Introduction to Intel Cache

Inktomi

Intel Cache Course Overview

- 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

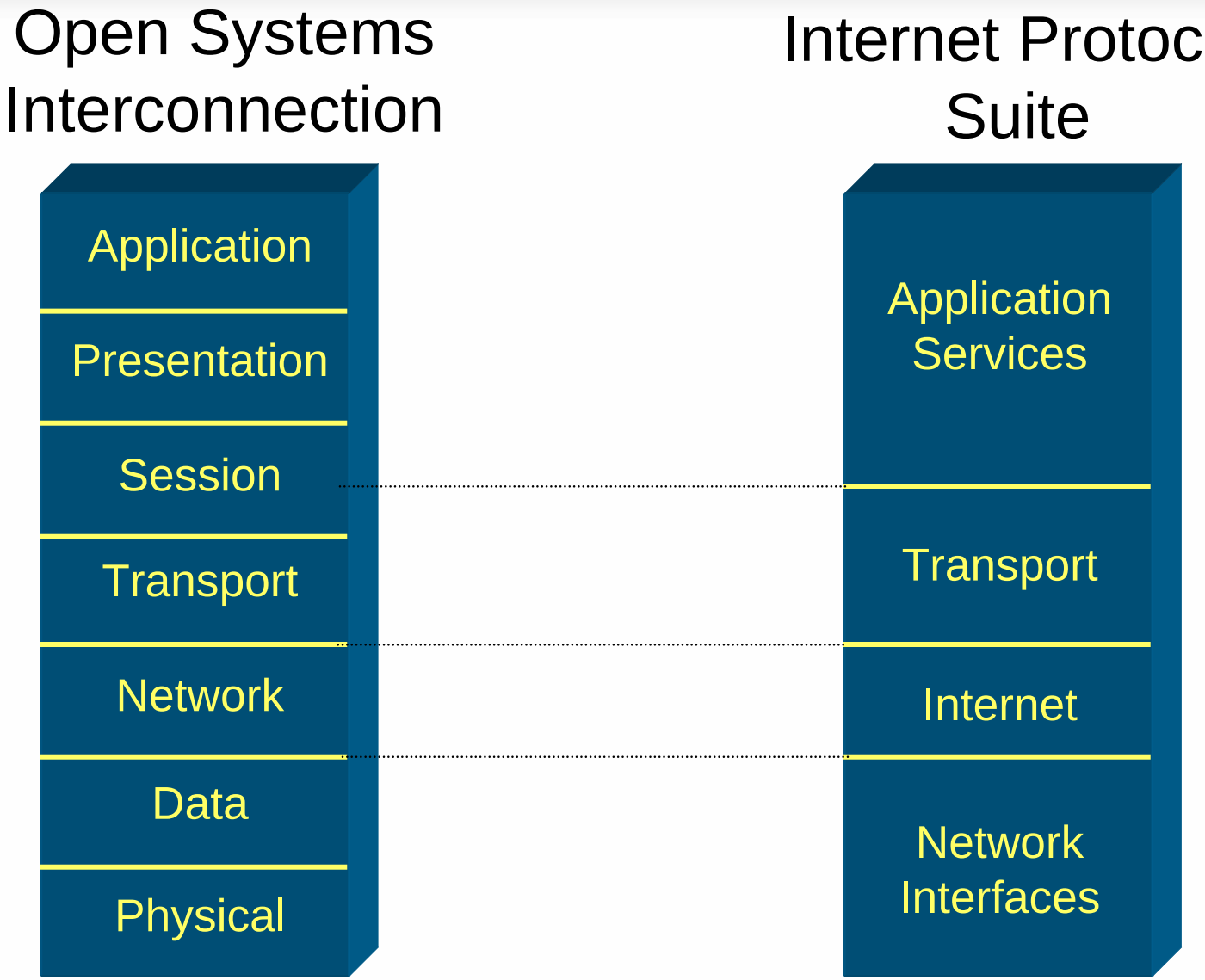
- Intel Cache: Configuration
- Intel Cache: Monitoring
- Intel Cache: Miscellaneous

Technology:
TCP/IP Overview

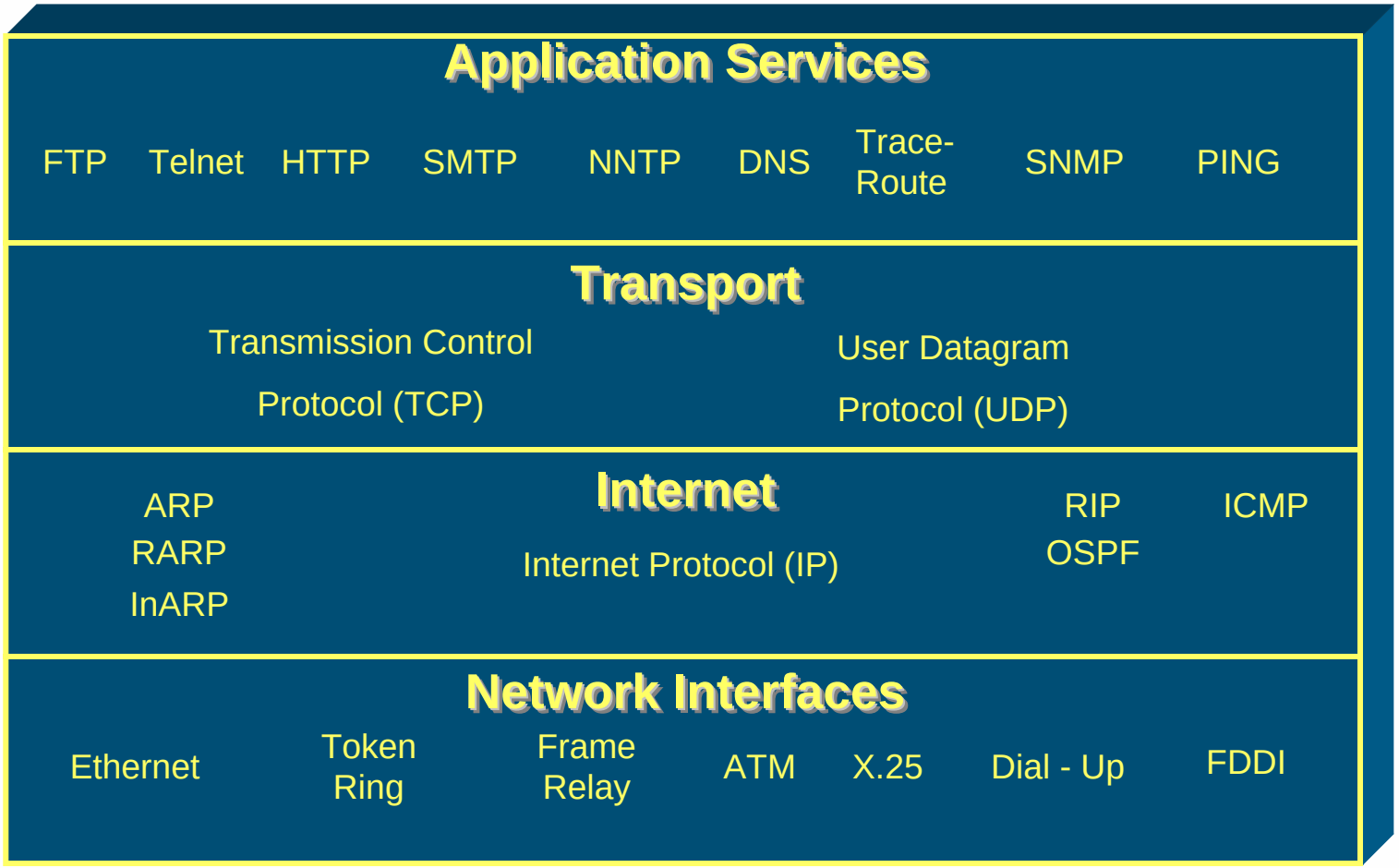
Chapter Overview

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

Comparing OSI & IP Suite



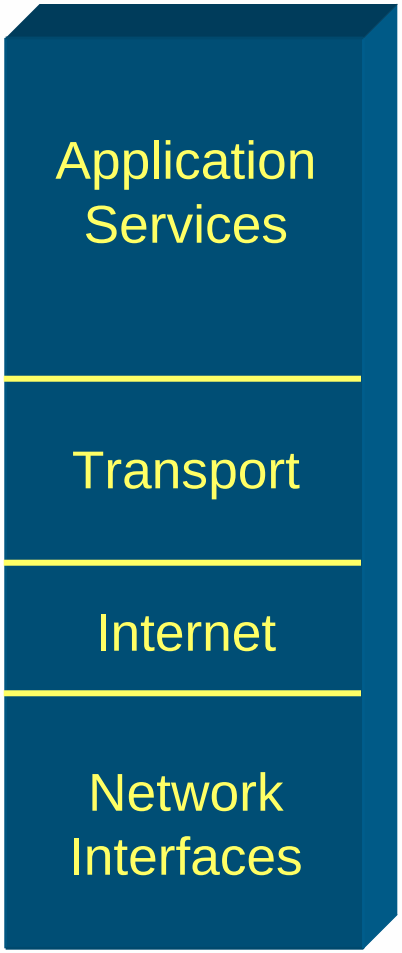
Internet Protocol Suite



Addressing

Inktomi

Internet Protocol Suite



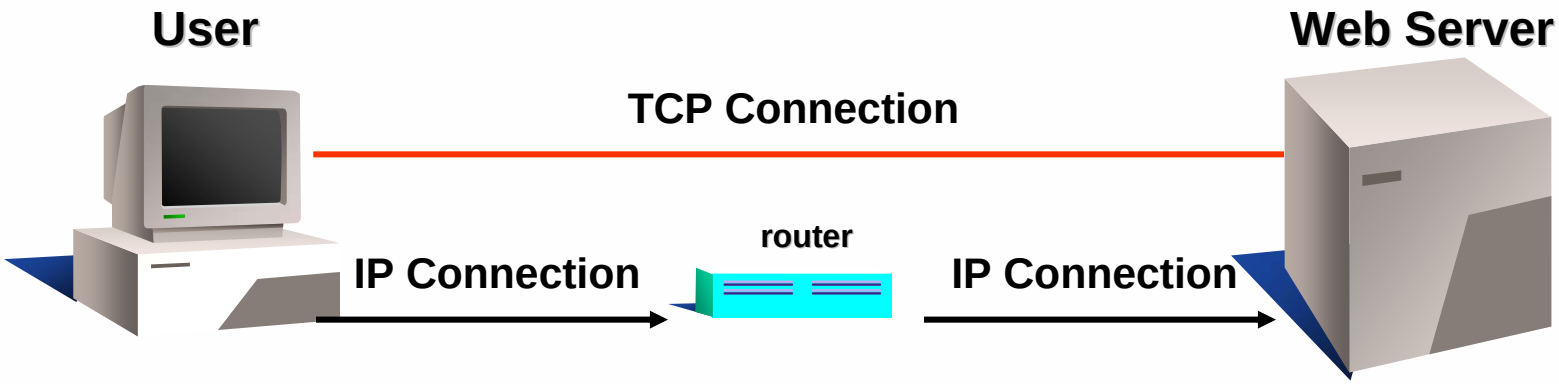
There is always a Source and Destination Address

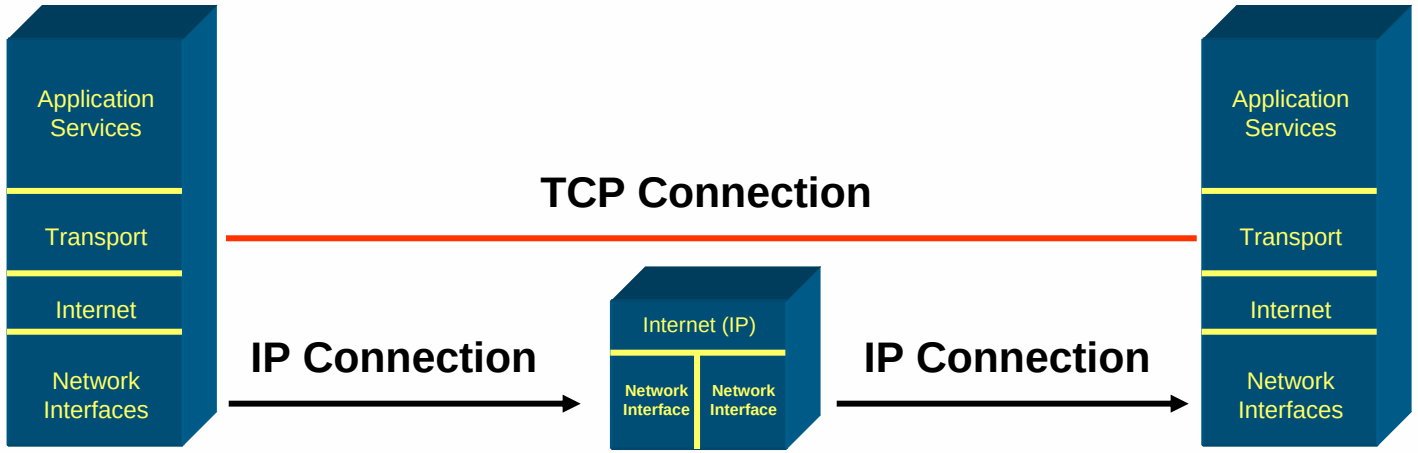
Transport → Port Address

Internet → IP Address

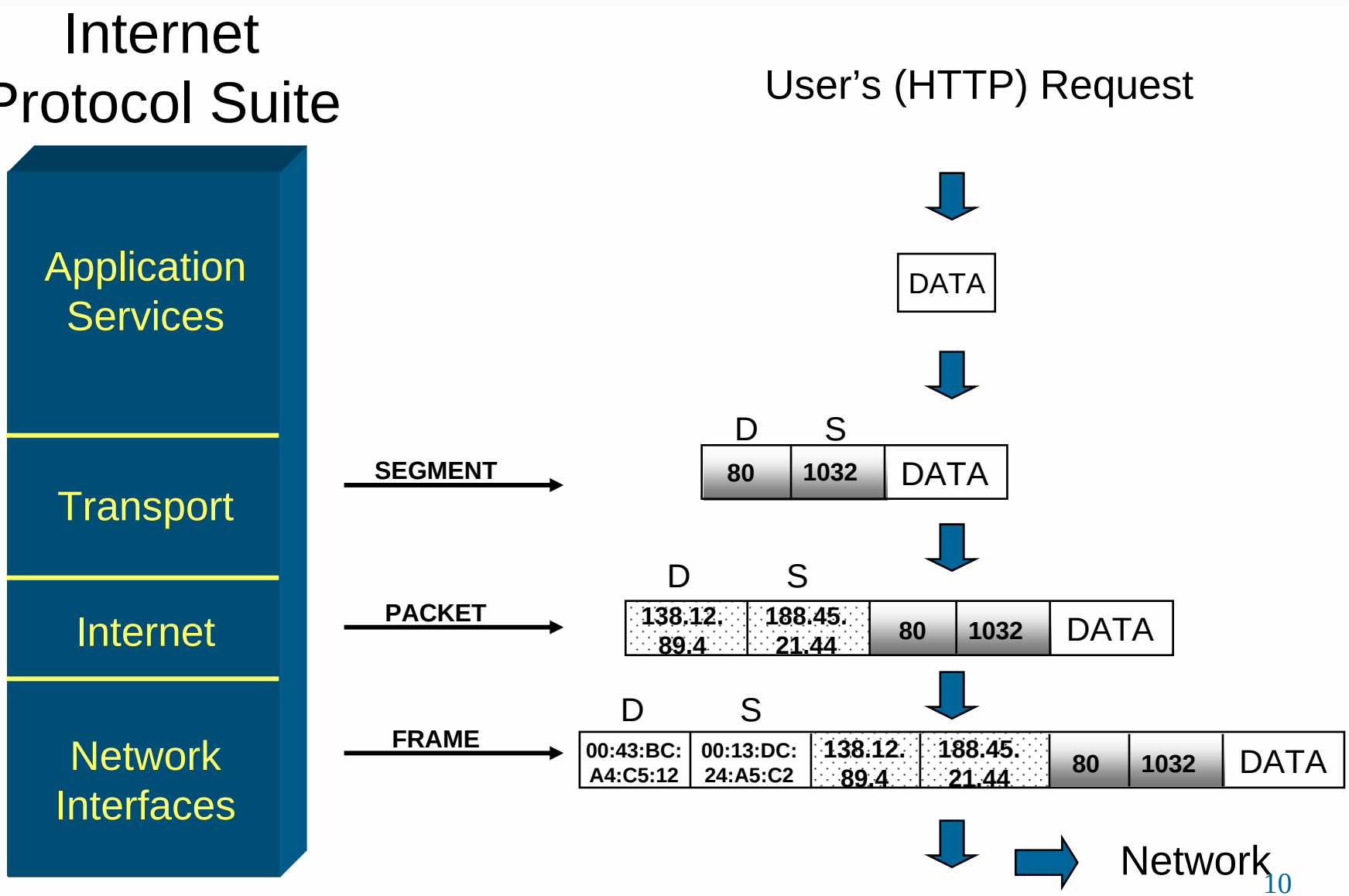
Network Interfaces →
Ethernet/Token Ring = MAC
Frame Relay = DLCI
ATM = NSAPA

Requesting A Web Page



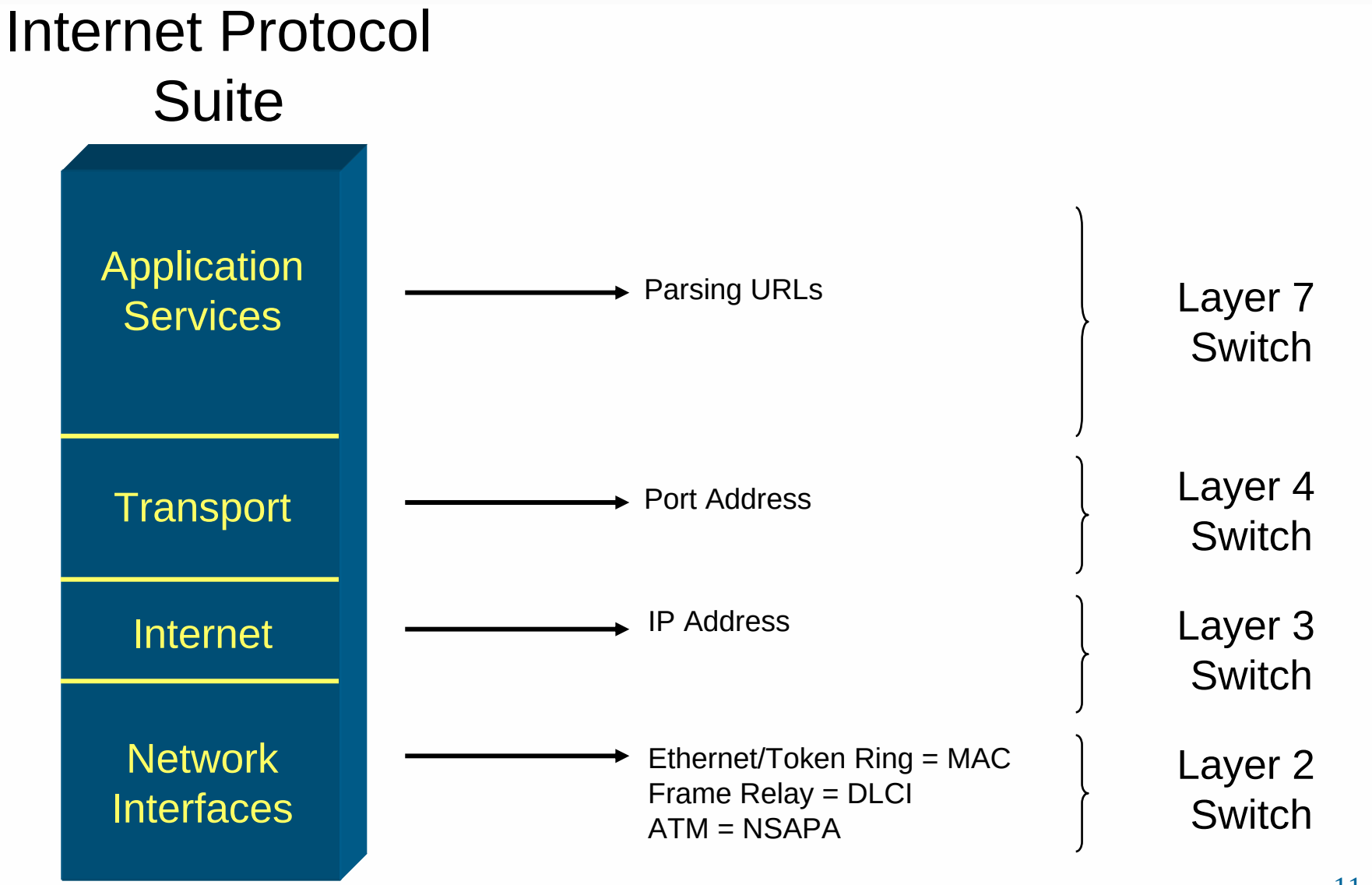


Building A Request

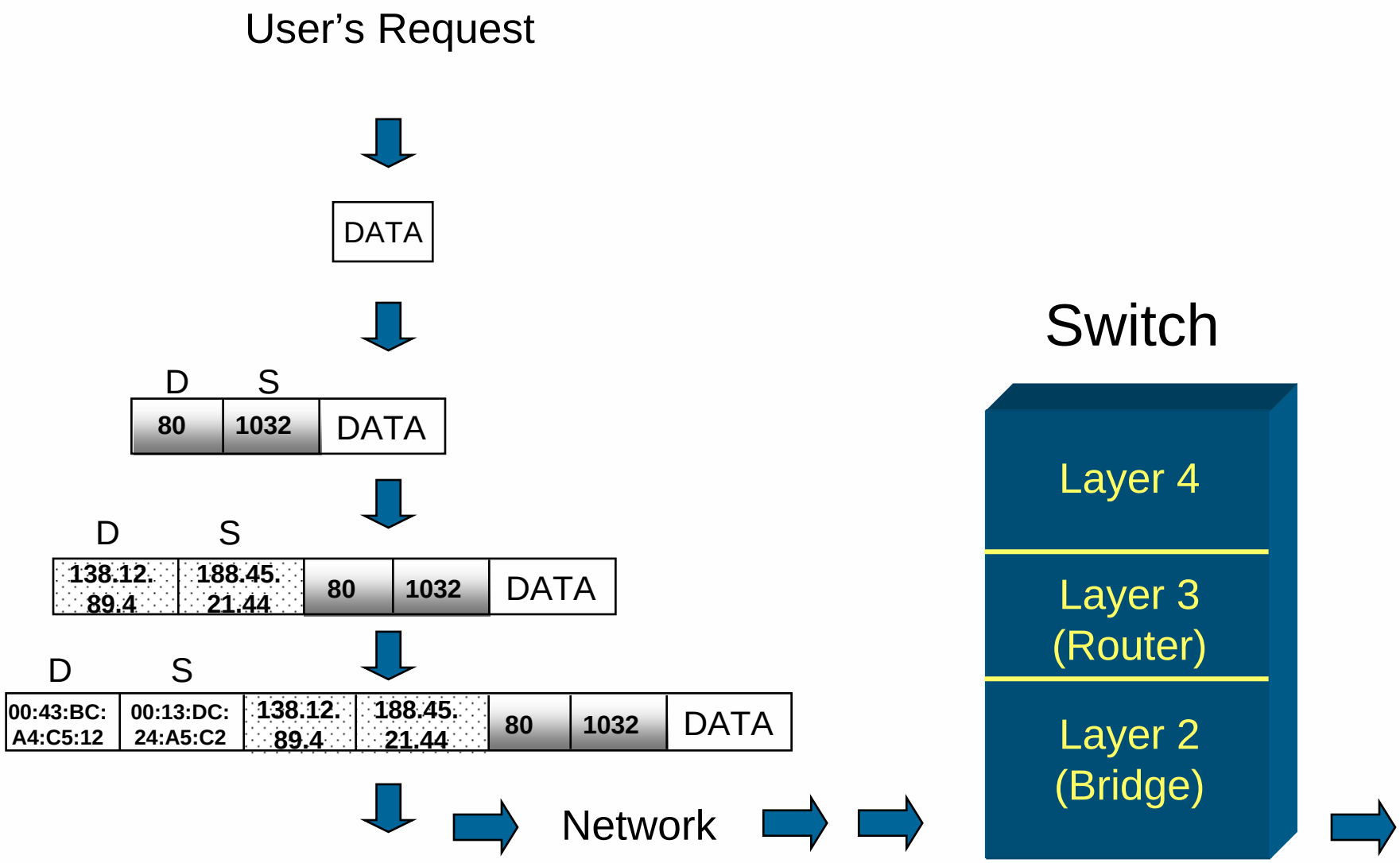


Layer Switches

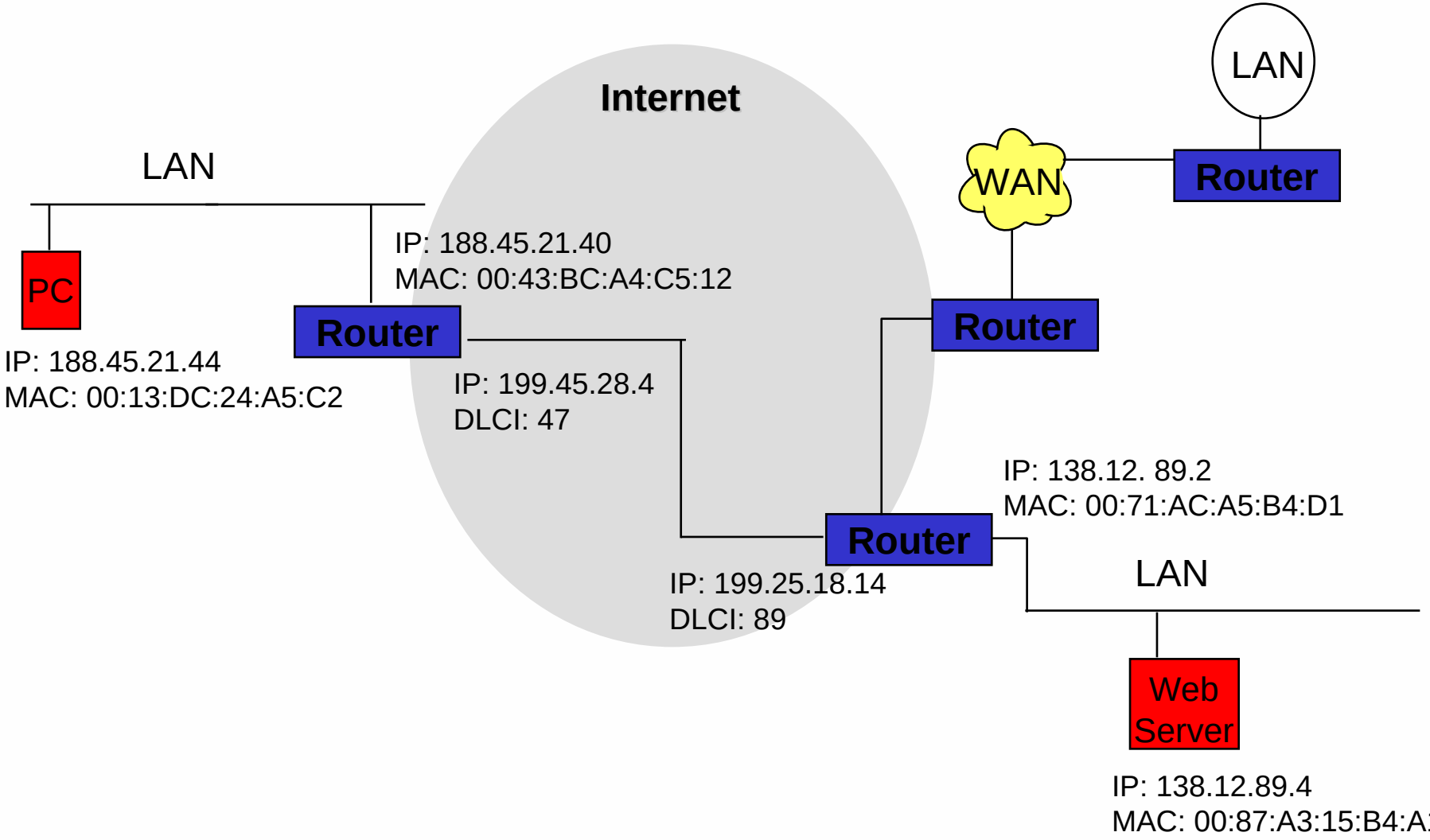
Inktomi



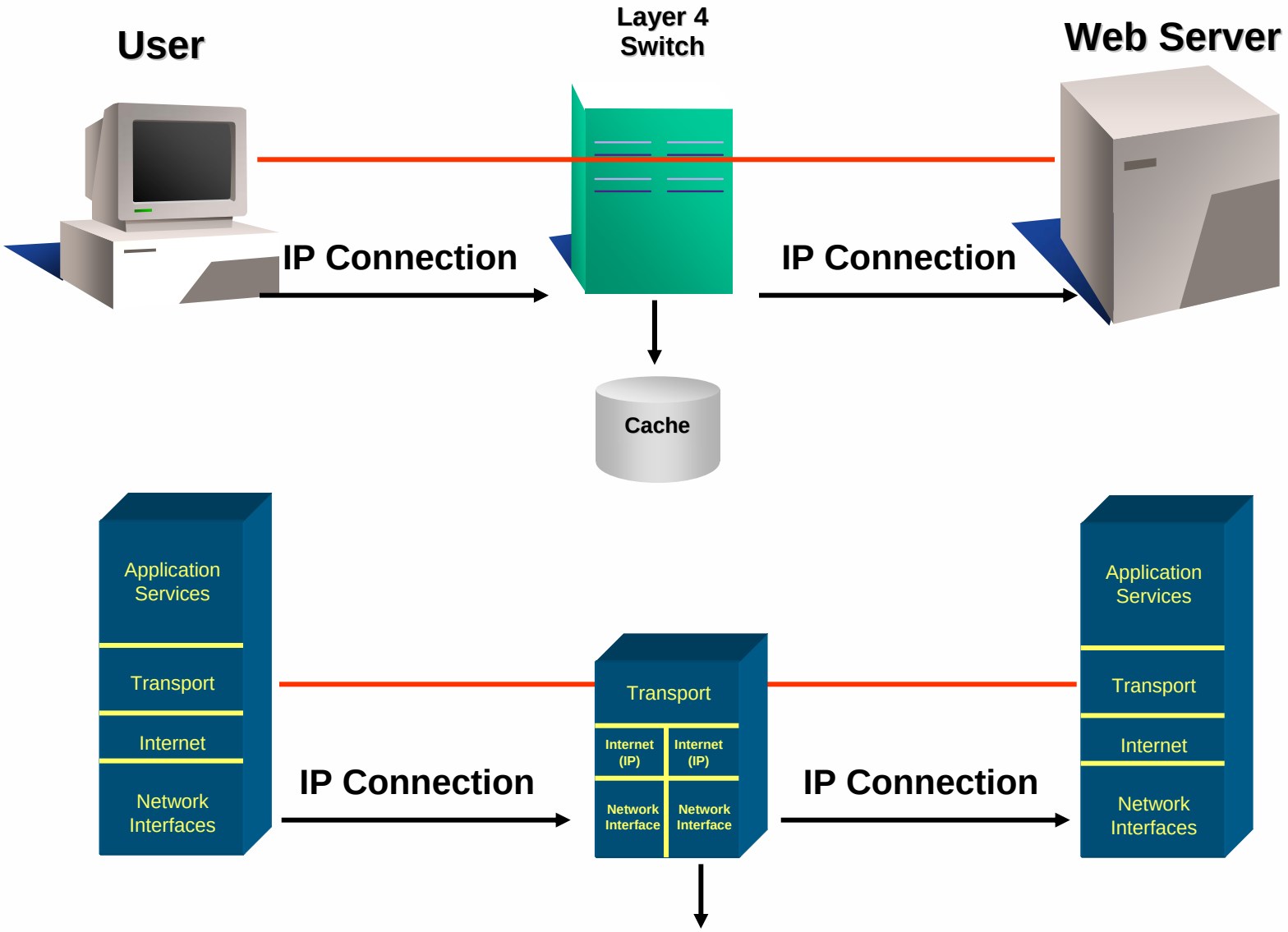
Switching


User's Request
DATA
D
S
80
1032
DATA
D
S
138.12.
89.4
188.45.
21.44
80
1032
DATA
D
S
00:43:BC:
A4:C5:12
00:13:DC:
24:A5:C2
138.12.
89.4
188.45.
21.44
80
1032
DATA
Network
Switch
Layer 4
Layer 3
(Router)
Layer 2
(Bridge)

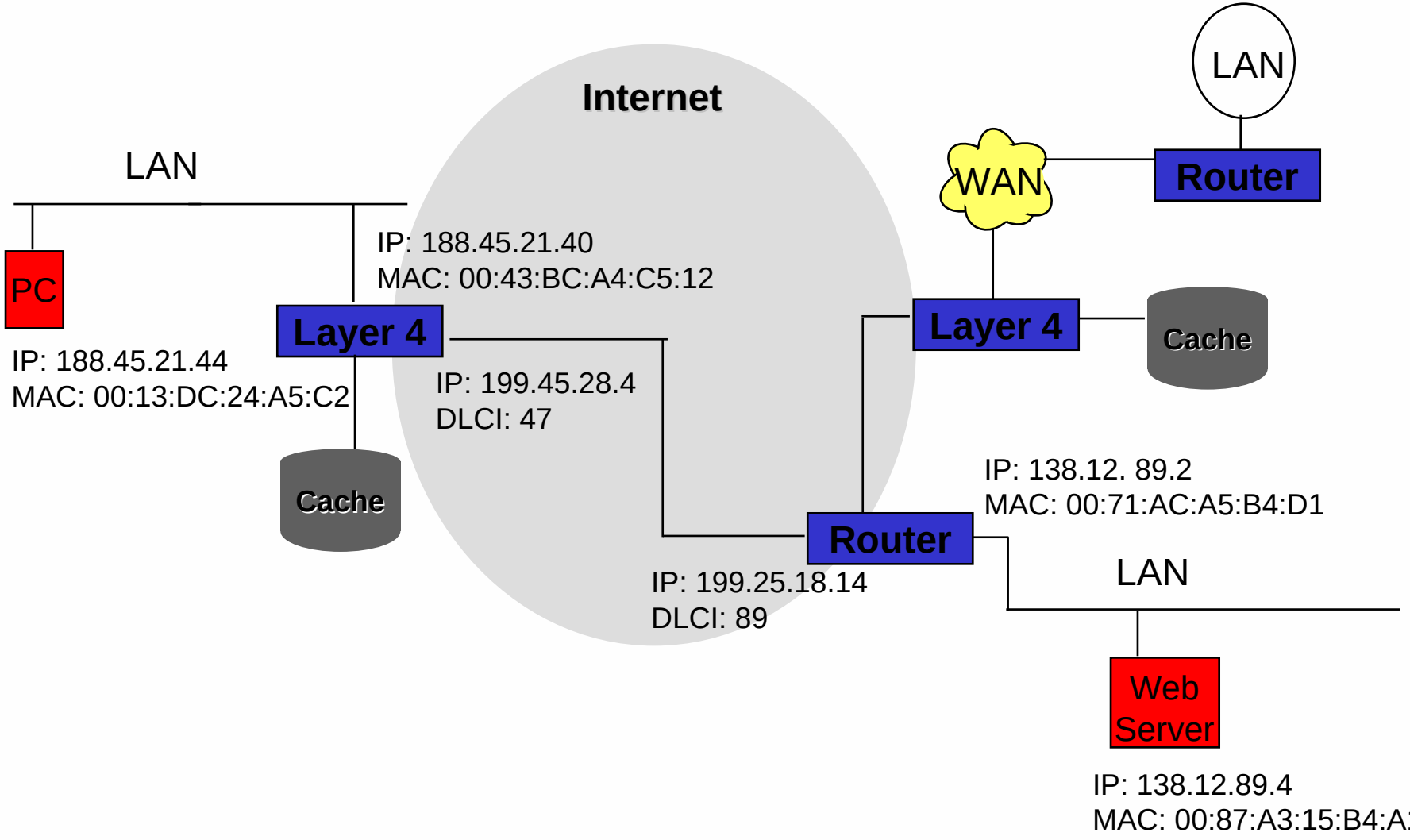
Routing A Request



Layer 4 Connection



Routing A Request



Review Questions:

- What is the benefit of the TCP/IP pairing?

- Why has TCP/IP become the protocol of choice?

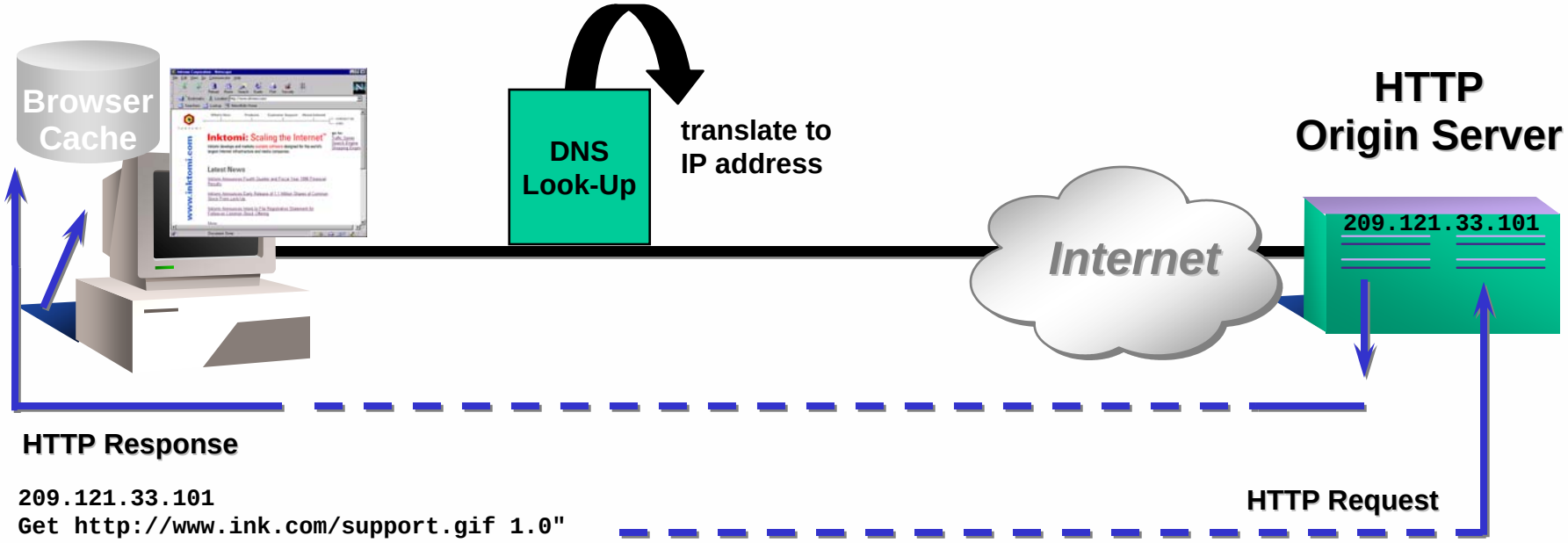
Proxy Service: Overview

Inktomi

Chapter Overview

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

The Direct Connection



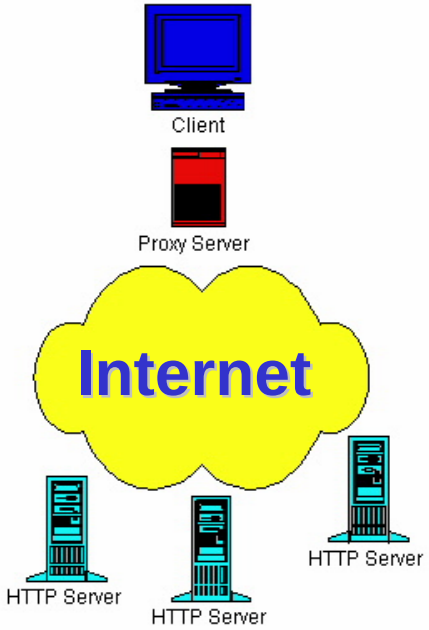
Drawbacks of Direct Connections

- **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**

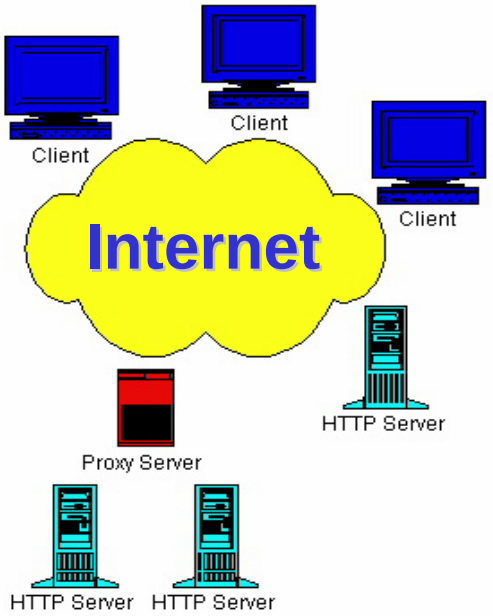
Proxy Cache: Forward & Reverse

Inktomi

Forward Proxy Cache
Cache server acts as a proxy for client requests.



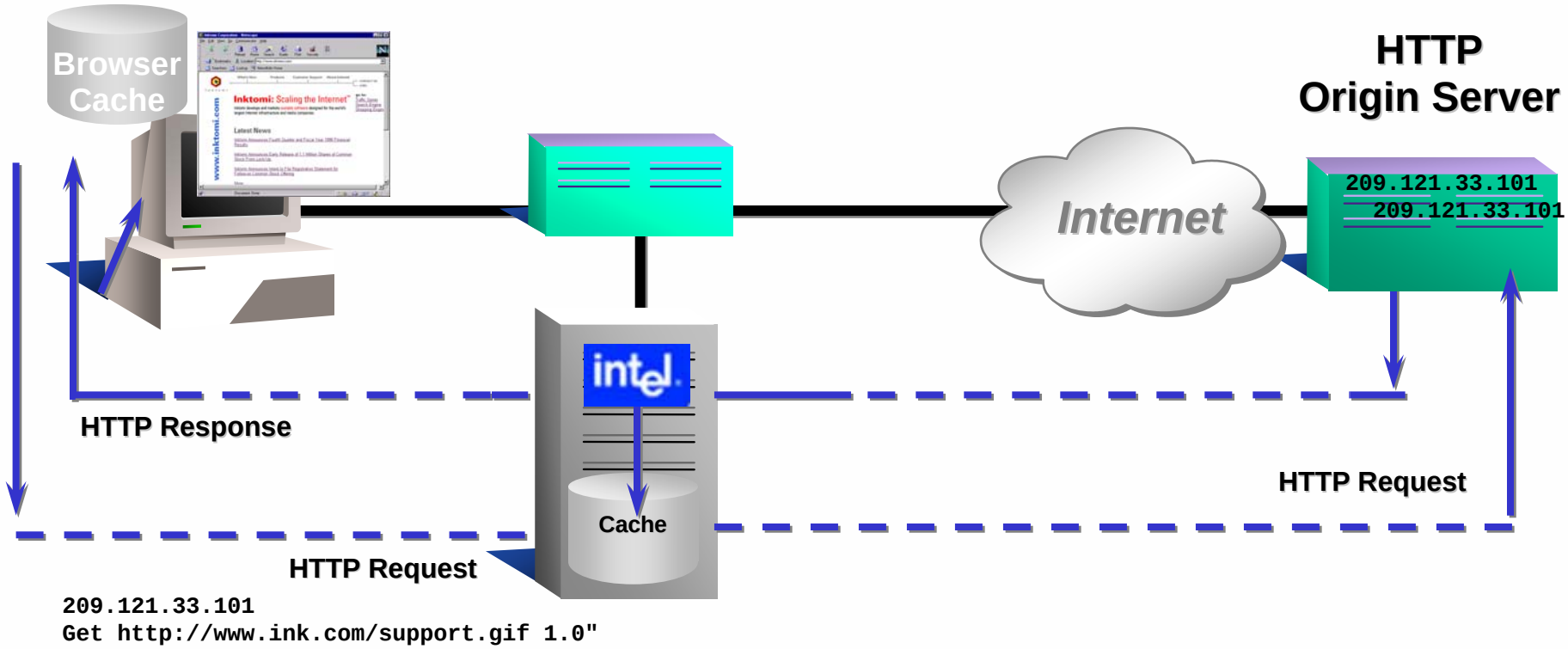
Reverse Proxy Cache
Cache server acts as a proxy for the origin server.



