Unit 1: Traffic Server Overview

- Caching Review
- Traffic Server Architecture & Design Goals
- Product Overview
  - Key Features
  - Configuration and Performance Monitoring Basics
  - Practice Lab

Unit 2: Installing the Traffic Server

- Server Preparation Activities
- Installing the Traffic Server
- Verifying Your Installation
- Practice Lab

Unit 3: Configuring Traffic Server

- Exploring Configuration Options
- Reviewing Configuration Files
- Practice Lab

Unit 4: Monitoring Performance

- Built-In Maintenance and Recovery Features
- Analyzing Performance Statistics
- Responding to Alarms
- Specifying Logging Parameters
- Tuning Your Traffic Server
- Practice Lab

Unit 5: Maintenance, Performance and Troubleshooting

- Traffic Server Maintenance
- Performance Tuning
- Error Messaging
- Tips and Techniques from the Inktomi Pros

Unit 6: Using Traffic Line

- Traffic Line Modes
- Features and Options
- Practice Lab

Unit 7: The Solutions Workshop

- Clustering
- Transparency
- Reverse Proxy
- ICP
- NNTP

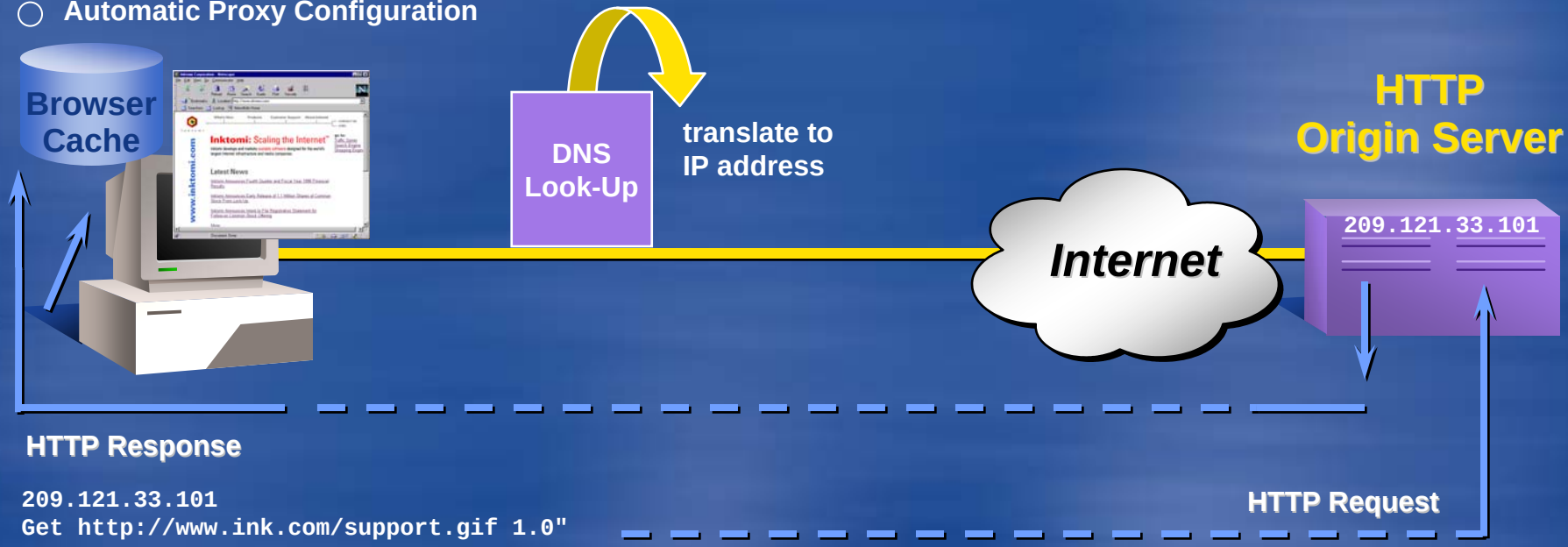
Traffic Server Overview

- Caching Review
- Traffic Server Architecture & Design Goals
- Product Overview
  - Key Features
  - Configuration and Performance Monitoring Basics
  - Practice Lab

