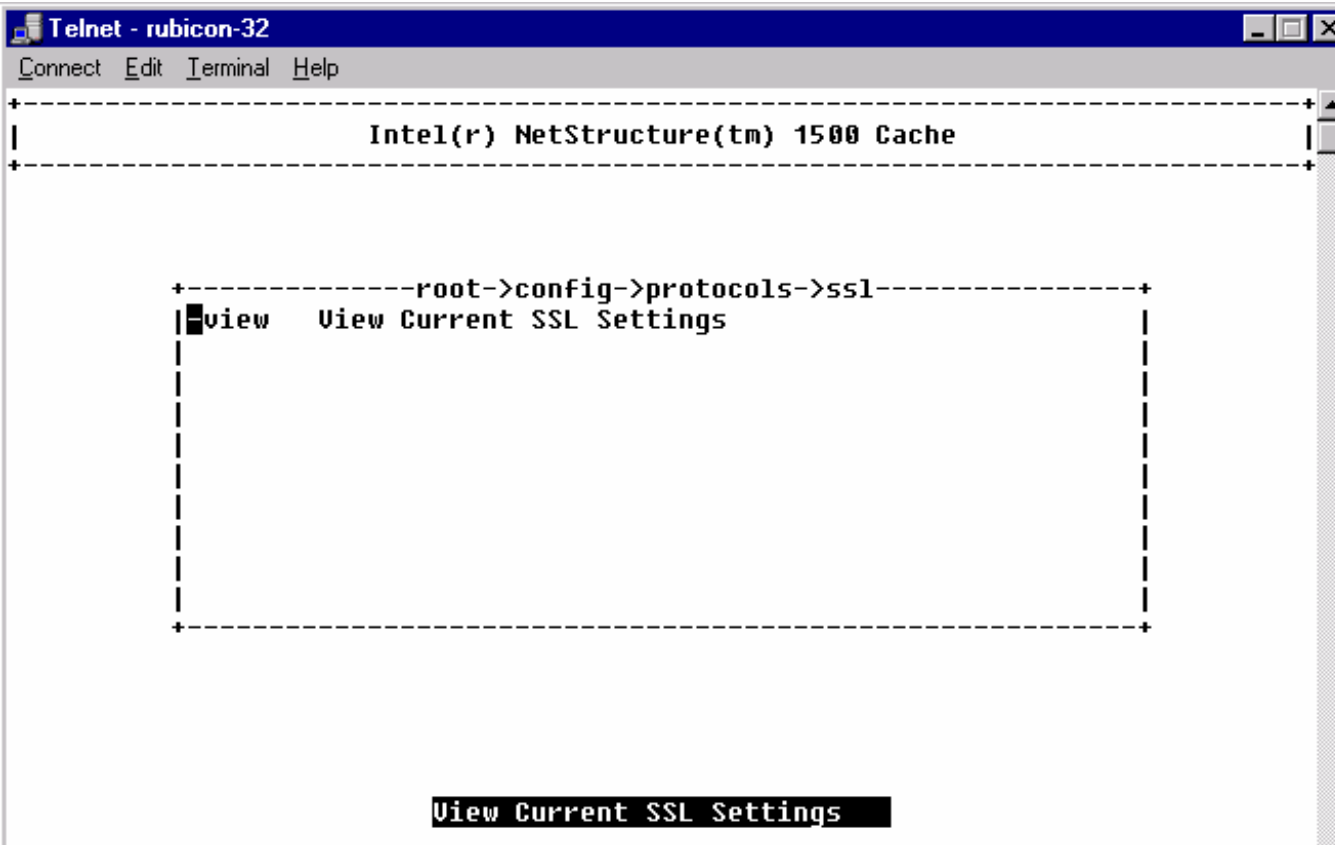
inktom i

```
Telnet - rubicon-32
Connect Edit Terminal Help
-----+-----
|                               |
| Intel(r) NetStructure(tm) 1500 Cache |
|                               |
-----+-----

19:# For 'basic', the "user" option is required and "pass" is optional.
20:#
21:# hostname = myhost.mydomain.com access="basic" user="joe" pass="bob"
22:#
23:# For 'generic', the "authenticator" option is optional.
24:#
25:# For 'custom', the "authenticator" option is required. The "user" and
26:# "pass" options are optional, but if present must be the string 'required'
27:# indicating that the user and password will be required of the user and
28:# then passed to the "authenticator" program through the "generic" interfac
29:#
30:# hostname = myhost.mydomain.com access="custom" authenticator = "homebrew"
31:#
32:# Clients can also be restricted to a subset of groups or prevented from
33:# posting.
34:#
35:# ip = 199.0.0.0-200.0.0.0 access=allow groups=*,!alt.* posting=deny
36:#
37:# Directives that appear earlier in the file take precedence.
38:#
39:# The default is no access.
40:#
```

CLI: config->protocols->ssl

- Only handled by CLI; no access via Manager UI



```
Telnet - rubicon-32
Connect Edit Terminal Help
+-----+
| Intel(r) NetStructure(tm) 1500 Cache |
+-----+

+-----root->config->protocols->ssl-----+
| view View Current SSL Settings |
+-----+

View Current SSL Settings
```