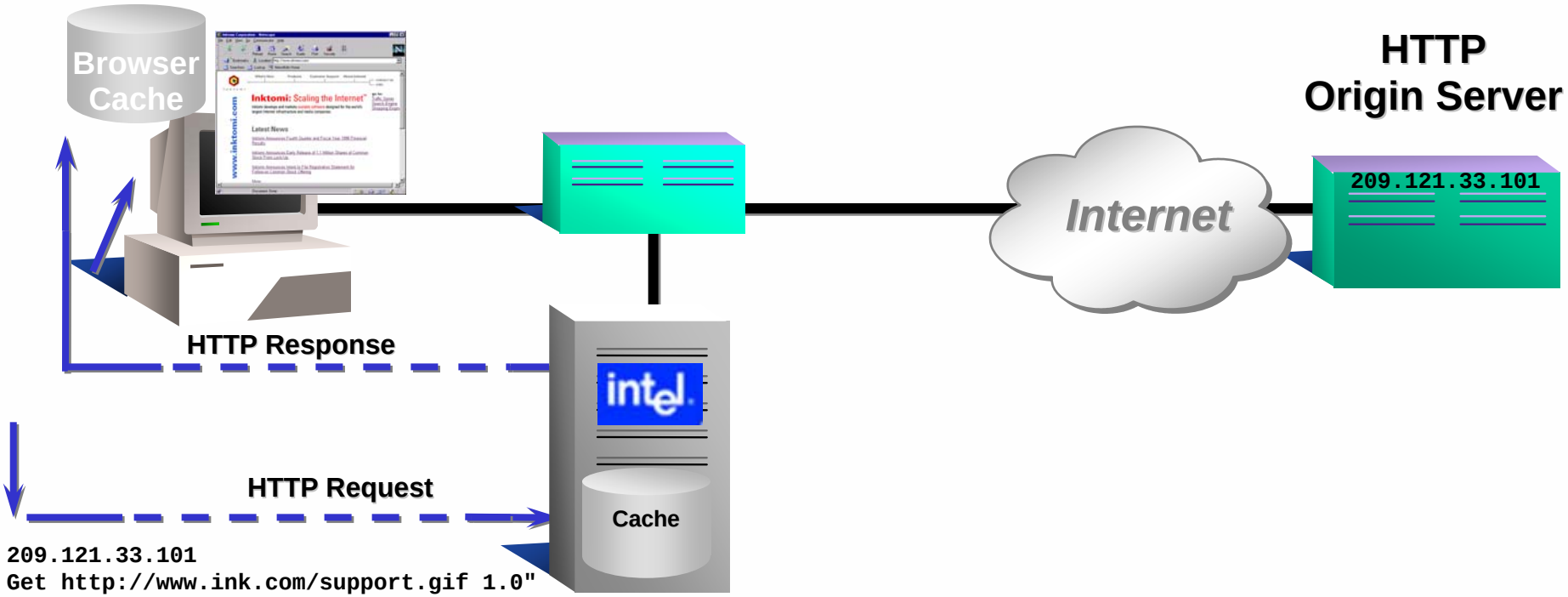
Benefits of Forward Proxy

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

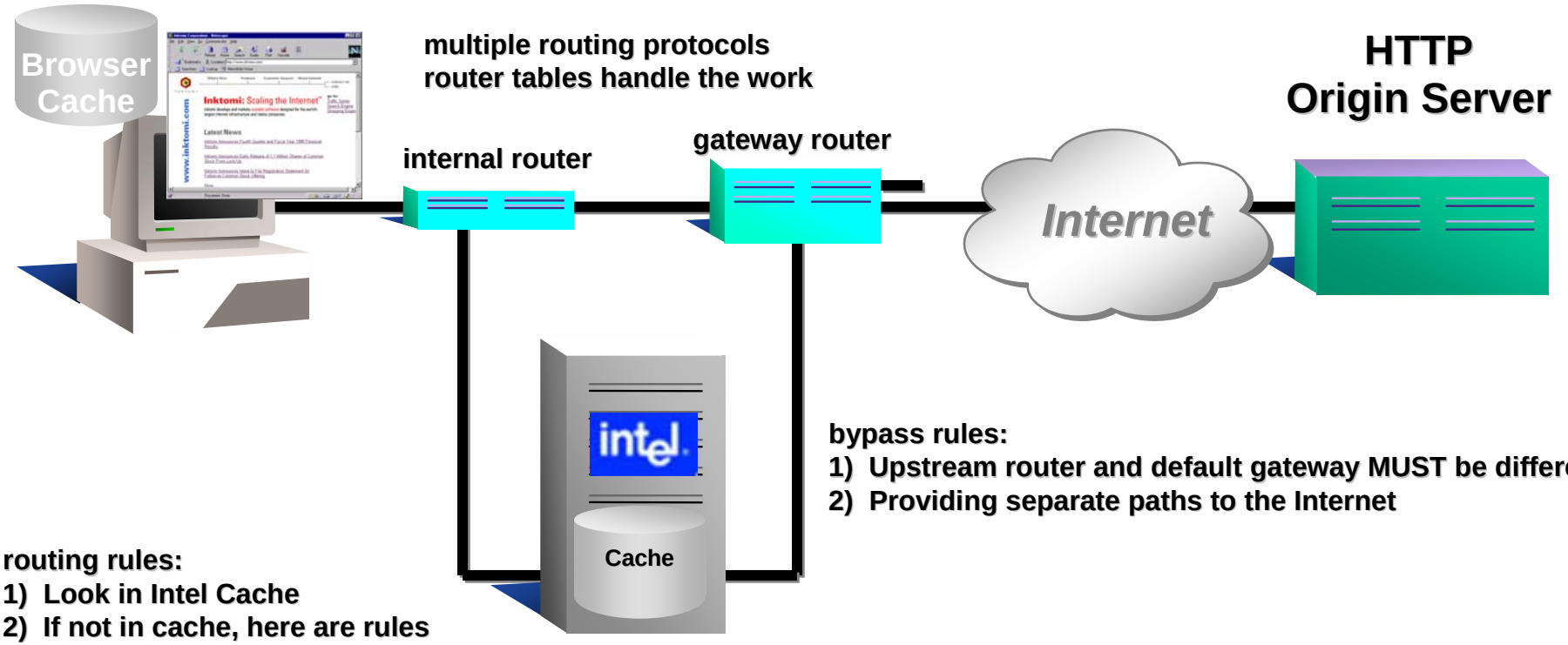
Origin Server Responds



Intel Cache Responds



Redirecting Requests via Router

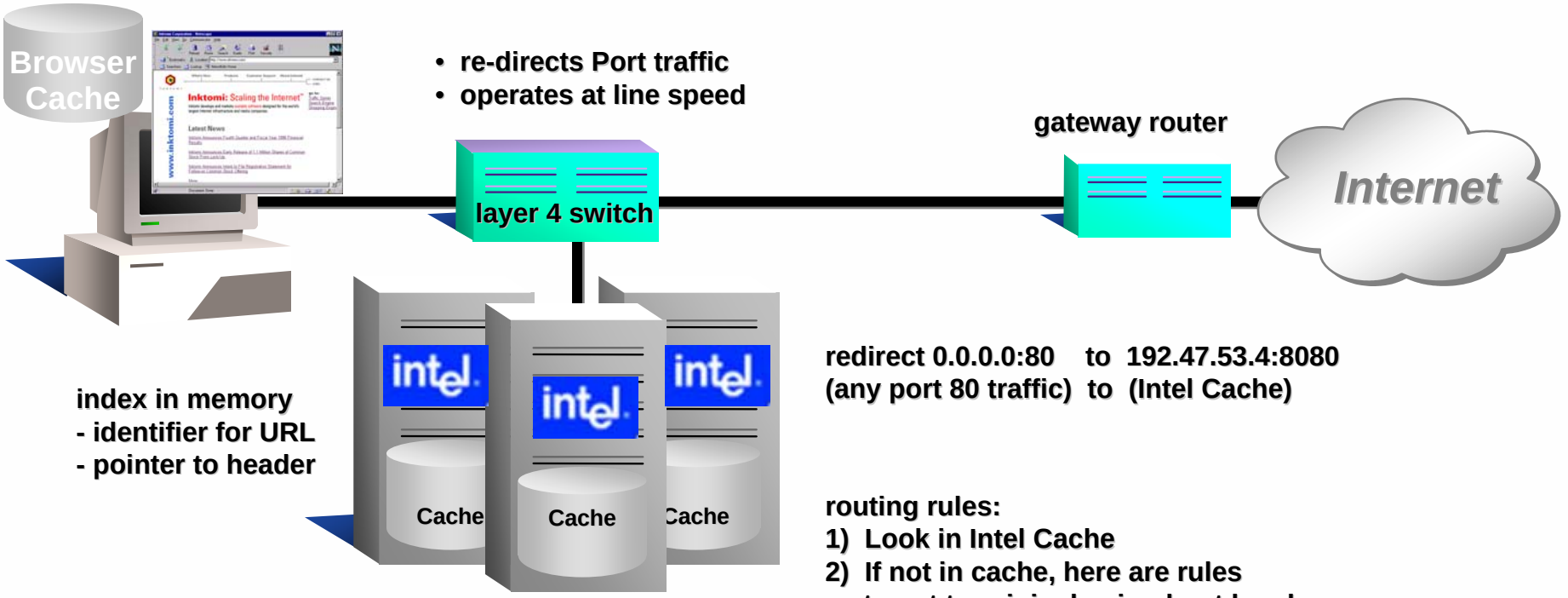


routing rules:
1) **Look in Intel Cache**
2) **If not in cache, here are rules to get to original using host header**

bypass rules:
1) **Upstream router and default gateway MUST be different**
2) **Providing separate paths to the Internet**

An environment with an interconnection device is optimal for transparency

Redirecting Requests via Layer 4 Switch



index in memory
- identifier for URL
- pointer to header

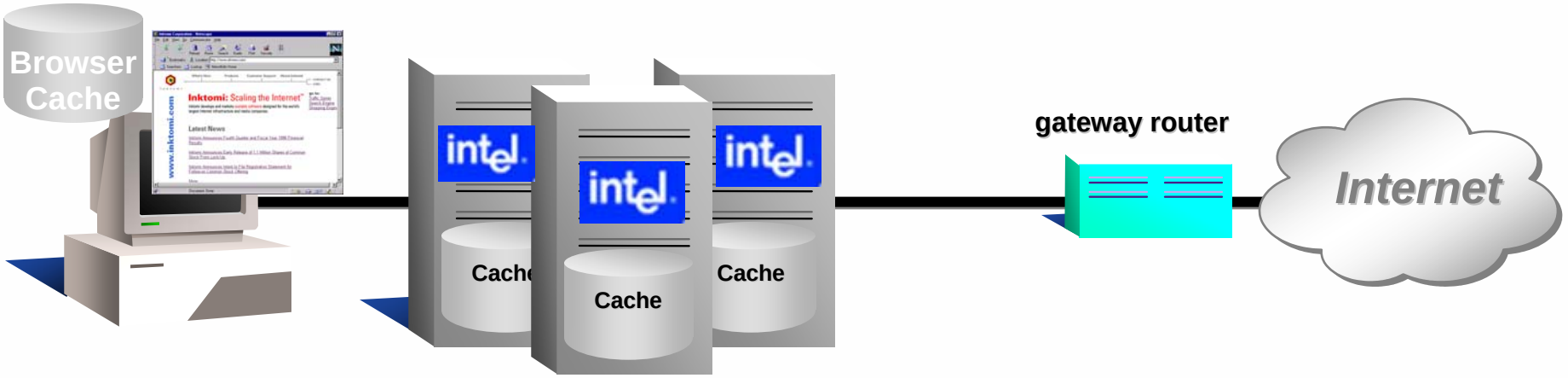
redirect 0.0.0.0:80 to 192.47.53.4:8080
(any port 80 traffic) to (Intel Cache)

routing rules:
1) Look in Intel Cache
2) If not in cache, here are rules
to get to original using host header

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



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!

Forward Cache

- **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

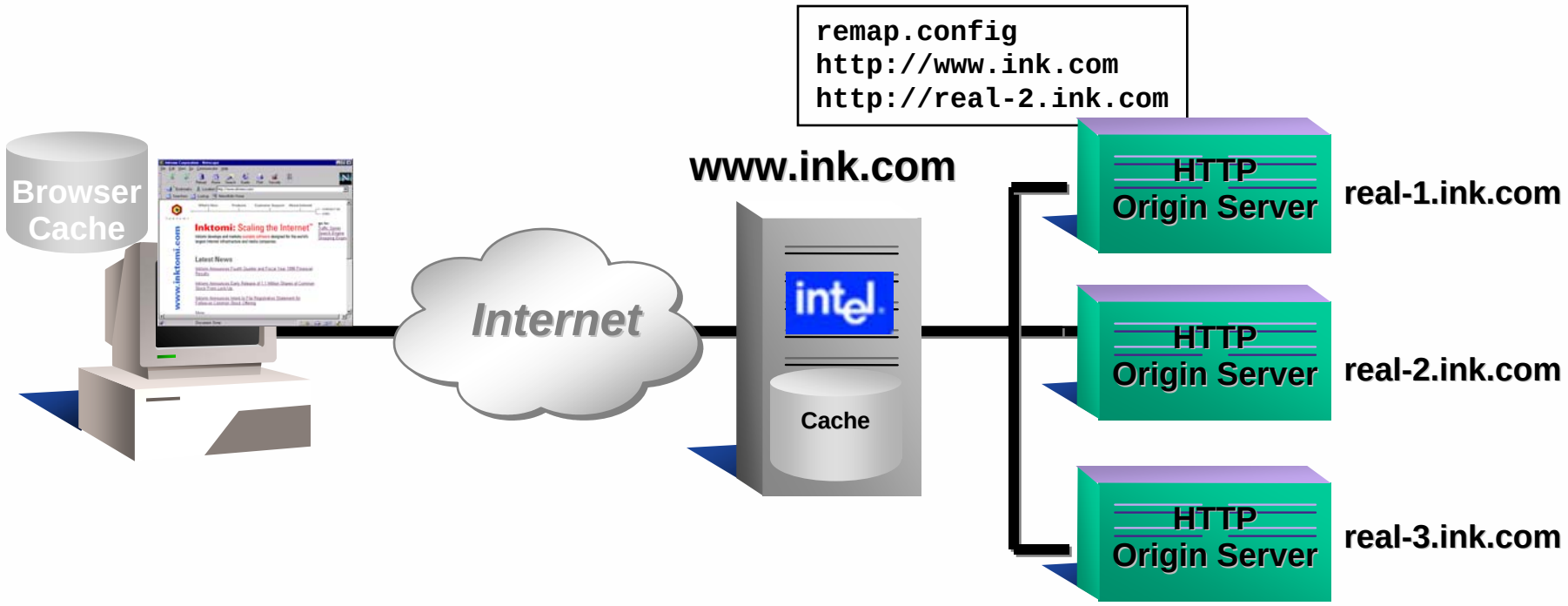
Transparency

- **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

- **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

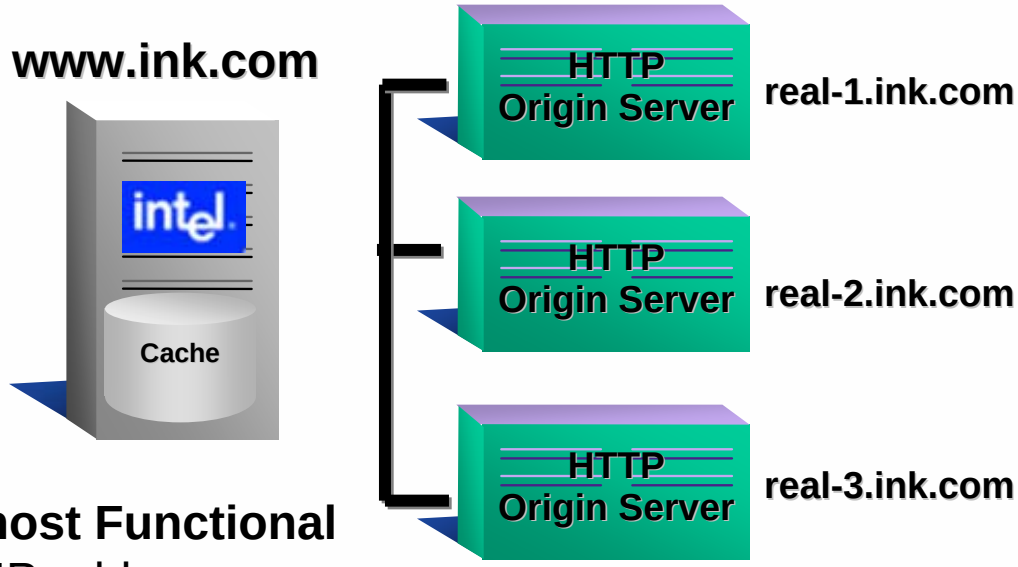


Intel Cache impersonates your web server to assume load and respond to requests that would normally be destined for your web servers

Reverse Proxy Environments

Option 1: Multi-host Round Robin
- 1 name with 3 IP addresses
- All have same content
- Optimal for load balancing

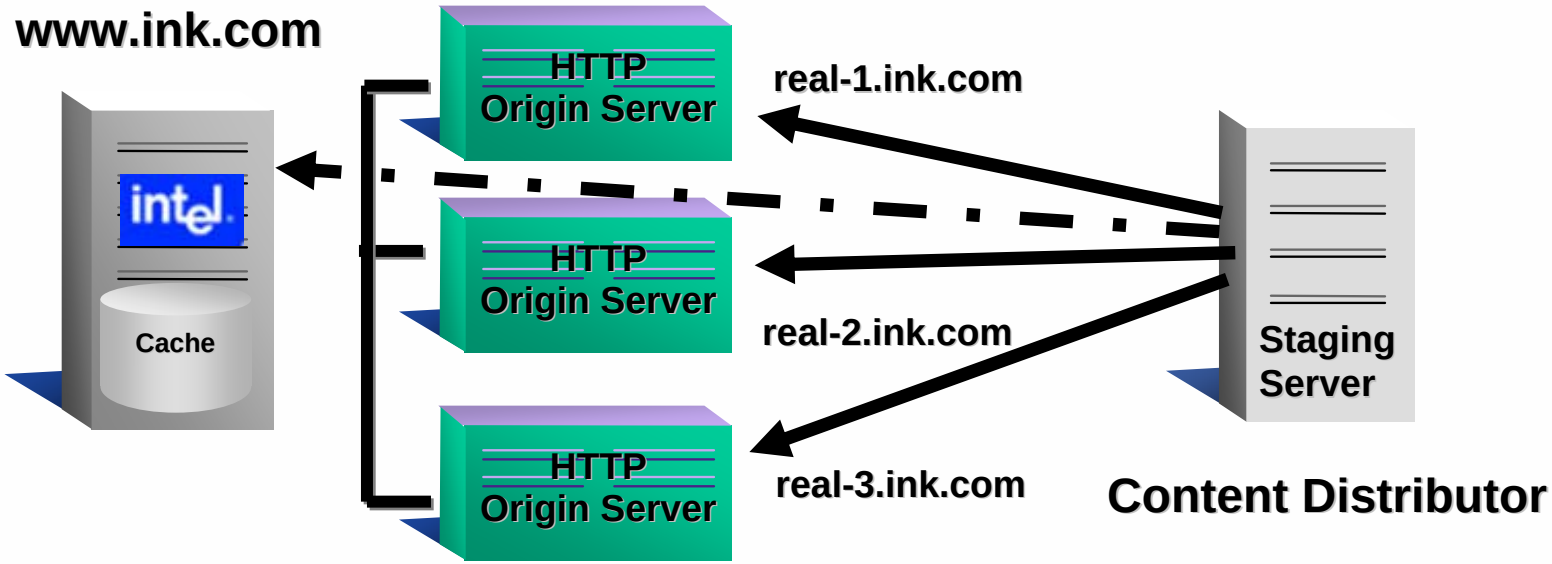
Option 2: Multi-host Load Director
- 1 name with 3 IP addresses
- Specify which machine responds
- All have same content
- Optimal for regional web servers



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

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**

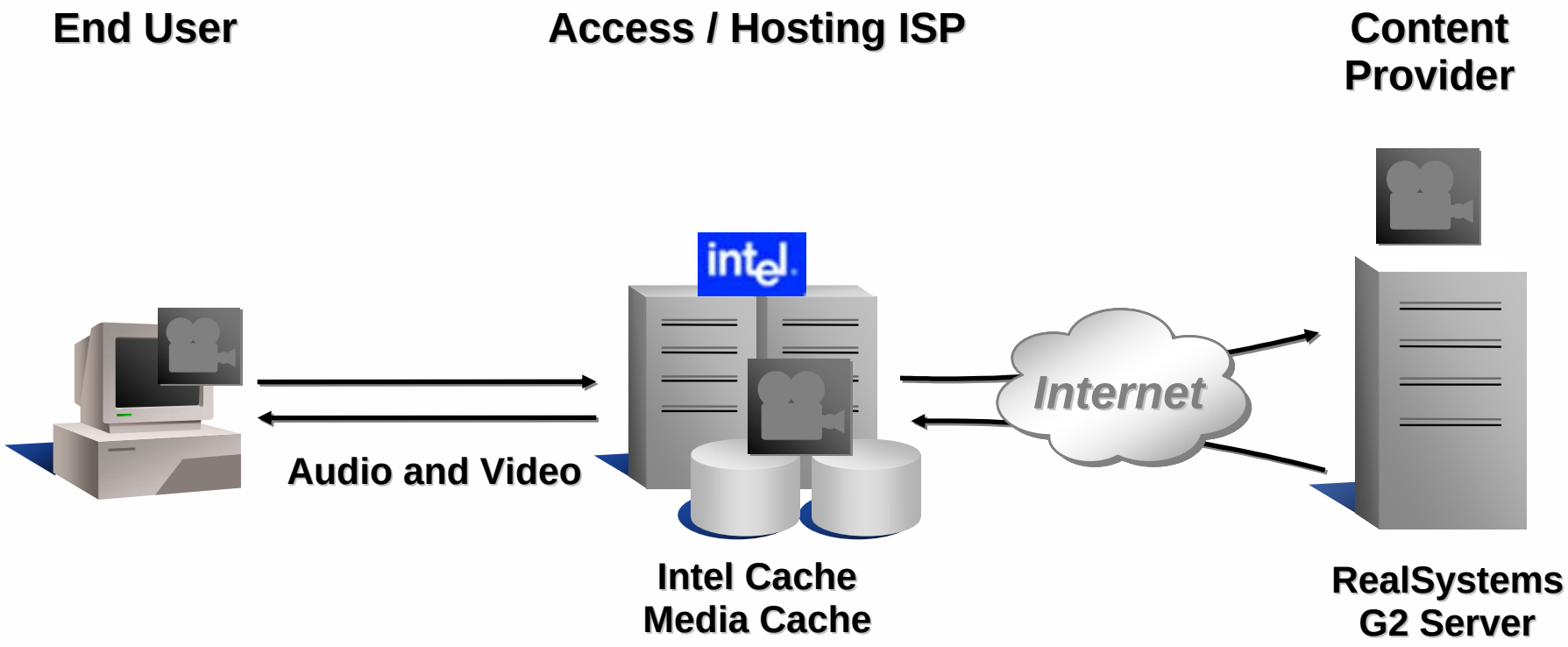
  E-commerce makes this feature more critical

- **Management and Reporting**

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

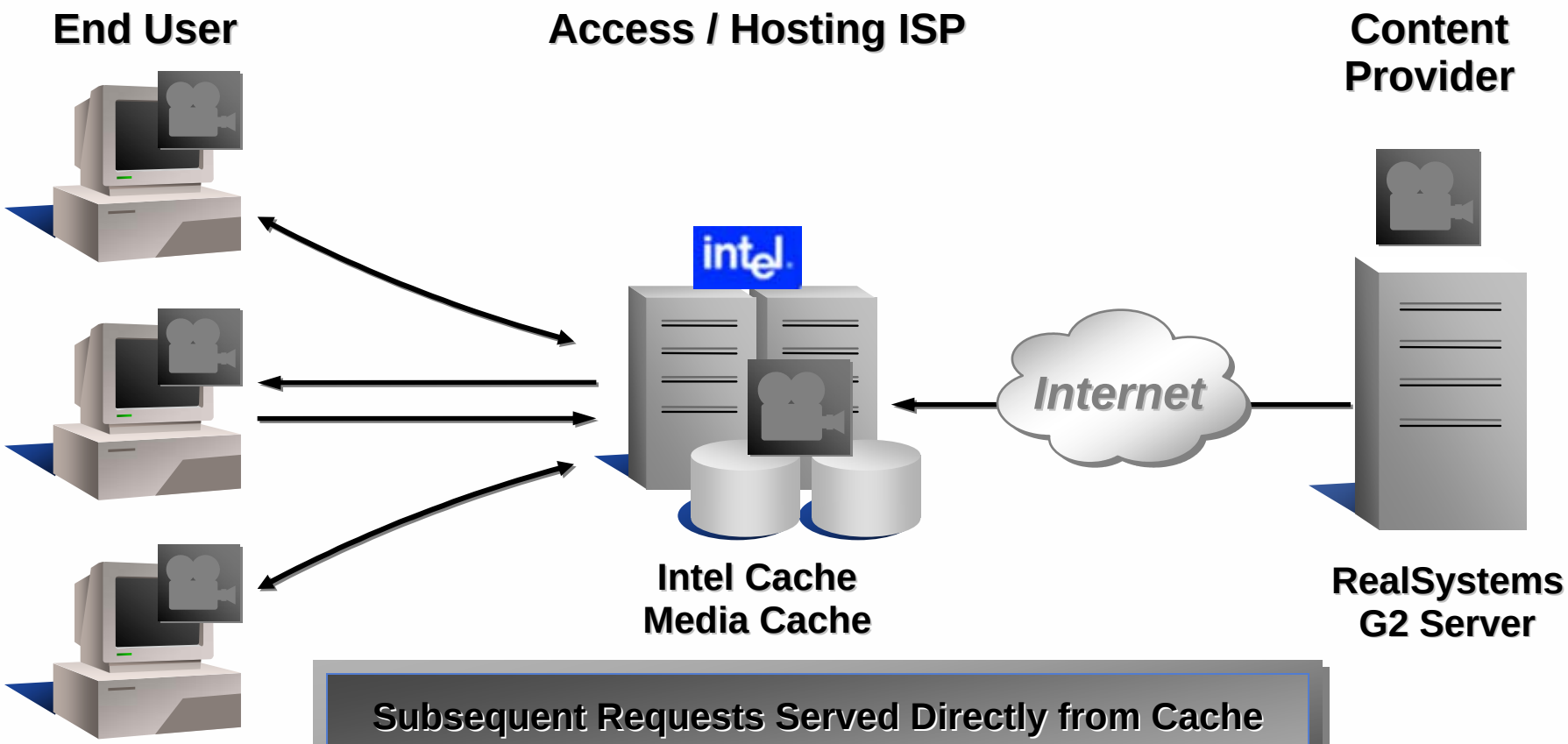
- **Fault Tolerance and Disaster Recovery**

On Demand with Media Caching



First Request Served Directly from RealSystems G2 Server

On Demand with Media Caching



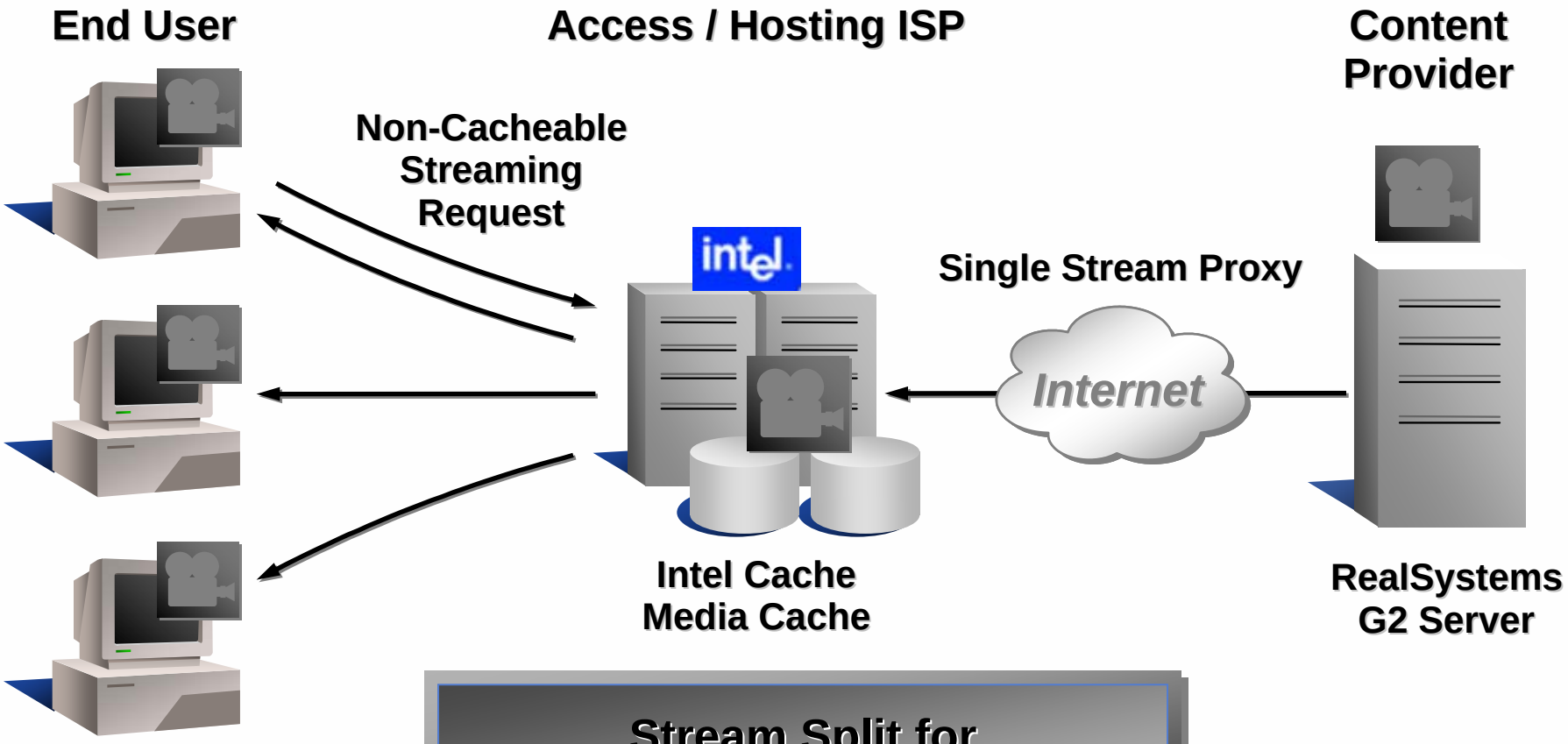
Subsequent Requests Served Directly from Cache

- **Improving Response Time**
- **Reducing Bandwidth Requirements**

Live Streams with Media Cache

Inktomi

Intel Cache supports live streams with Media Cache Option

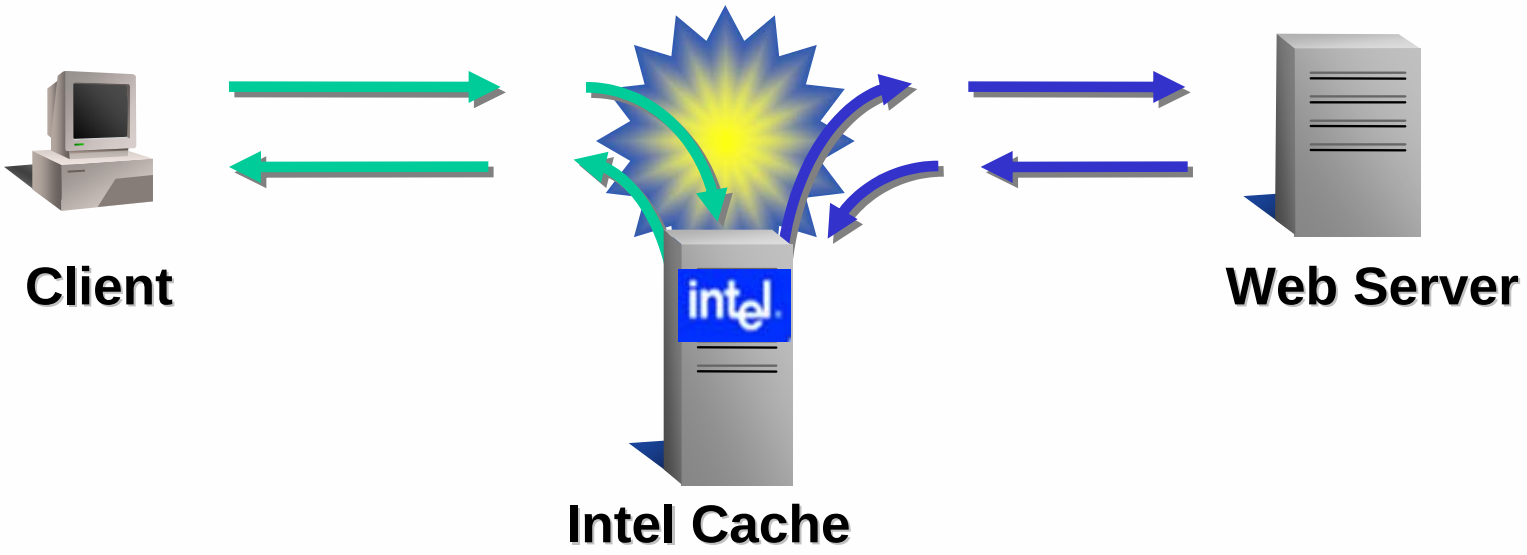


Benefit of Distributed Streaming Media

- **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**

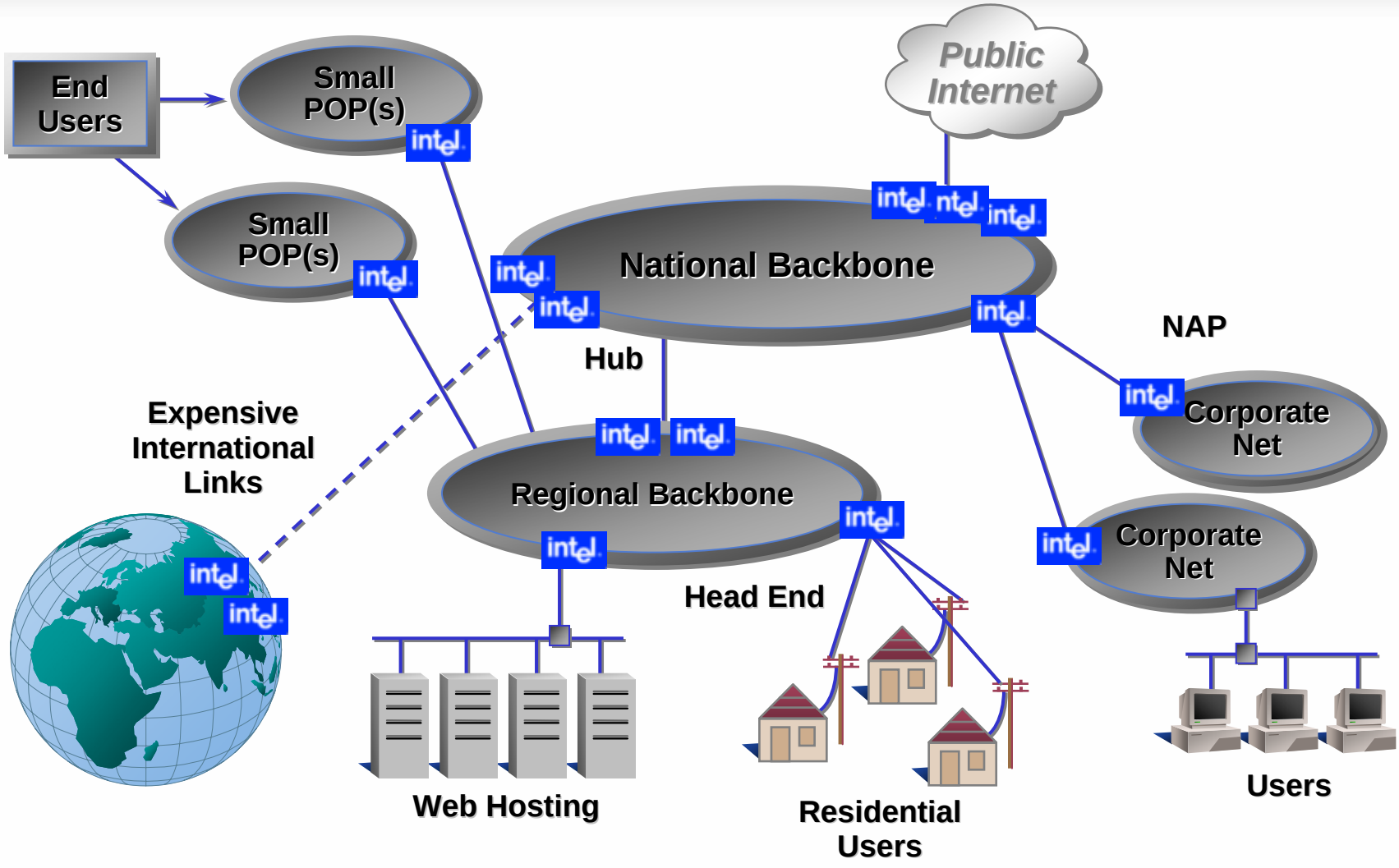
Caching Summary

Intel Cache provides multiple solutions for intercepting and caching Internet traffic



Creative configuration can address many unique problems

Meets Any Network Needs



Review Questions:

- 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?

Practice Lab

- **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

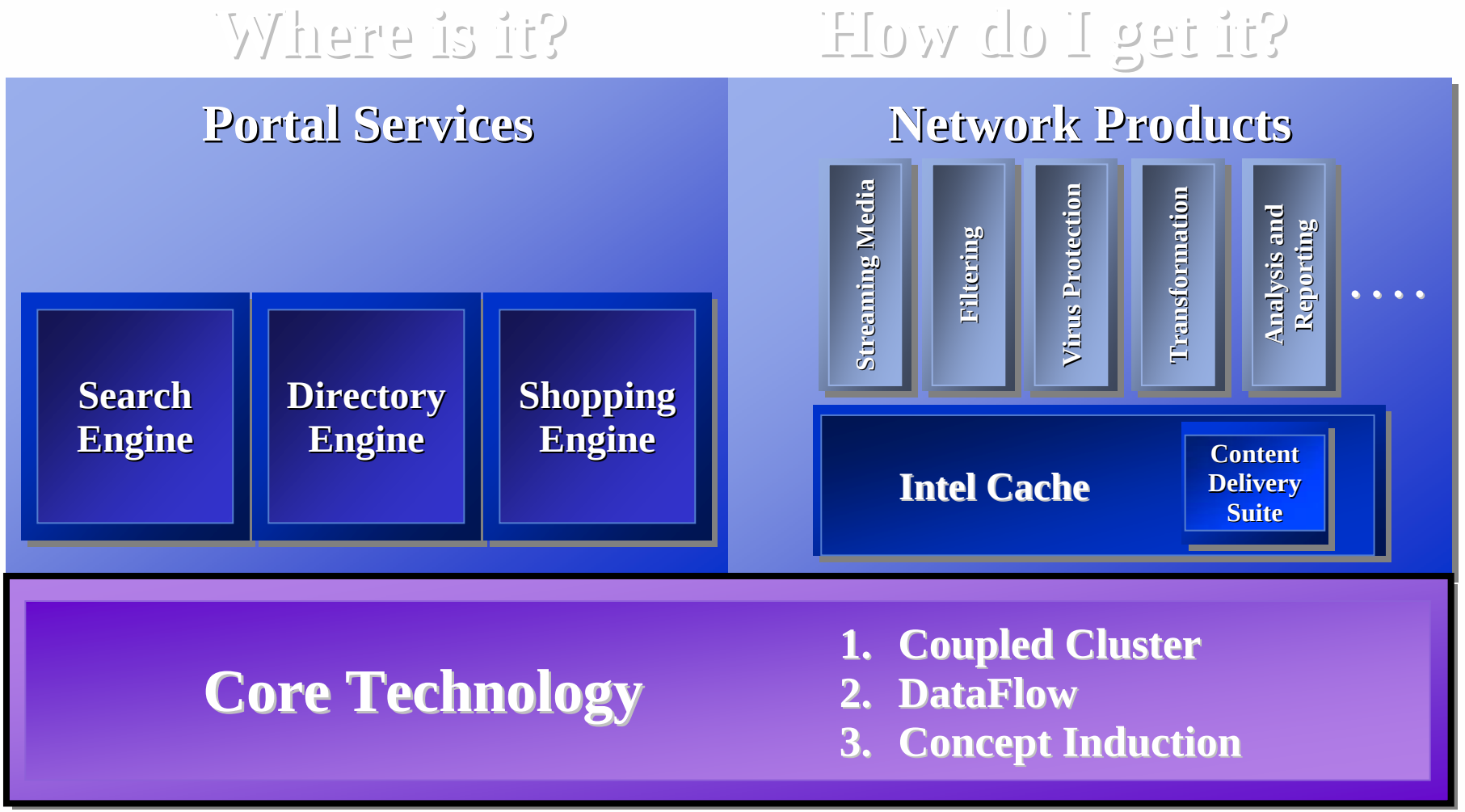
Intel Cache: Overview

Inktomi

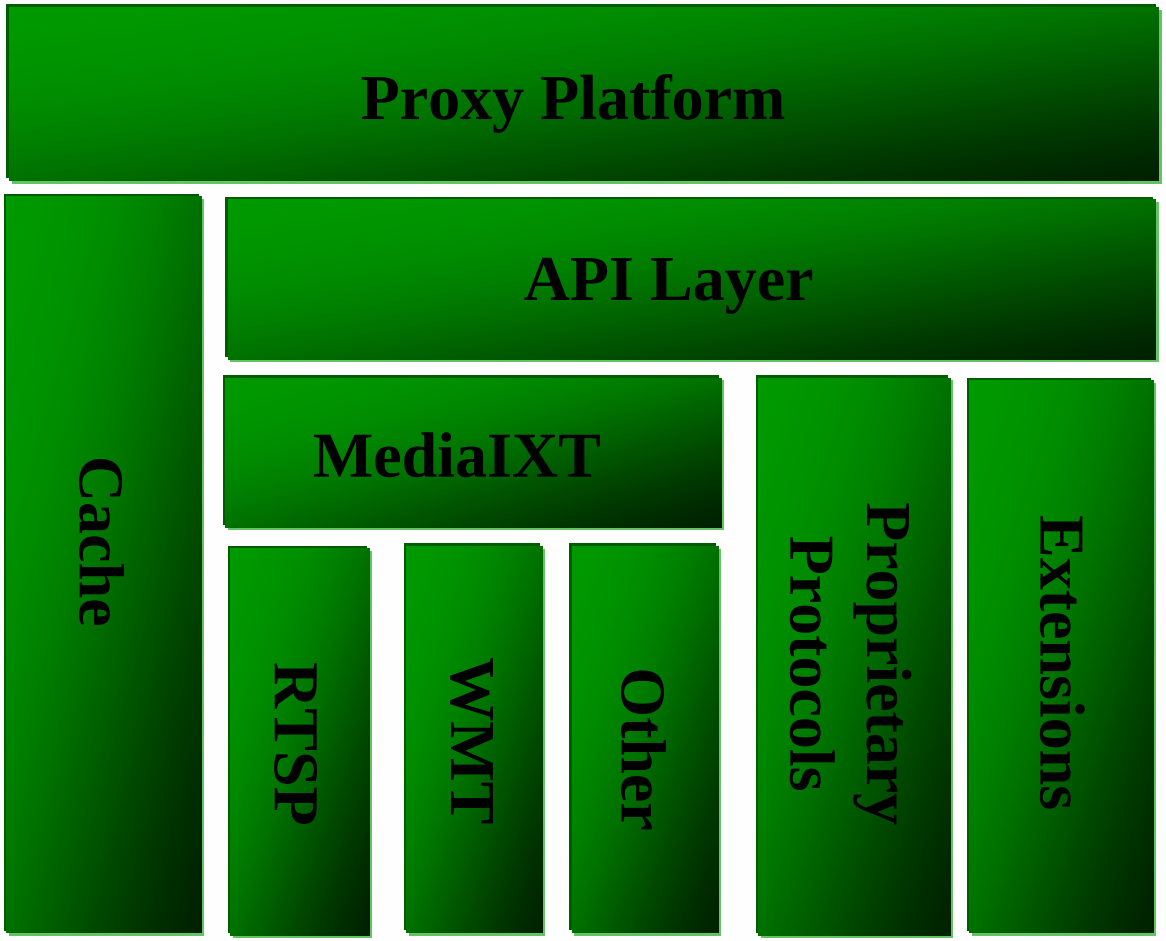
Chapter Overview

- Intel Cache Releases
- Intel Cache Applications
- Extensions

Inktomi Infrastructure Solutions



Intel Cache - An Extensible Platform for Rich Media



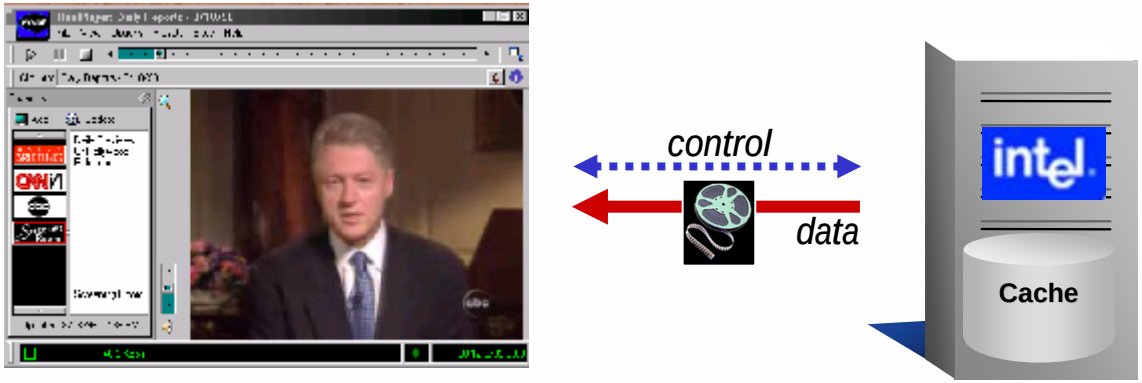
Intel Cache Partners and Services

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

Extension - Media Delivery

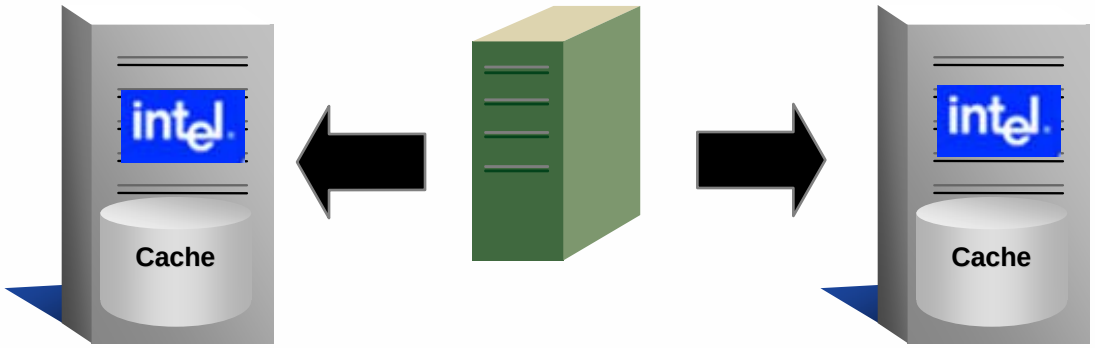
Inktomi

- **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

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

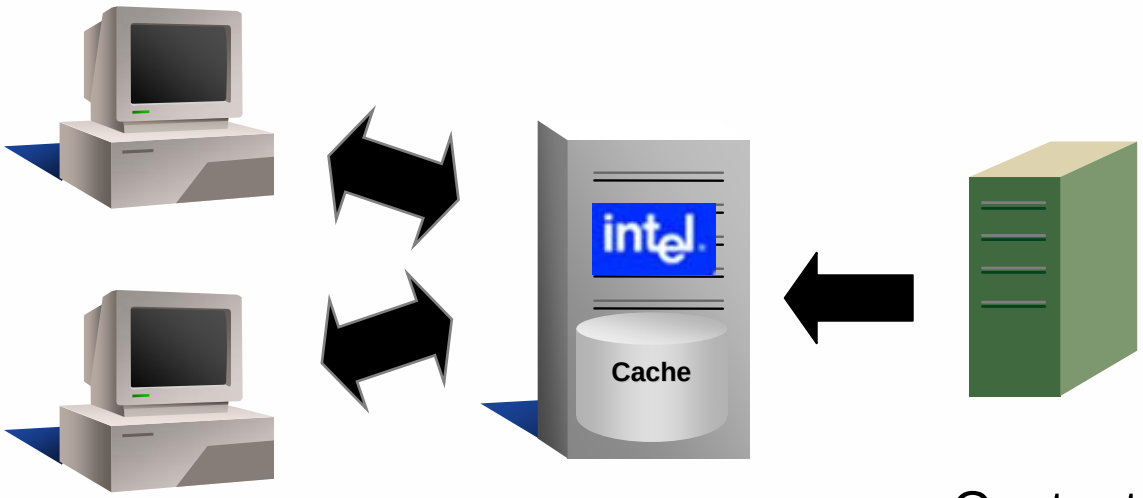


Content Server

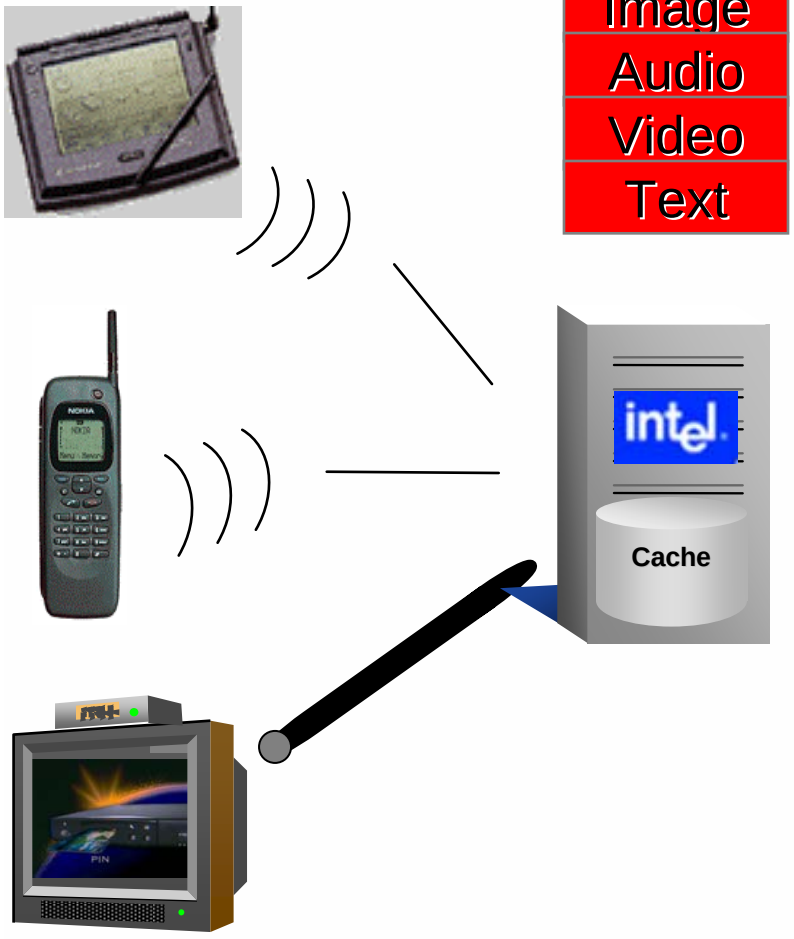
Extension - Interactive Applications

Inktomi

- **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



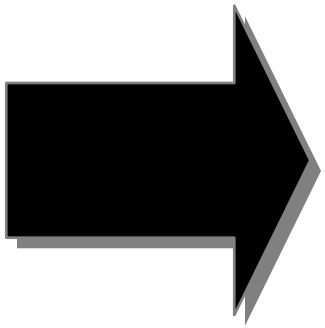
Content Transformation for:

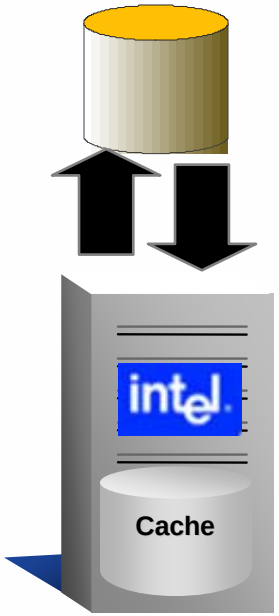
- **Unique content delivery by connection speed**
- **Language translation services**
- **Specially formatted content for “thin clients”**

Benefits:
Single copy of content capable of supporting multiple needs

Extension - Filtering

URL filtering





Only content that is not filtered is accessed by the user

Content Server

Benefits:

Incremental Revenue for Service Providers

Restricts content to appropriate users

Review Questions:

- What types of environment would extensions be beneficial?