Caching Review: The Direct Connection

Inktomi

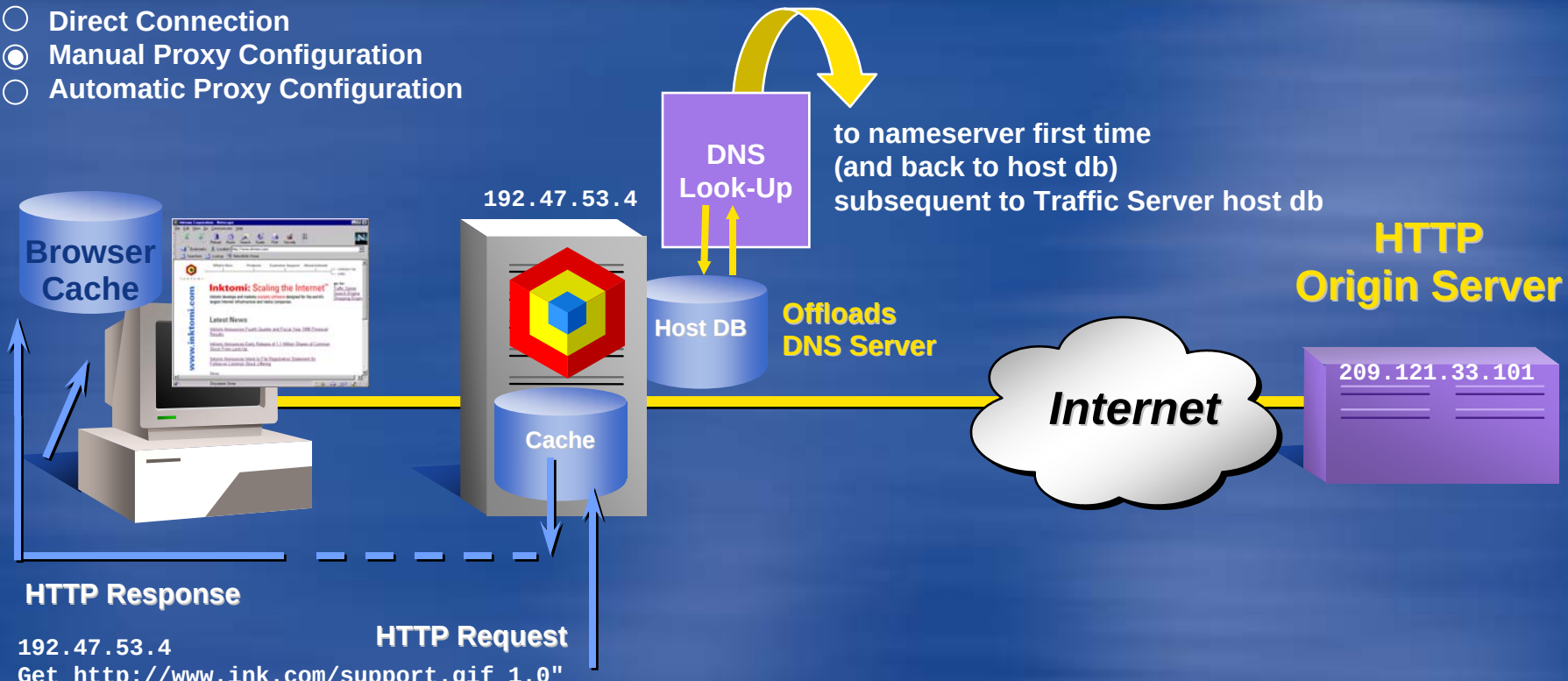
- (•) Direct Connection
- ( ) Manual Proxy Configuration
- ( ) Automatic Proxy Configuration