Intel Cache: Installation

Inktomi

Chapter Overview

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

Intel Cache:
Features and Benefits

Inktomi

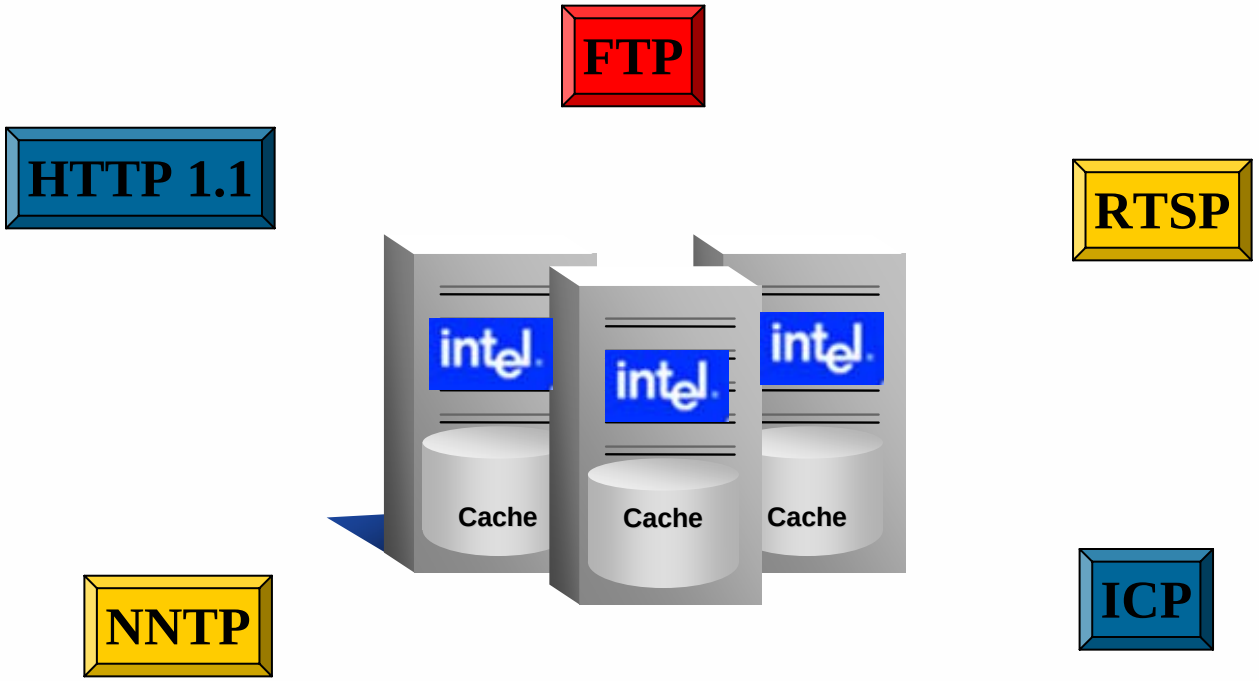
Chapter Overview

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

Development Environments

- **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



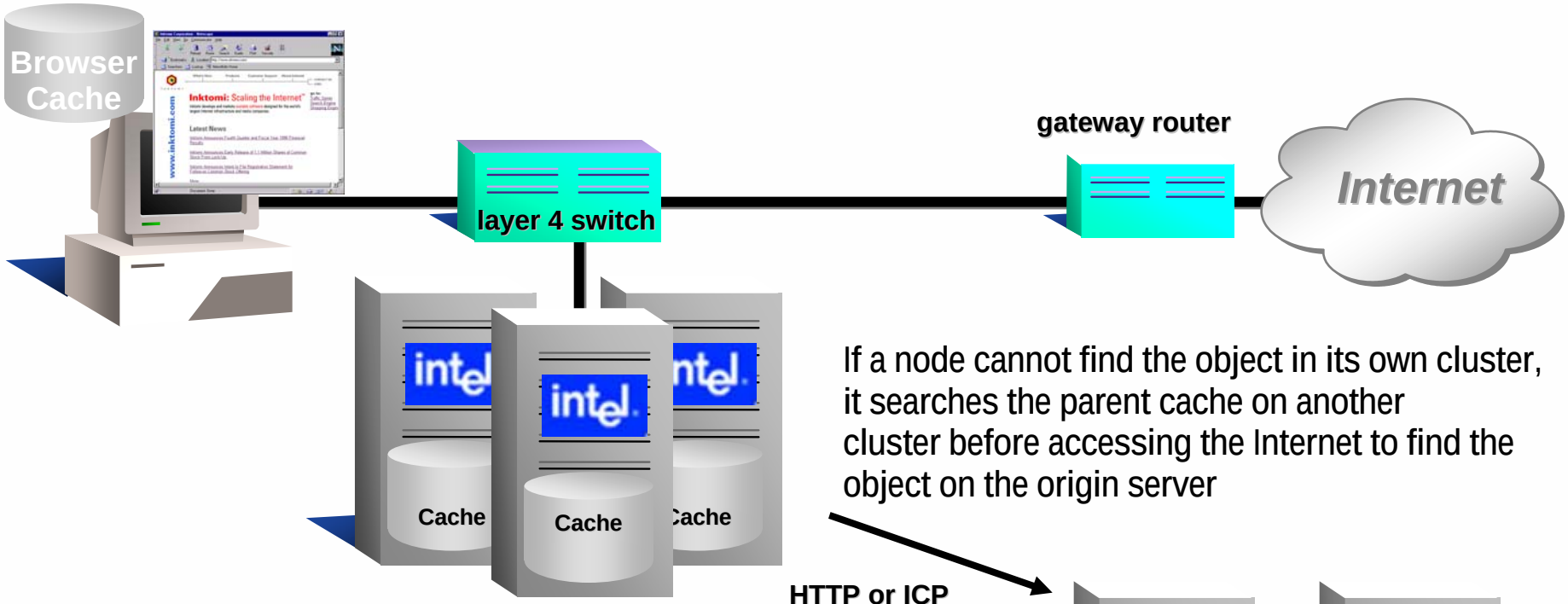
Manageability and Ease of Use

- **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

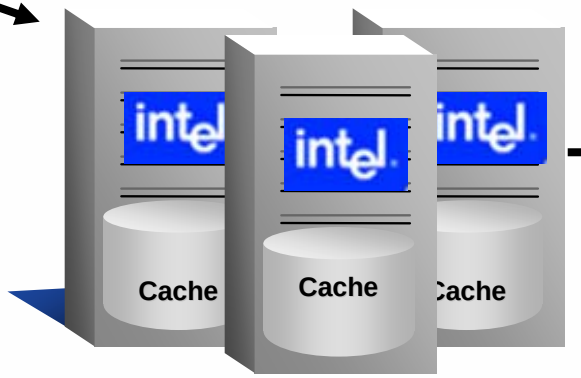
- **Since Intel Cache is composed of “look-alike” nodes, you can**
  - Easily add one or more nodes
  - Bring nodes up and down for maintenance
  - Remove a node

Hierarchical Caching



If a node cannot find the object in its own cluster, it searches the parent cache on another cluster before accessing the Internet to find the object on the origin server

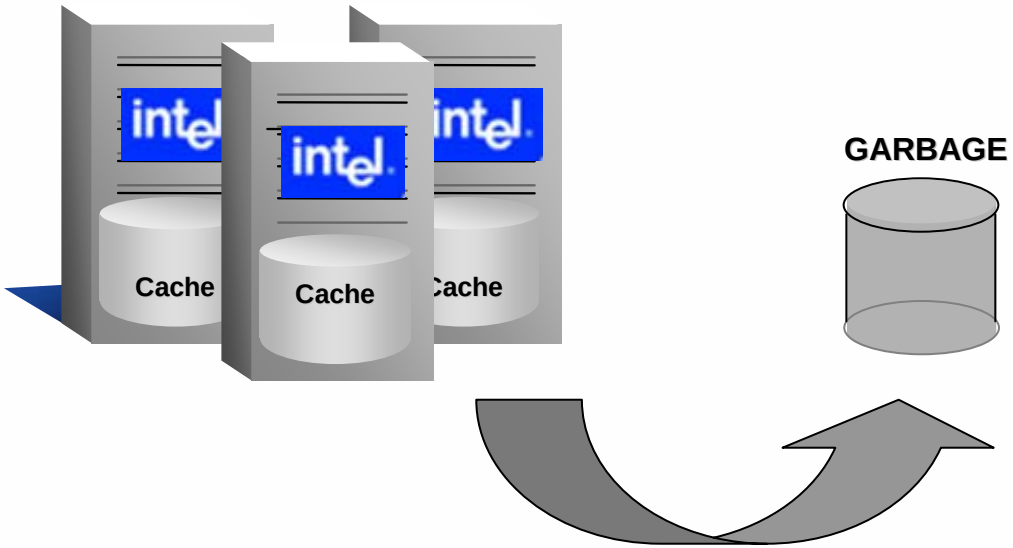
HTTP or ICP



PARENT CACHE

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

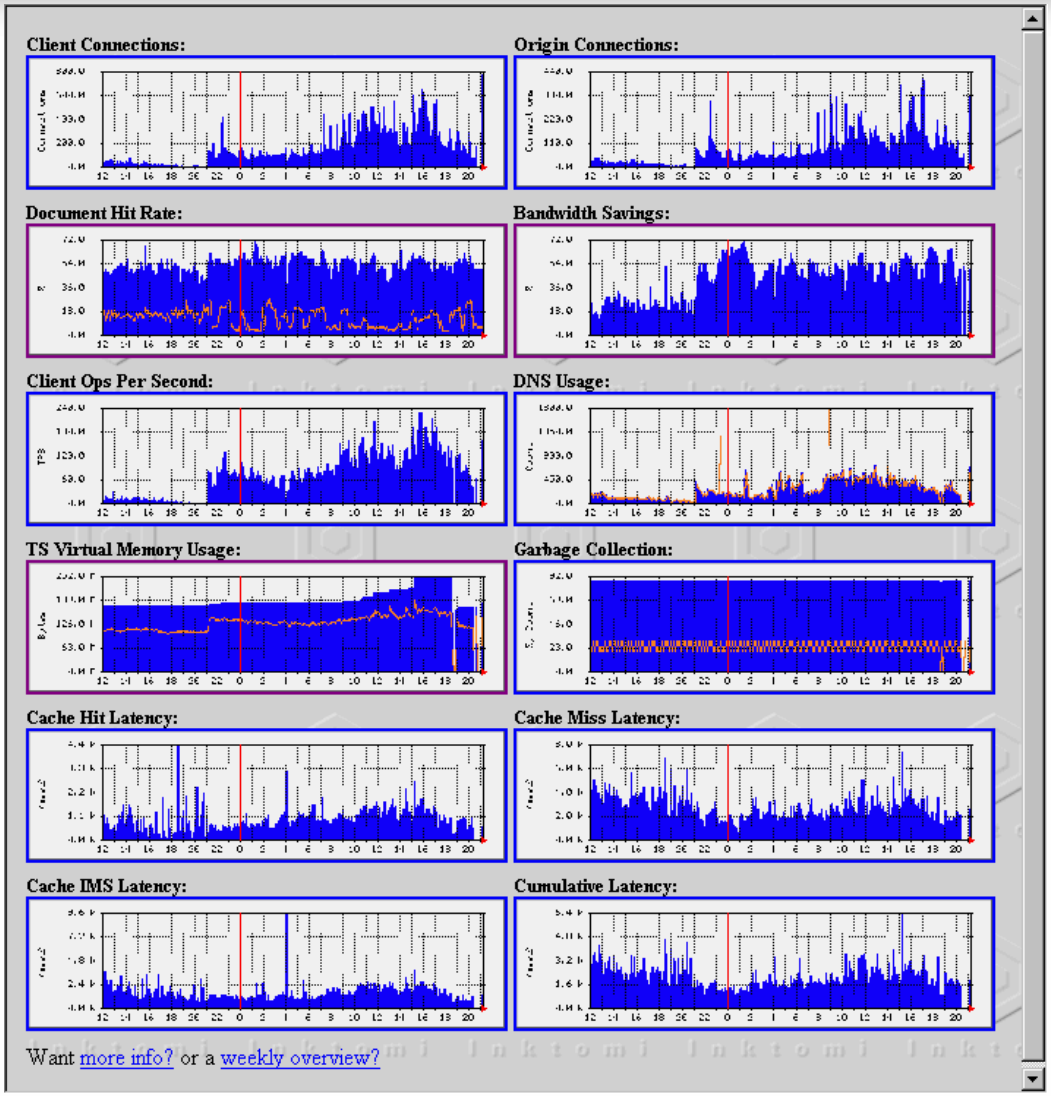


Sophisticated Garbage Collectors and Auto-Deletion functions remove stale data

Flexible Logging

- **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



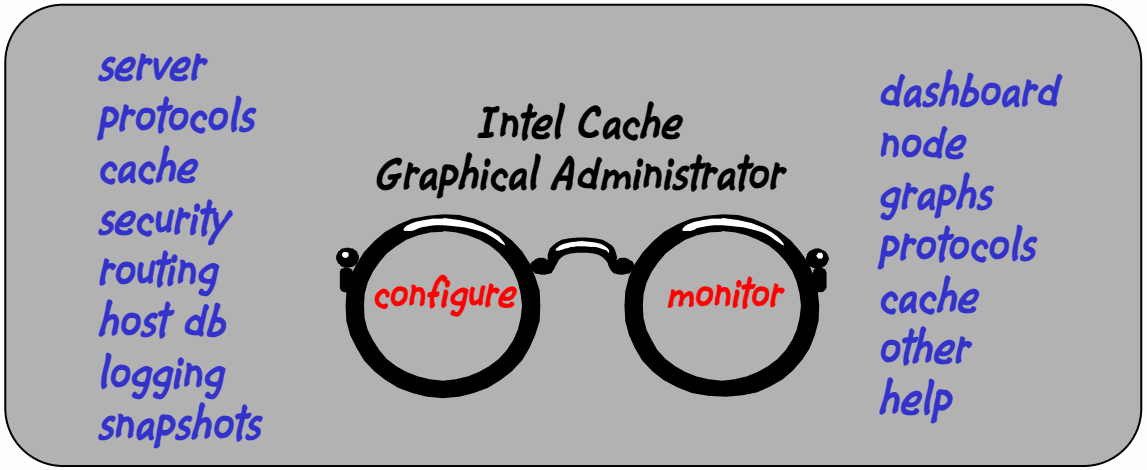
Each data point corresponds to the execution of a perl script, once every 5 minutes.

Security

- **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

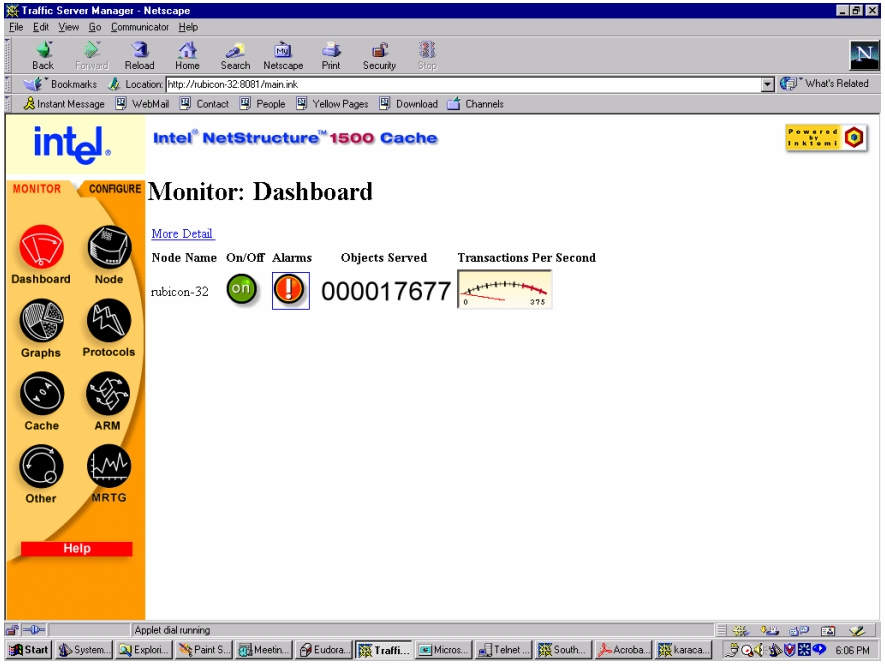
Graphical Administration

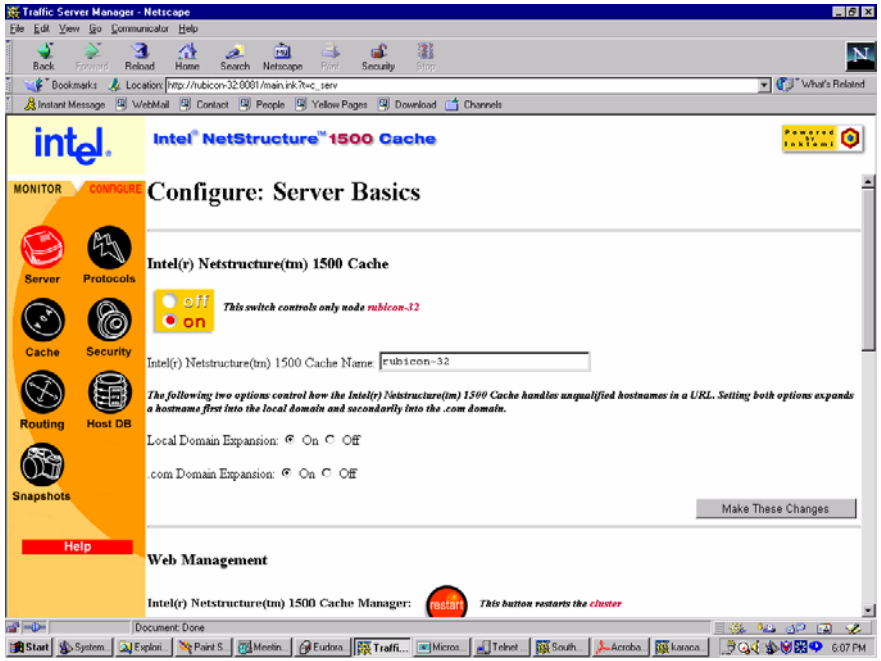


- **Provides secure *single-point* administration**
  - Configure, monitor and tune features and services
  - Encrypted remote administration
  - Efficient for clusters

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

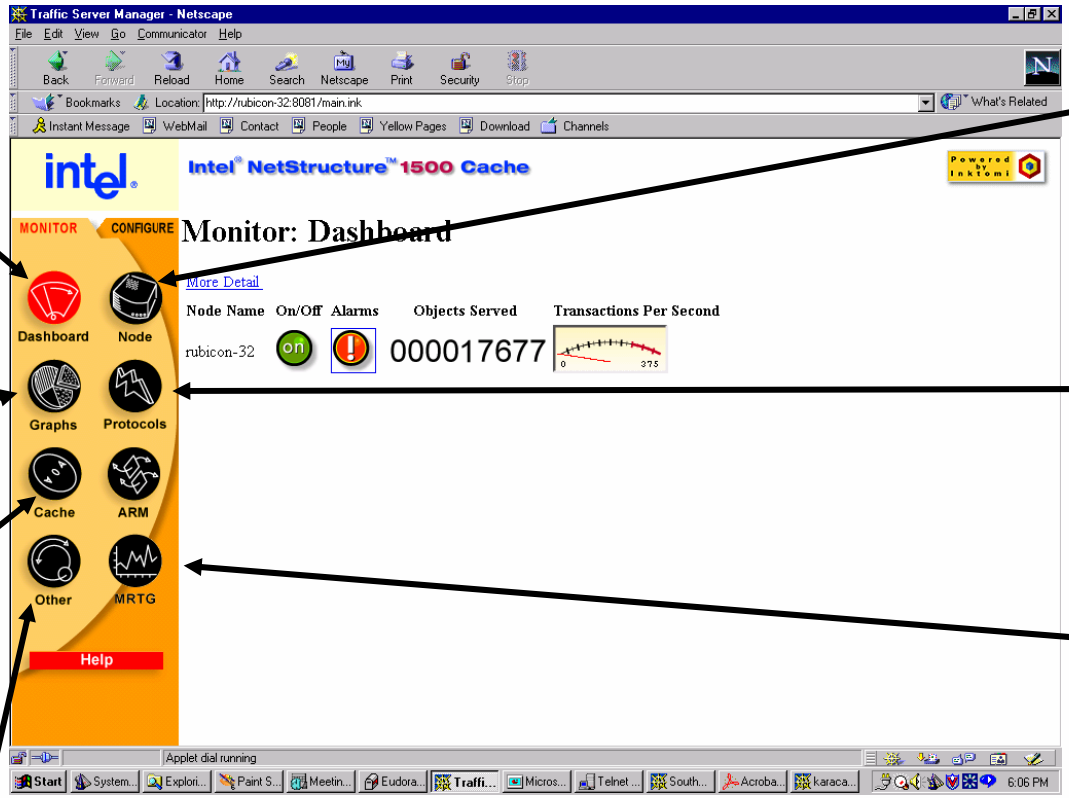
Inktomi

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

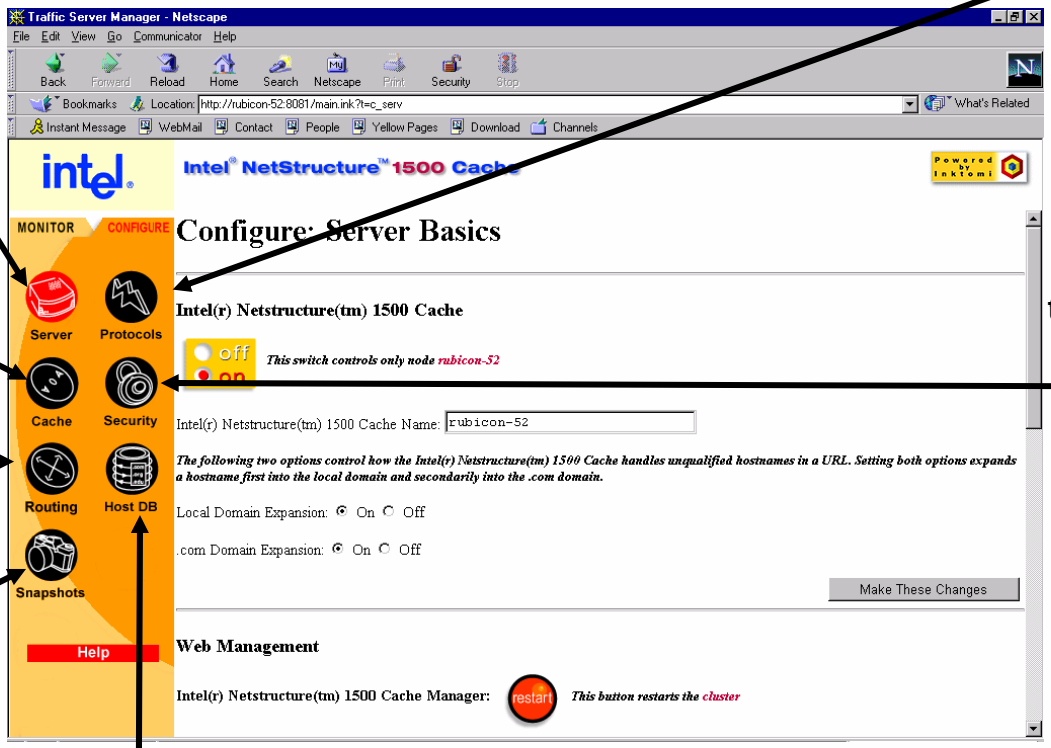
Turn Intel Cache On and Off, restart or reconfigure Traffic Manager, auto-configure browser, and enable SNMP

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



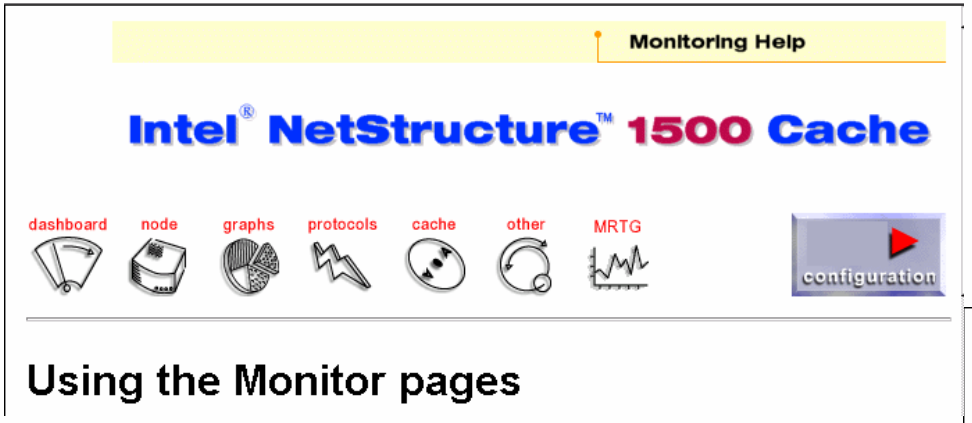
Controlling access to the UI, set admin and guest Ids and passwords

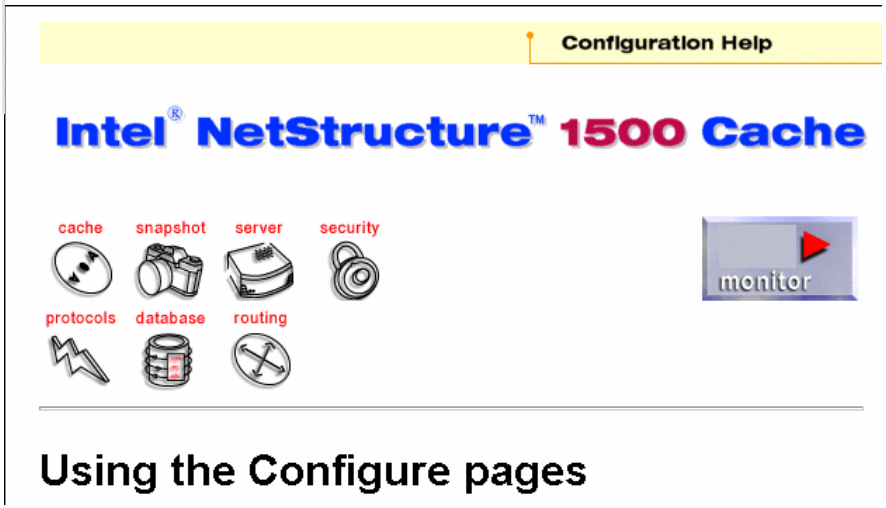
Host database options and DNS lookups

Default (recommended) configuration values are assigned during installation

Traffic Manager Help

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

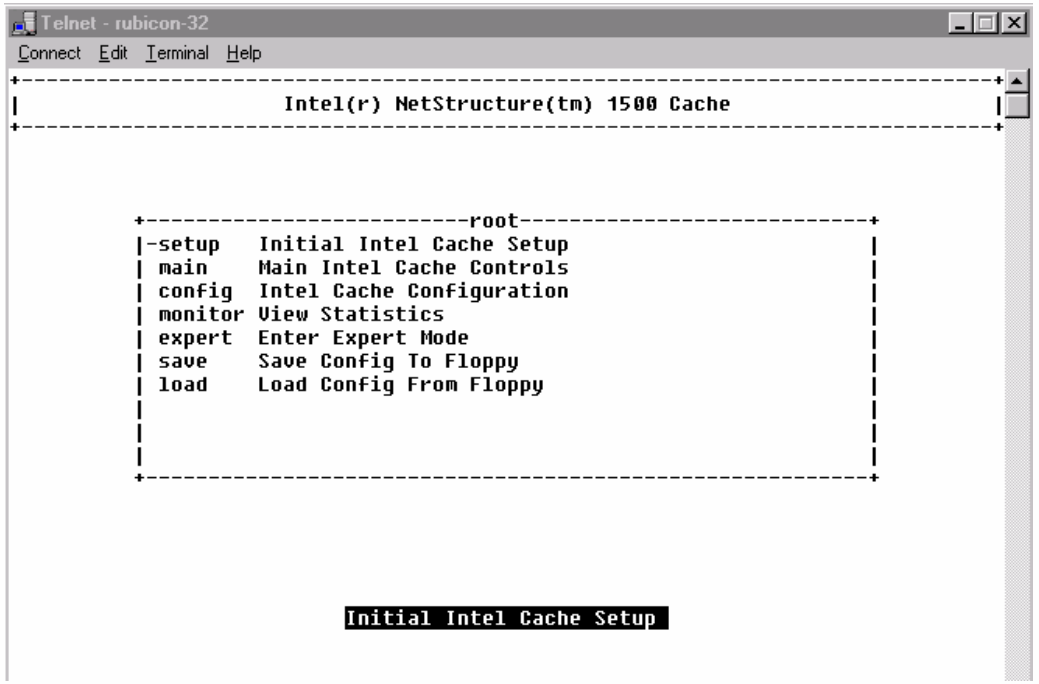




Online HELP allows you to learn more about particular pages in the Traffic Manager

Command Line Interface

- **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**



Review Questions:

- What protocols can be cached by Intel Cache?

- What two ways can one modify Intel Cache configuration?

Intel Cache Reverse Proxy

Inktomi

Chapter Overview

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

Reverse Proxy

- **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

- **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

- **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/     http://real.hopalong.com/
reverse_map  http://real.hopalong.com/    http://www.hopalong.com/
```

maps incoming requests lacking a `host:` *header*

Reverse Proxy Setup

- **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:

- What are the benefits of reverse proxy?

- What environment would you install reverse proxy?

Practice Lab

- **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

Practice Lab

- **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**

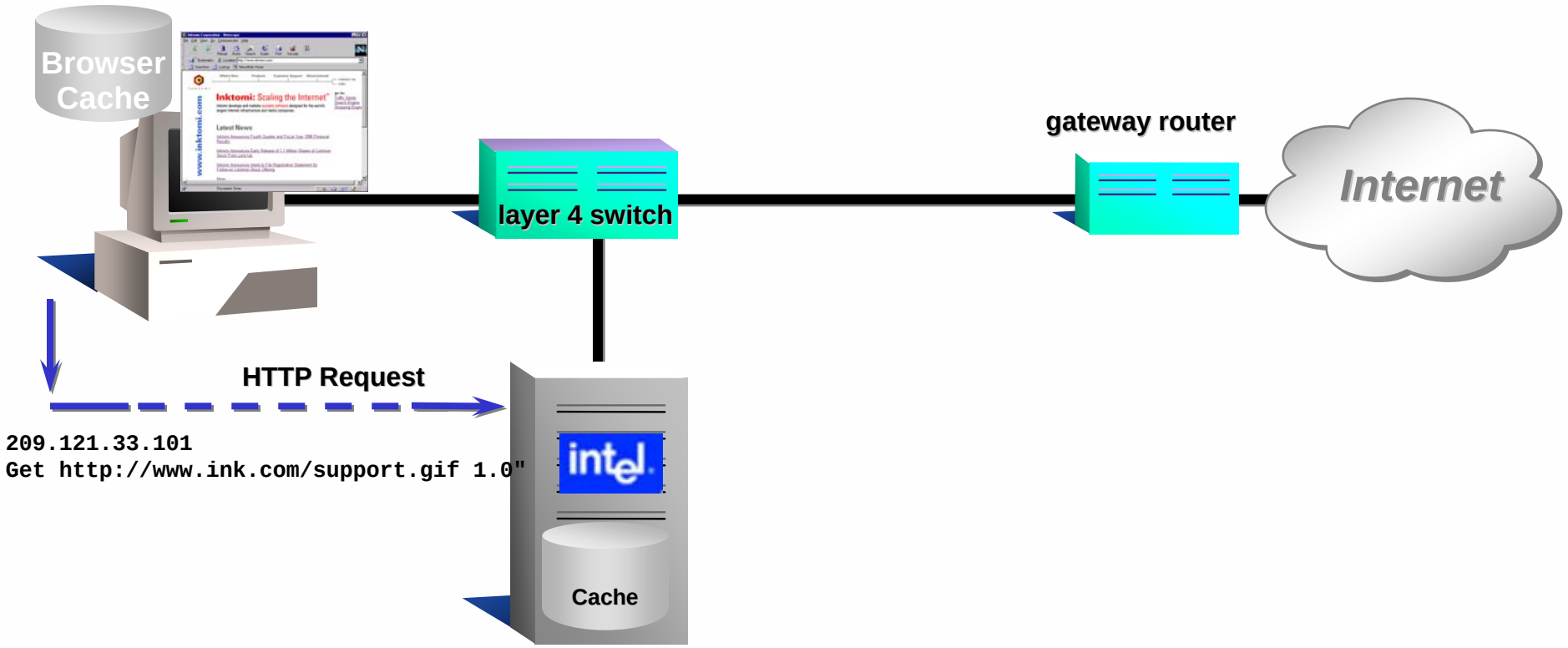
Intel Cache: Processing Requests

Inktomi

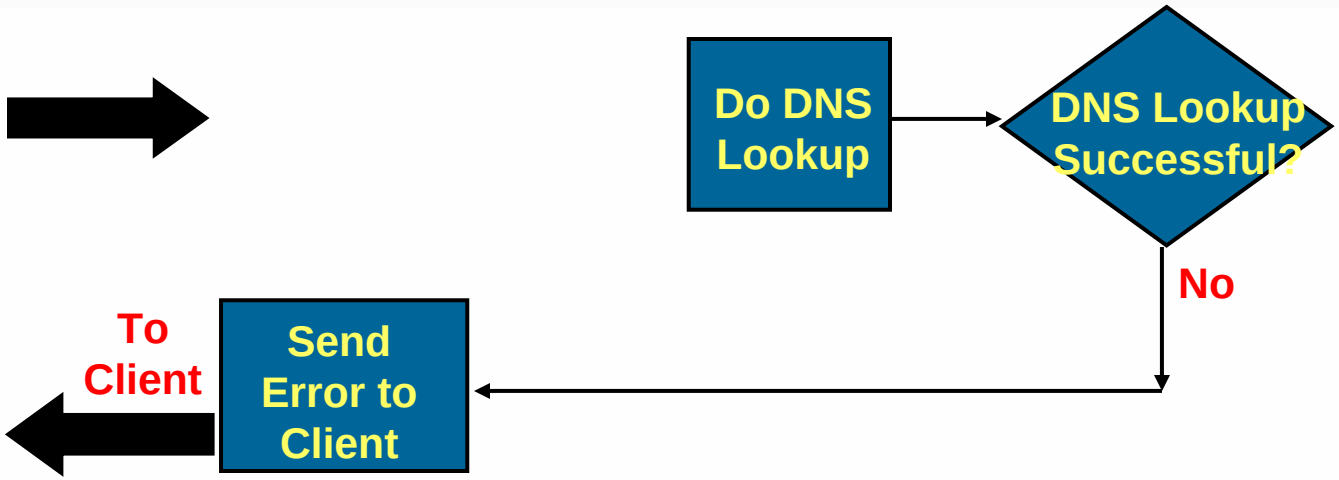
Chapter Overview

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

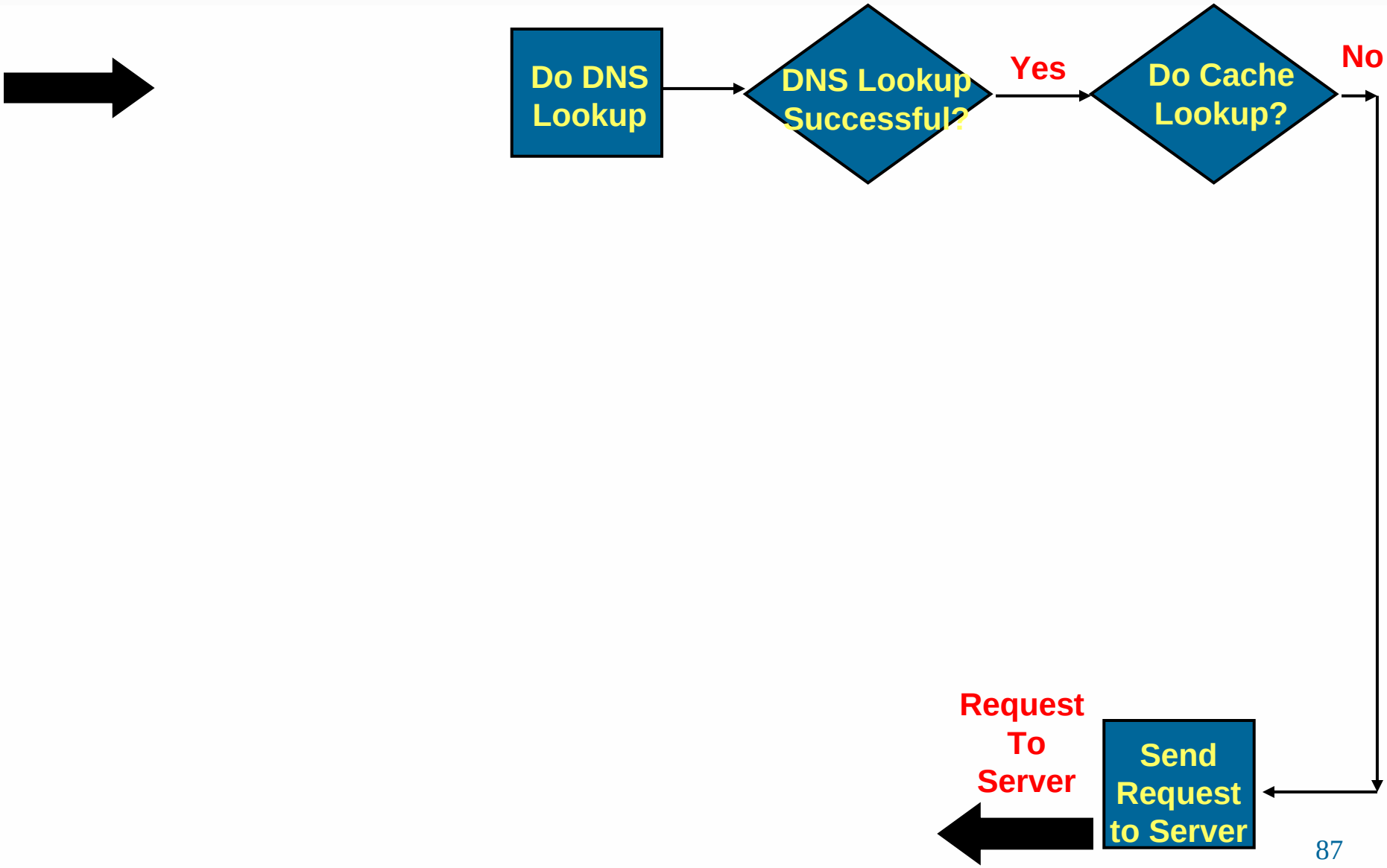
Client Request



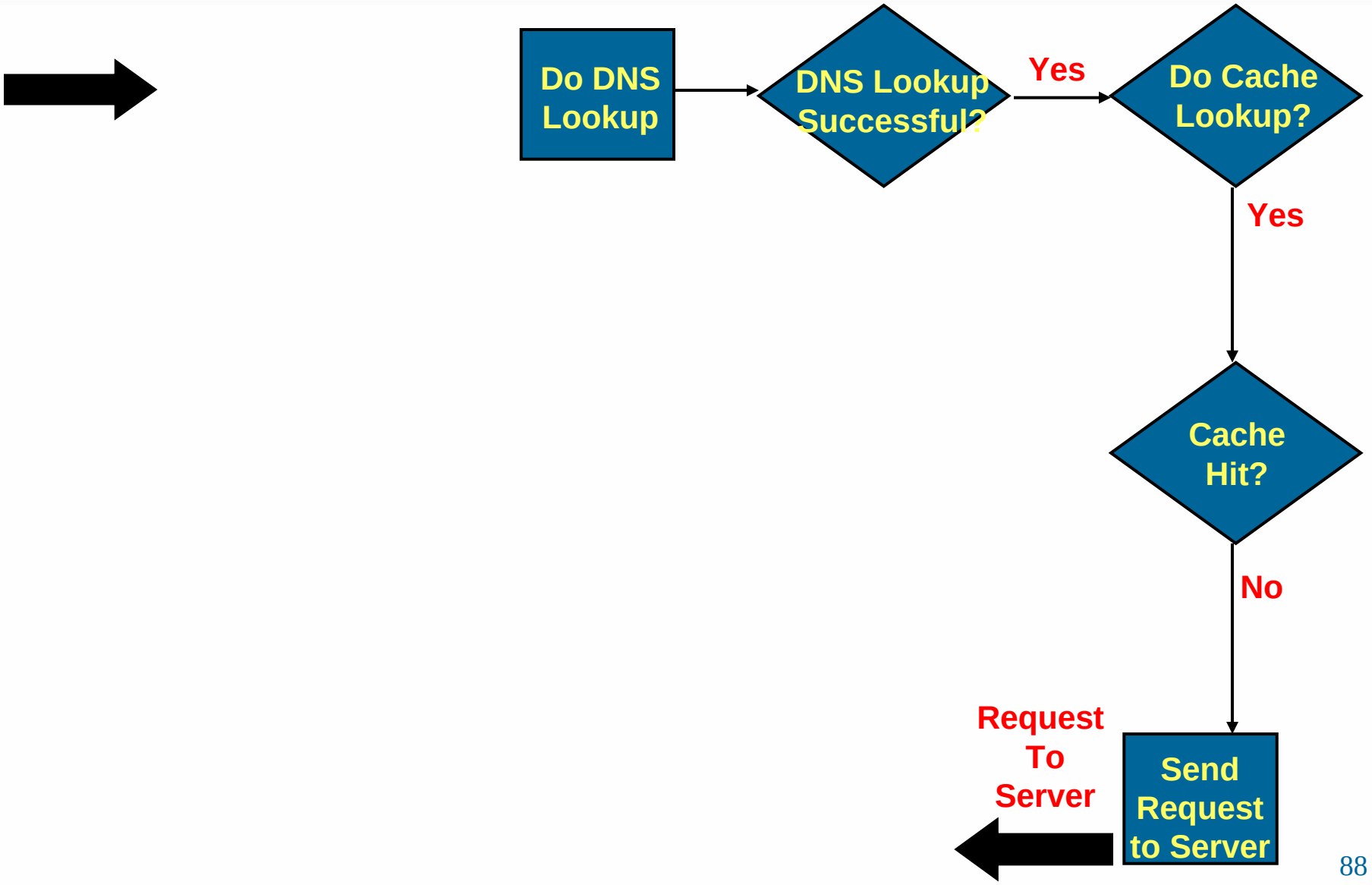
Client Request



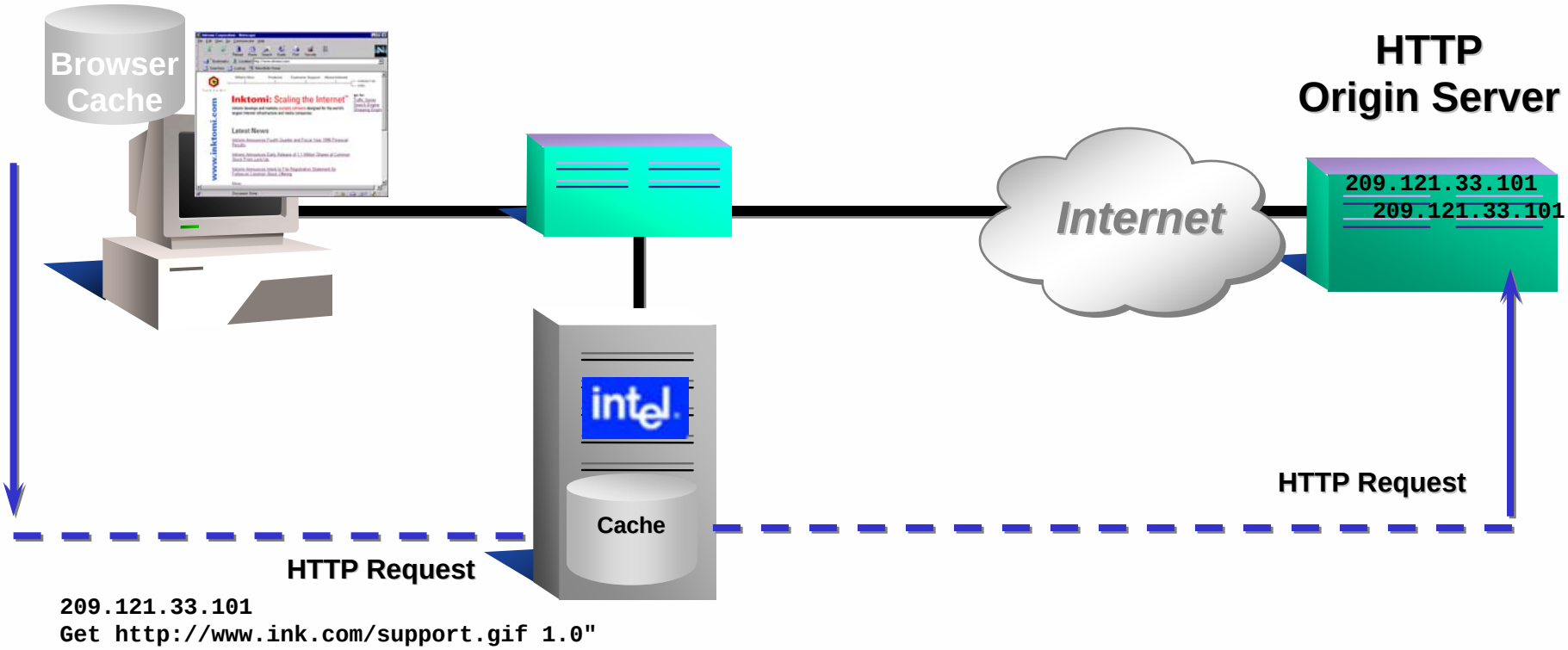
Client Request



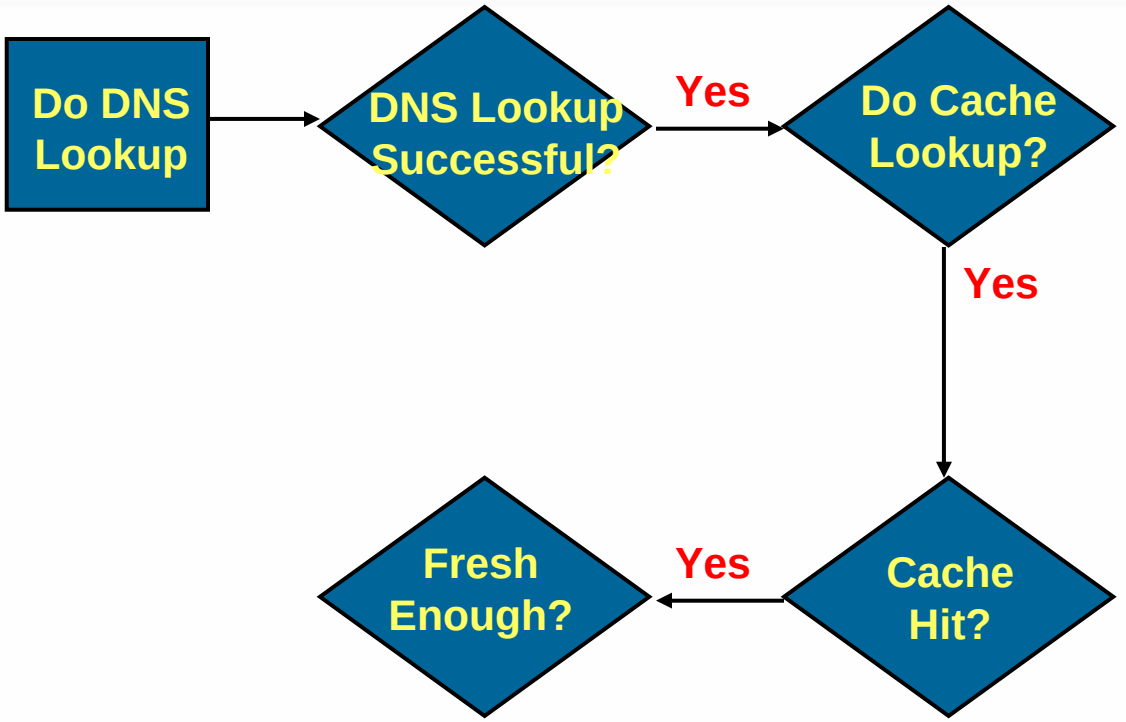
Client Request



Cache Miss


Browser Cache
HTTP Origin Server
Internet
209.121.33.101
209.121.33.101
Cache
HTTP Request
HTTP Request
209.121.33.101
Get http://www.ink.com/support.gif 1.0"

Client Request



Content Freshness

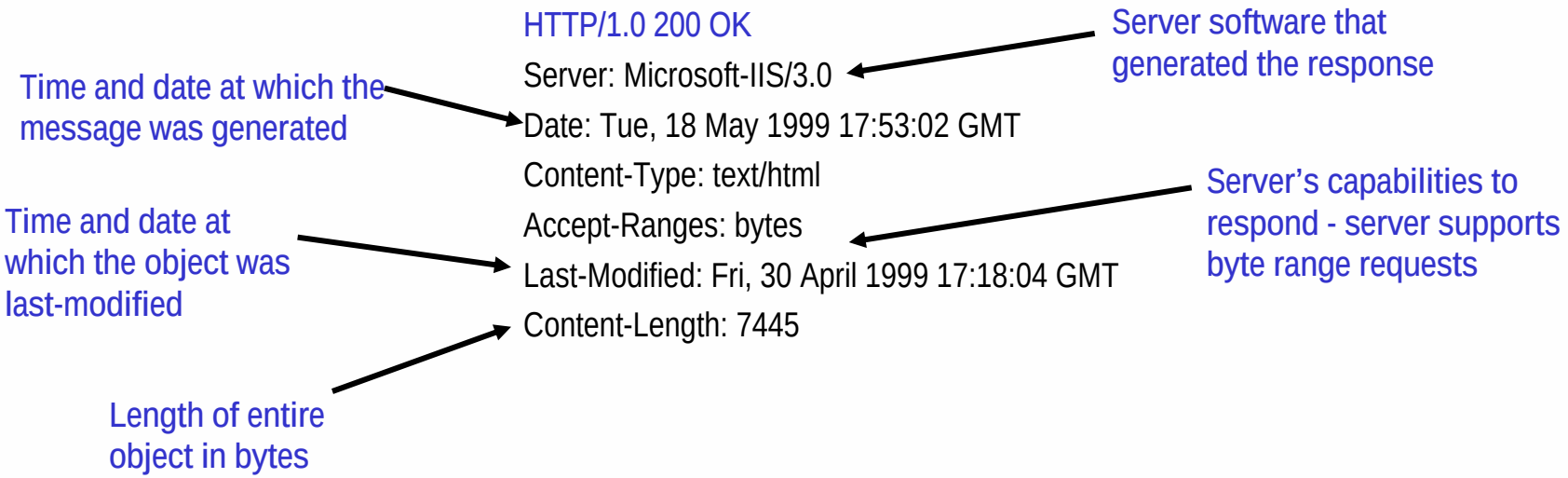
- **Keeping the Content Fresh is the Goal of all Parties involved in Object Content and Delivery**
  - Designers that maintain the content
  - Origin Server that stores and delivers the content
  - Intel Cache that stores and deliver the content
  - Client Browser that generates the request

Maintaining Freshness via Designer

- **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

Intel Cache utilizes header information to determine content freshness



(Last-Modified -Date) *10% = Cache Freshness Limit

Maintaining Freshness via Origin Server

Intel Cache utilizes header information to determine content freshness

Time and date at which the message was generated

Origin Server's instructions to the Intel Cache: Max-age overrides Expires header, Public, Private, No-cache, No-cache by field name, No-transformation, Must revalidate