Caching Review: Manual Proxy Connection

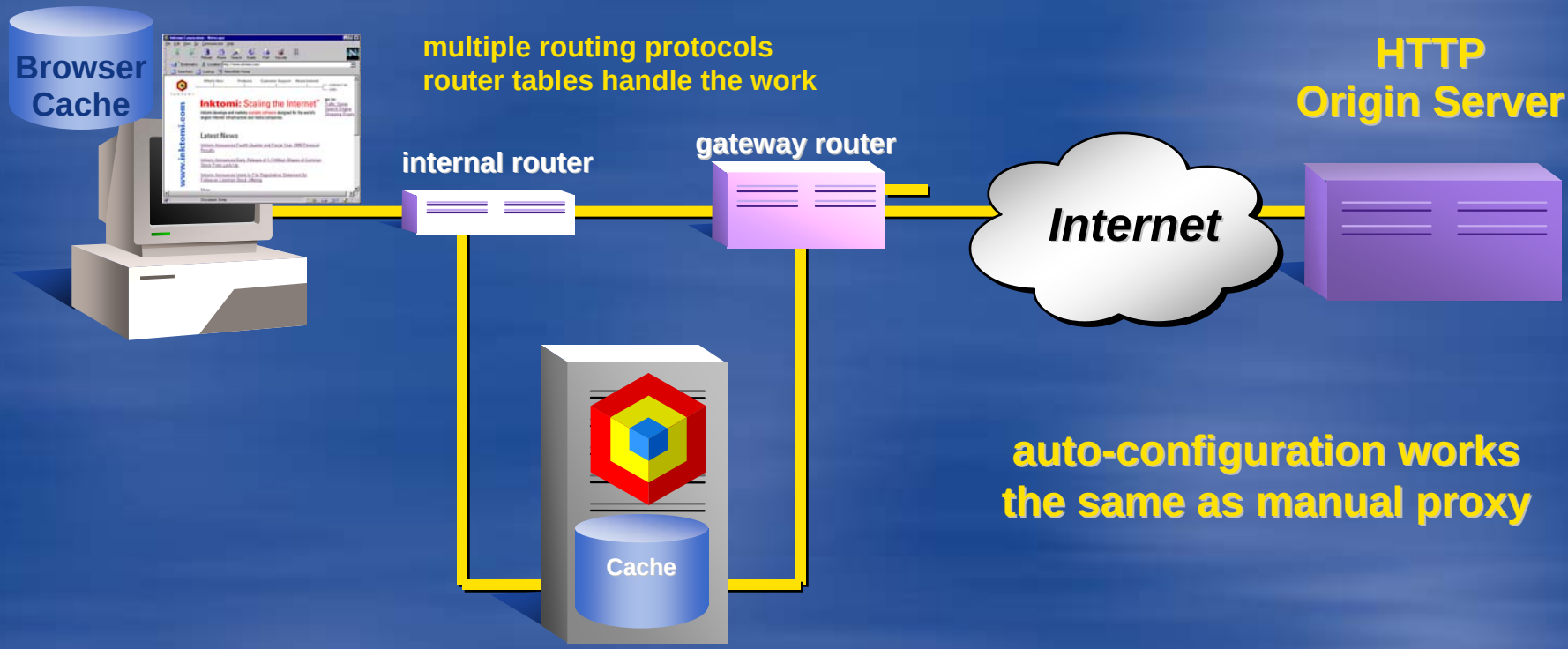
Inktomi

- ○ Direct Connection
- ◉ Manual Proxy Configuration
- ○ Automatic Proxy Configuration