```
HTTP 1.1 200 OK
Date: Sat, 20 Feb 1999 13:19:41 GMT
Server: Apache/1.3.3 (Unix)
Cache-Control: max-age=3600, must-revalidate
Expires: Sat, 20 Feb 1999 14:19:41 GMT
Last-Modified: Fri, 29 Jan 1999 02:28:12 GMT
ETag: "3e86-410-3596fbbc"
Accept-Ranges: bytes
Content-Length: 1040
Content-Type: text/html
```

Tells all caches the time and date until which an object should be considered fresh

Server's capabilities to respond - server supports byte range requests

Entity tag is generated by the origin server and is utilized during an object revalidation

Length of entire object in bytes

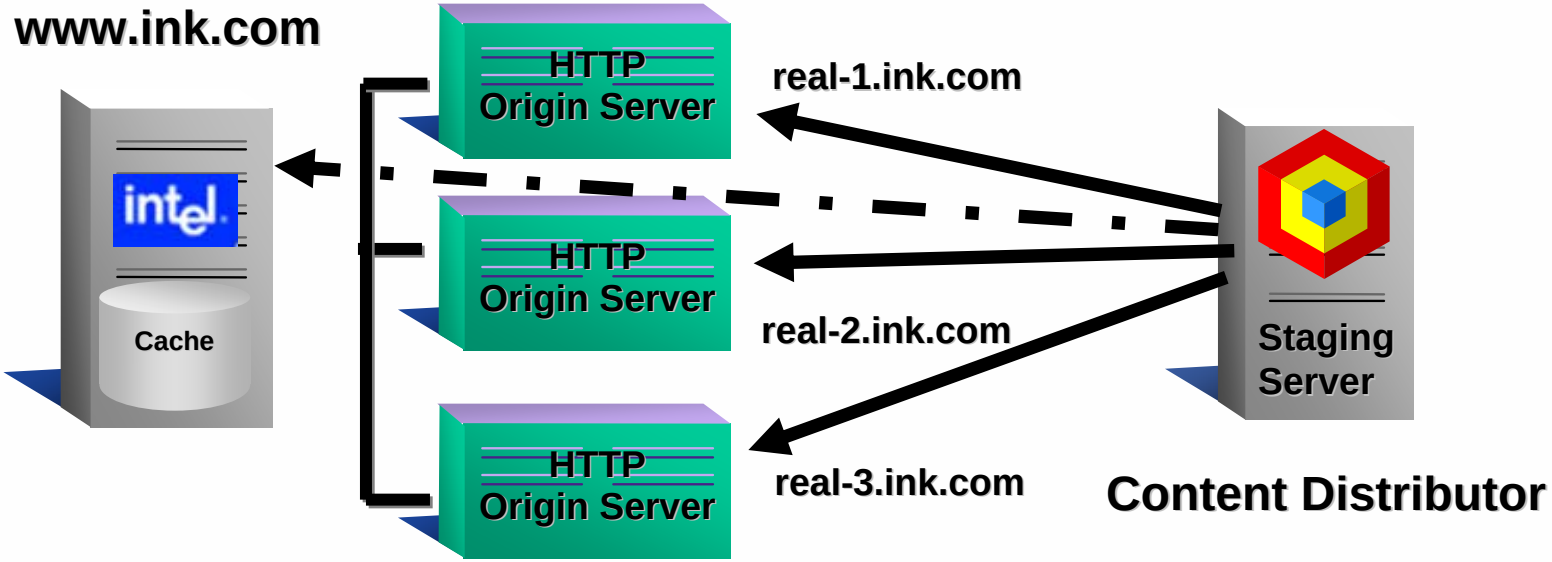
HTTP Request also contains Cache-Control. This allows the client to monitor the source of a requested object. No-cache, No-store, Max-age, Max-stale, Min-fresh, Only-if-cached

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

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



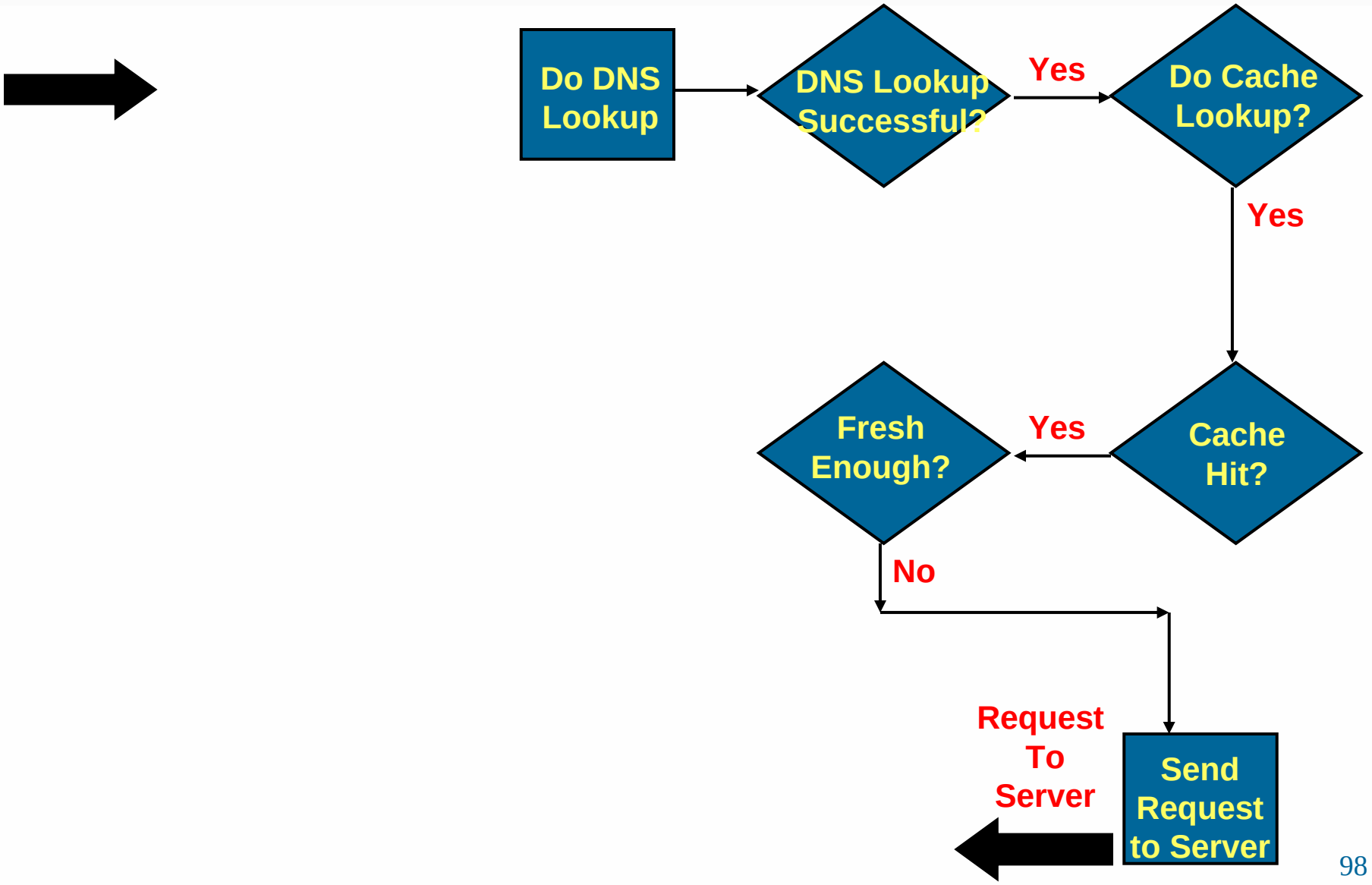
1. Content is updated at the staging server.
2. Content is replicated to origin servers.
3. After update, Intel Cache is told to delete stale content.

➡ Users always receive fresh content

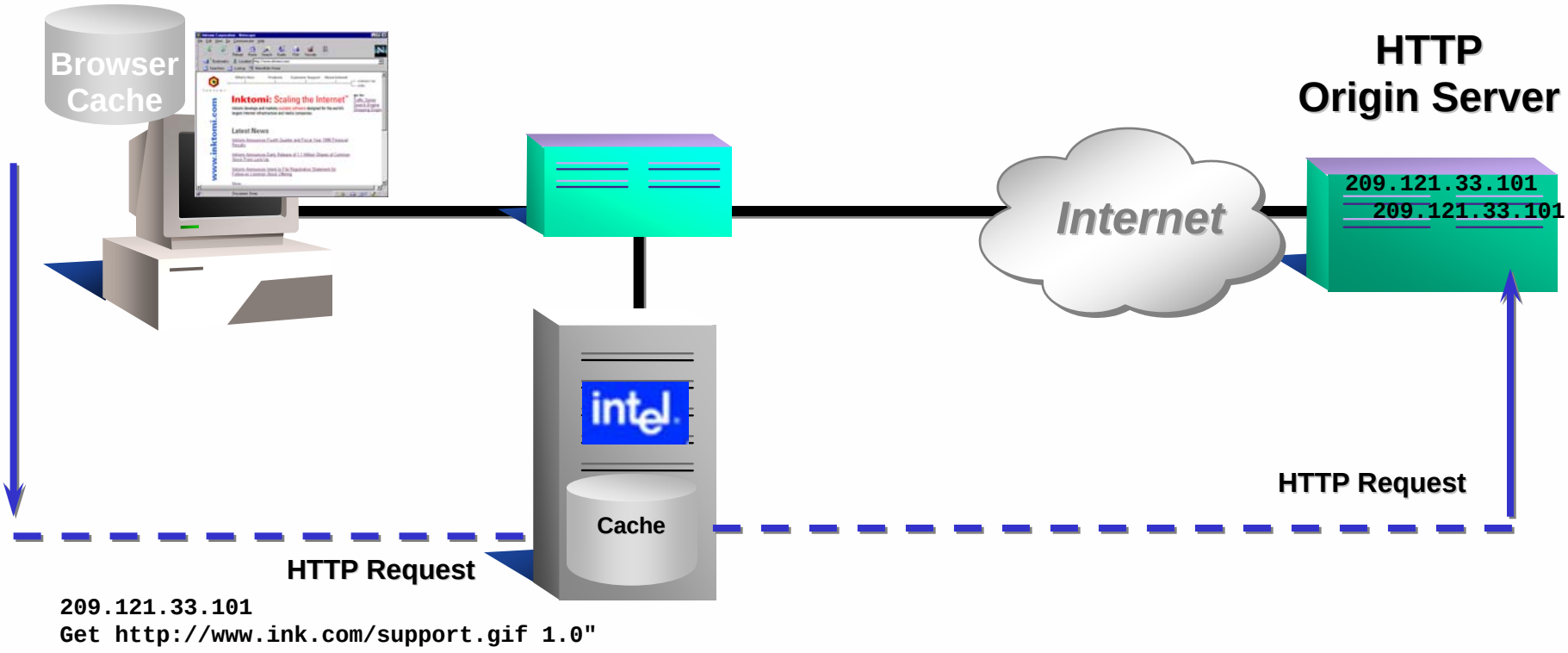
Client Freshness

- **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.**

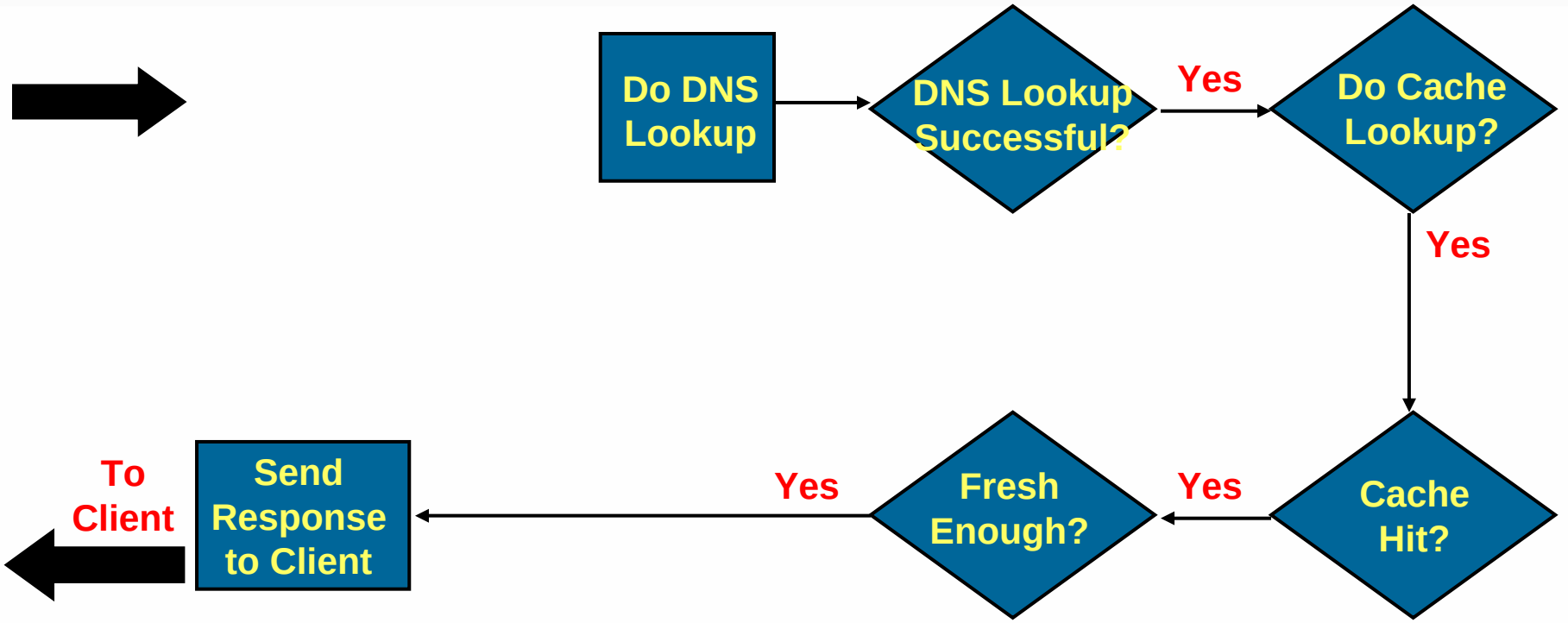
Client Request



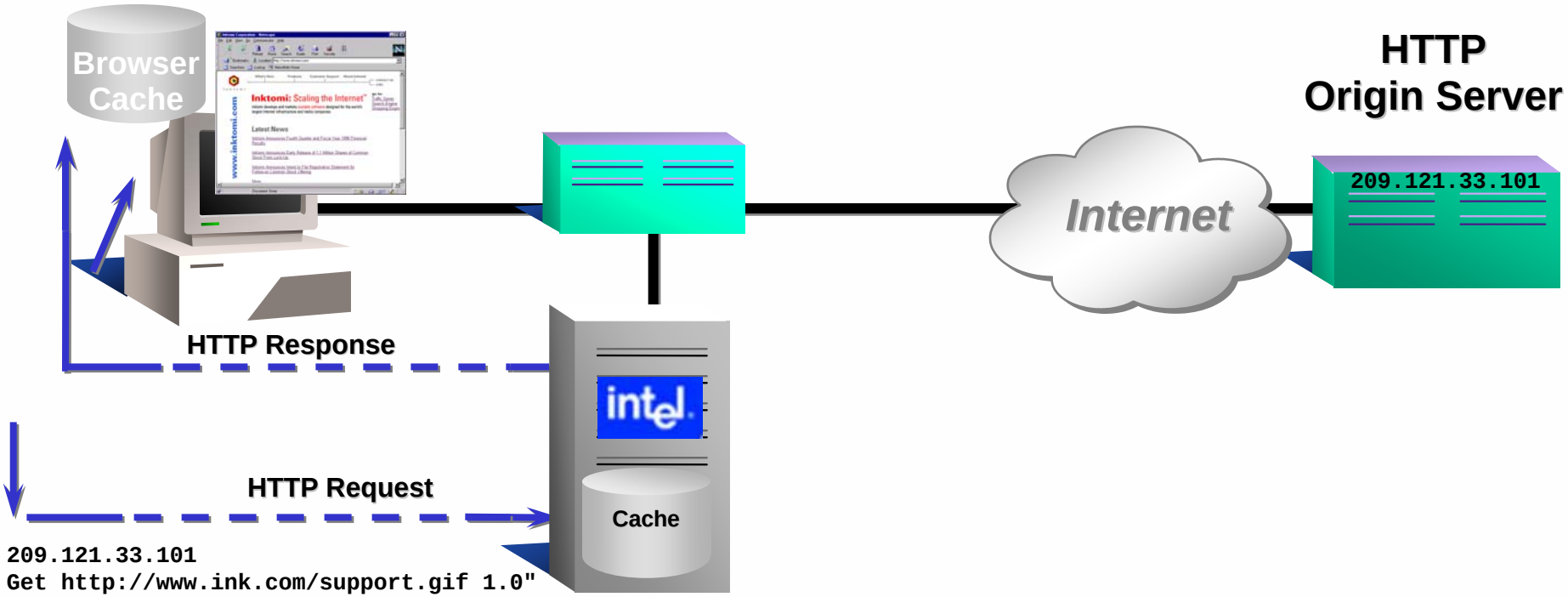
Cache Refresh



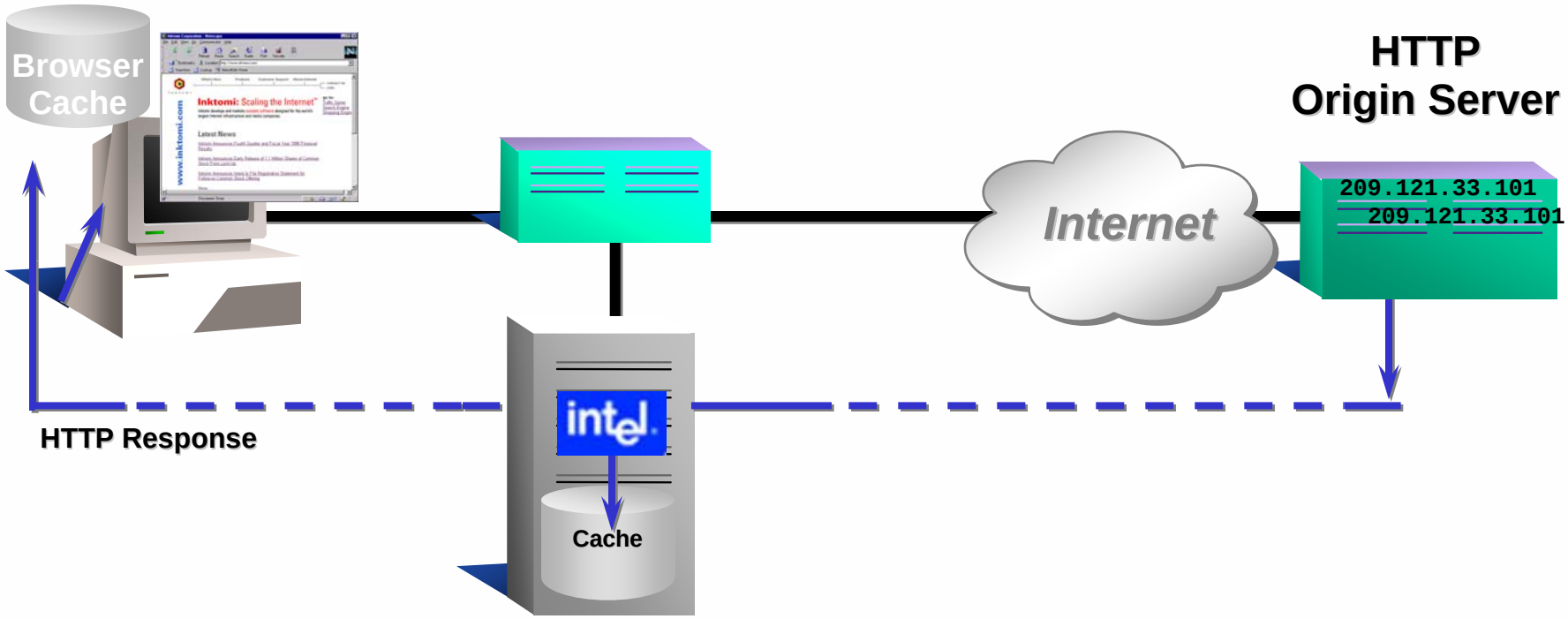
Client Request



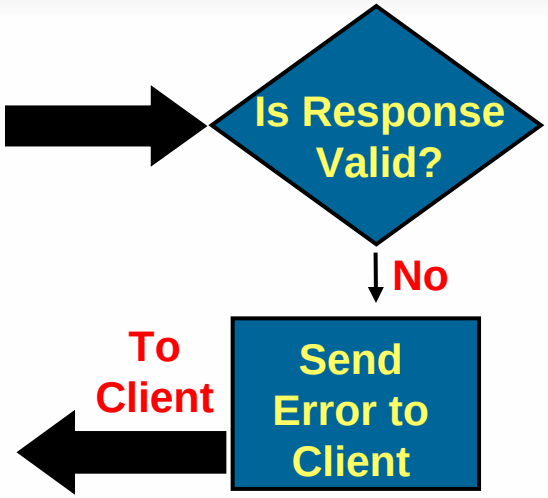
Cache Hit


Browser Cache
HTTP Origin Server
Internet
209.121.33.101
HTTP Response
HTTP Request
Cache
209.121.33.101
Get http://www.ink.com/support.gif 1.0"

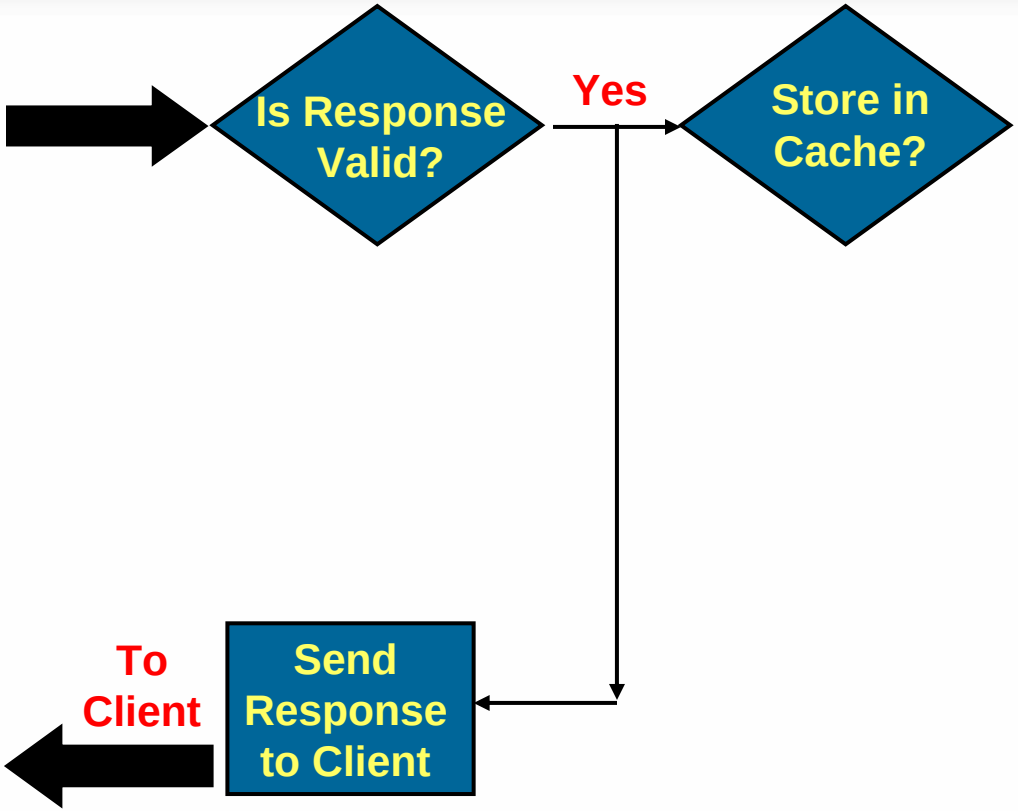
Origin Server Response



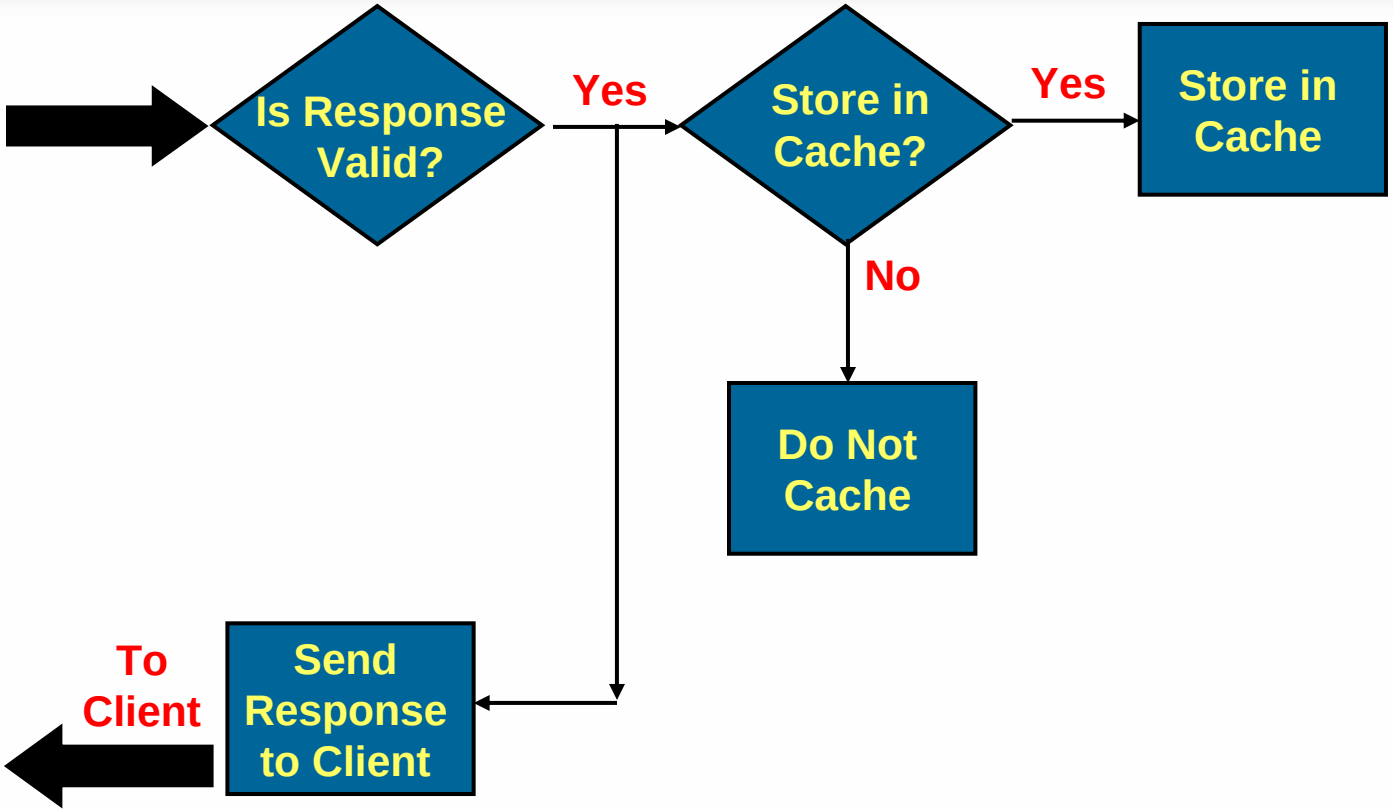
Response from Server



Response from Server



Response from Server



Benefits of Forward Caching

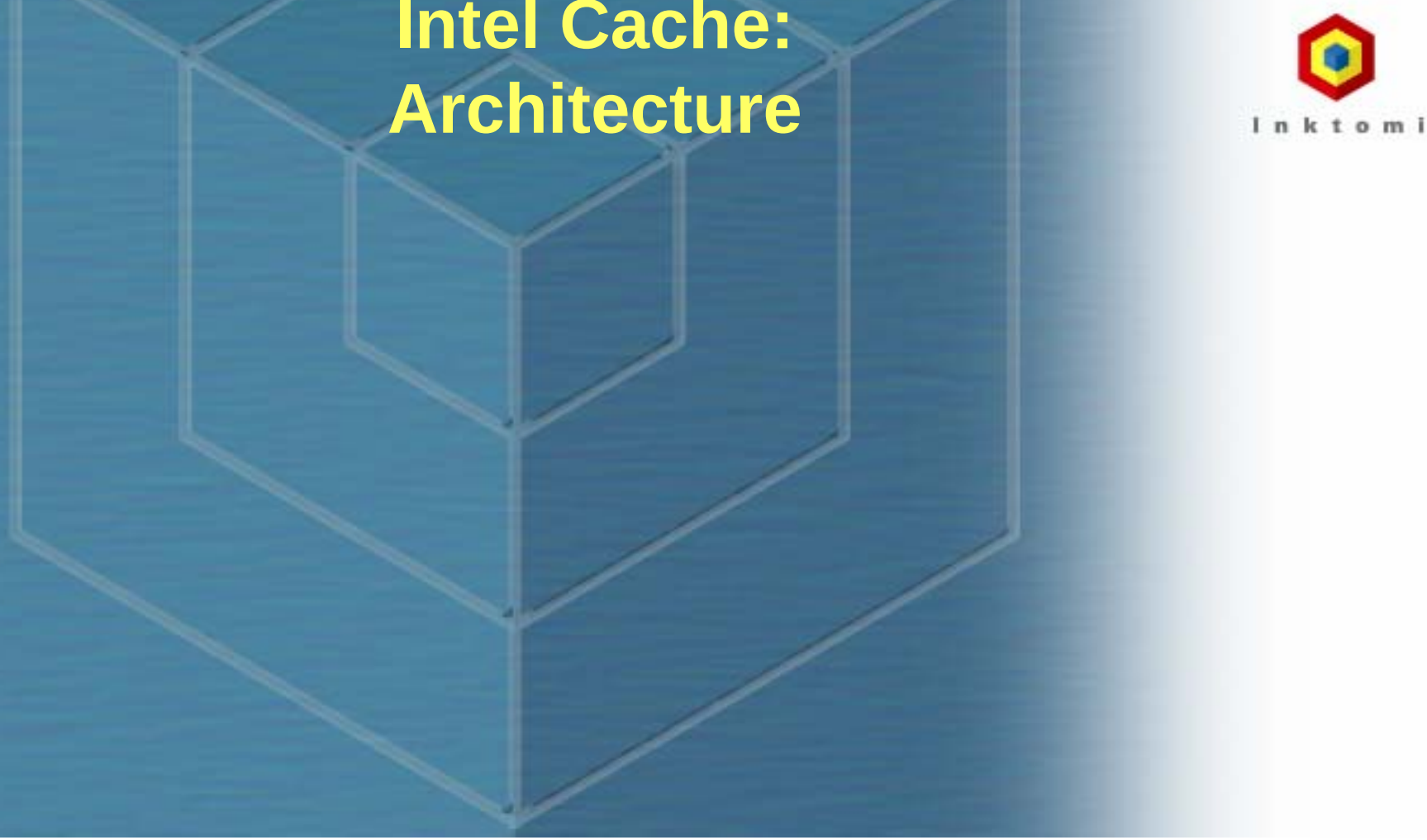
- **Alleviates network congestion by significantly reducing the transmission of redundant Internet data**
  - lower bandwidth costs
  - more controlled network build-out
  - reduced load on critical network routers and origin Web servers
- **Increased quality of service for the end user by storing frequently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

Review Questions:

- What are the benefits of installing a Intel Cache?

- What is the difference between a Hit and Miss?

Intel Cache: Architecture



Chapter Overview

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

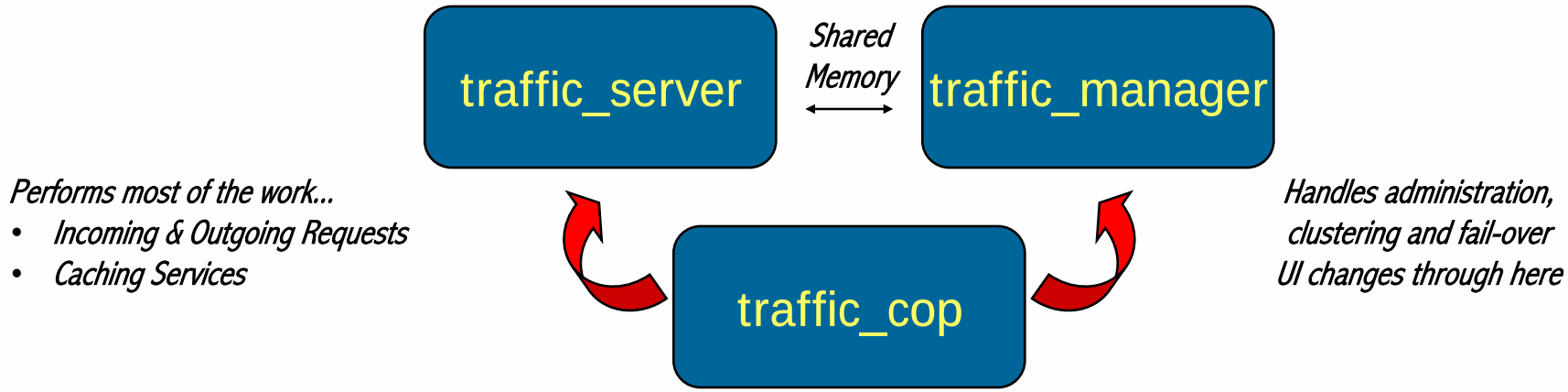
Intel Cache Innovations

- **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

- **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

- **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

- **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**

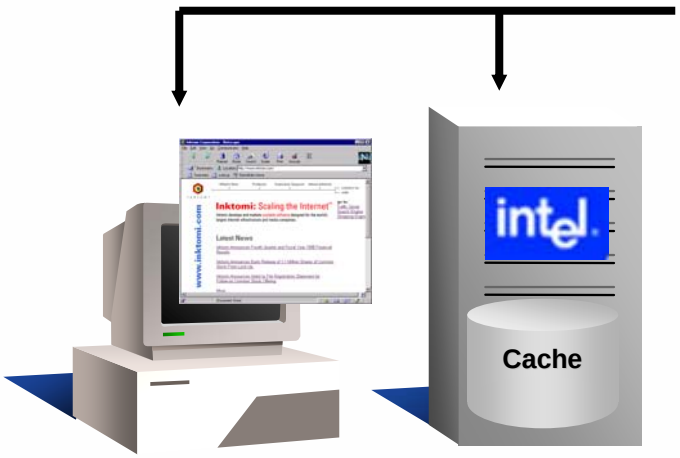
- **Responsibilities:**
  - Heartbeat tests (fetches synthetic.txt)
    - Occurs every 10 seconds
    - Heartbeat is logged to Intel Cache's access log

      `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

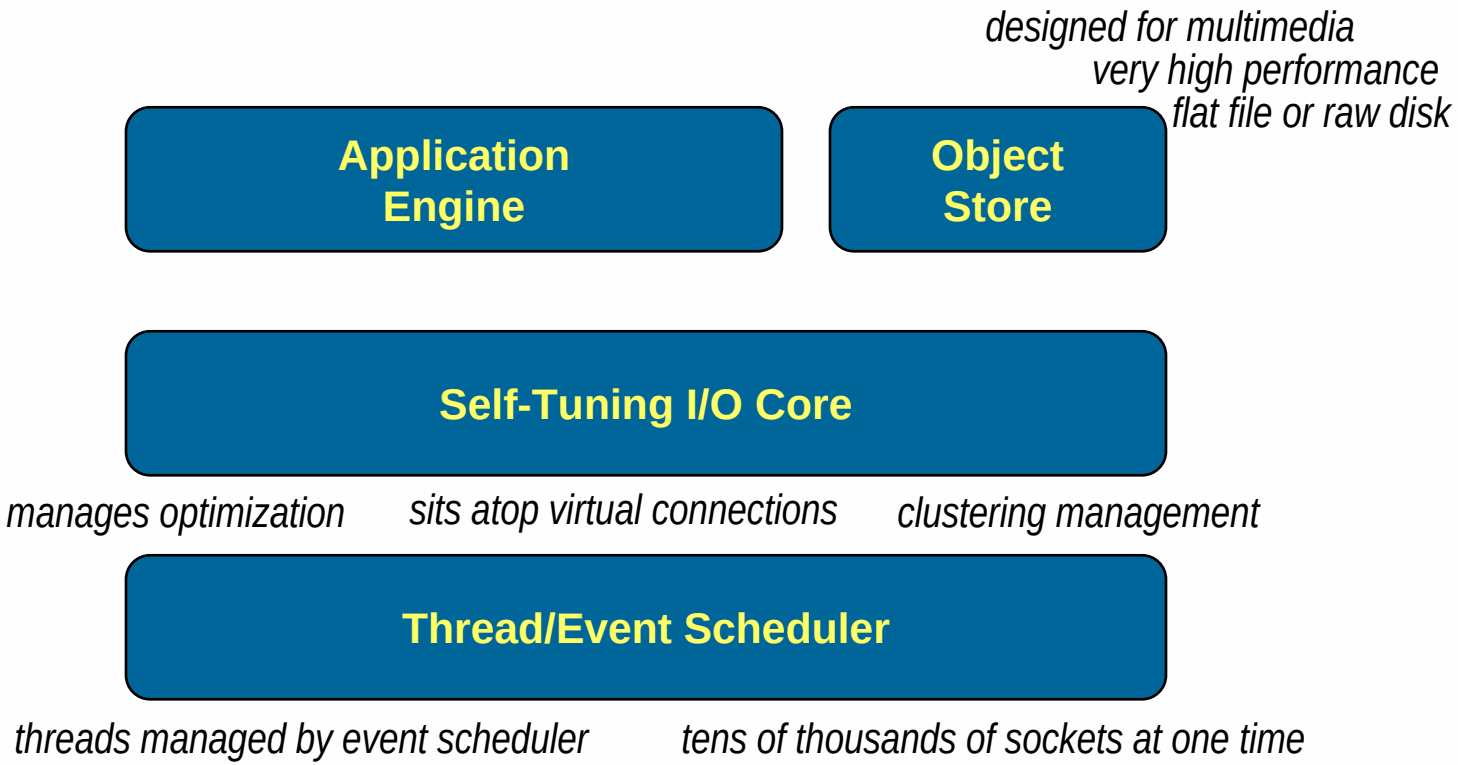
- **Designed for Performance**
  - Streaming dataflow engine rapidly transfers data to and from disk and network connections
    - Adapts to network and disk performance
    - Minimizes use of system resources



Pipeline streams data objects from web hosts to users <u>while</u> it caches them

Intel Cache Architecture

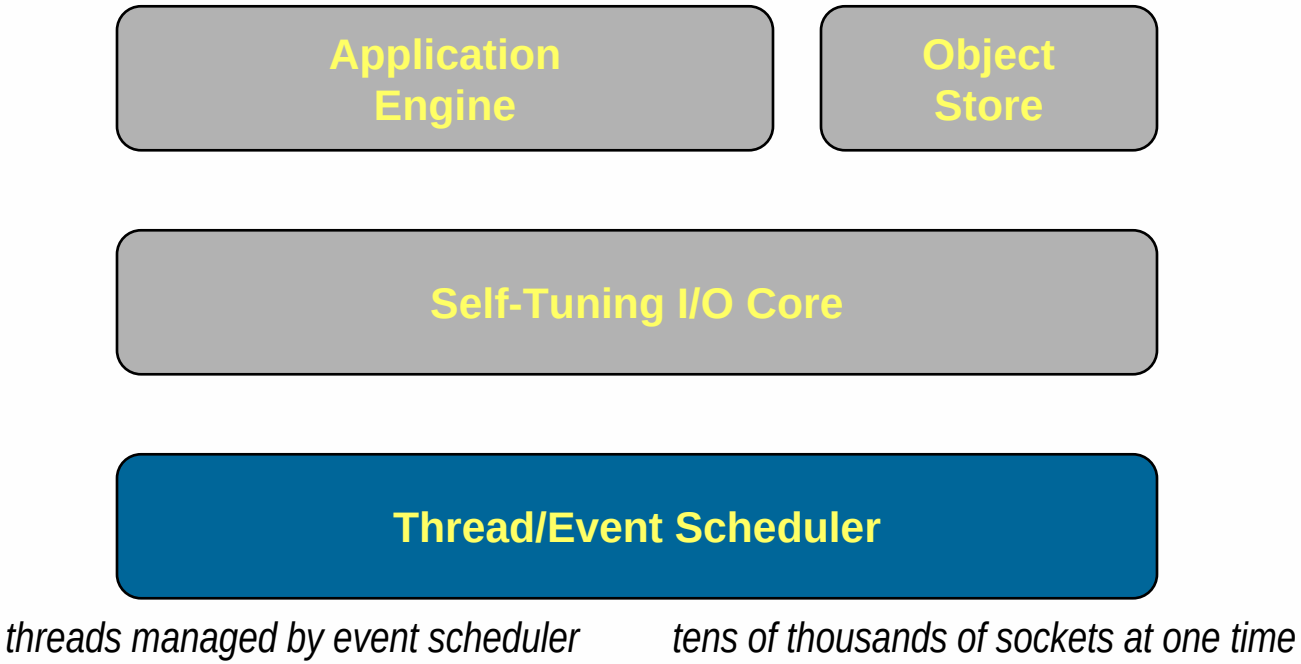
- **Modular, layered system**
- **Uniform API for network, disk, cluster and cache**



Thread/Event Scheduler

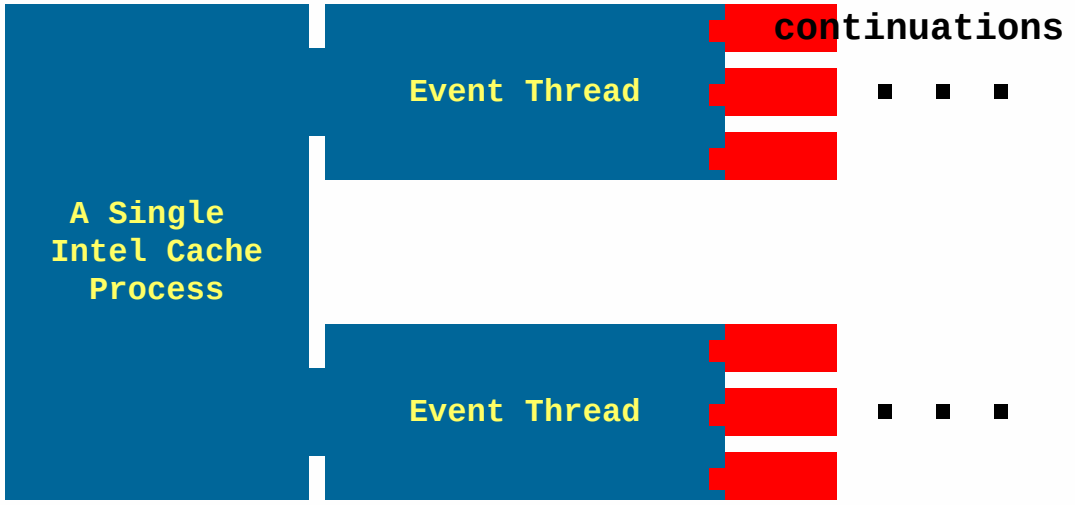
Inktomi

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



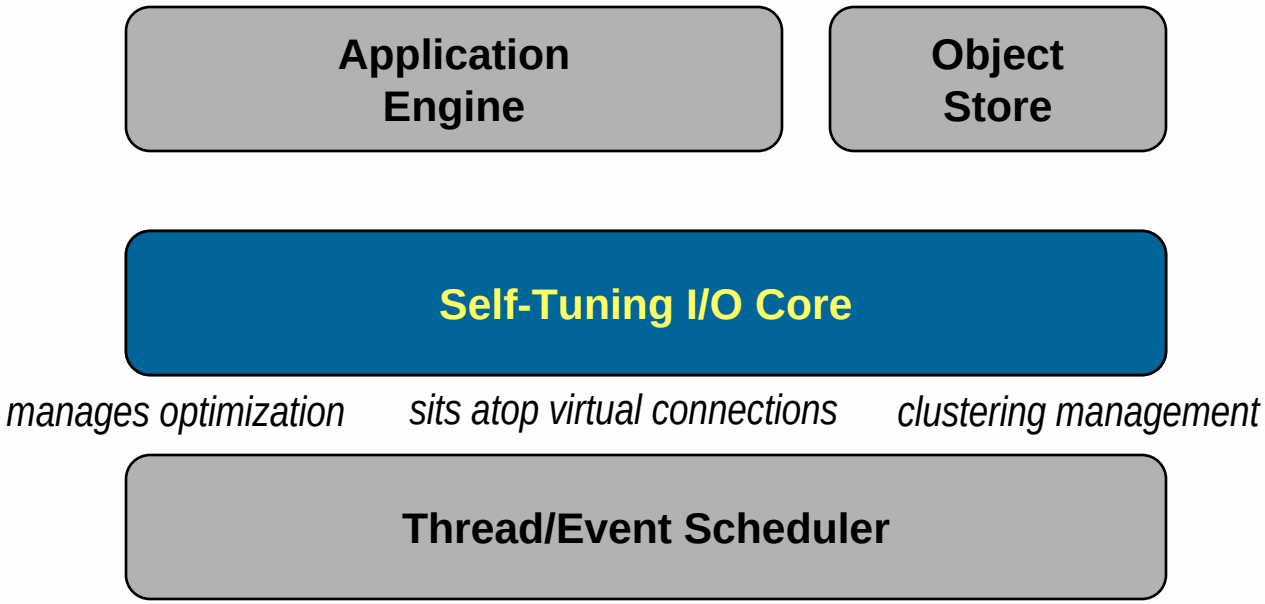
Continuations

- **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**



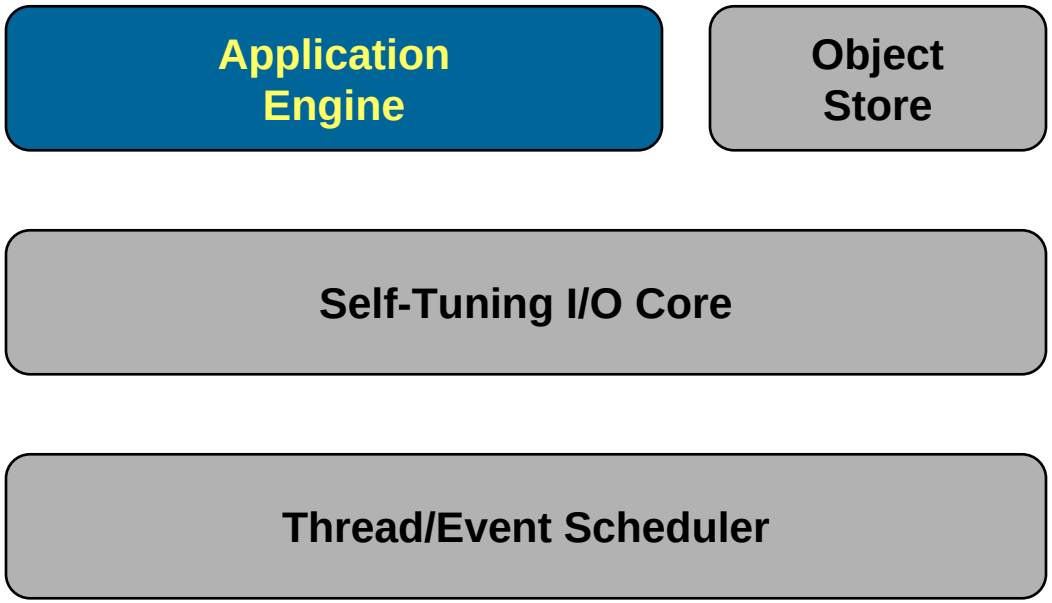
Self-Tuning I/O Core

- **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!

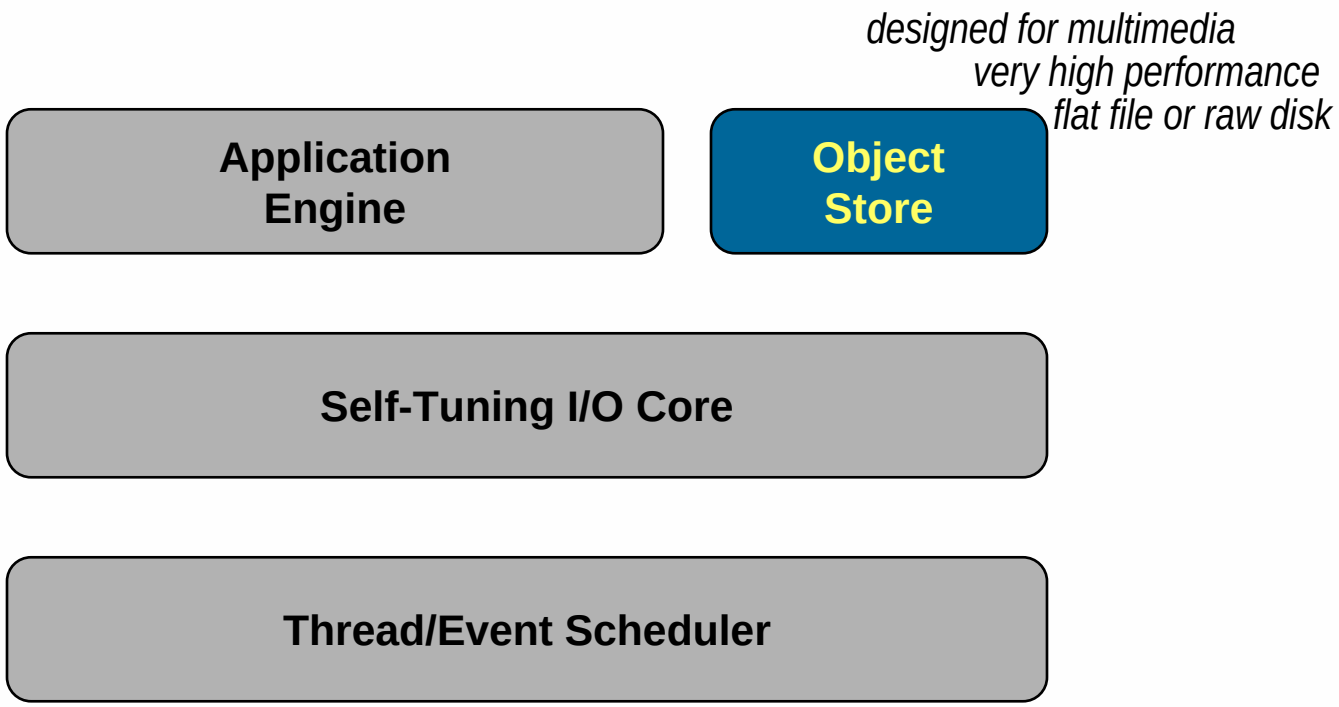


Application Engine

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



Object Store



Object Database

- **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

- **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

- **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:

- What are three processes in Intel Cache?

- What are the benefits of the Data Flow Architecture?

Intel Cache: Configuration and Monitoring Interfaces

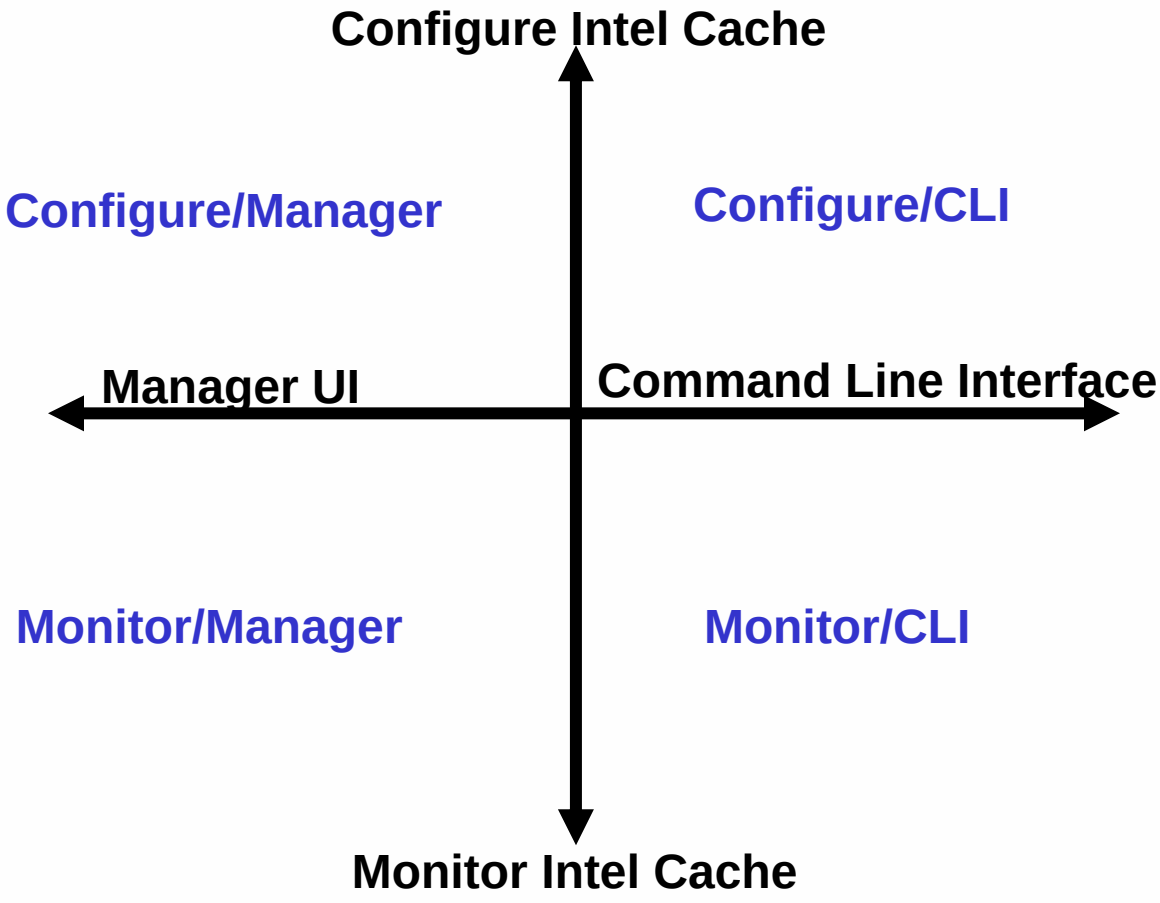
Inktomi

Chapter Overview

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

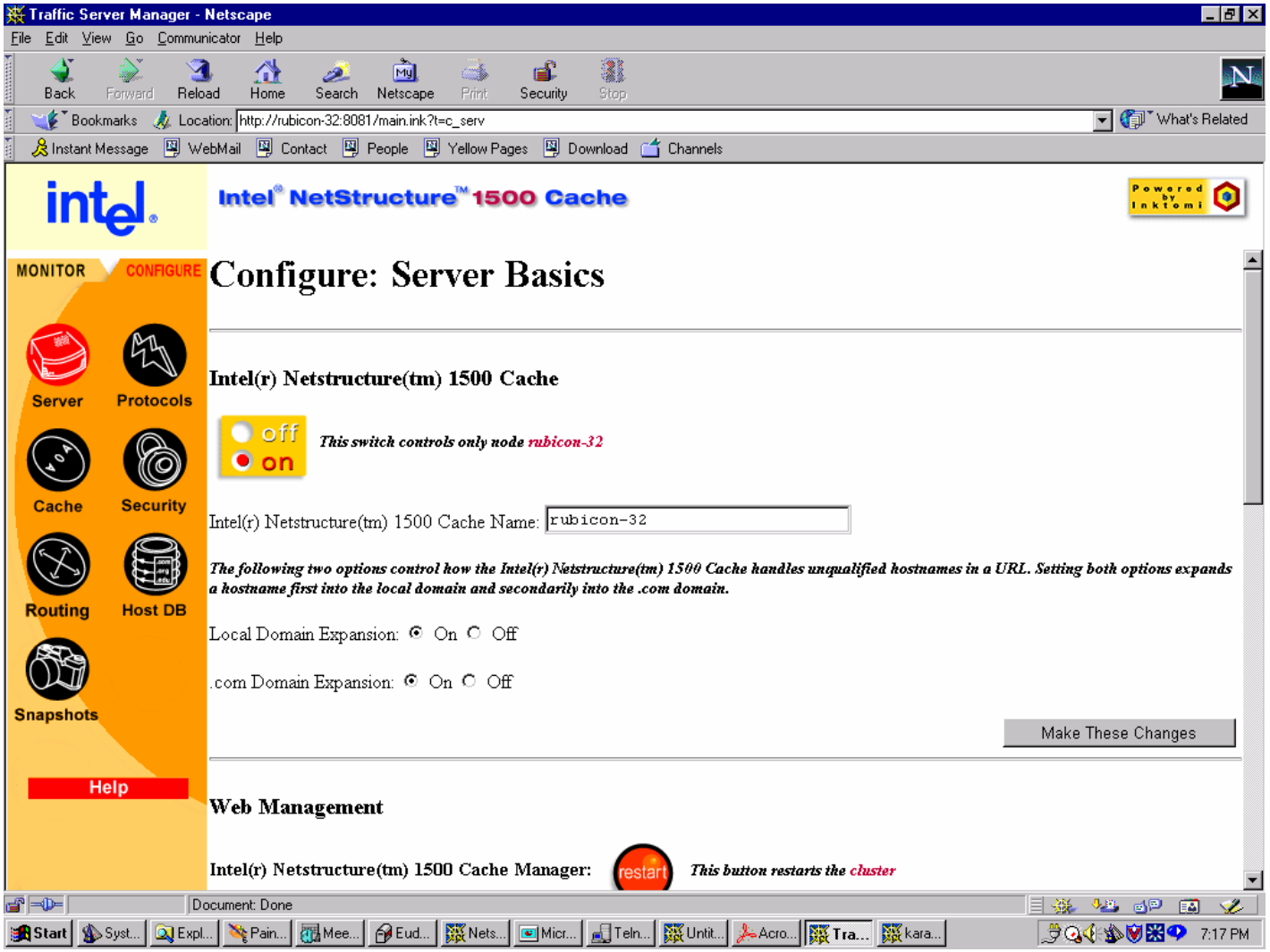
Interfaces and Usage

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

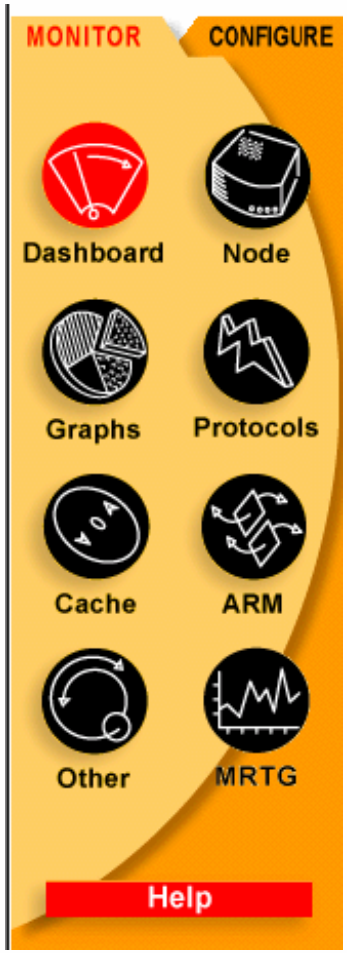


Manager User Interface

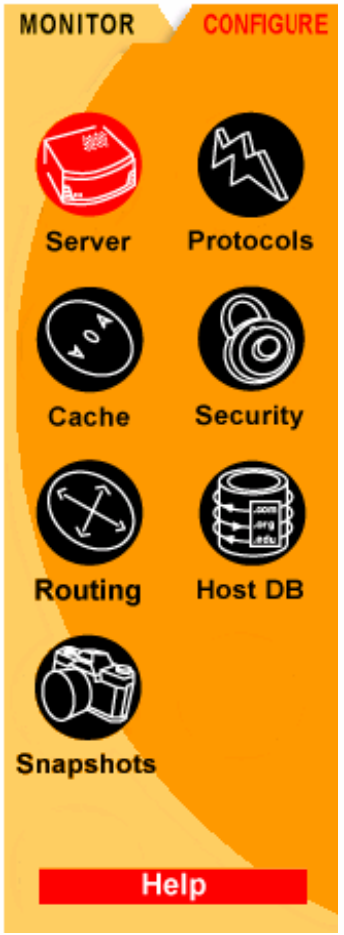
- **Enter URL of Intel Cache and proxy port (usually 8081)**



Manager UI Usage

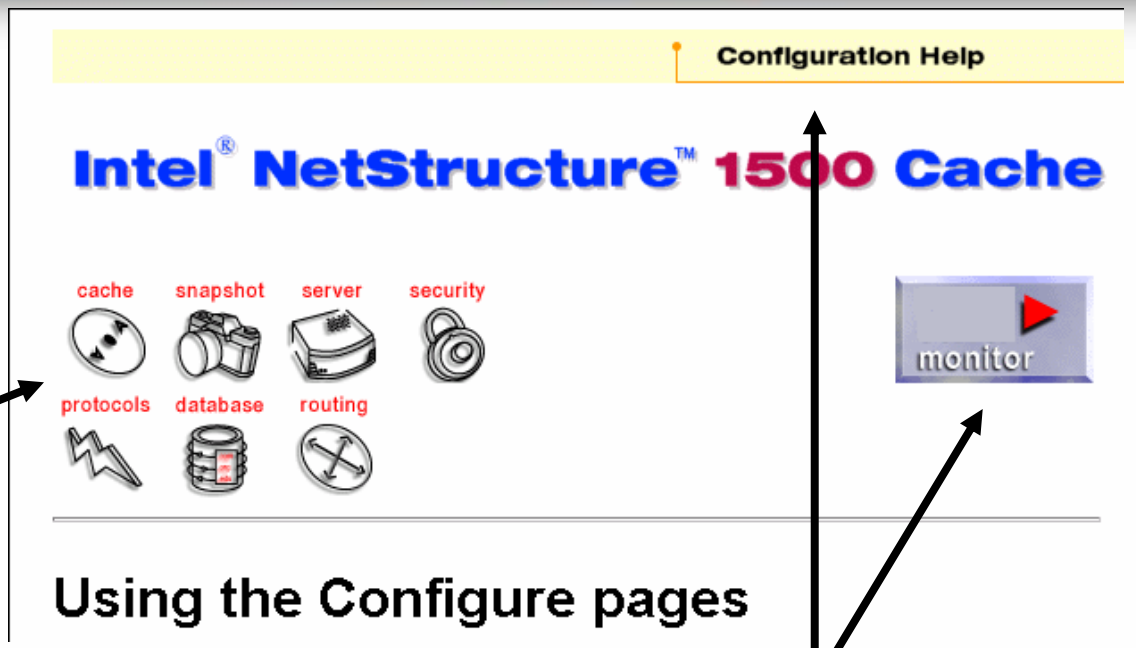


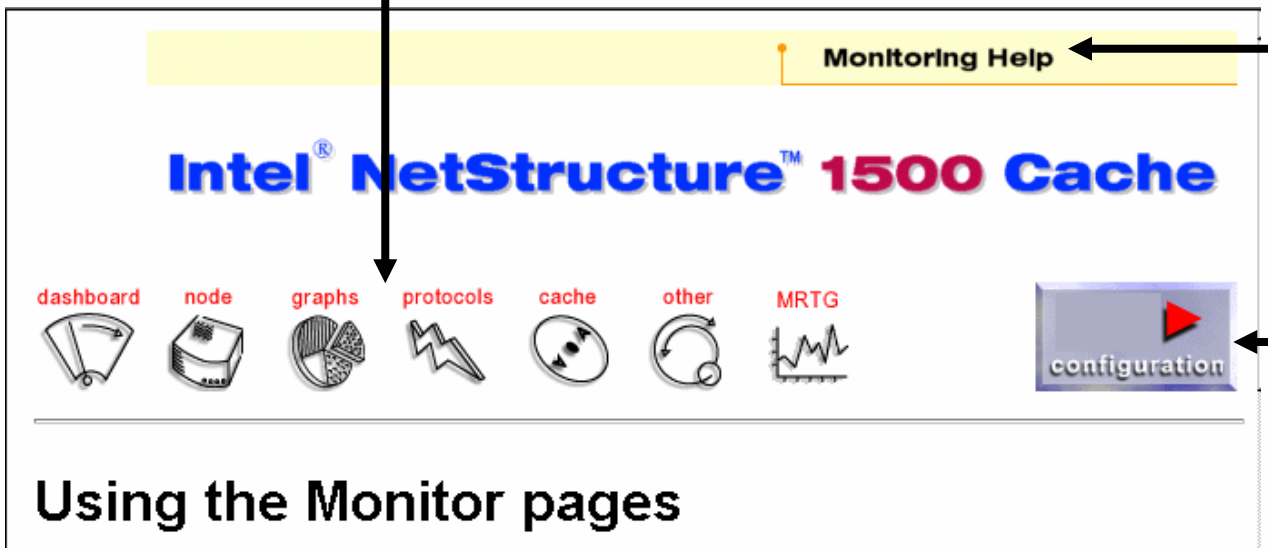
- 2 menus: Monitor and Configure
- Select menu by clicking on title (Monitor or Configure)
- Red text indicates which menu is selected
- Select menu items by clicking on them
- Red icon indicates which menu item is selected
- *Help* icon opens a new browser window with a help page for the currently selected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



Manager UI Help Pages

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





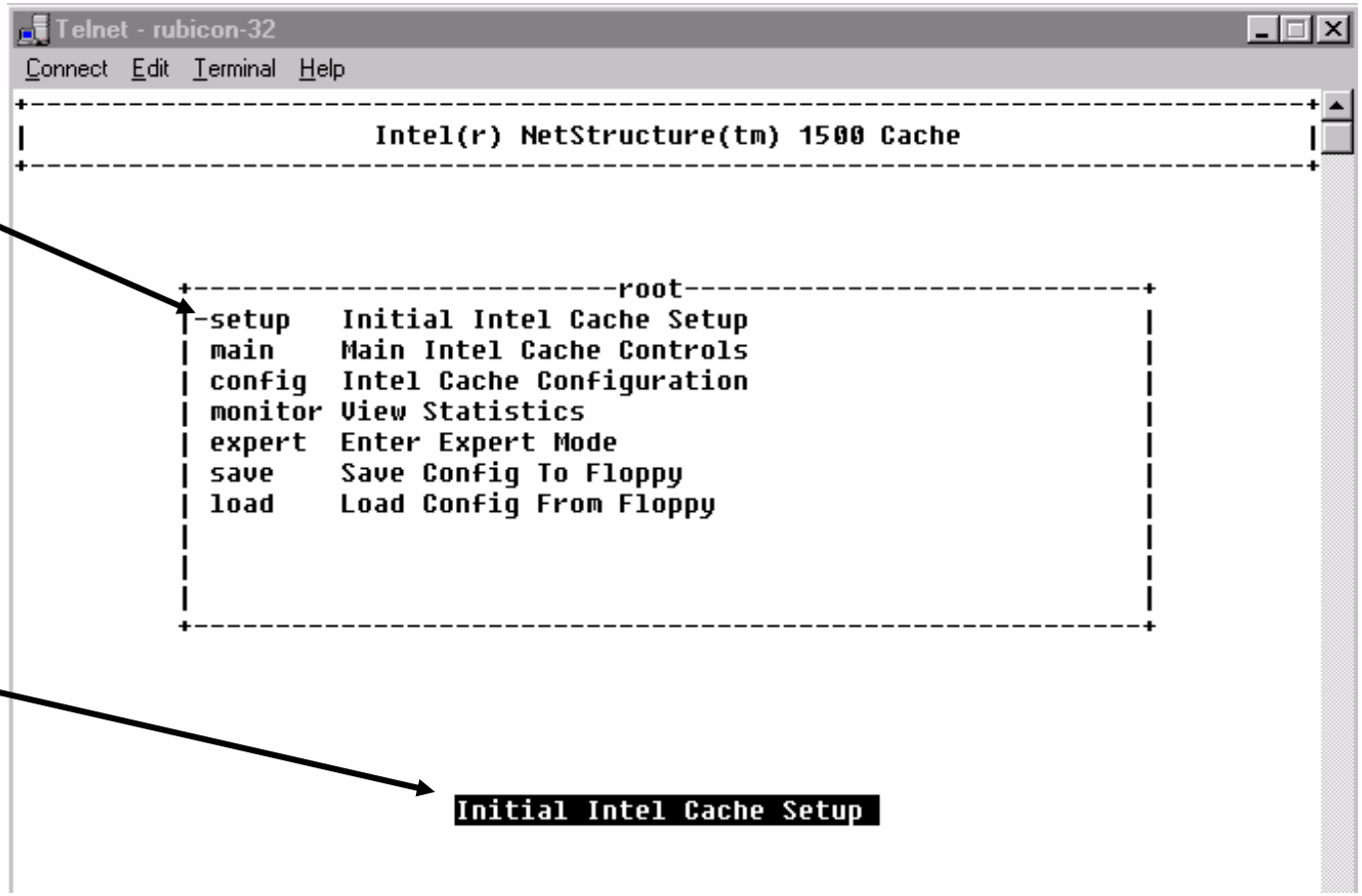
- Get to overall help pages by clicking either <Mode> Help in upper right corner or by clicking labeled arrow boxes

Command Line Interface

- `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

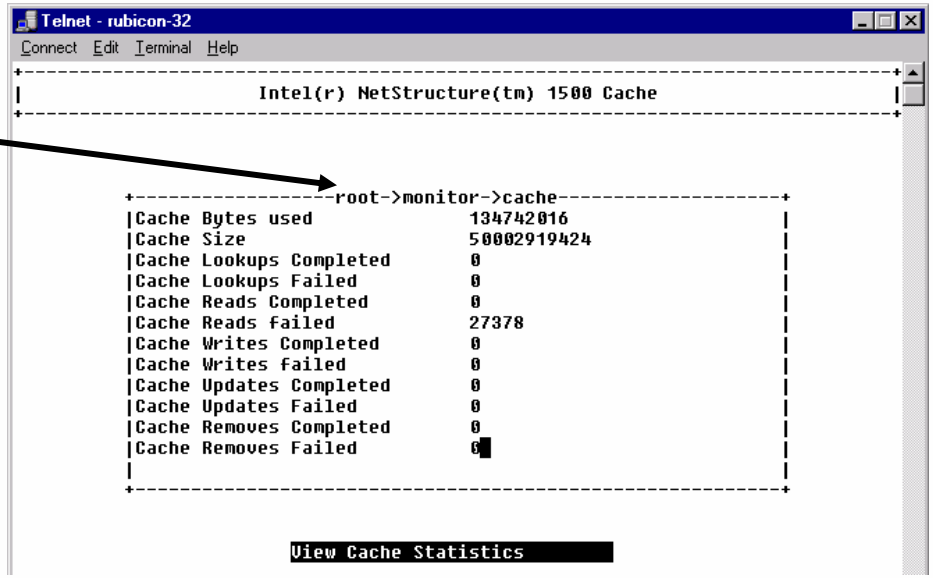
Brief description of selected menu item



Navigating the Command Line Interface

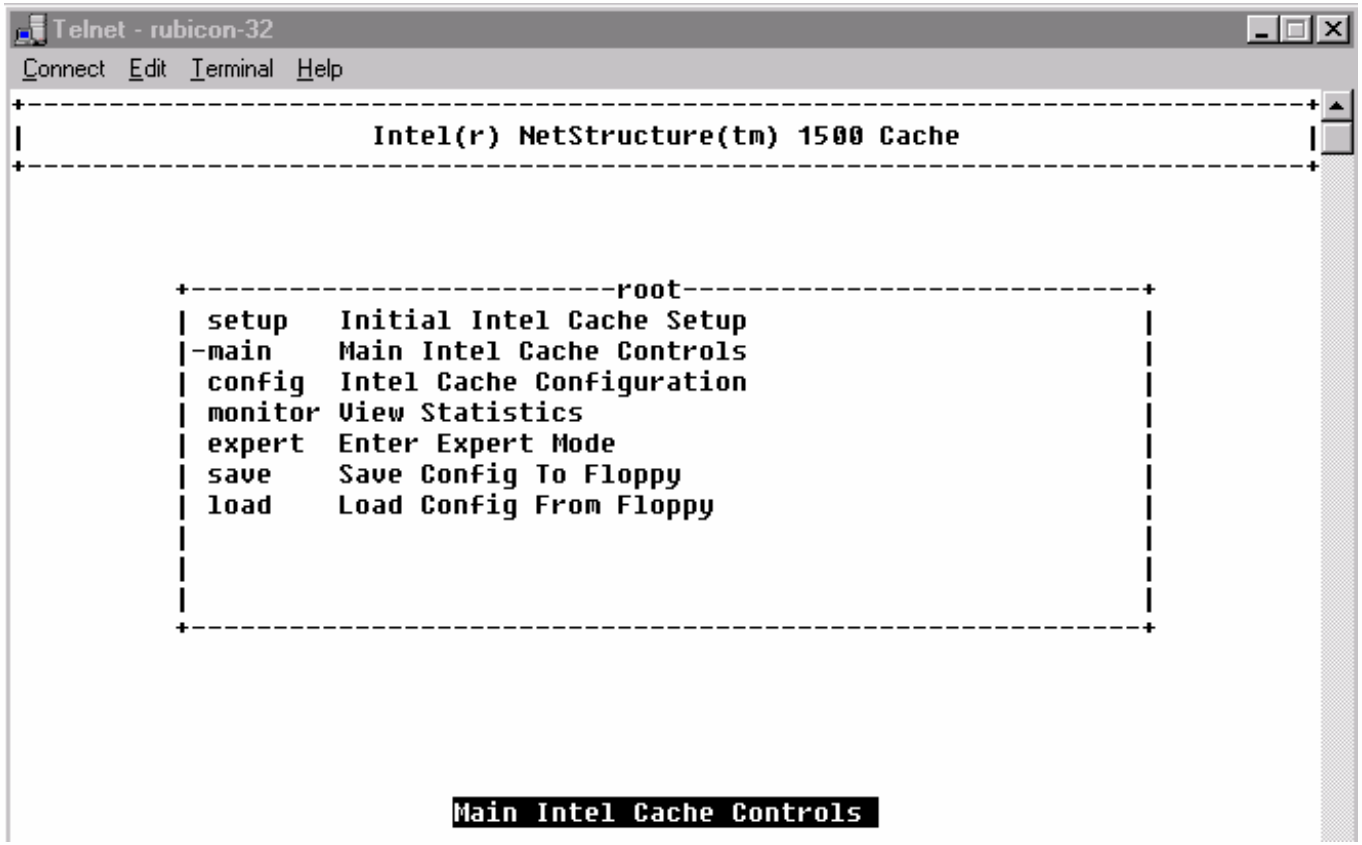
- **Move within a menu:** **Up and Down Arrow Keys**
- **Select a menu item:** **Enter**
- **Return to previous menu:** **Ctrl-x**
- **Save information field in a form window** **Enter**
- **Accept data confirmation box after entering information** **Ctrl-x**
- **Exit a form window** **Ctrl-x**

Current location and path in menu hierarchy

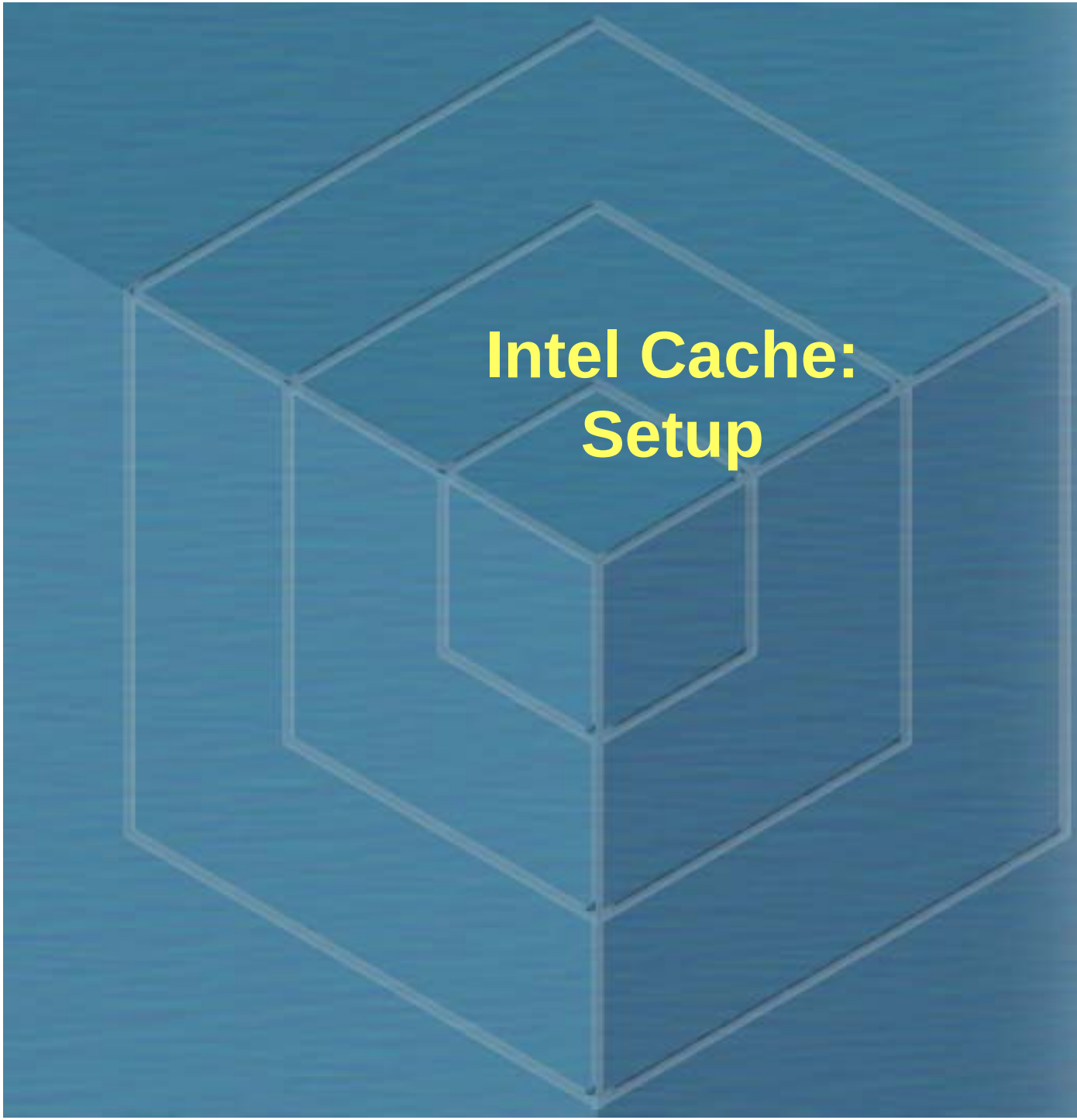


Command Line Interface: Initial Menu

- **Top level menu with seven submenus/commands**



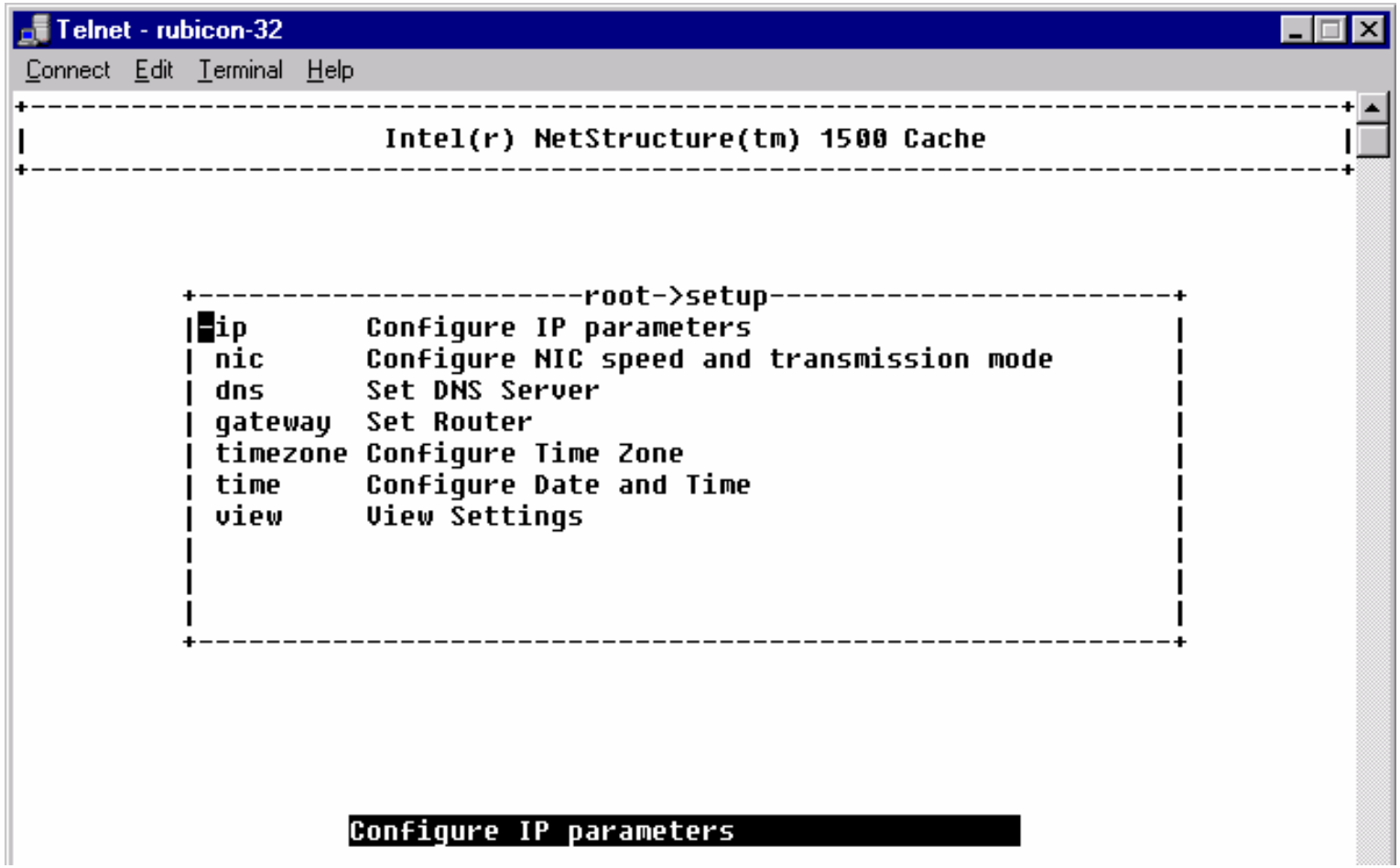
Intel Cache: Setup



Inktomi

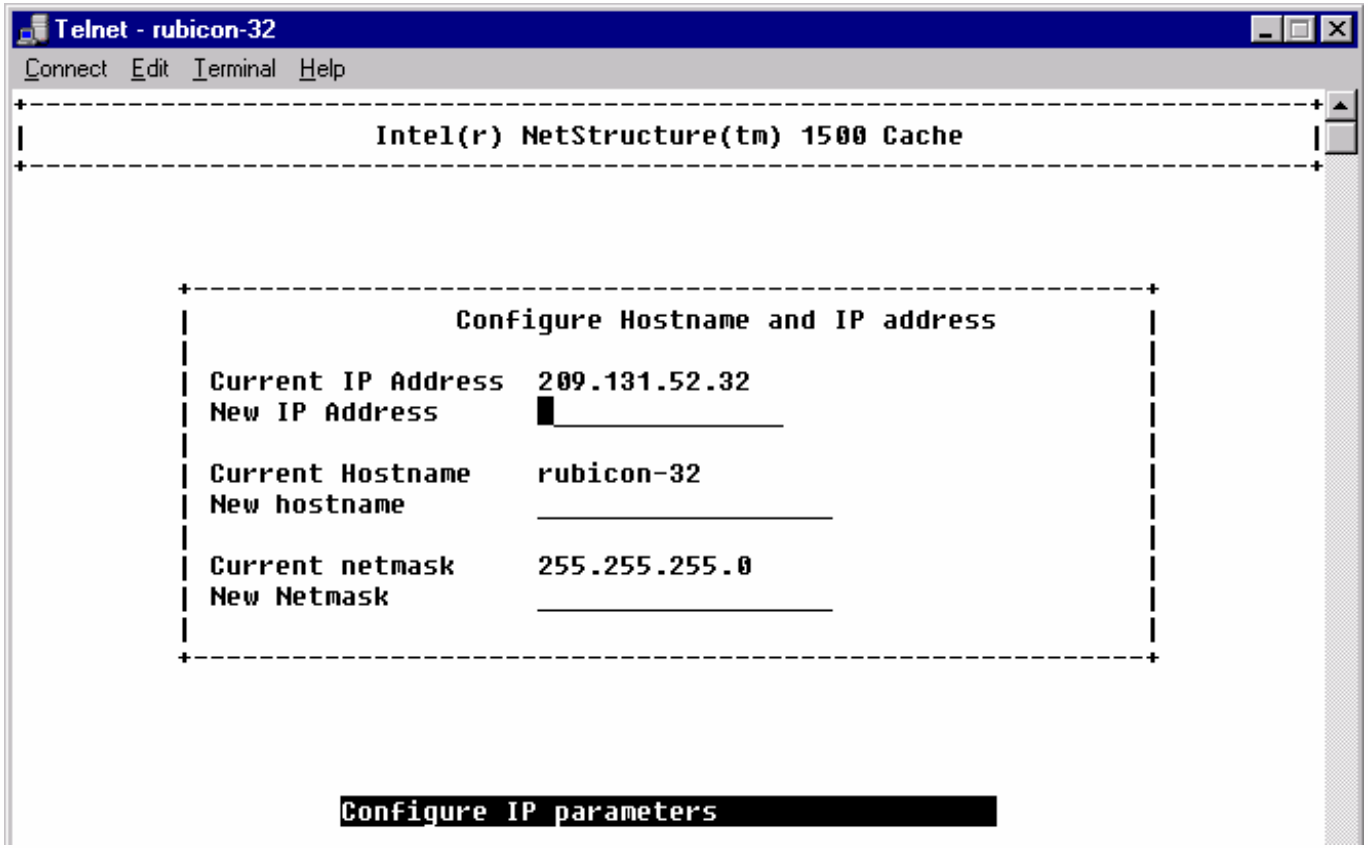
CLI: Setup Menu

- **Used to configure basic network settings for the Intel Cache after startup**



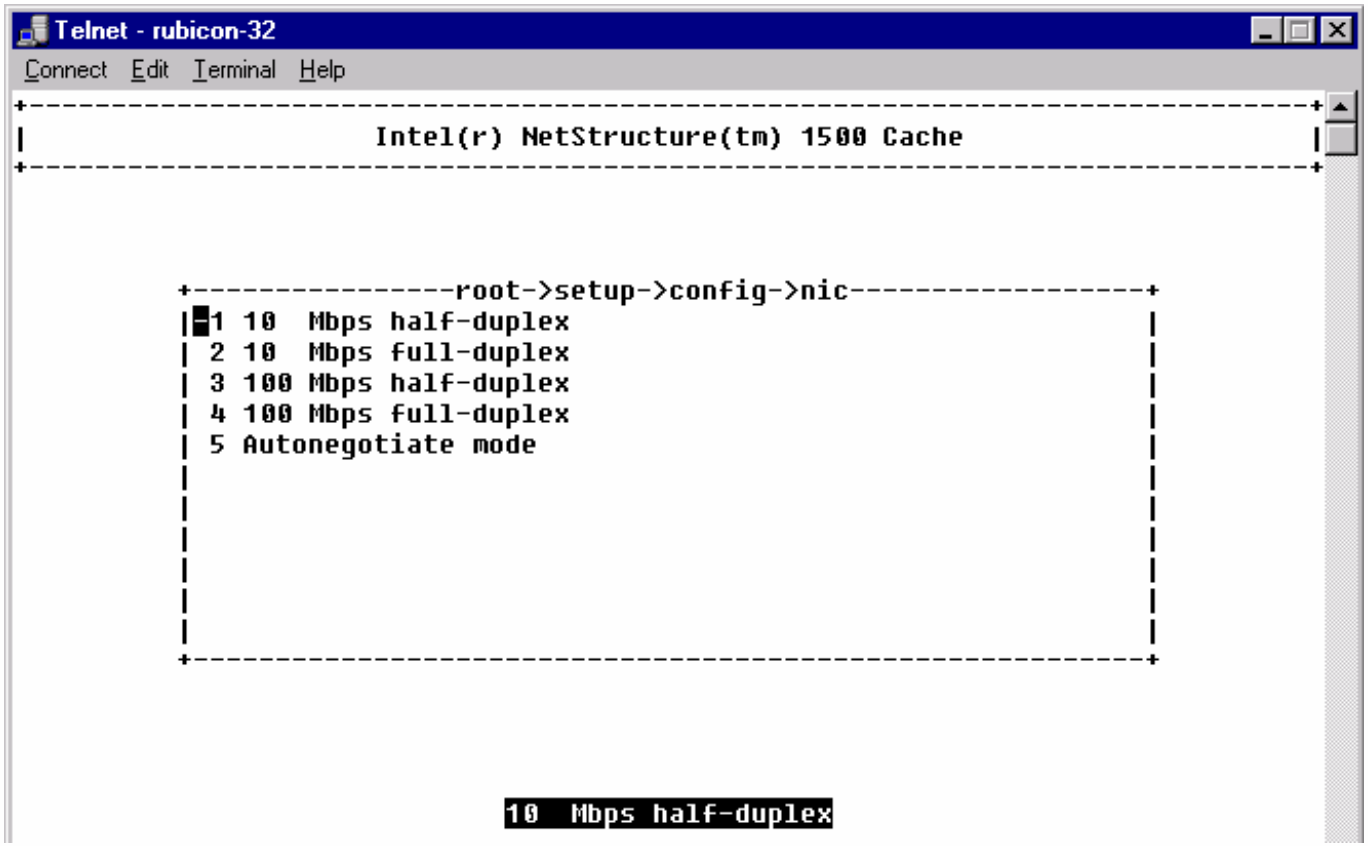
CLI: Setup->IP Parameters

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



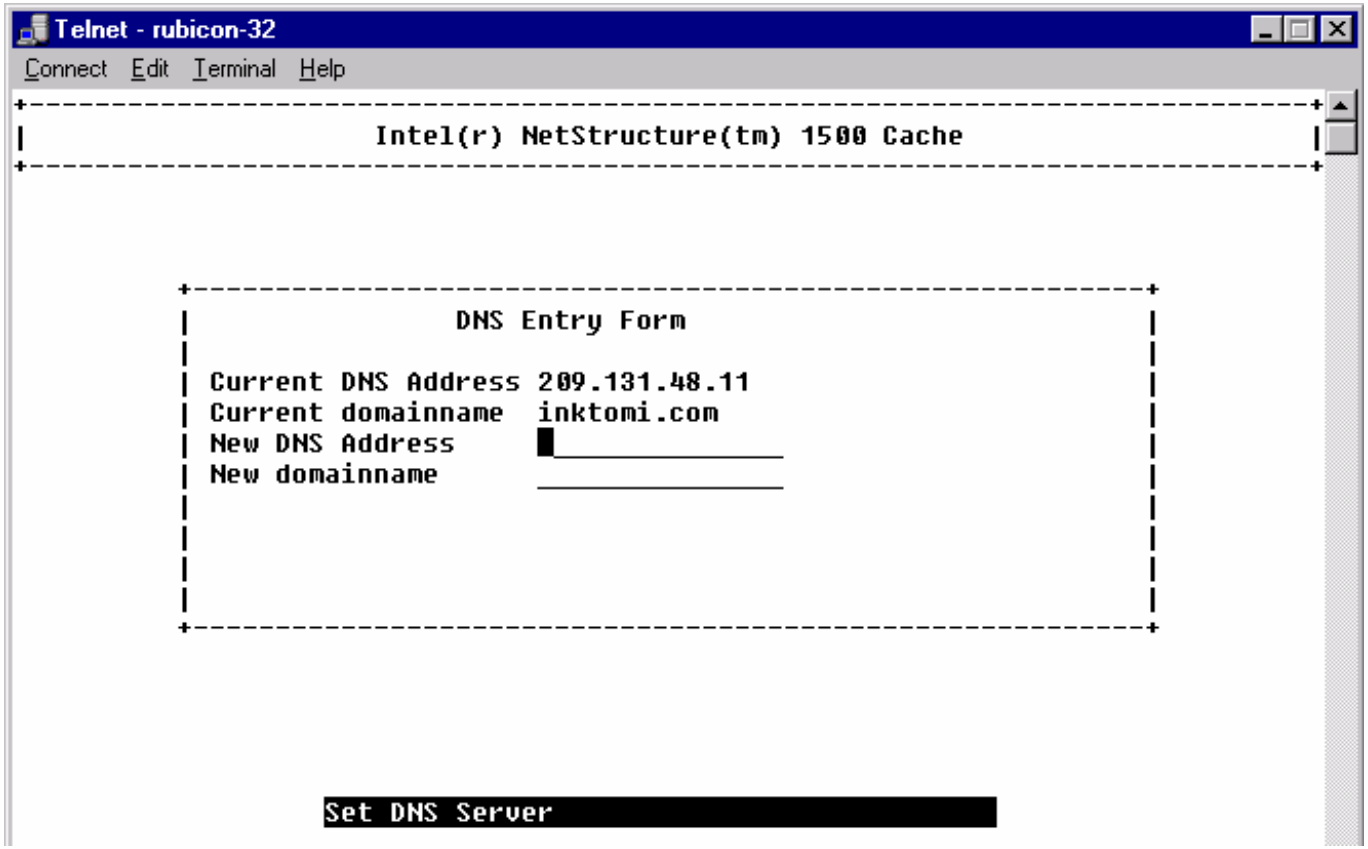
CLI: Setup->NIC

- **Set the speed and mode of the Network Interface Card's connection**



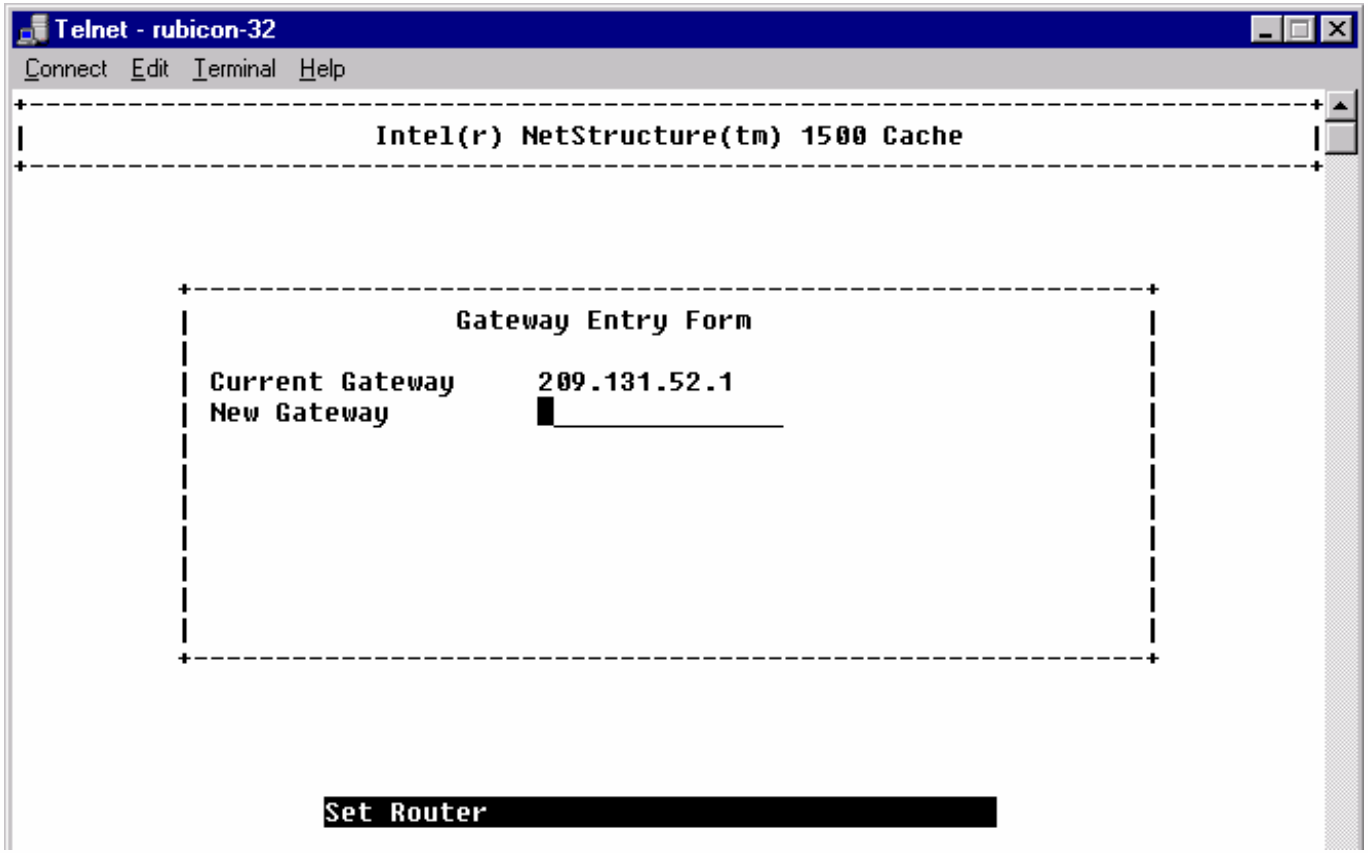
CLI: Setup->DNS

- **Change the DNS name and address**