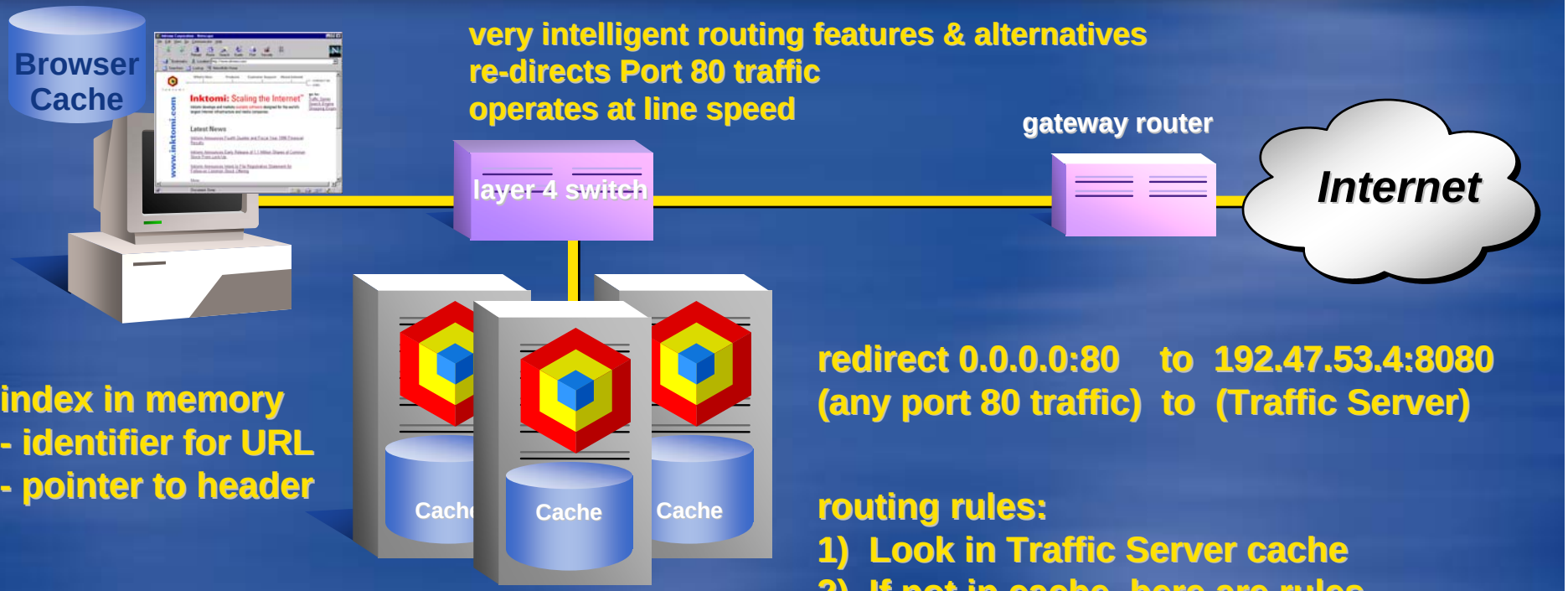
Caching Review: Manual Proxy with Router

- ○ Direct Connection
- ◉ Manual Proxy Configuration
- ○ Automatic Proxy Configuration



Caching Review: Transparency

"I know what you've asked for but I'm going to do this instead..."



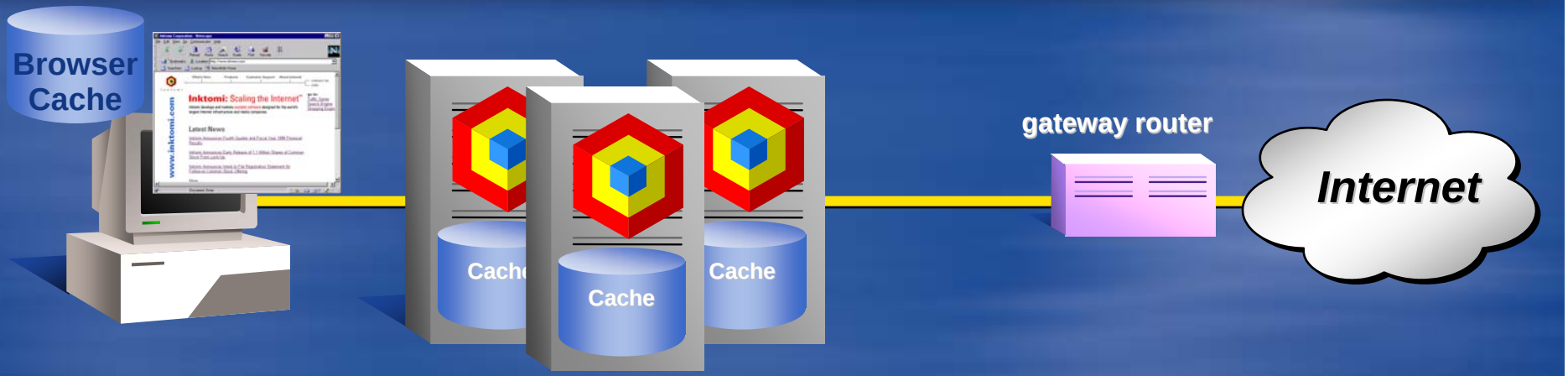
index in memory
- identifier for URL
- pointer to header

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

redirect 0.0.0.0:80 to 192.47.53.4:8080
(any port 80 traffic) to (Traffic Server)

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

Caching Review: Transparency Without Switch

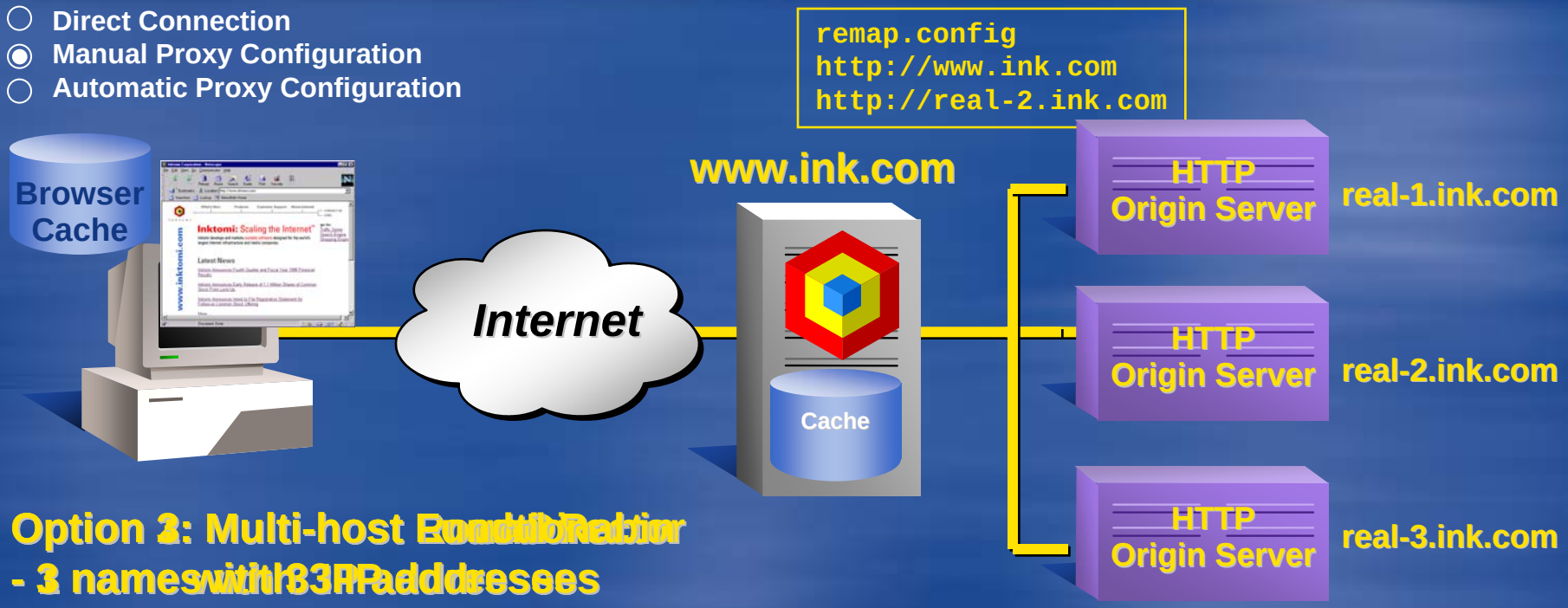


loads lots of work on Traffic Server to do router work
additional software required to handle port redirection
must write policy based rules
requires browser that does host/header
slower opening gateway
greater potential for problems