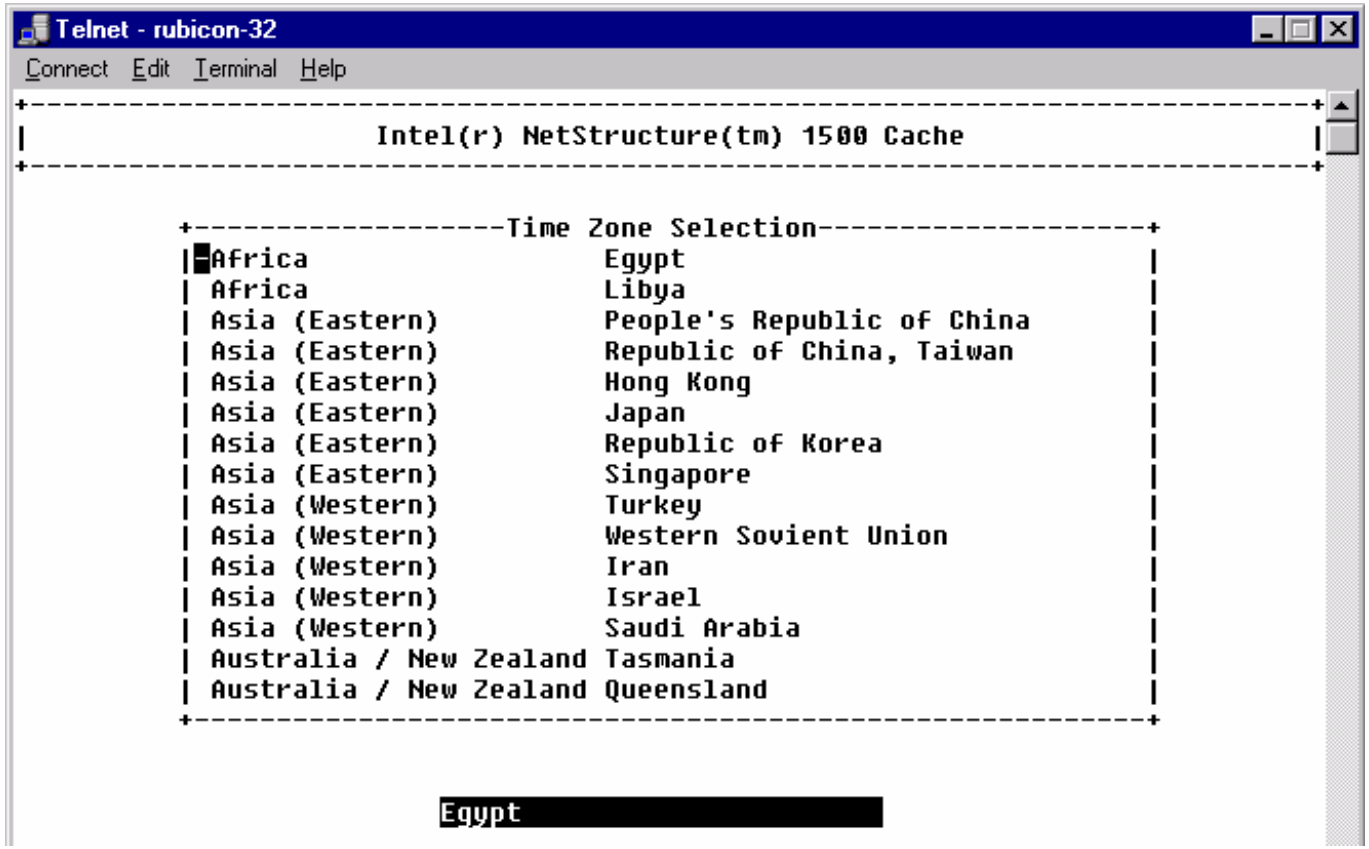
CLI: Setup->Gateway

- **Change the Gateway address used by the Intel Cache**



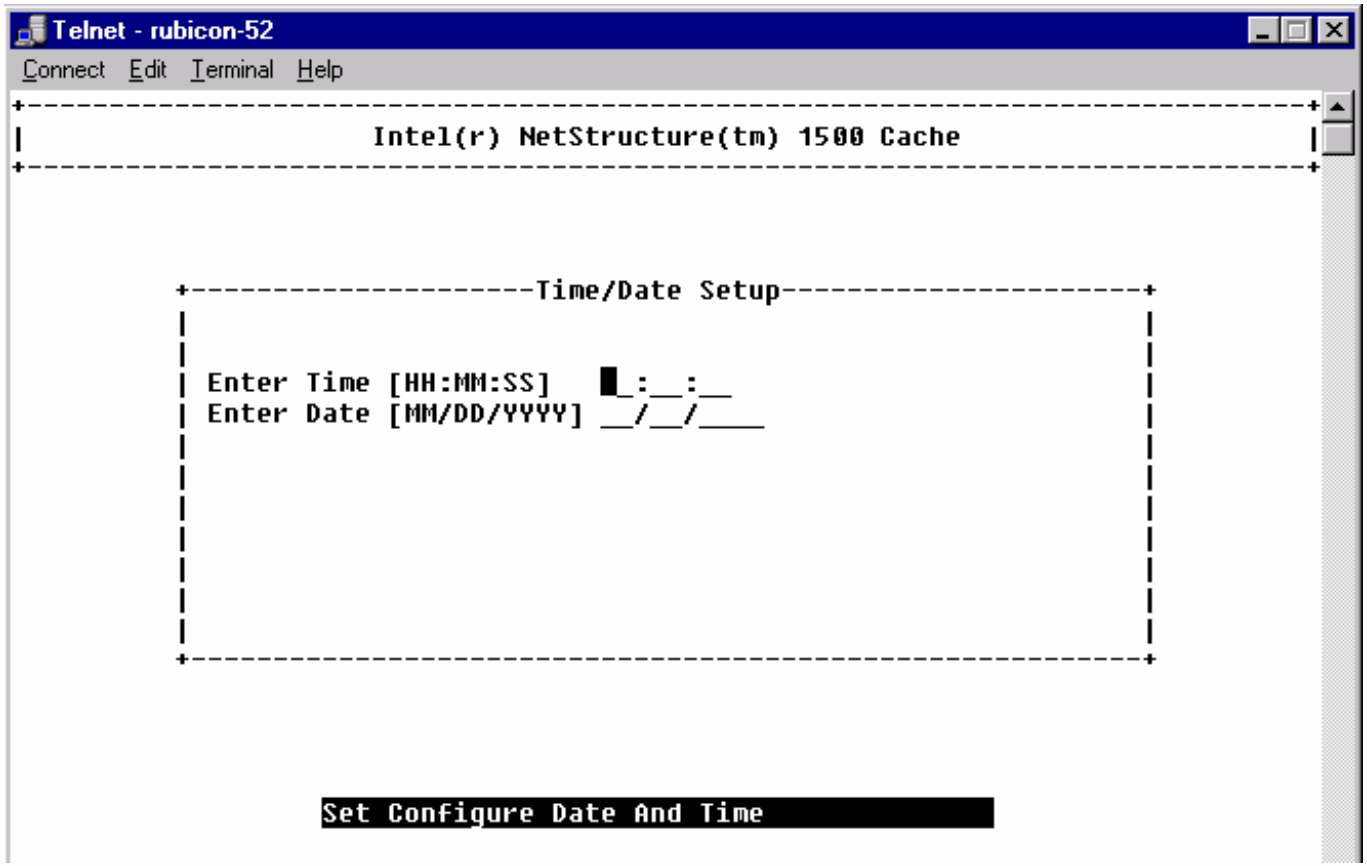
CLI: Setup->Timezone

- **Select the time zone the Intel Cache is in**



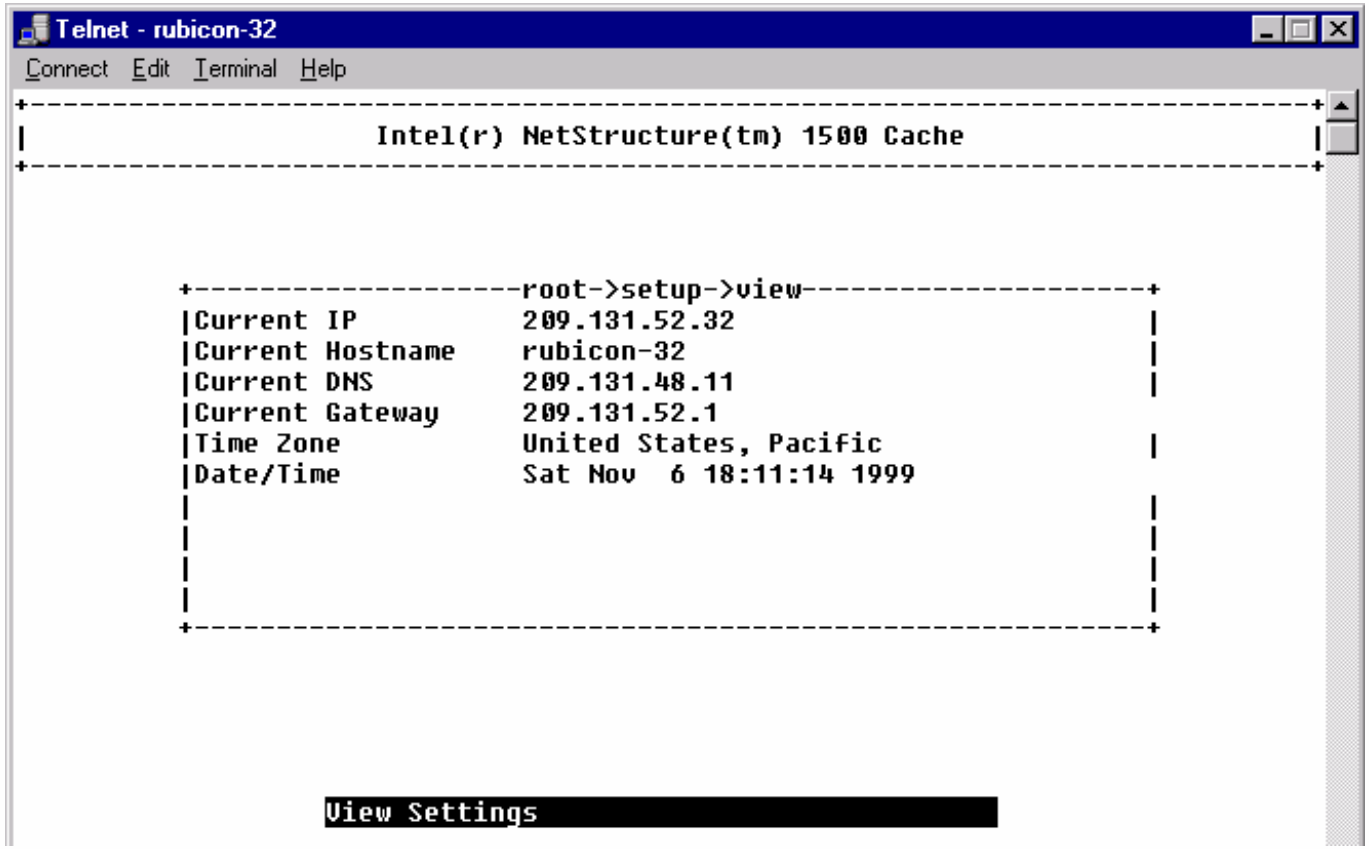
CLI: Setup->Time/Date

- **Enter the current time and date**



CLI: Setup->View Settings

- **View the current Setup settings**



Intel Cache:
Manager UI Server / CLI Main Menu Configuration

Inktomi

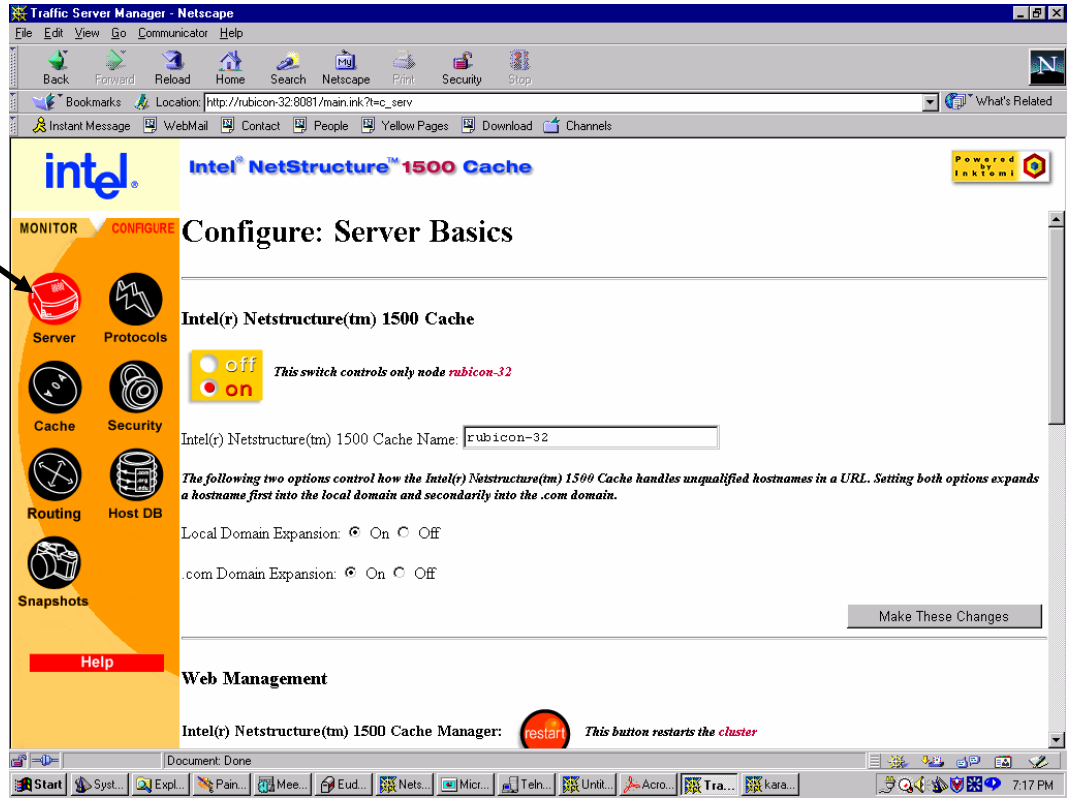
Chapter Overview

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

Manager UI: Server Basics

Inktomi

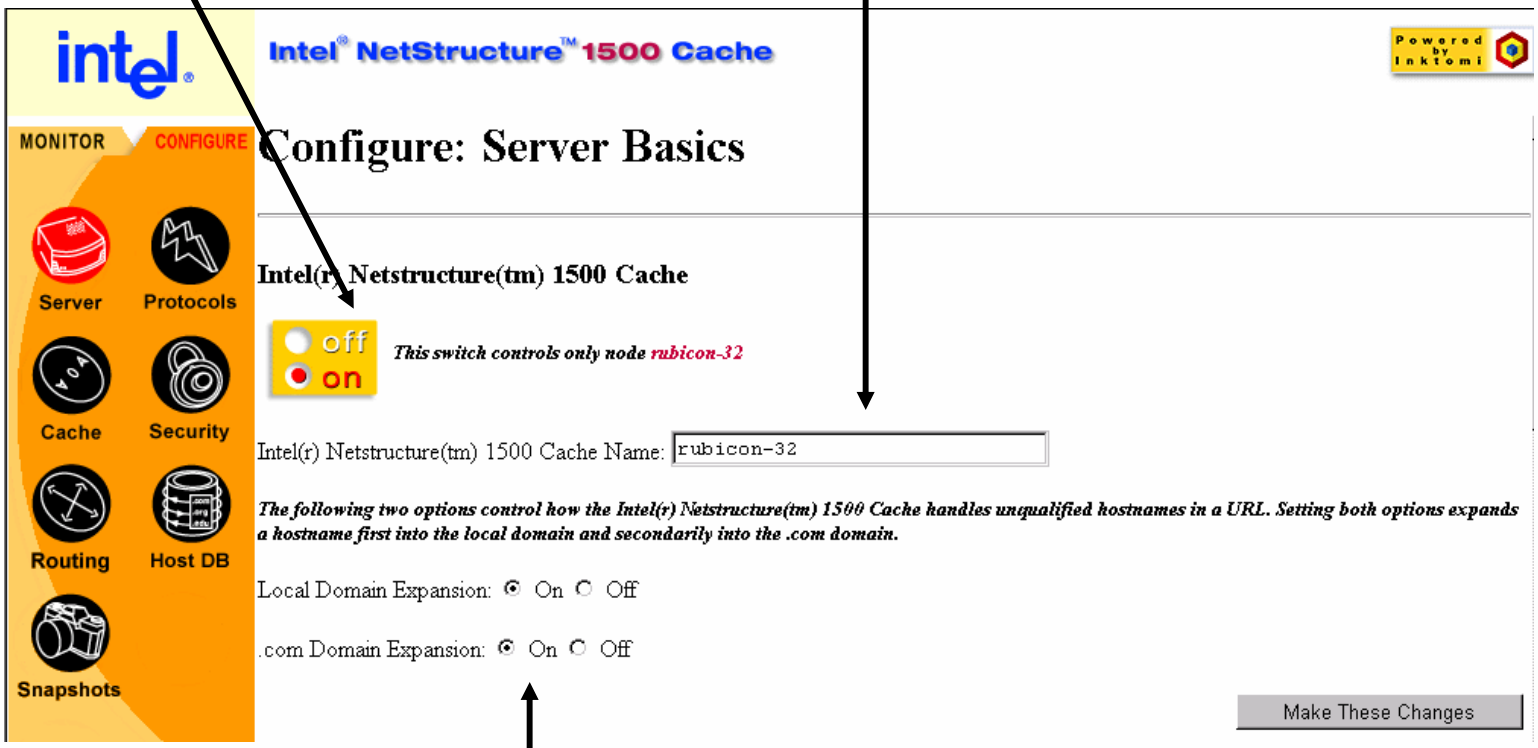
Turn Intel Cache On and Off, enable domain name expansion, enable load shedding, restart or reconfigure Traffic Manager, configure VIPs, auto-configure browser, and enable SNMP



Manager UI: Server Basics Details

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

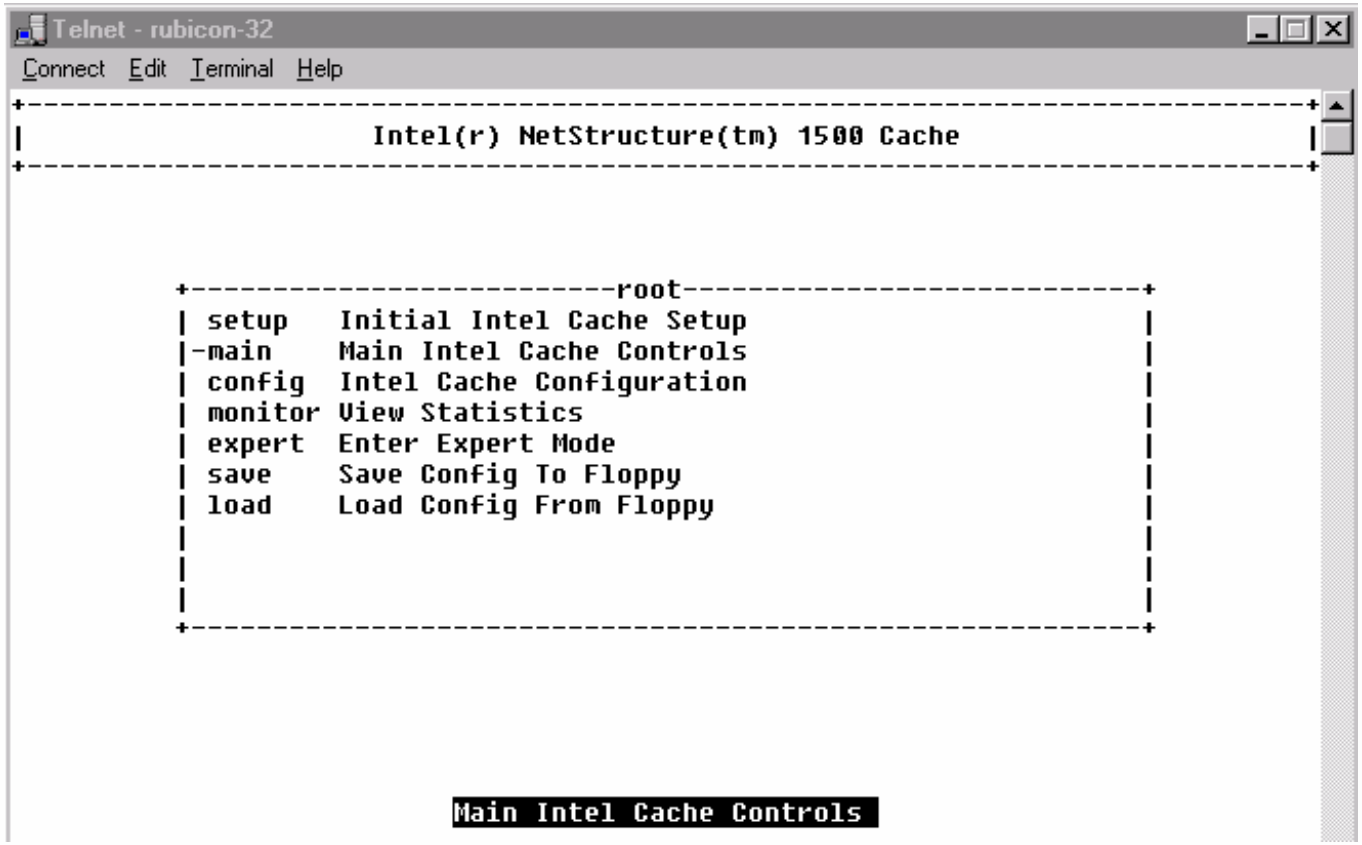
Cache name is the proxy name



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

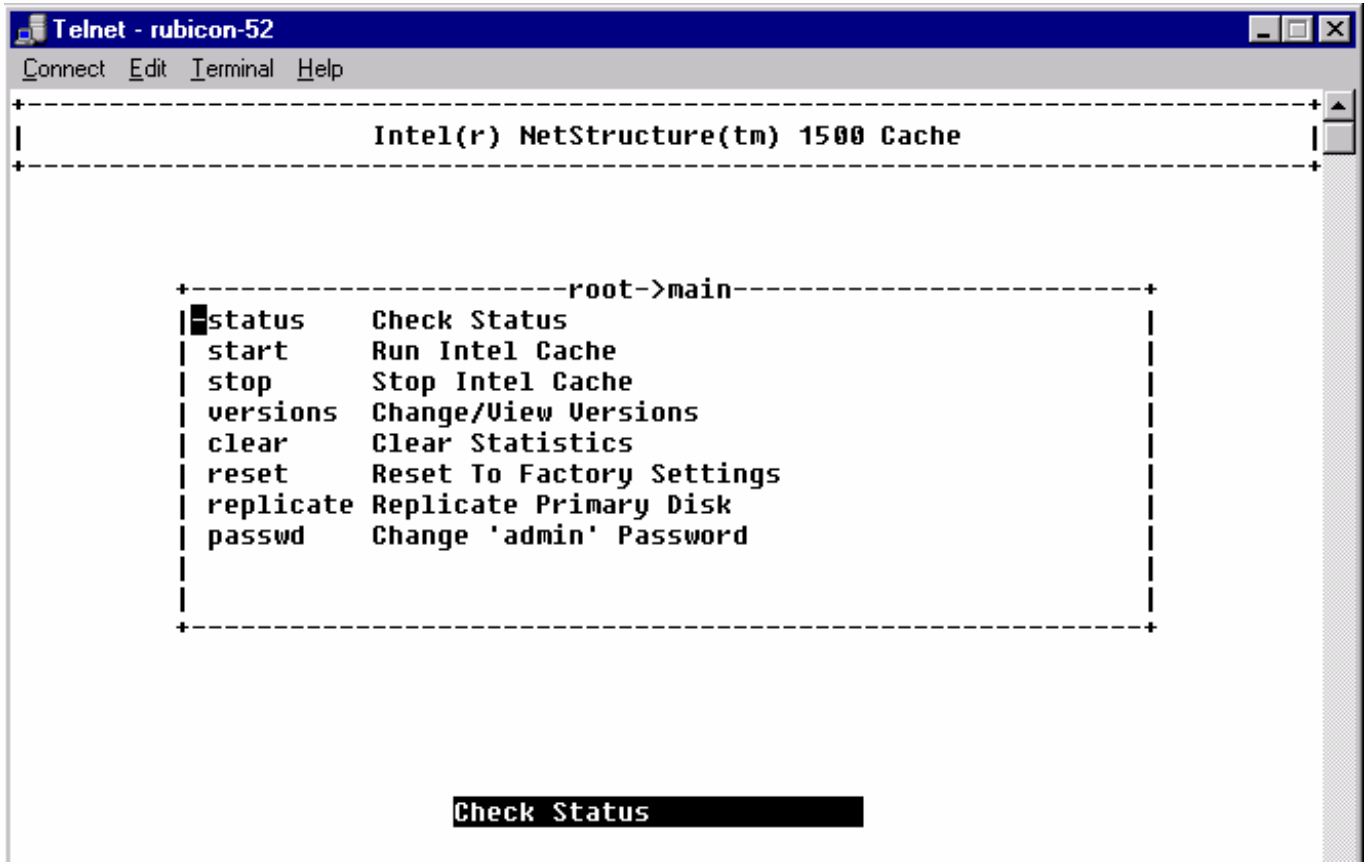
Command Line Interface: Main

- **Closest CLI equivalent to Server button in Manger UI**



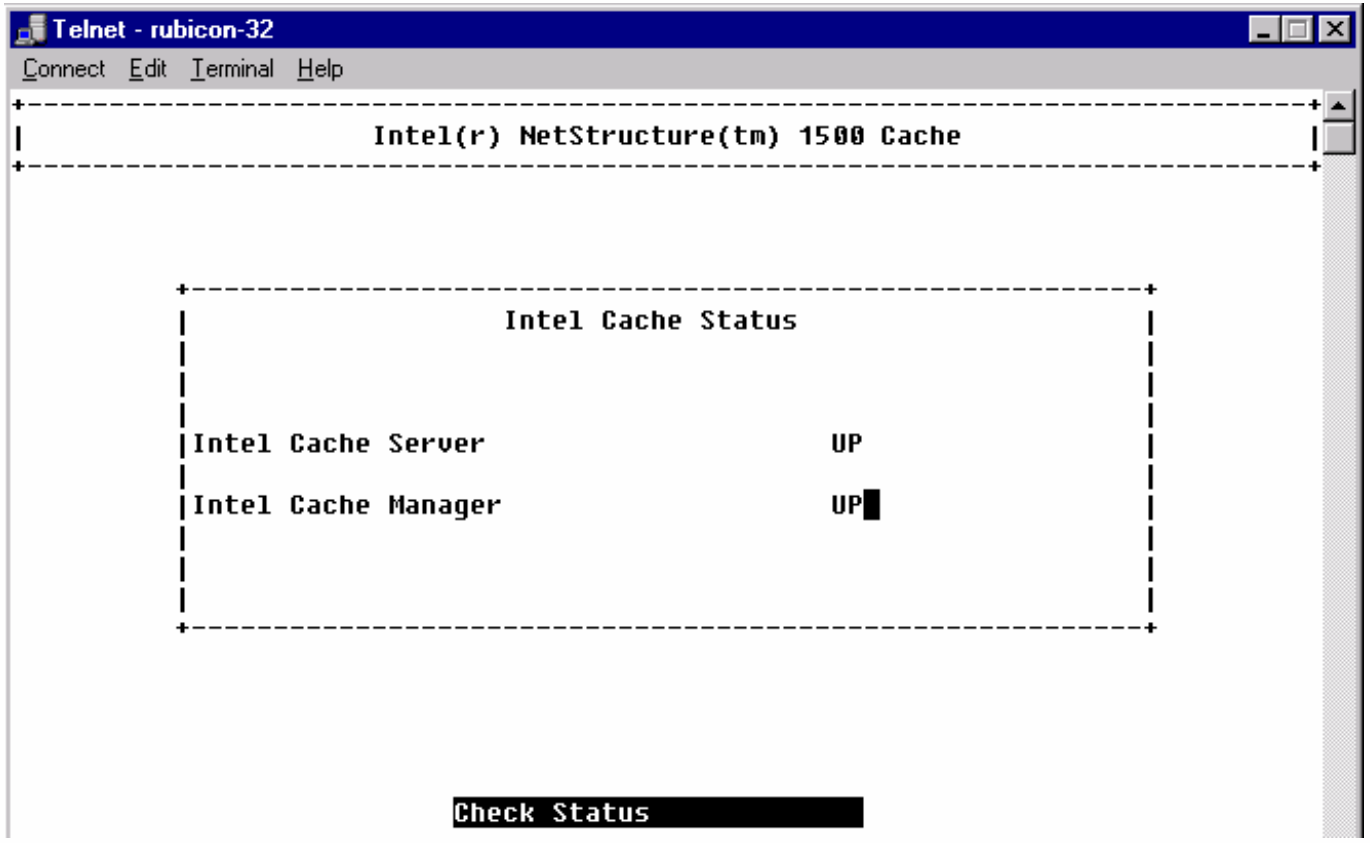
CLI: Main Menu

Main menu contains commands to start and stop the Intel Cache and do large scale configuration



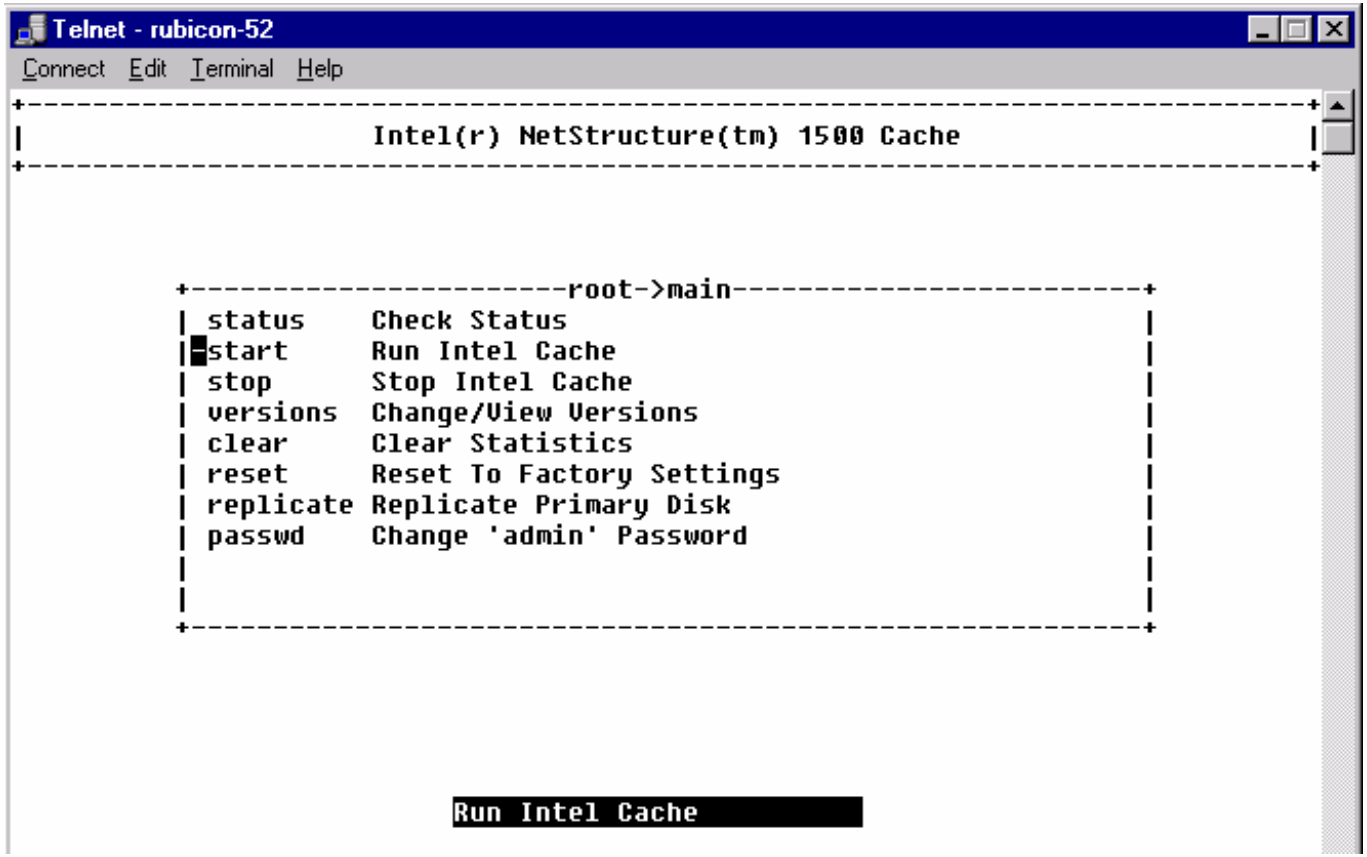
CLI: Main->Status

- **Check current Intel Cache status**



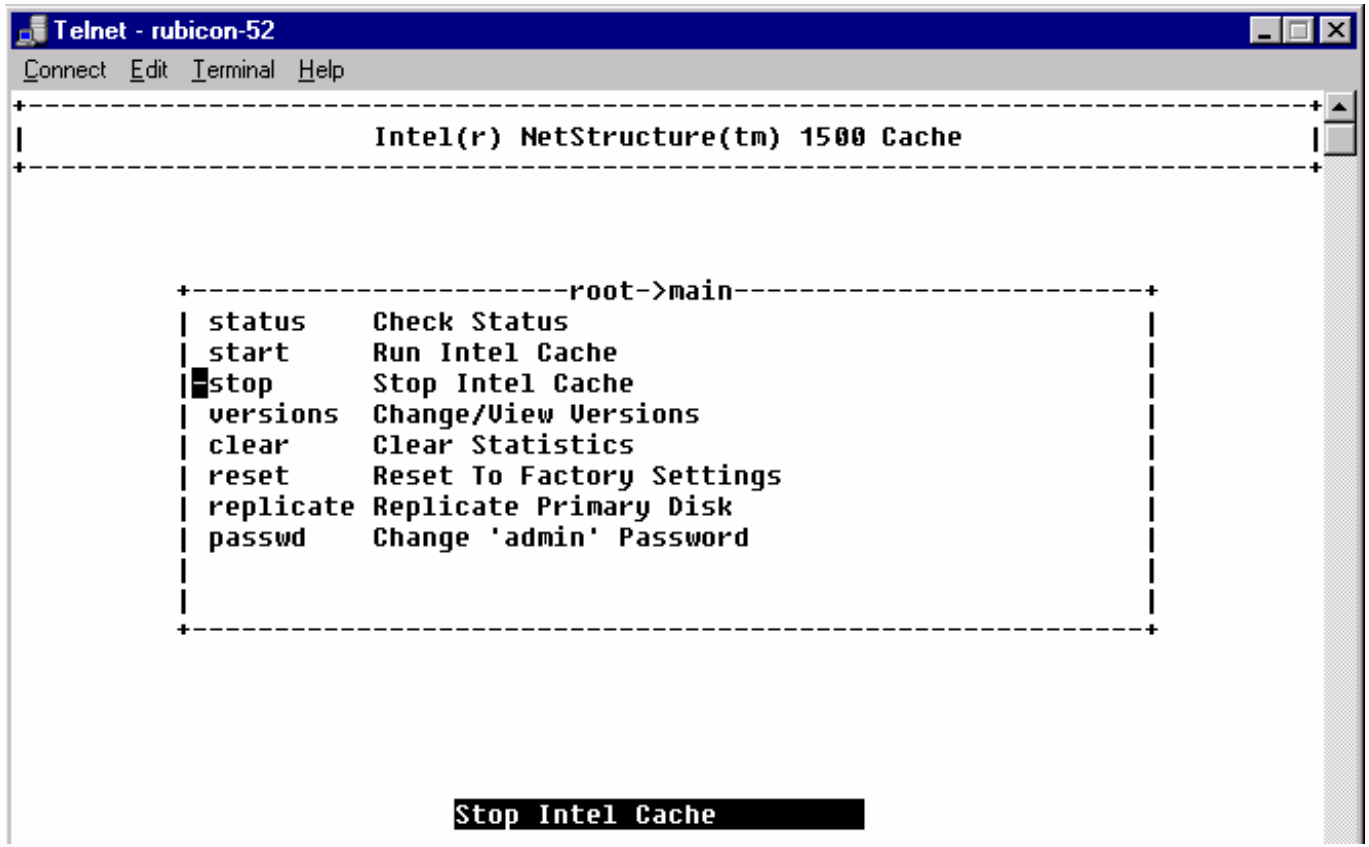
CLI: Main->Start

- **Start the Intel Cache**



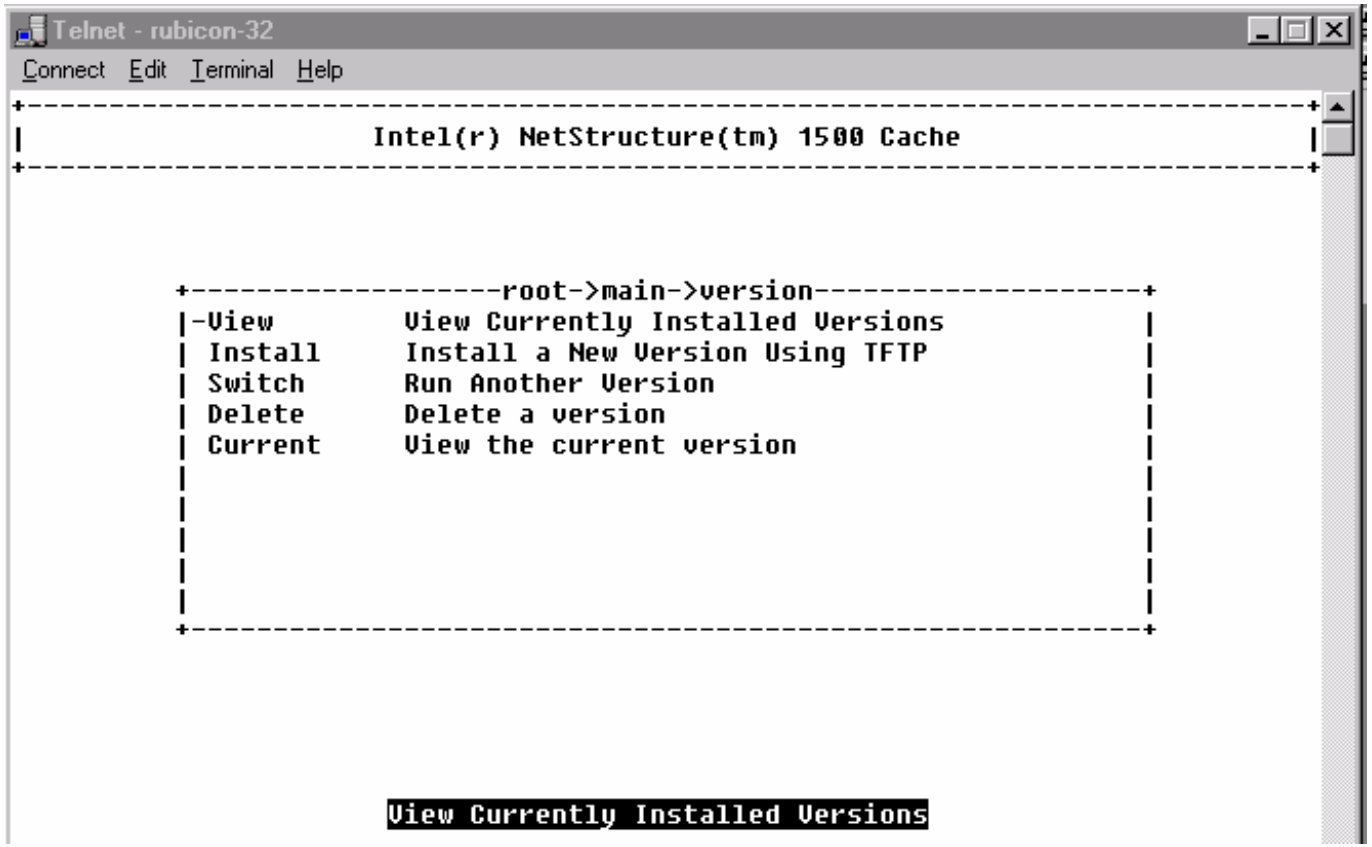
CLI: Main->Stop

- **Stop the Intel Cache**



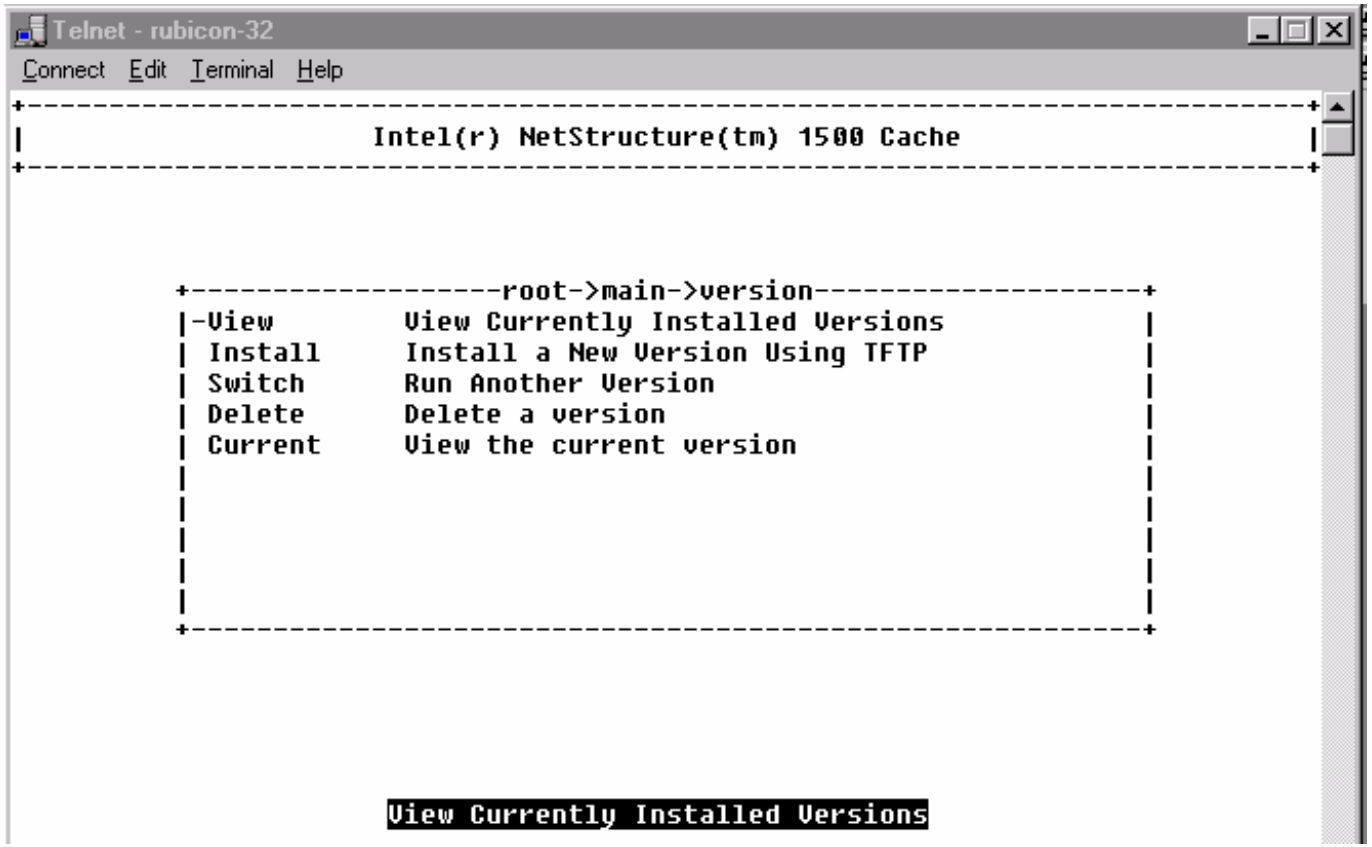
CLI: Main->Versions

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



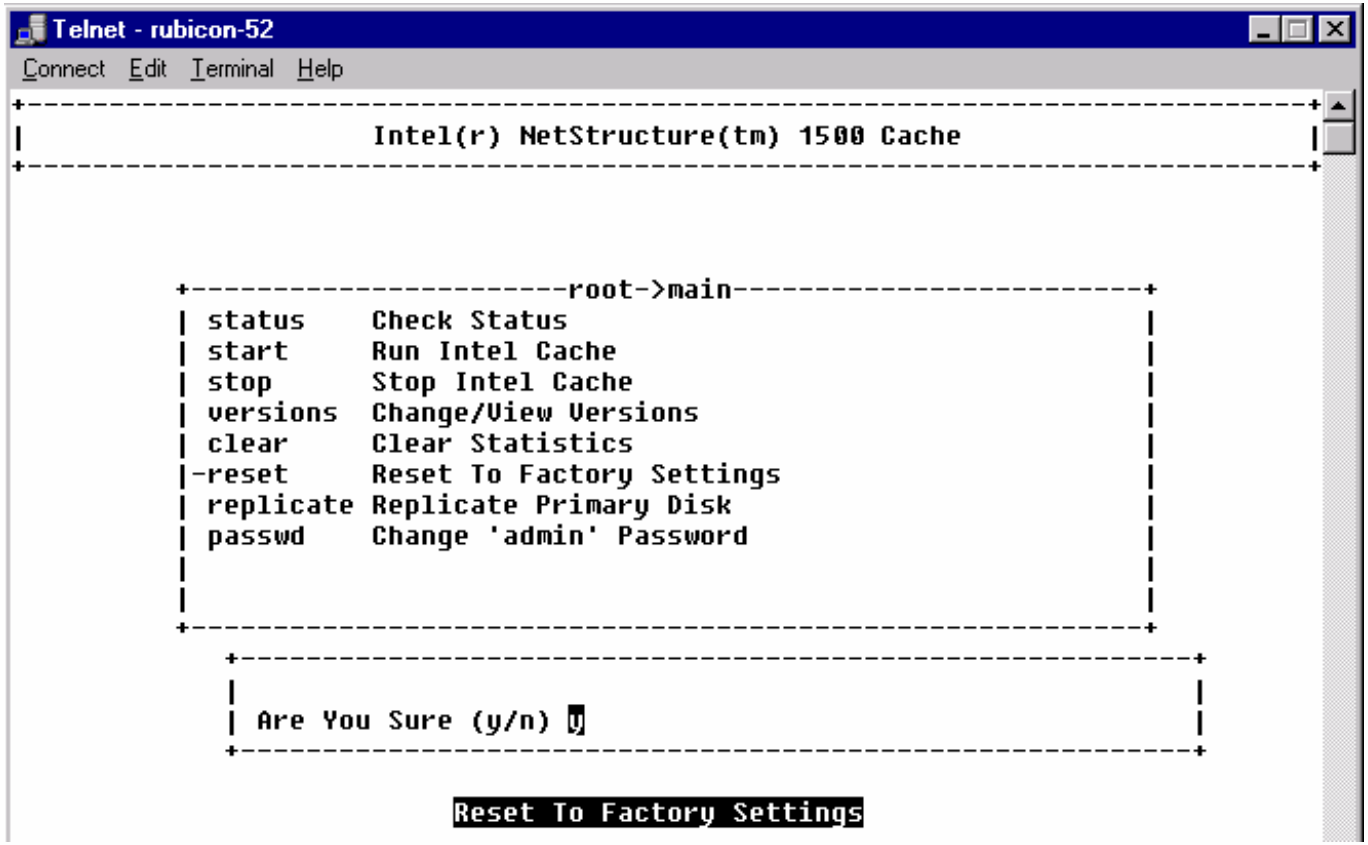
CLI: Main->Versions

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



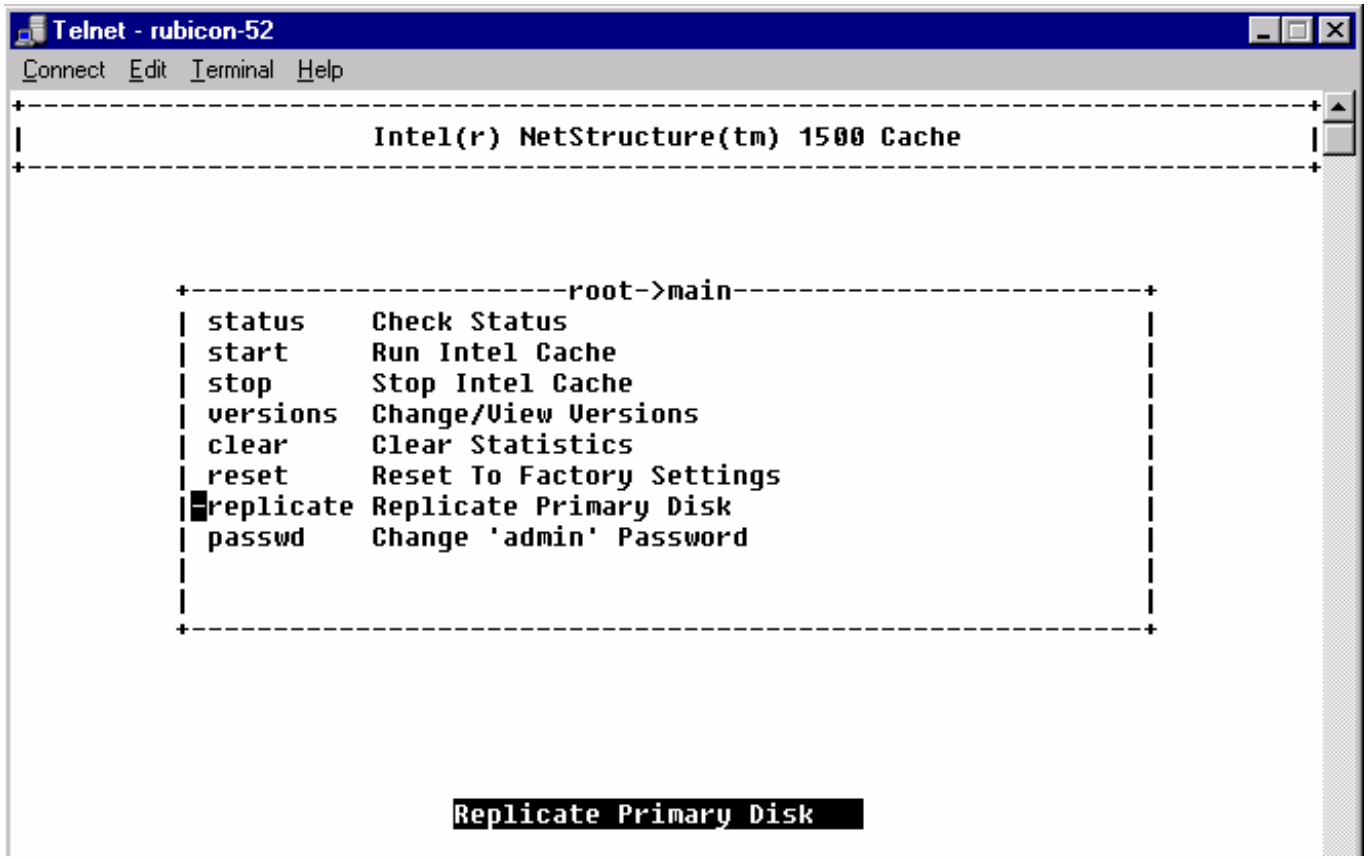
CLI: Main->Reset

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



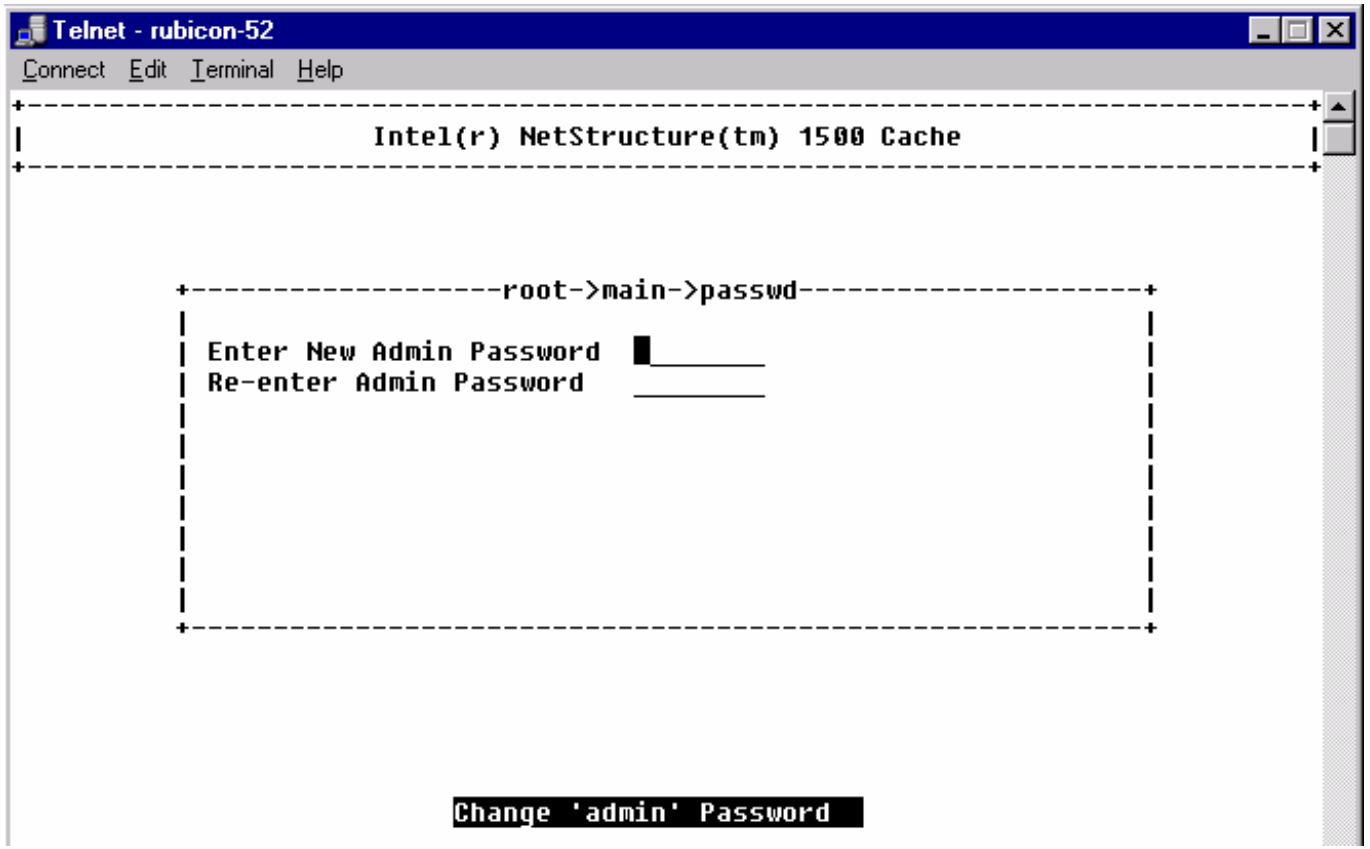
CLI: Main->Replicate

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



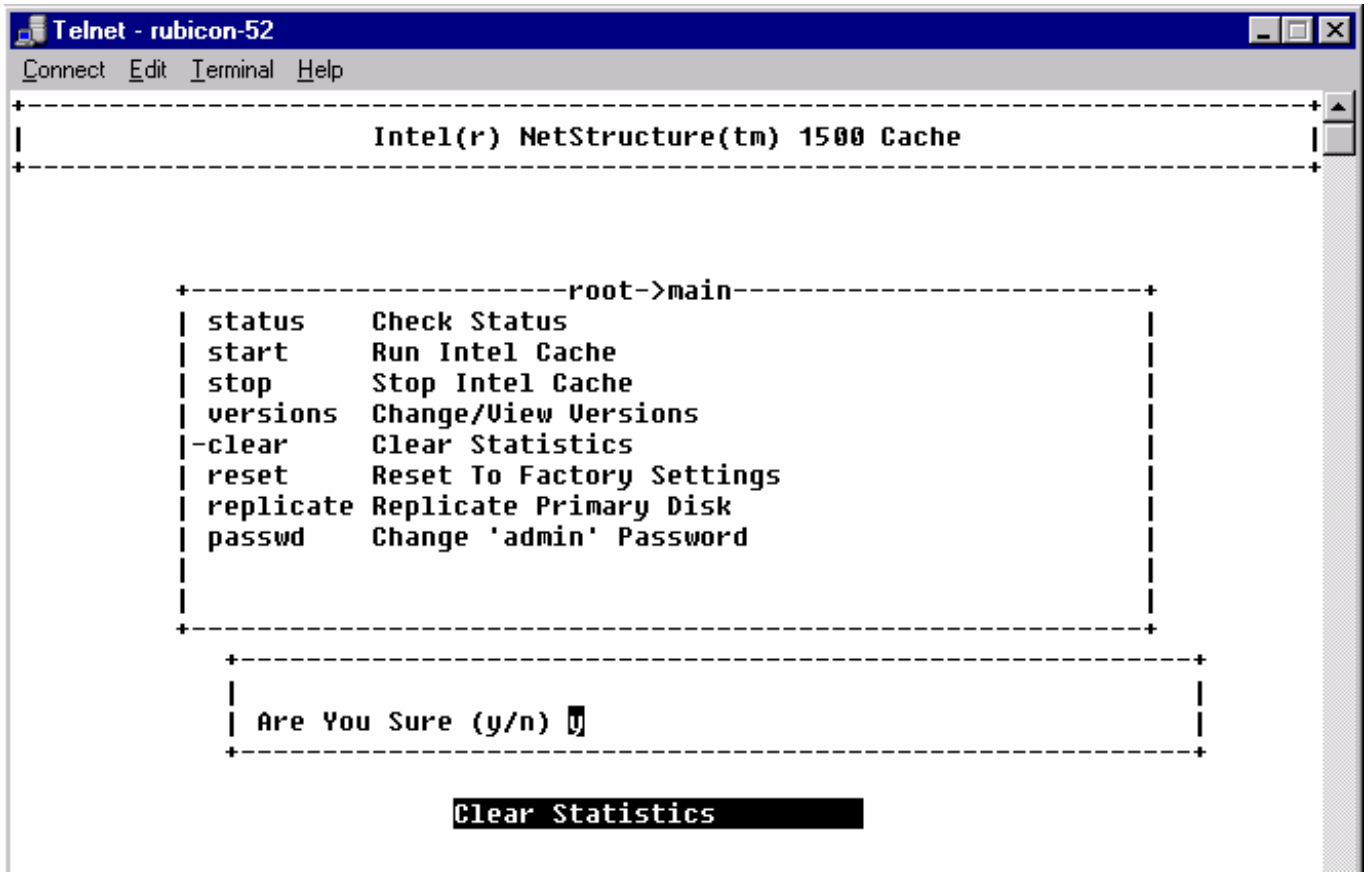
CLI: Main->Passwd

- **Set the admin password for the Manager UI**



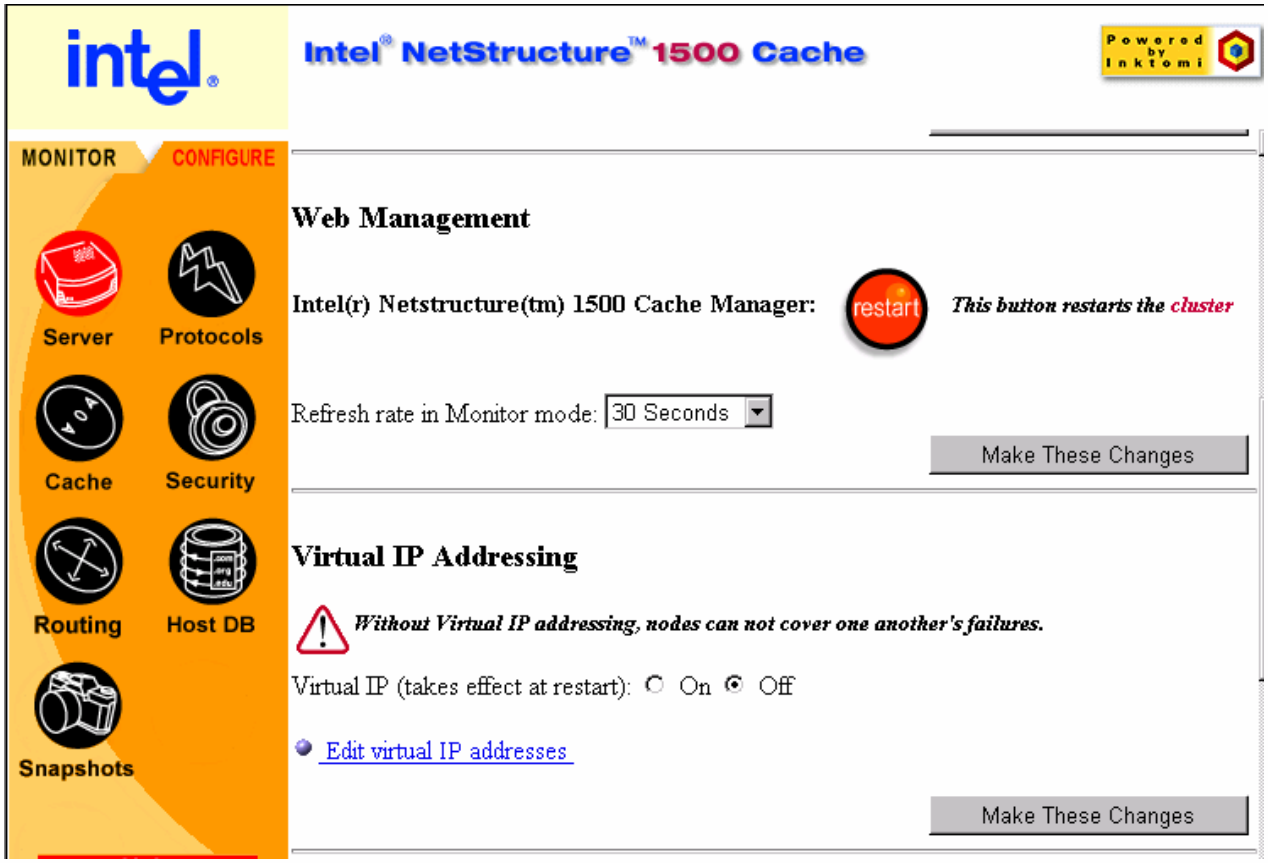
CLI: Main->Clear

- **Clear out stored statistics used in monitoring the Intel Cache**



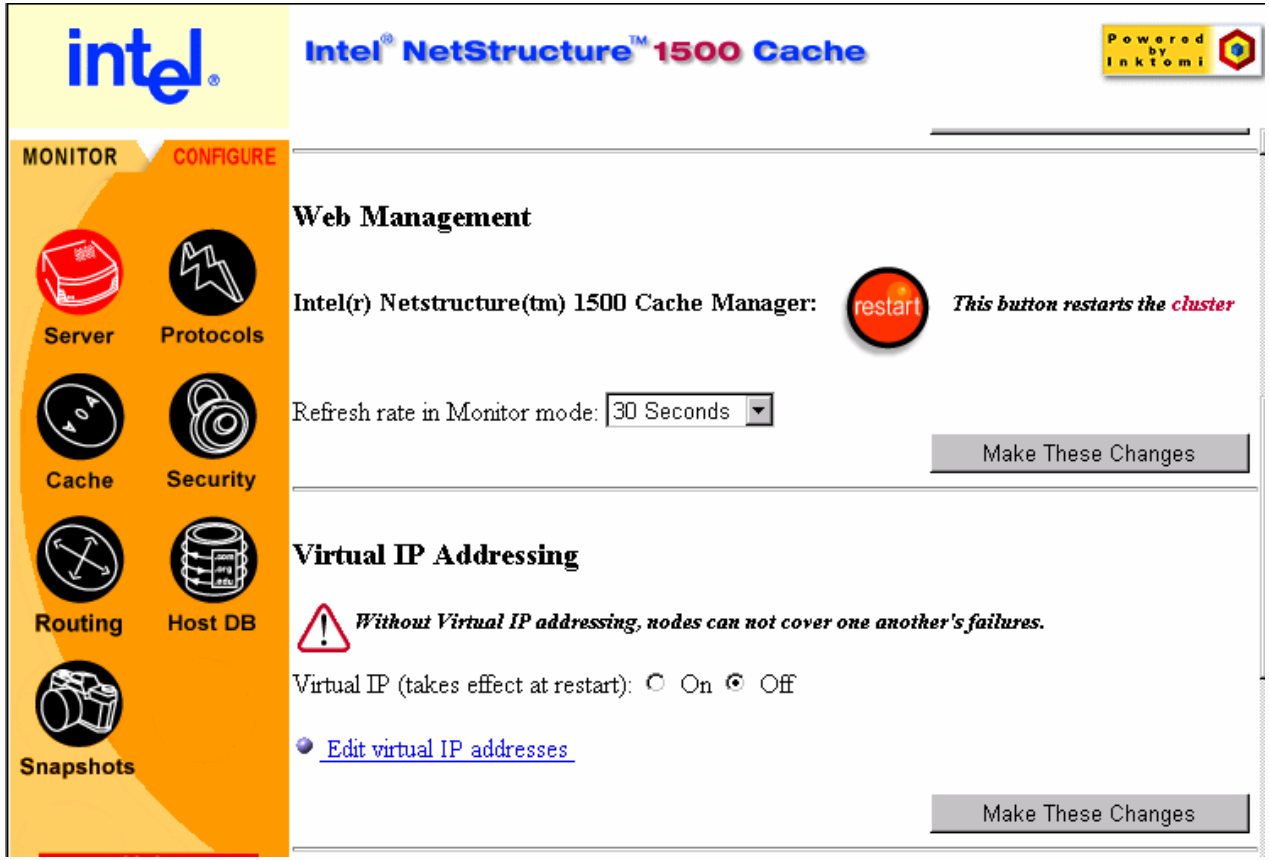
Manager UI: Server Web Management

- 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



Manager UI: Server Virtual IP Addressing

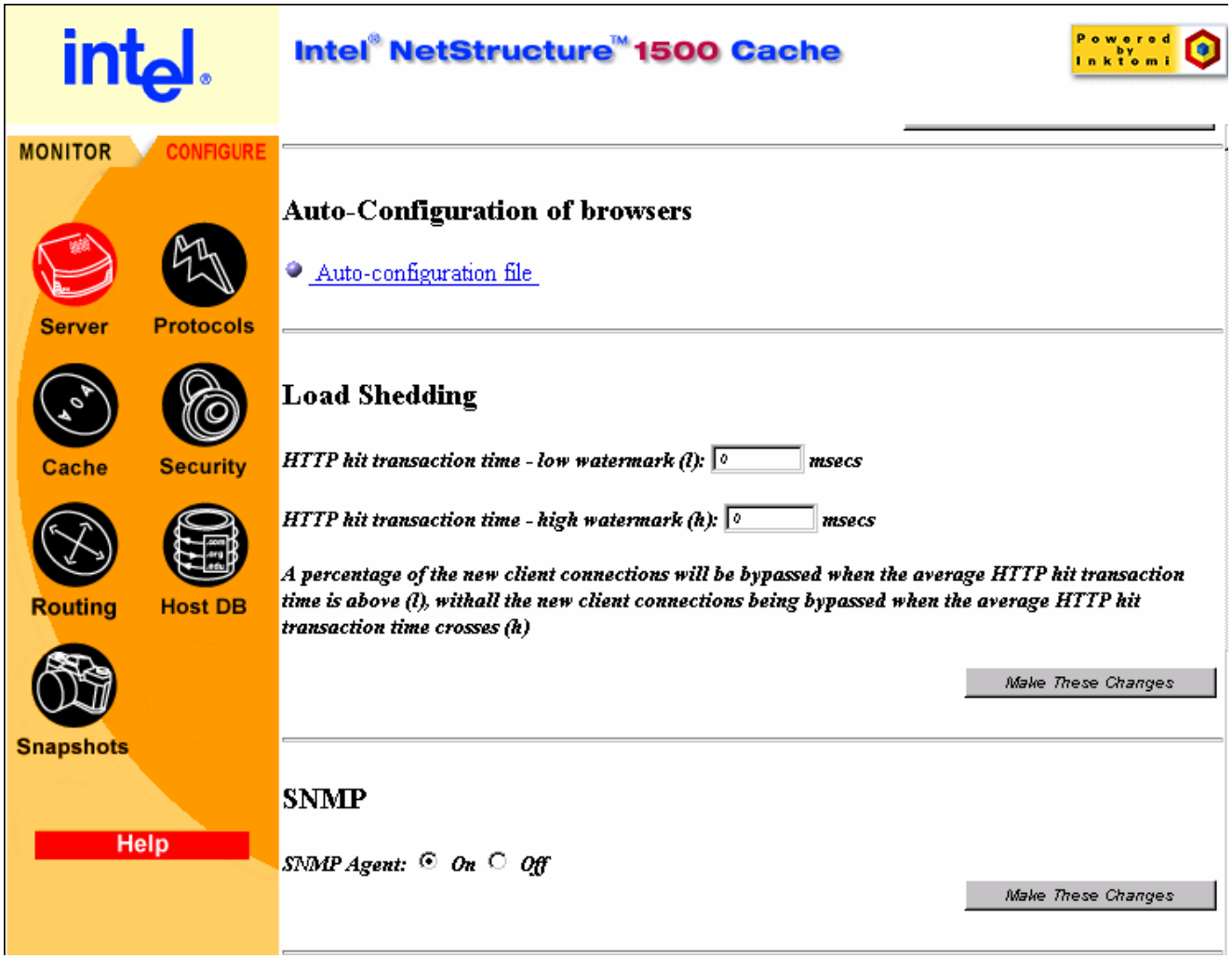
- Virtual IP addresses are additional IP addresses not oriented to any particular machine, but assigned dynamically within the cluster
- You can set up a DNS round robin so client requests will rotate among available nodes
- In the event a node fails, a peer node can take over the failed node's virtual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



Incorrect IP Addresses can effectively disable your system

Manager UI: Server: Browser Auto-Configuration

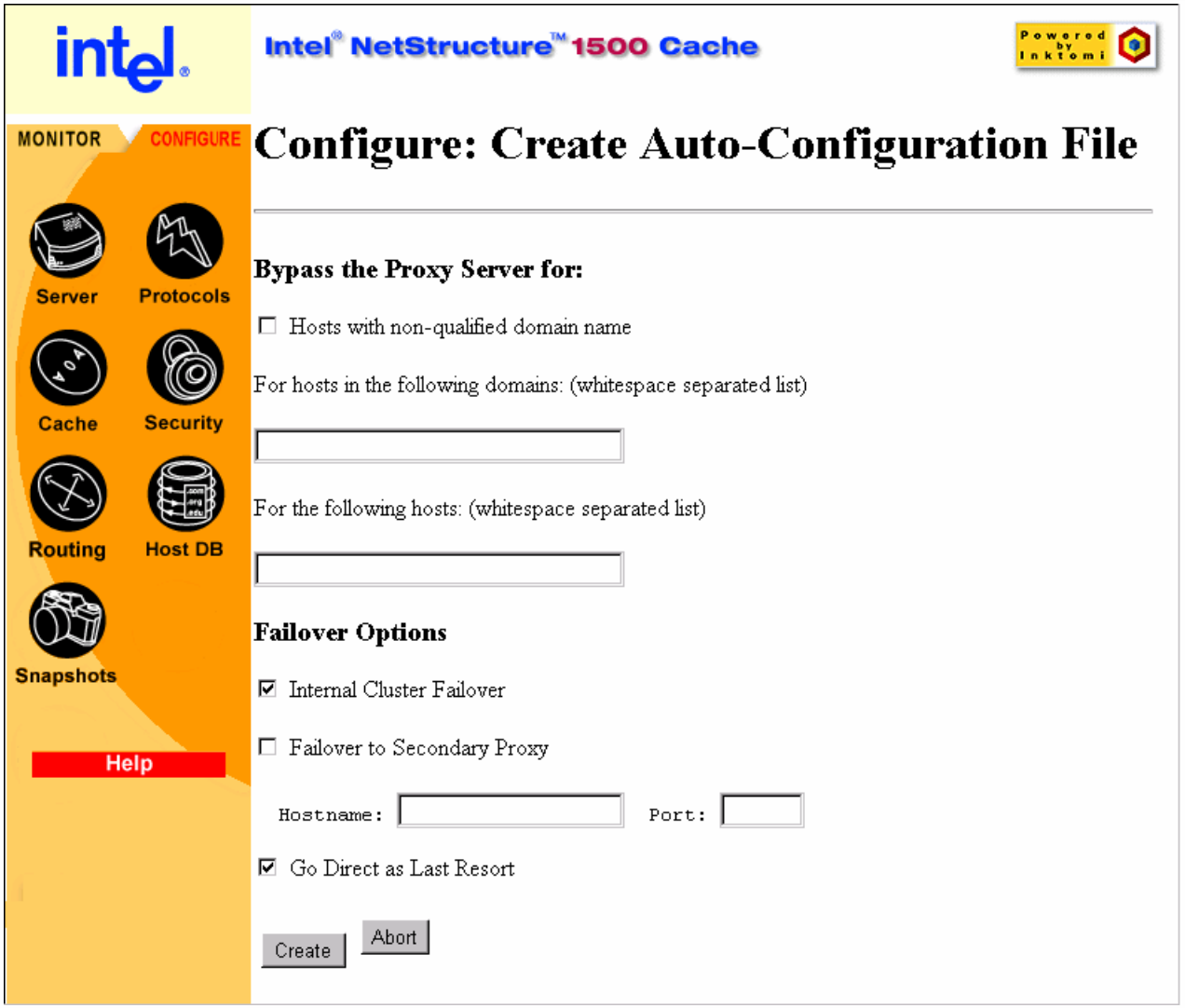
- If you are not using transparency options, you may specify a preference to use the proxy server through an auto-configuration file
  - If none exists, it will be created
  - If the server detects an auto-configuration file you will have options to view, replace or delete the existing file



Setting Browser Auto-Configuration

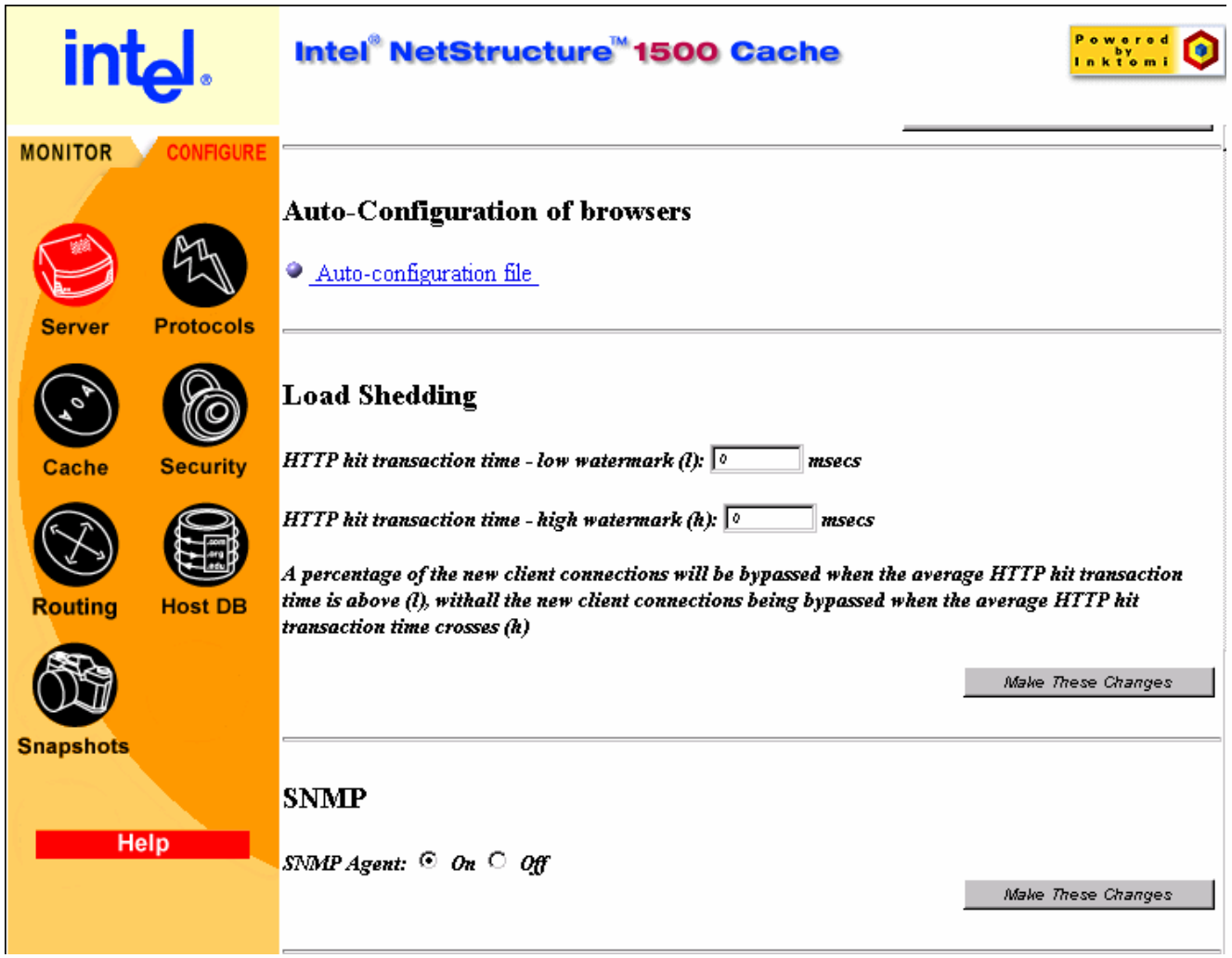
- **Users will need to set their browsers to connect to your Intel Cache auto-configuration file as appropriate for each browser**

Setting Browser Auto-Configuration



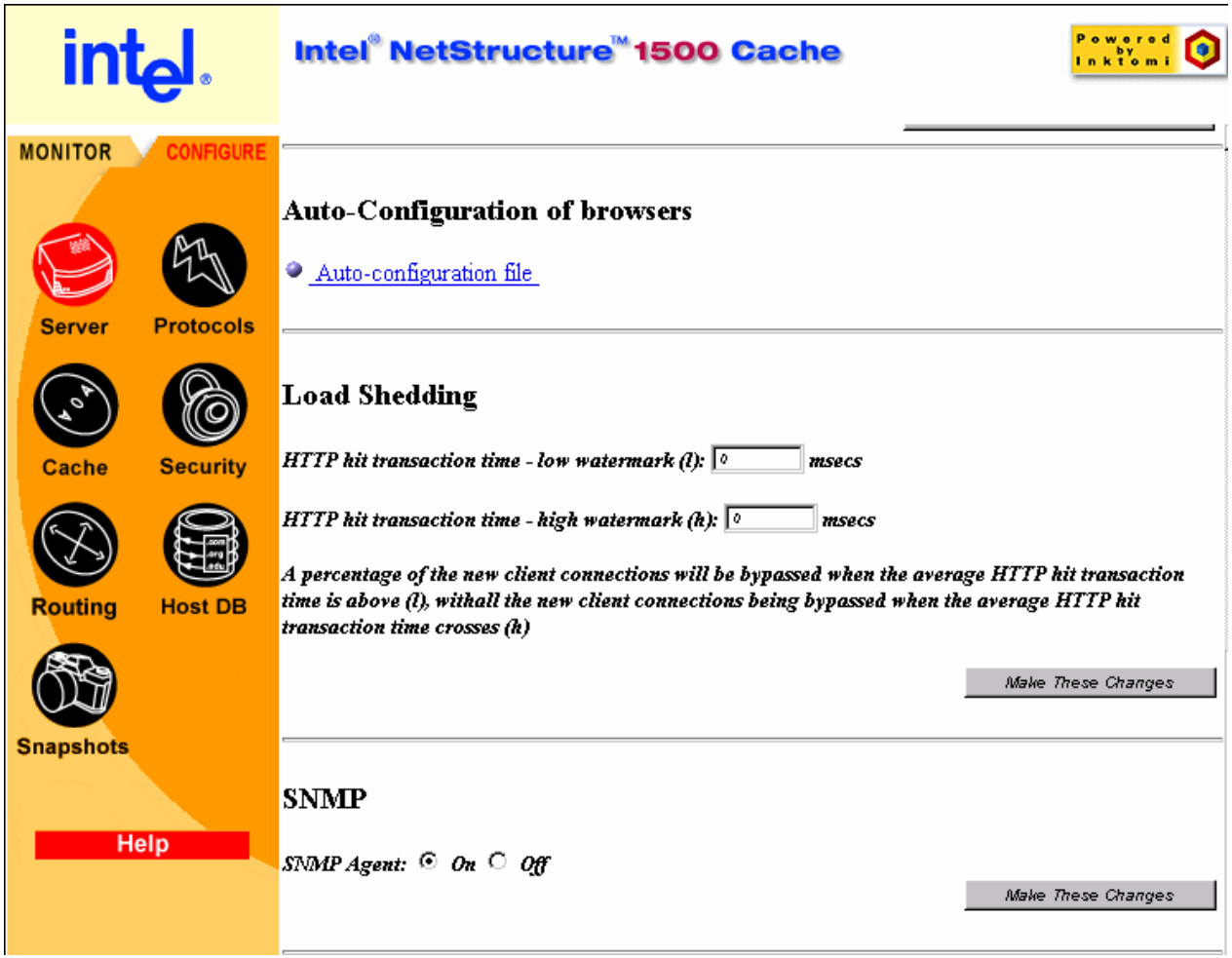
Manager UI: Server: Load Shedding

- 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



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)



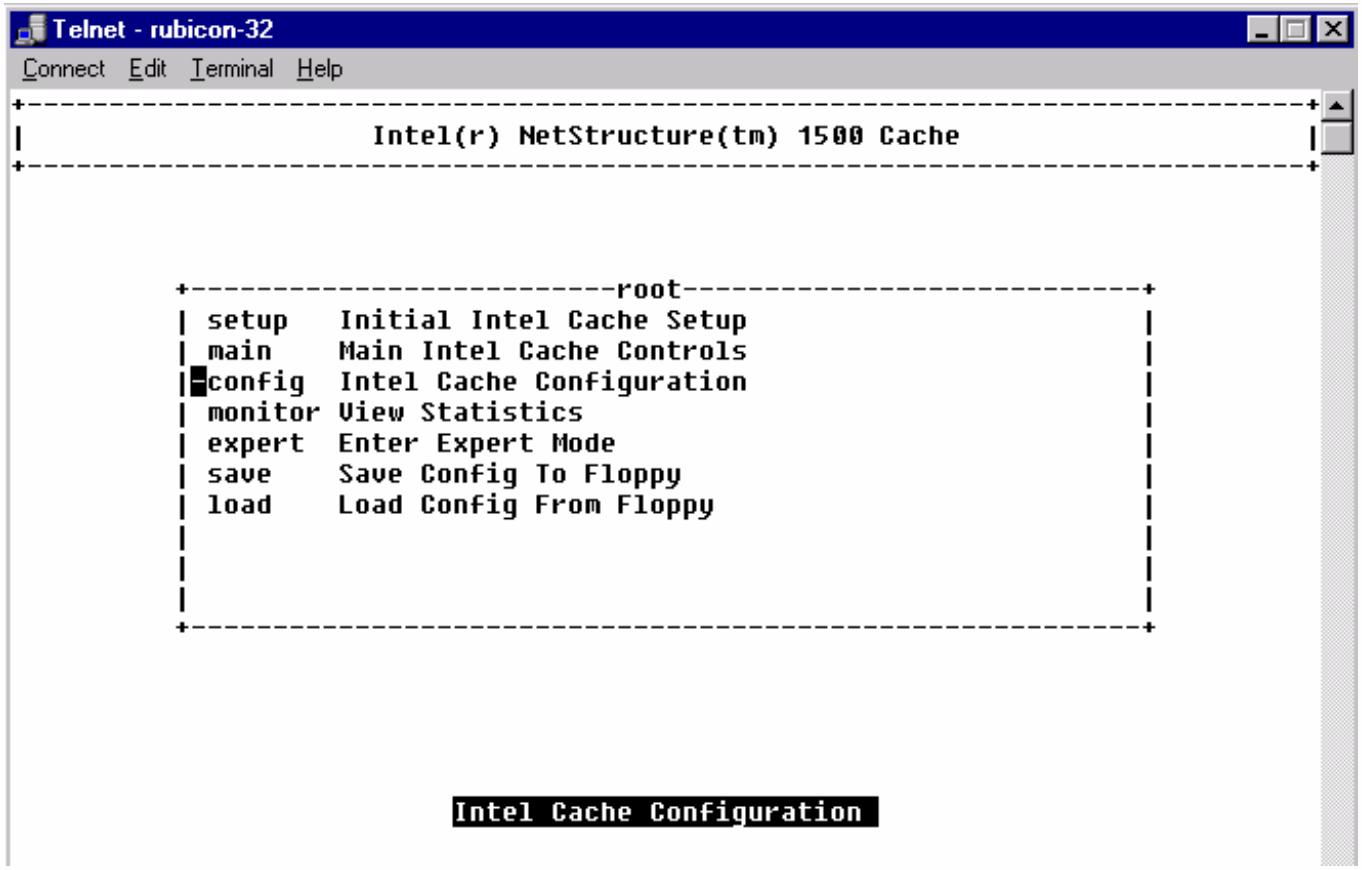
Intel Cache:
Manager UI Protocols/ CLI
Config->Protocols

Chapter Overview

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

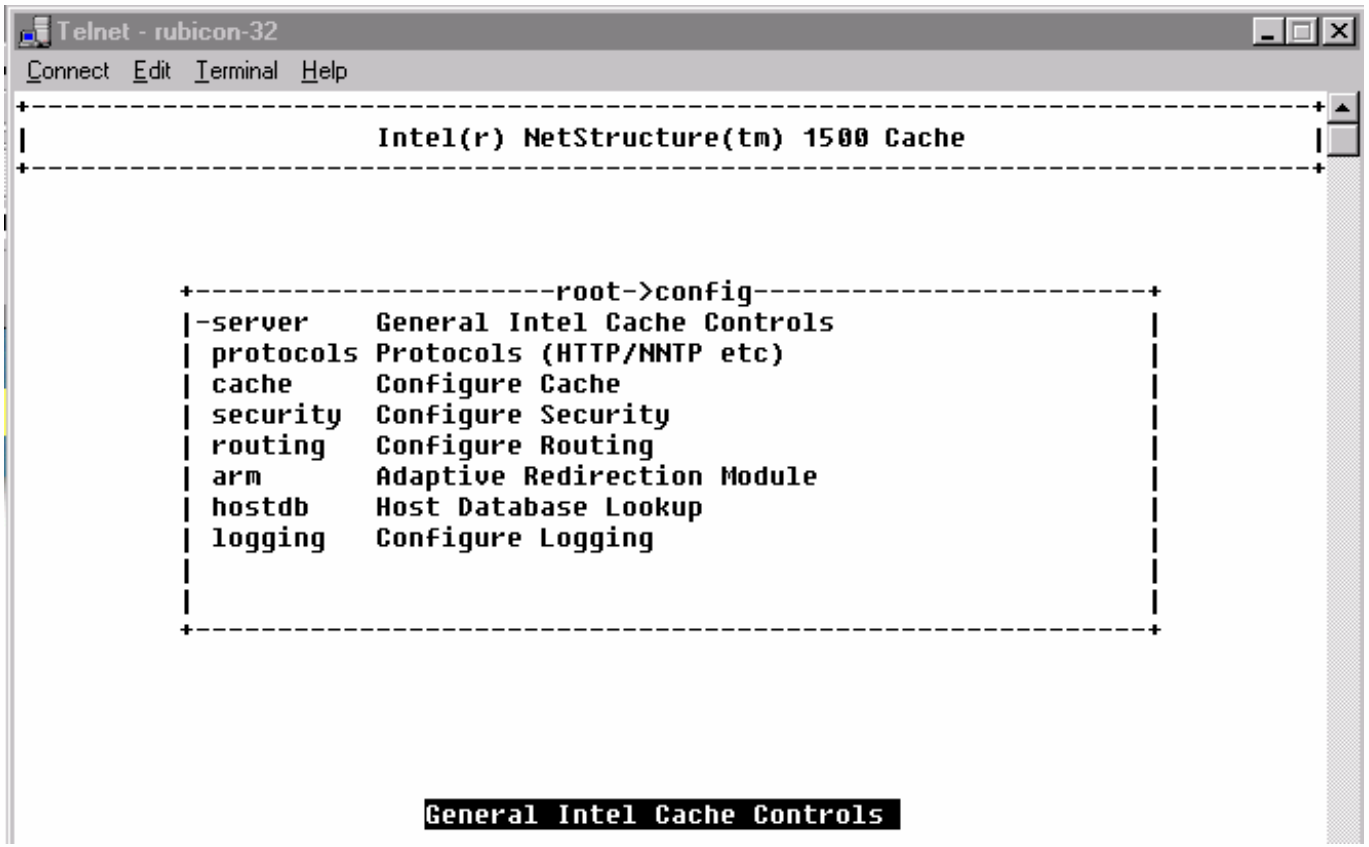
CLI: config

- **Majority of CLI configuration commands located here**



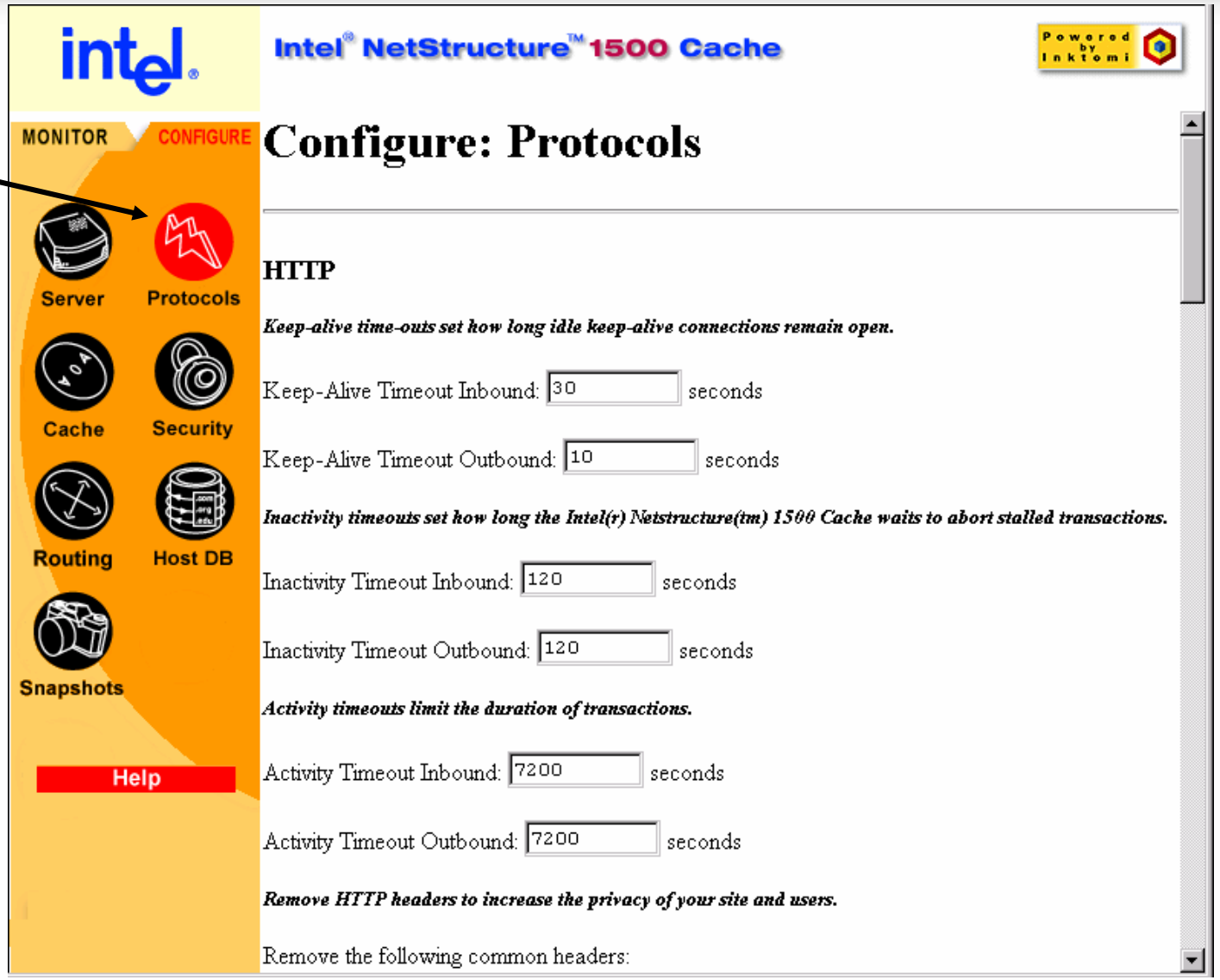
CLI: config menu

- **Similar menu items to buttons in Manager UI Configure menu**



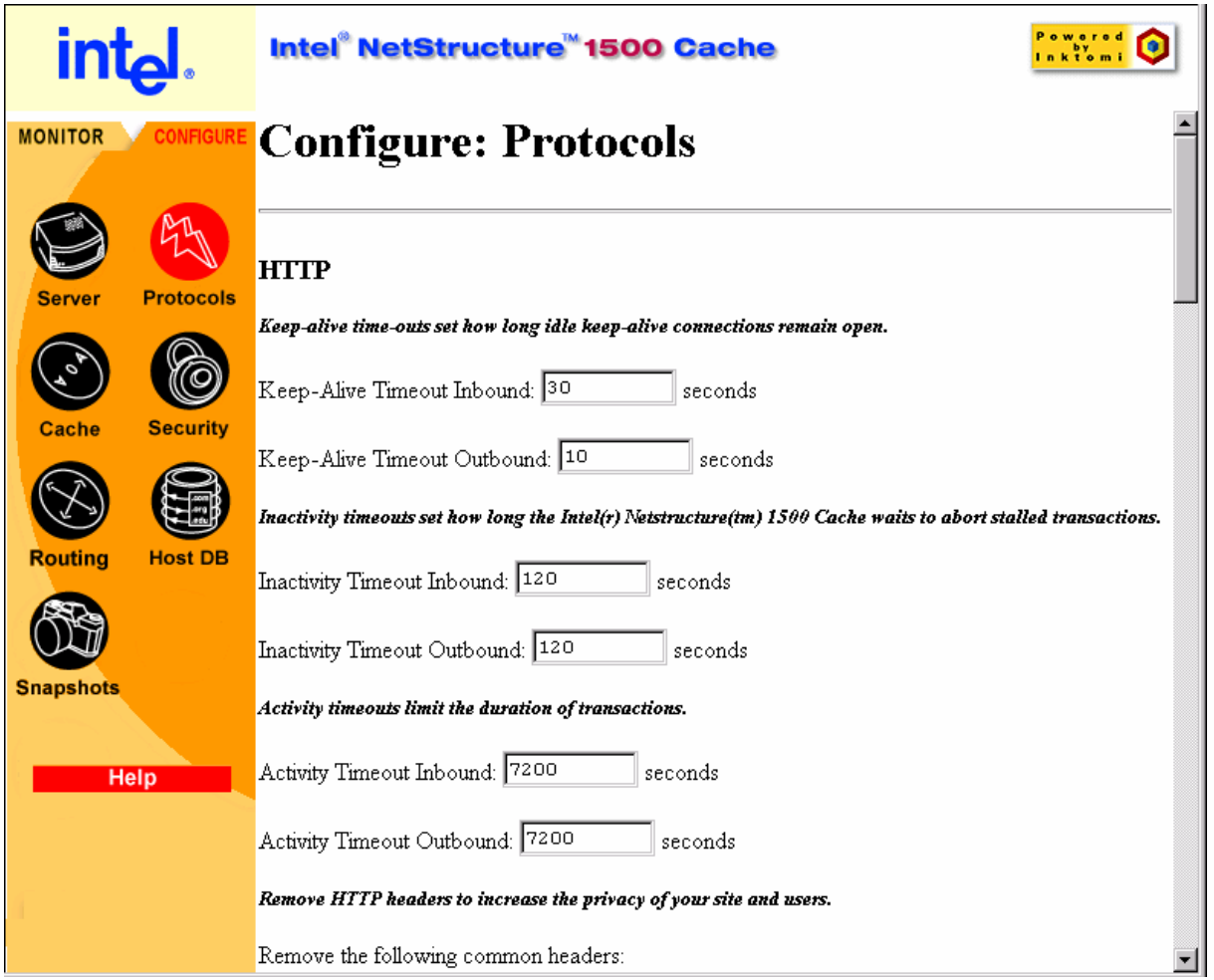
Manager UI: Protocols

Configuring HTTP, NNTP, HTTPS, and FTP



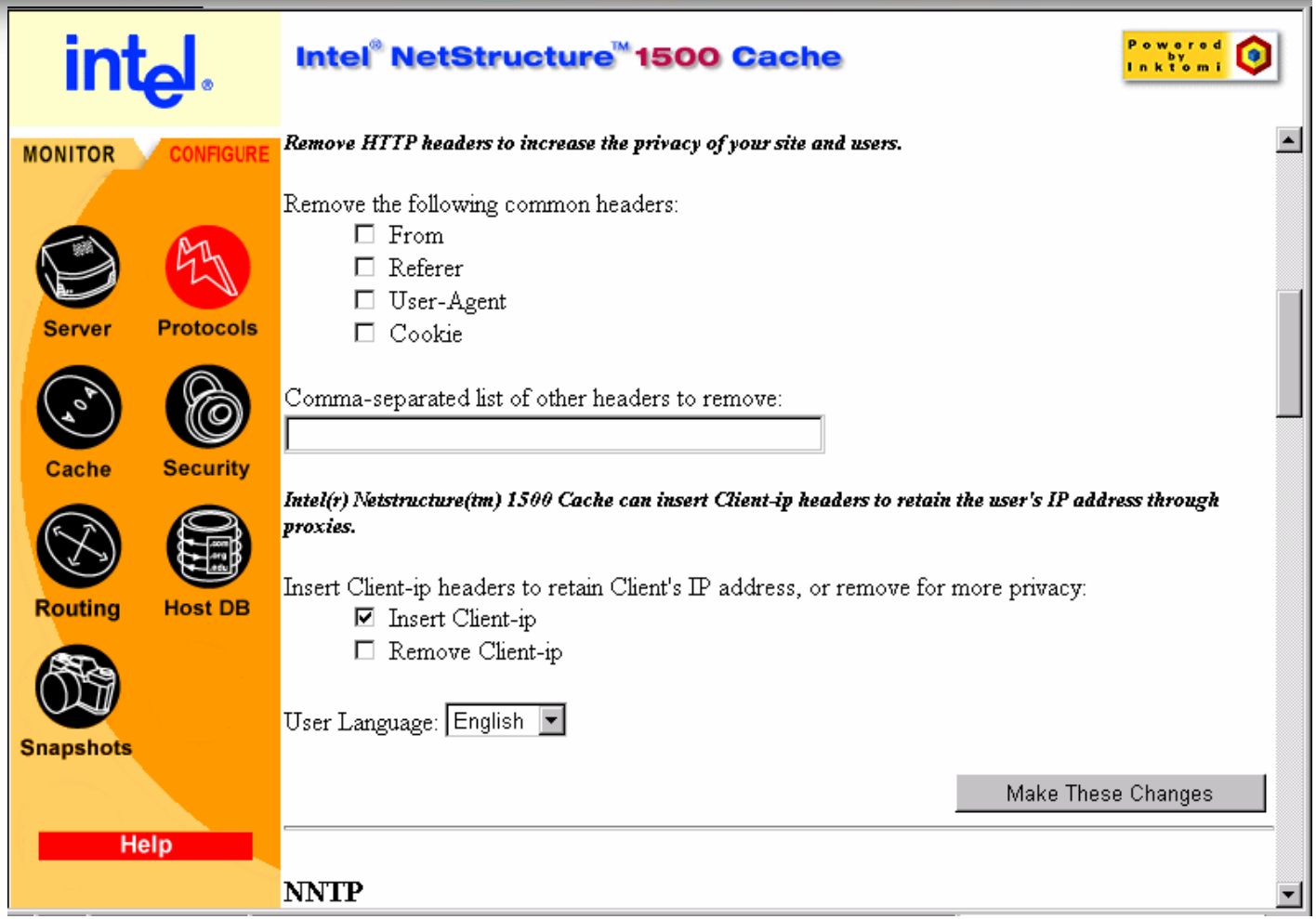
Manager UI: Protocols HTTP Details

- **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)**



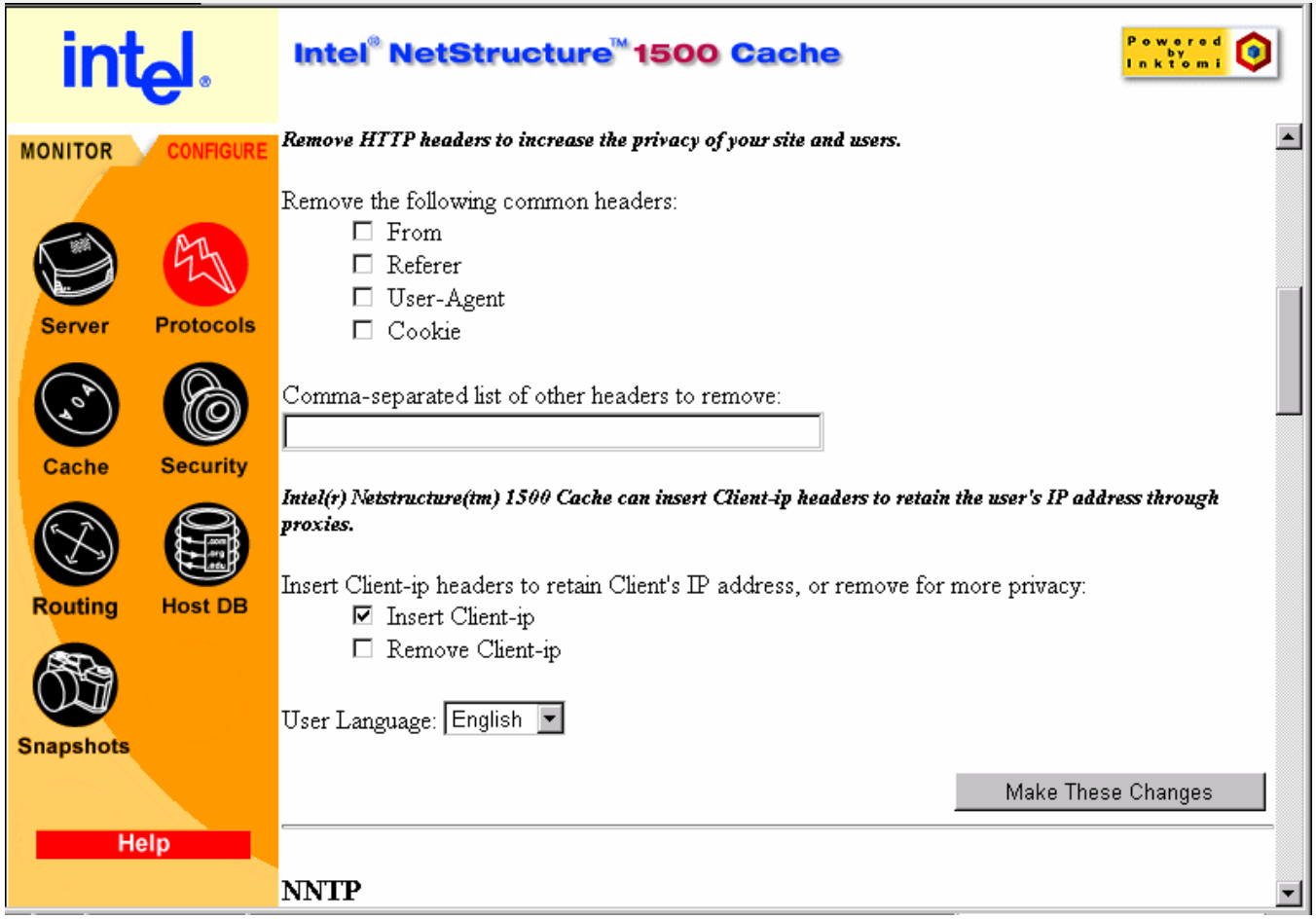
Manager UI: Protocols HTTP Privacy

- **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**



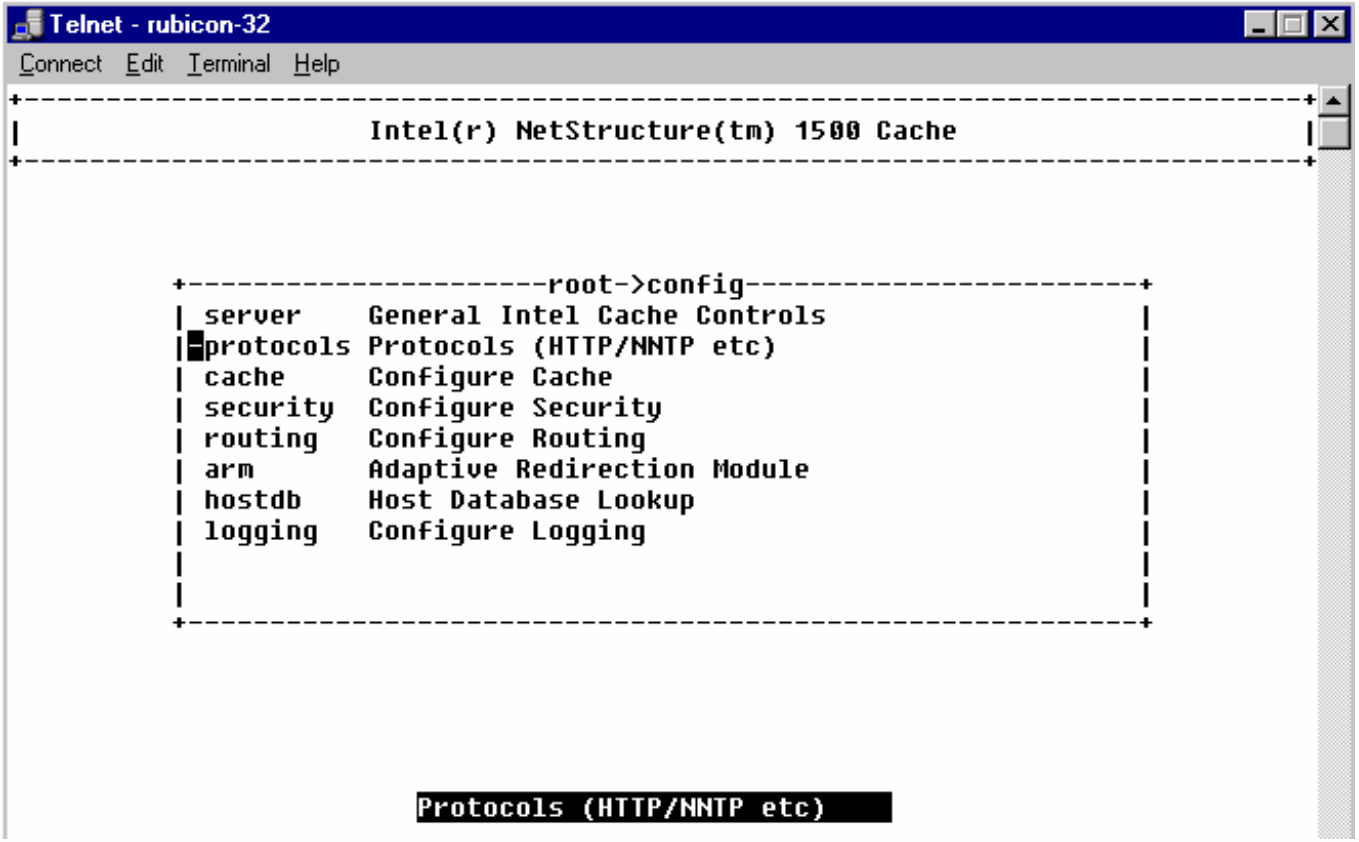
Retaining HTTP Headers

- 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



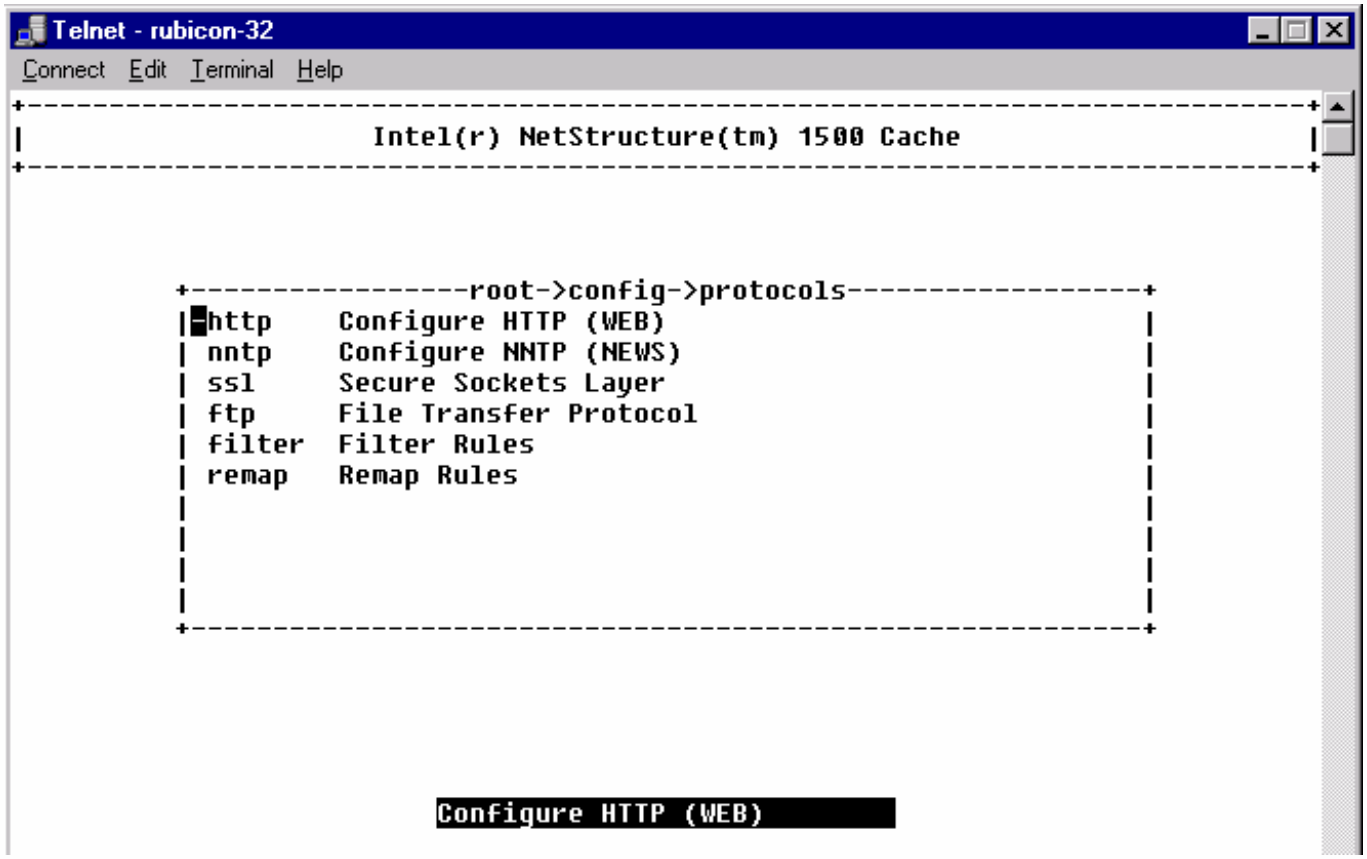
Client-ip headers and User Language can only be configured through the Manager UI

CLI: config->protocols



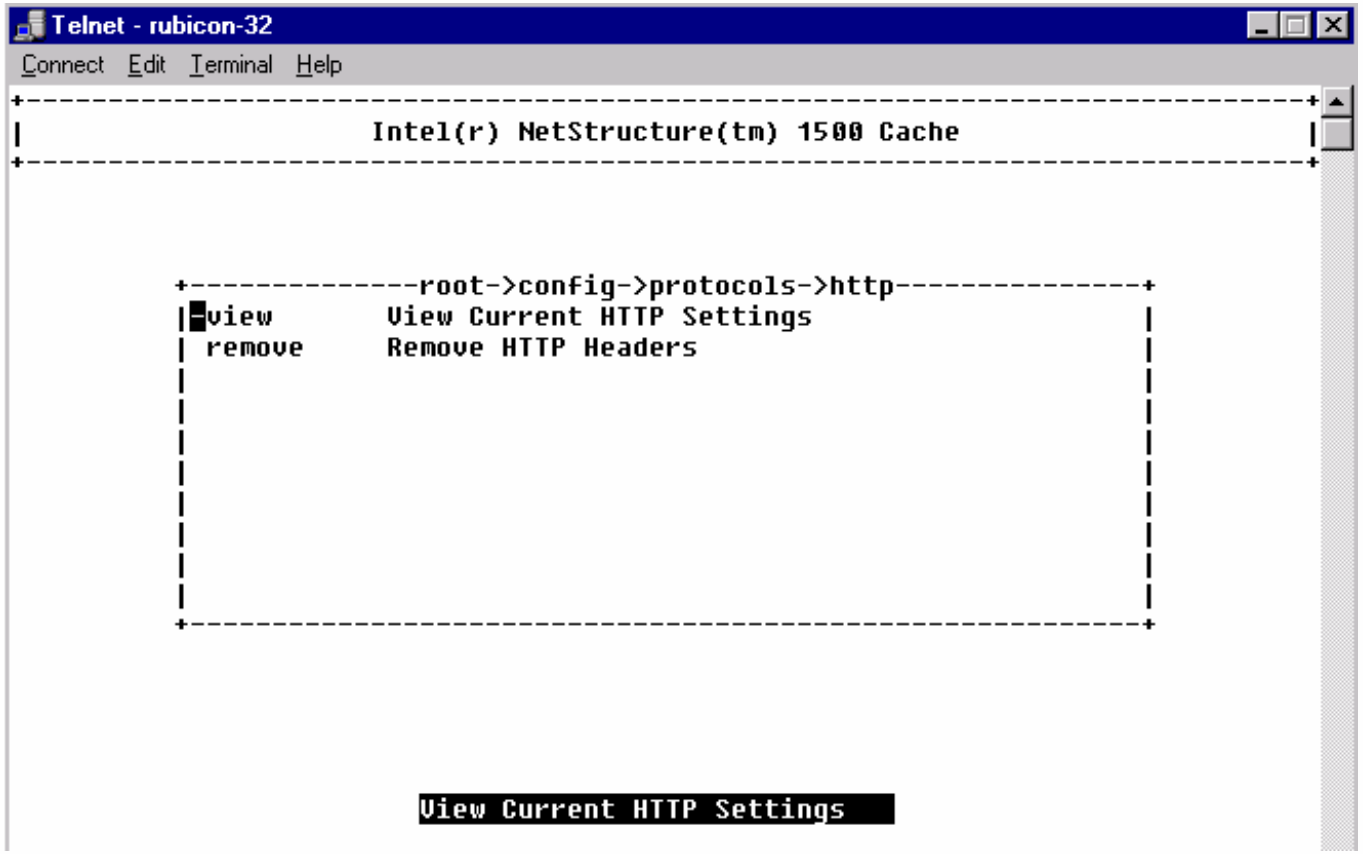
CLI: config->protocols menu

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



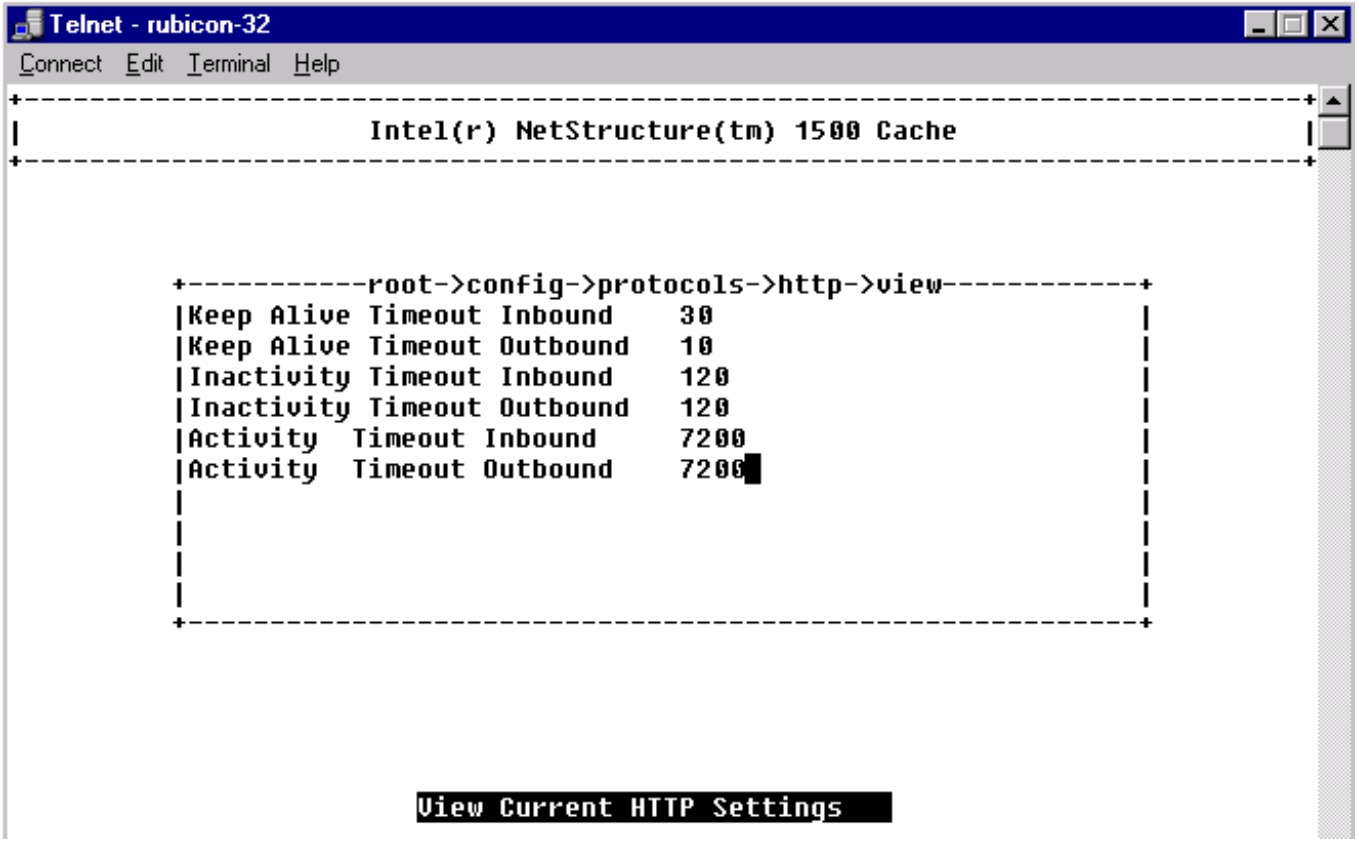
CLI: config->protocols->http

- **Options to view or change HTTP settings and headers**



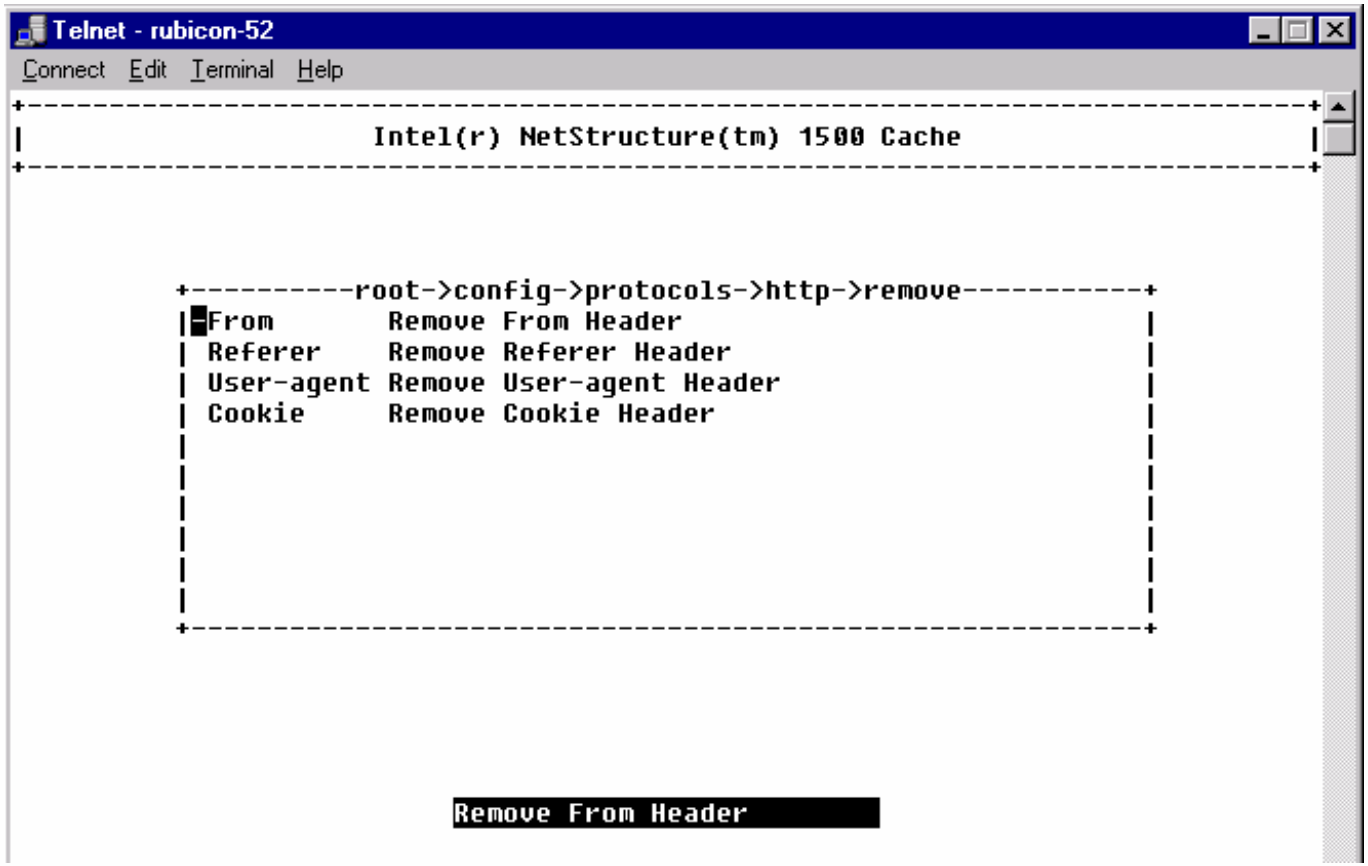
CLI: config->protocols->http->view

- **Same settings as in Manager UI**



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

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



Manager UI: Protocols NNTP

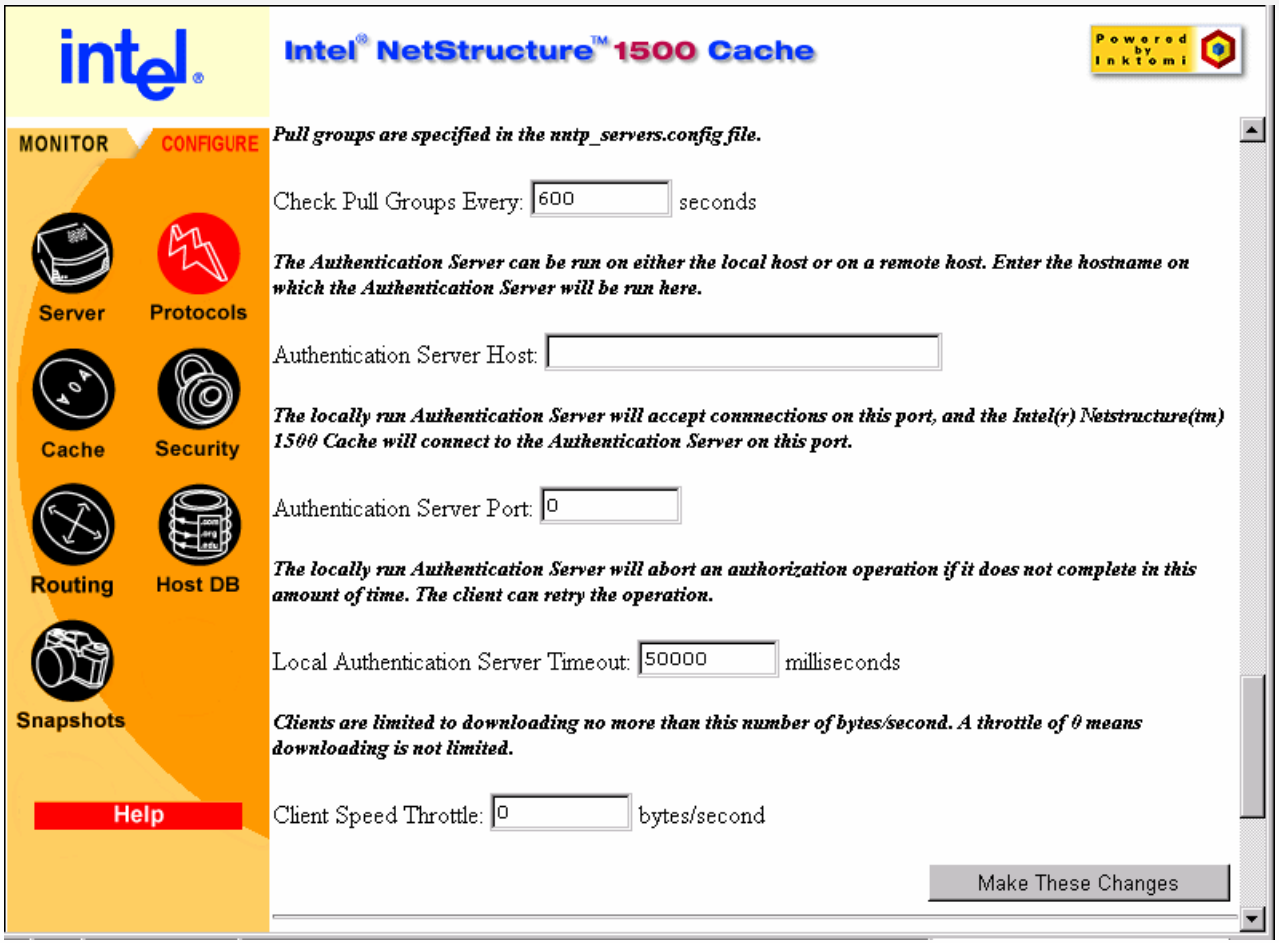
- **Enable/Disable Intel Cache to cache and serve news articles by turning NNTP server on or off**
- **Caution: you must click the Restart button to activate this change**
- **This page allows you to configure basic NNTP options to servers**

Configuring NNTP Polling

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

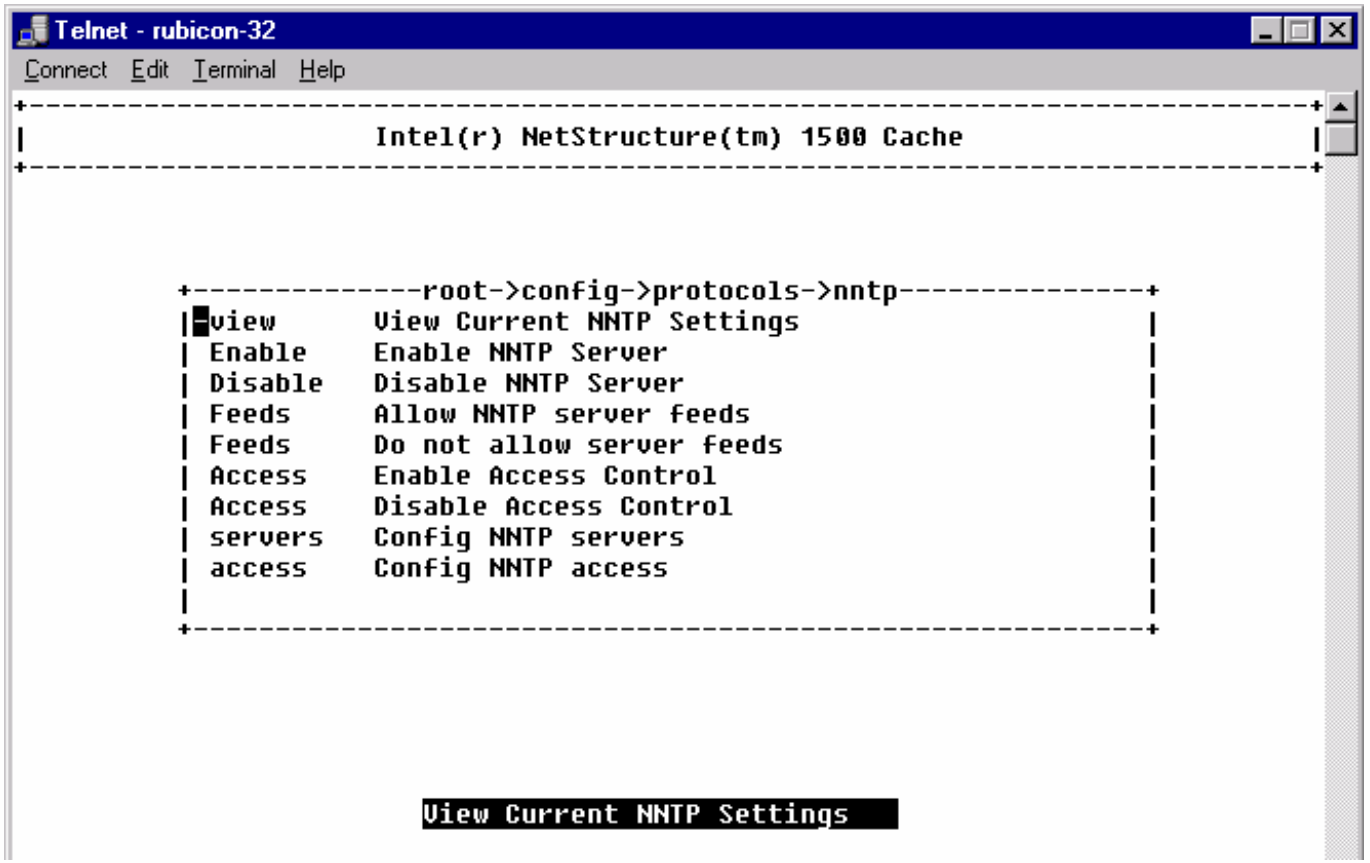
Configuring NNTP Polling

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



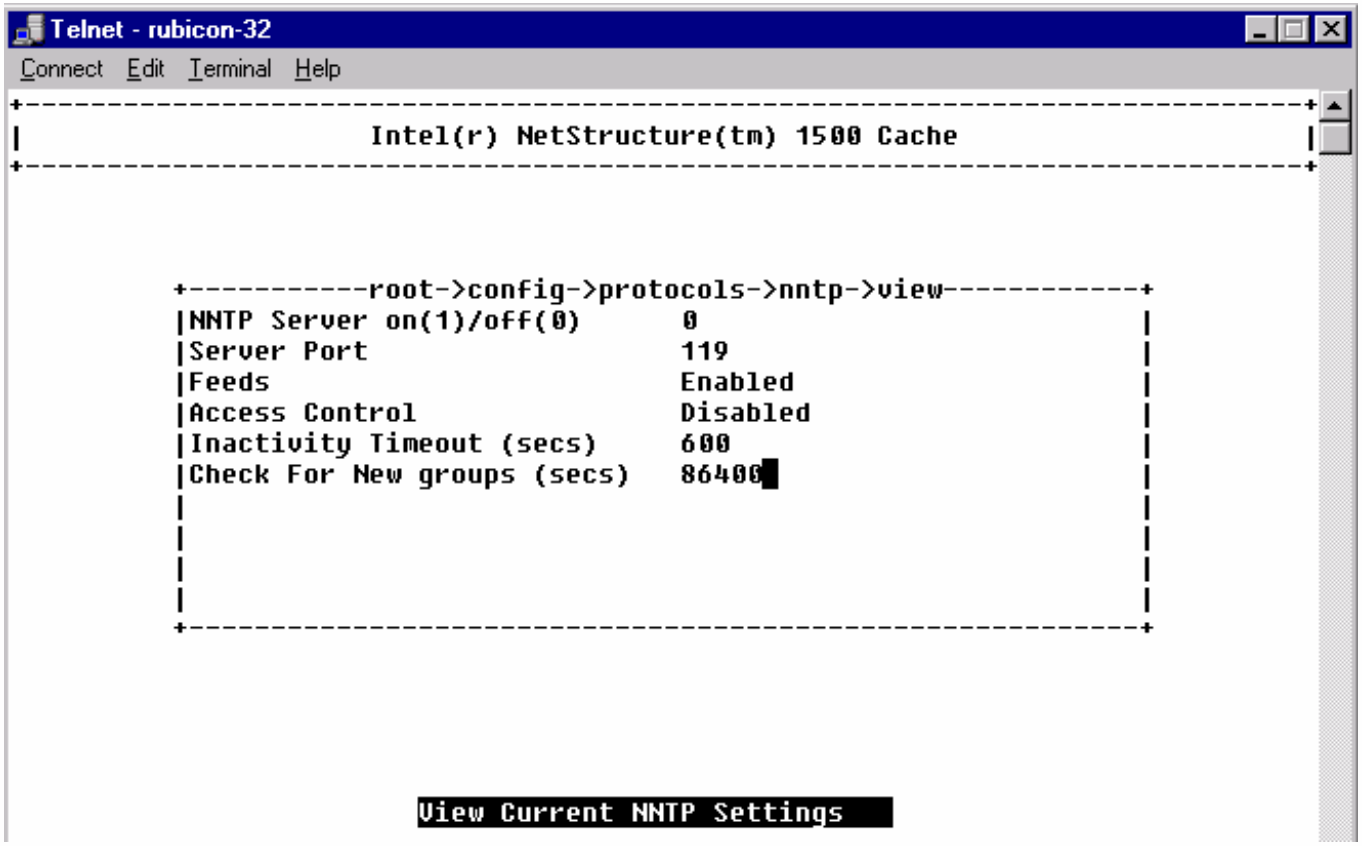
CLI: config->protocols->nntp

- **Control NNTP server enablement, server feeds, and access control from this menu**



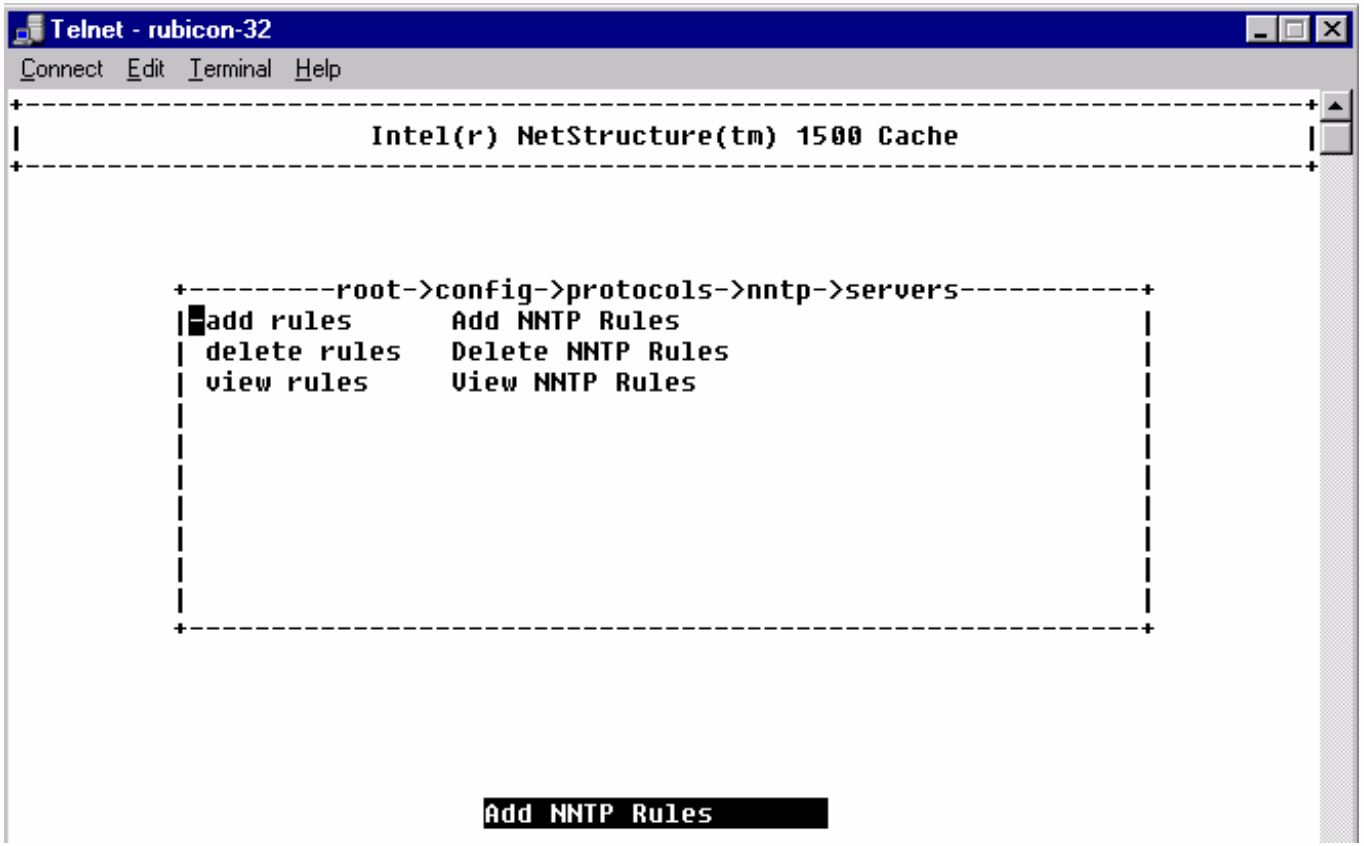
CLI: config->protocols->nntp->view

- **Show current NNTP configuration settings**



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

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



NNTP Server Rules

- **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

- **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

- **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

- **`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

- **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

- 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

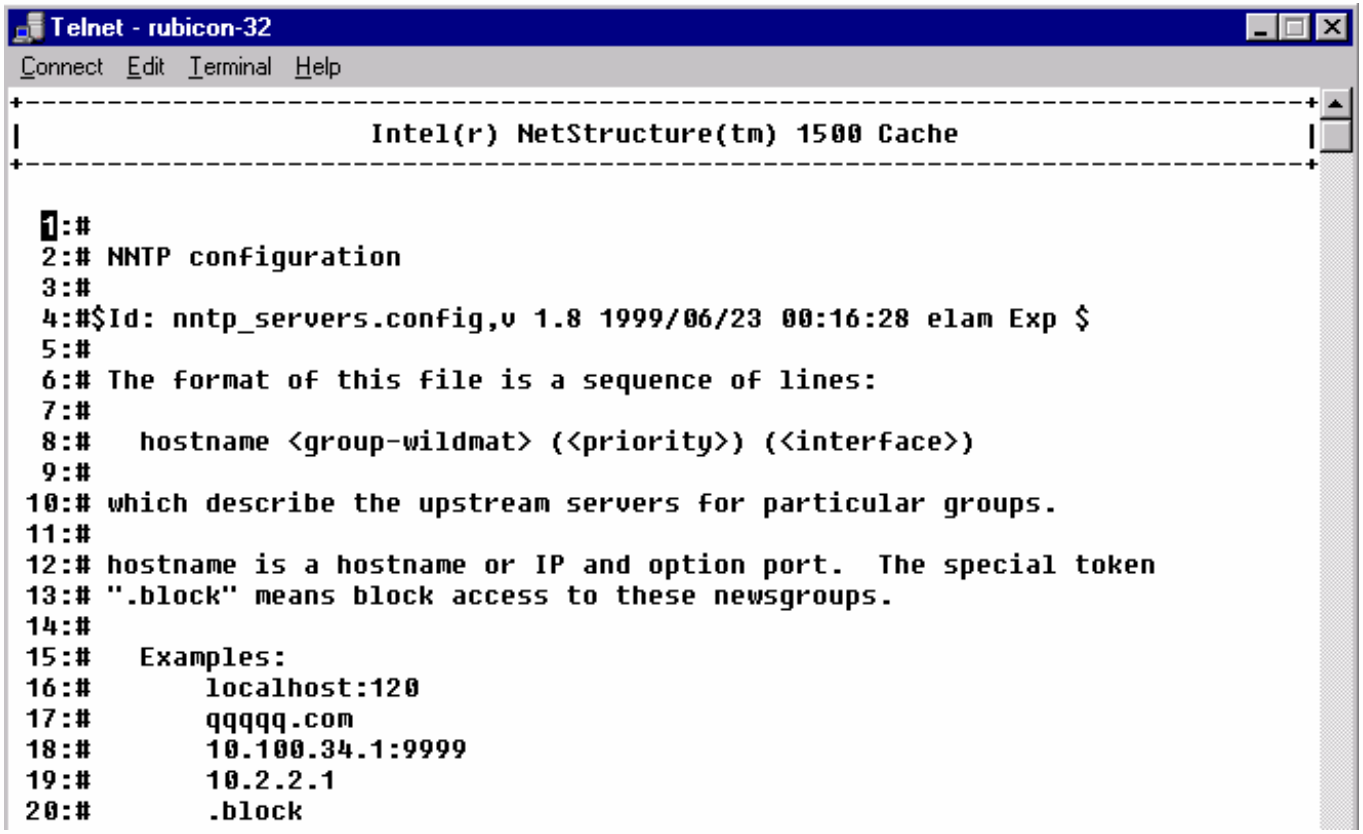
```
news.webhost.com * 0
news.backup.com * 1
```

- Define a pull group