Caching Review: Reverse Proxy

Inktomi

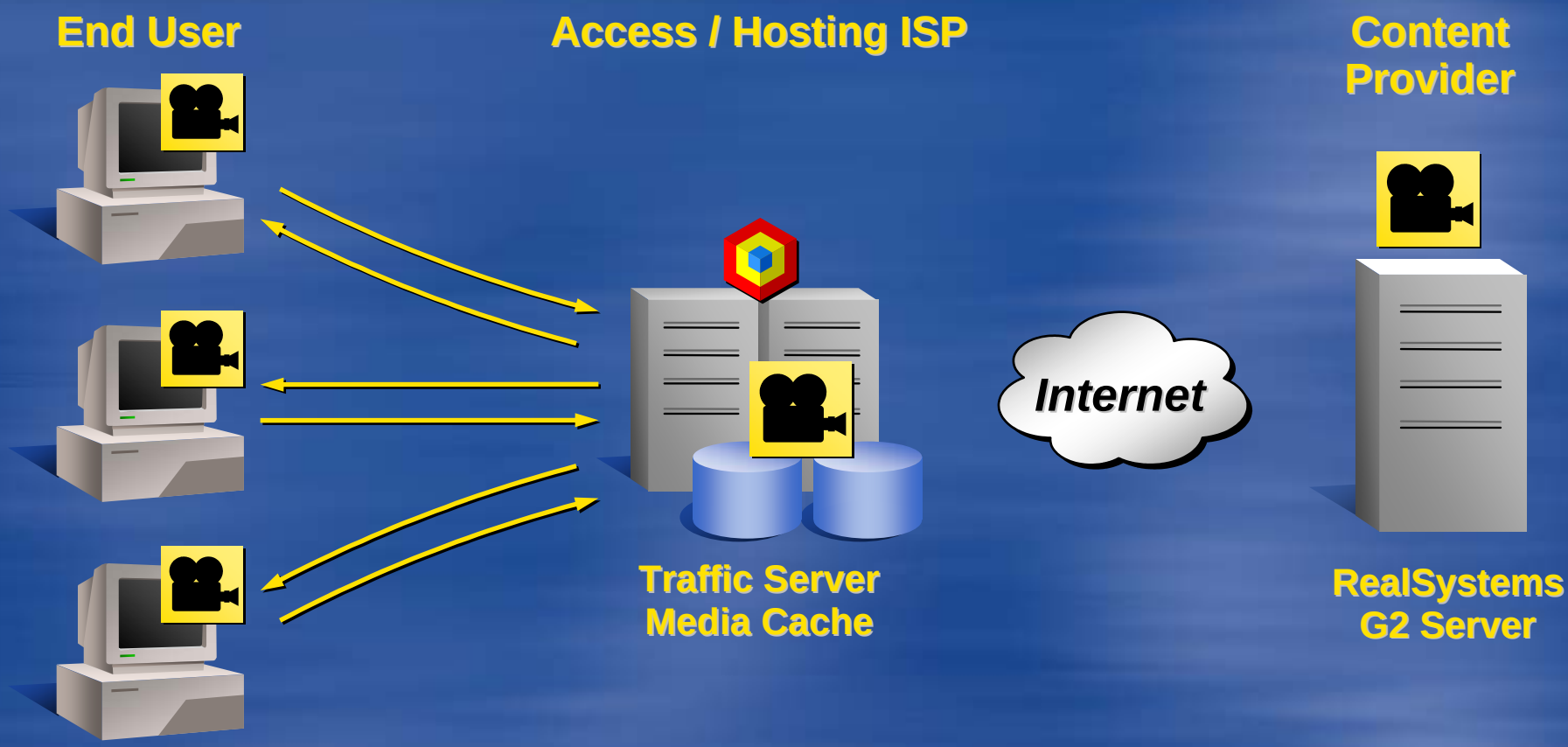
- ○ Direct Connection
- ◉ Manual Proxy Configuration
- ○ Automatic Proxy Configuration



Option 2: Multi-host Reverse Proxy

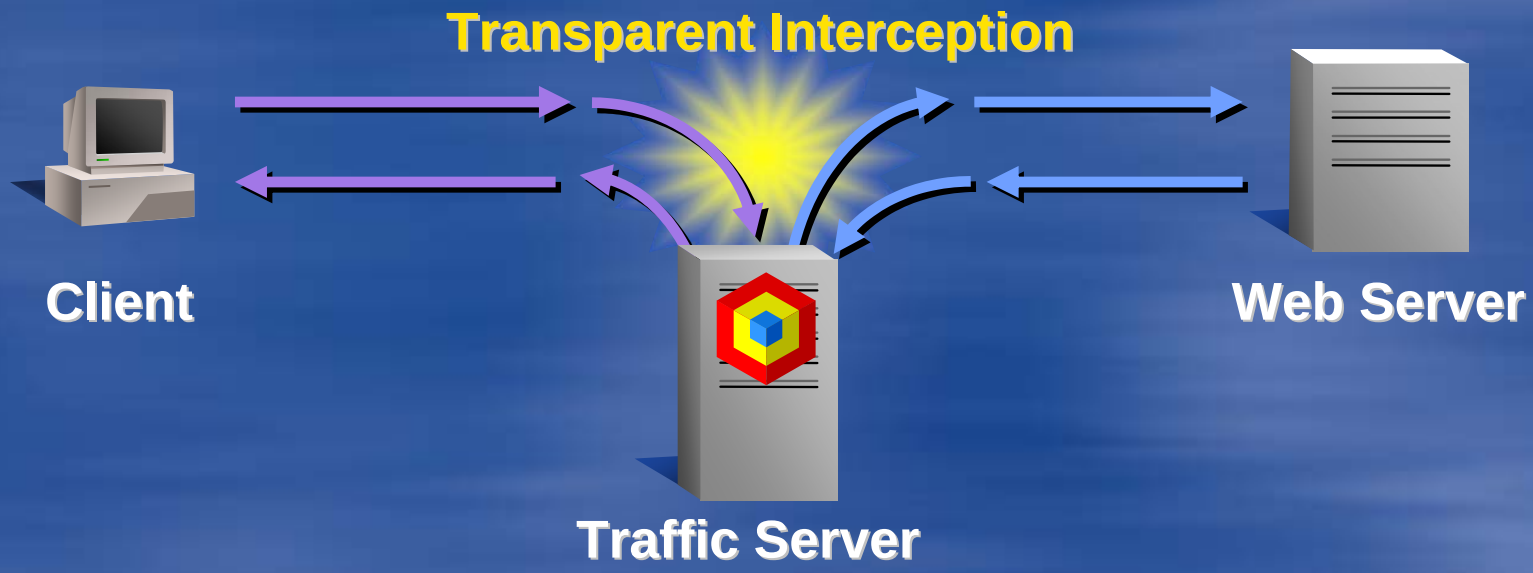
- 3 names with 3 IP addresses
- Specify which machine takes it
- All have different content
- Good for internal workers
- support.ink.com, sales.ink.com...