```
comp.webhost.com alt.*
comp.webhost.com alt.bicycles pull
```

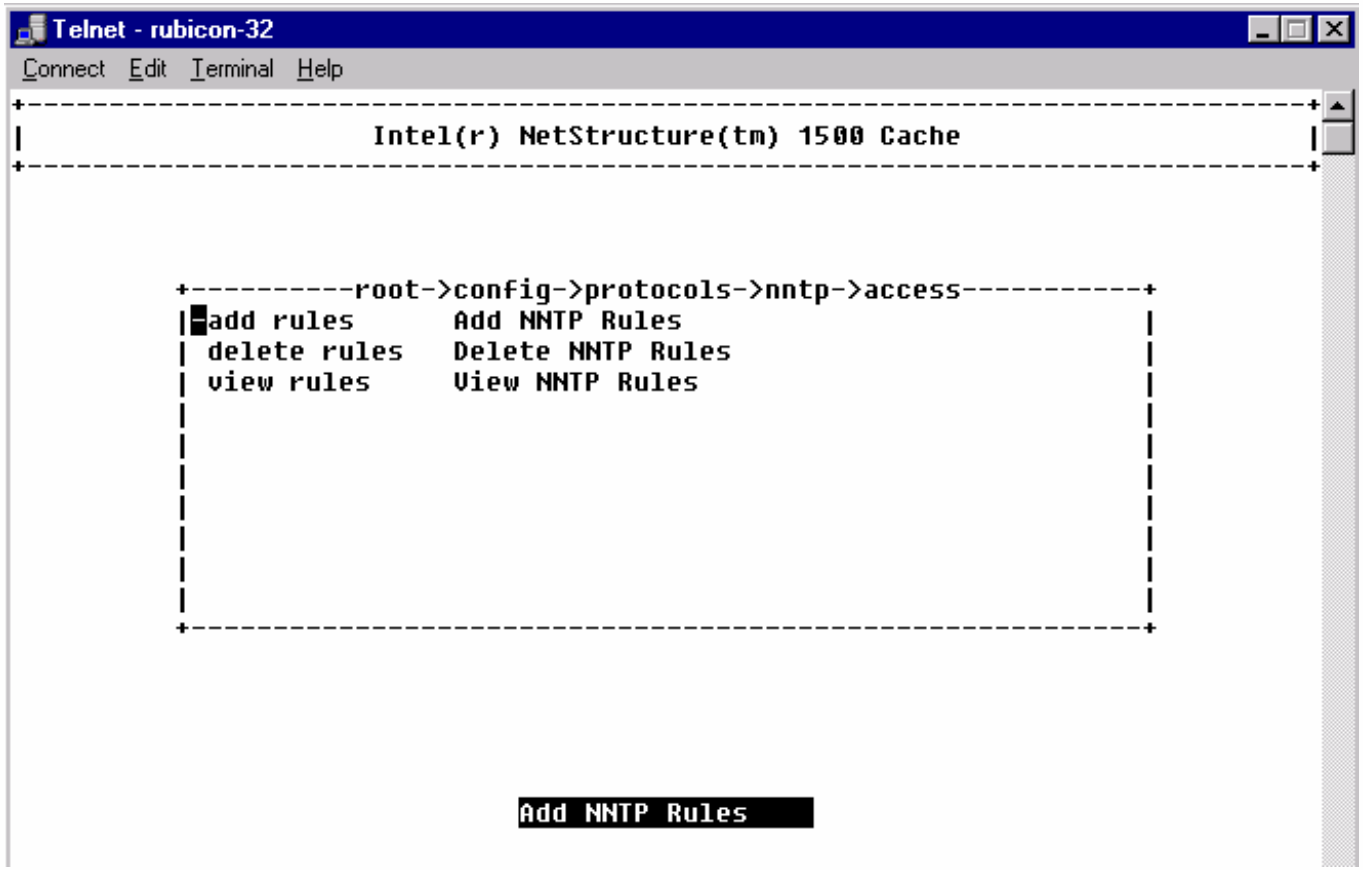
CLI: config->protocols->nntp->servers->view rules

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



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

- **NNTP access rules control user access to cached news articles**



NNTP Access Rules

- **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

- **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

- **NNTP access rules are of the format**

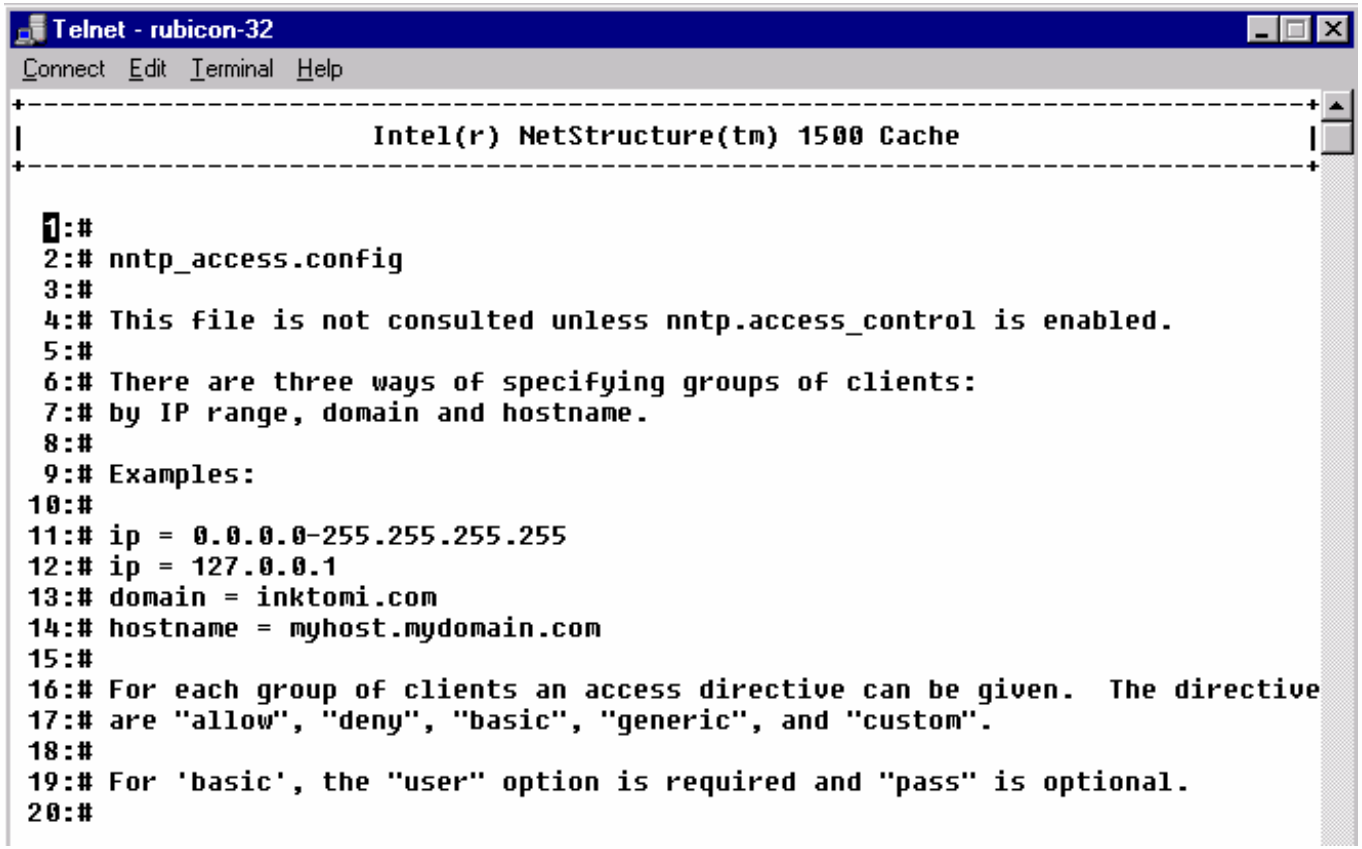
  client-group **access=*value authenticator users passwords***

  where *authenticator, users,* and *passwords* are optional

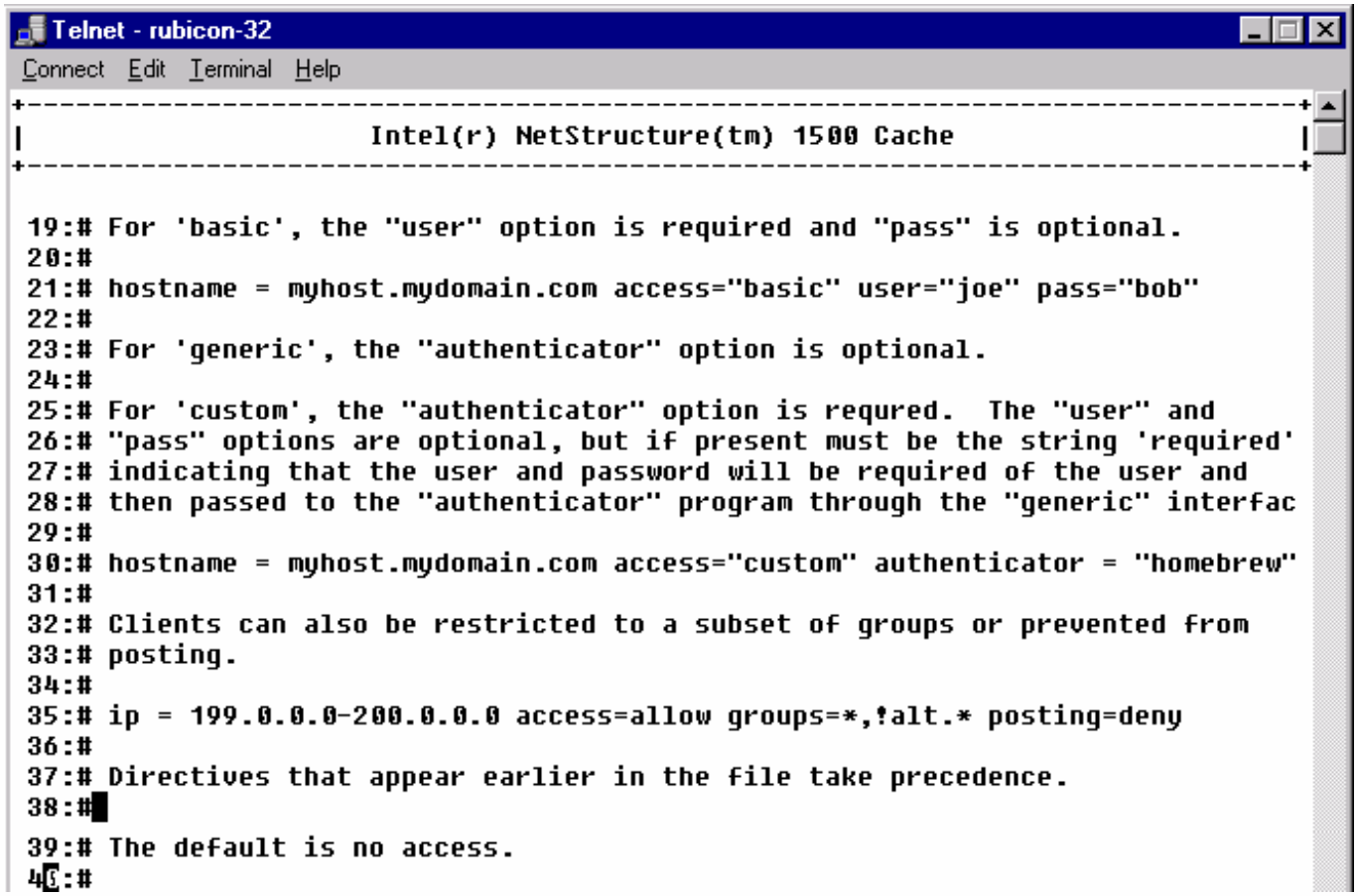
- **`ip=127.0.0.1 access="custom" authenticator="hb" user=required pass=required`**

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

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



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



CLI: config->protocols->ssl

- **Only handled by CLI; no access via Manager UI**