Caching Review:
On Demand Media Caching



Caching Review: Summary

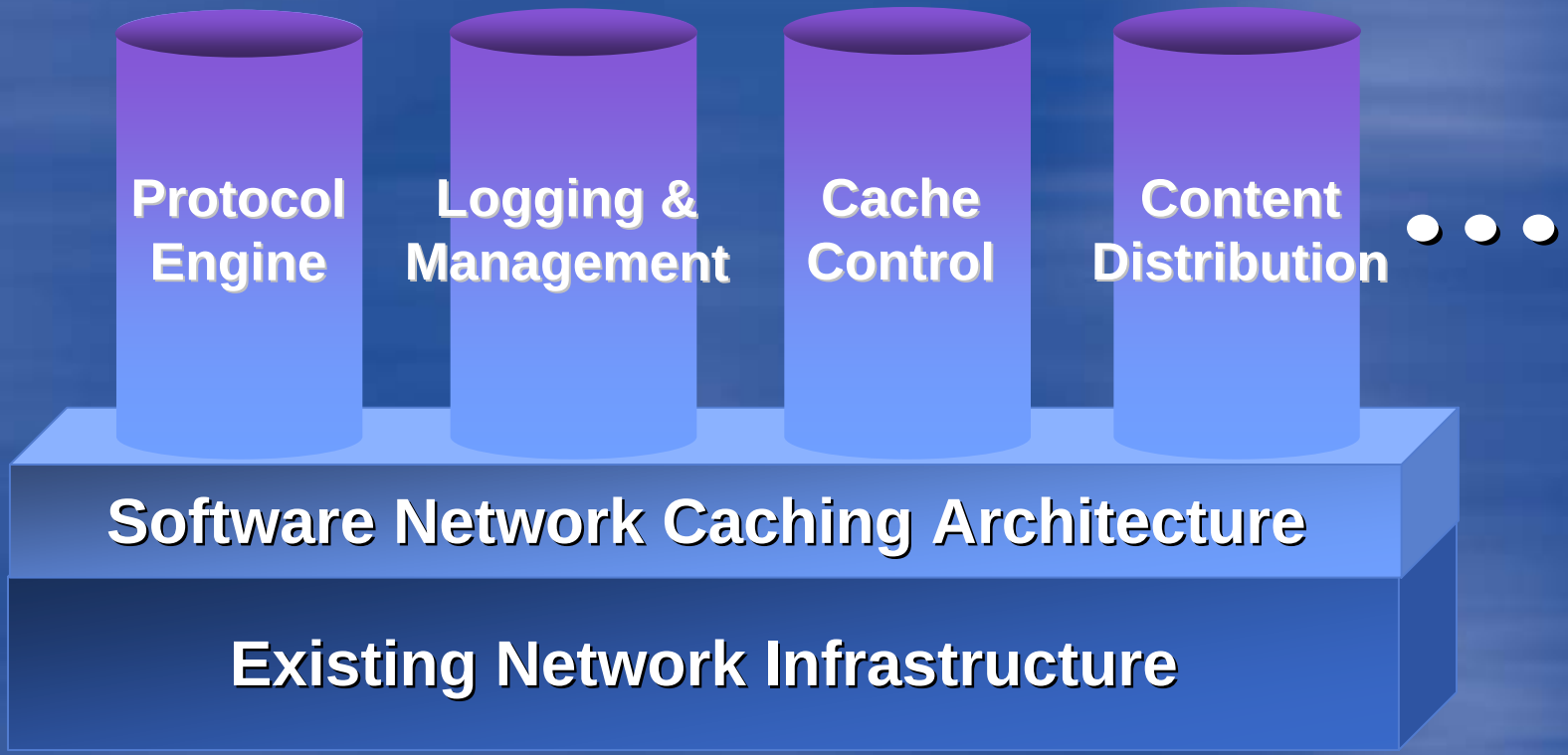
- Traffic Server provides multiple solutions for transparently intercepting and caching Internet traffic
- Creative configuration can address many unique problems



Traffic Server Architecture

Inktomi

Scalable Software Network Infrastructure for Building Smarter Networks



Application Features

- First Streaming Media Cache
- Rich Protocol Support
- Highest Performance - Proven for > 300 Million Hits / Day
- Most Scalable - Cache Sizes Over a Terabyte
- Completely Fault Tolerant
  - Single Node Rapid Failover and Recovery (< 30 sec)
- Simple Manageability
  - Single Interface for Cluster
  - Real Time Statistics
- HTTP 1.1 Compatible
- Transparent Caching Solutions
- Reverse-Proxy Capability

Protocol Support

- HTTP 1.1
  - Further leveraging current caching investment
- FTP
- Real Media: RTSP and PNA
- Network News Transport Protocol: NNTP
  - Improves news reading experience
  - Reduces bandwidth and news server load
- Inter-Cache Protocol: ICP
  - Interoperability with legacy and popular products

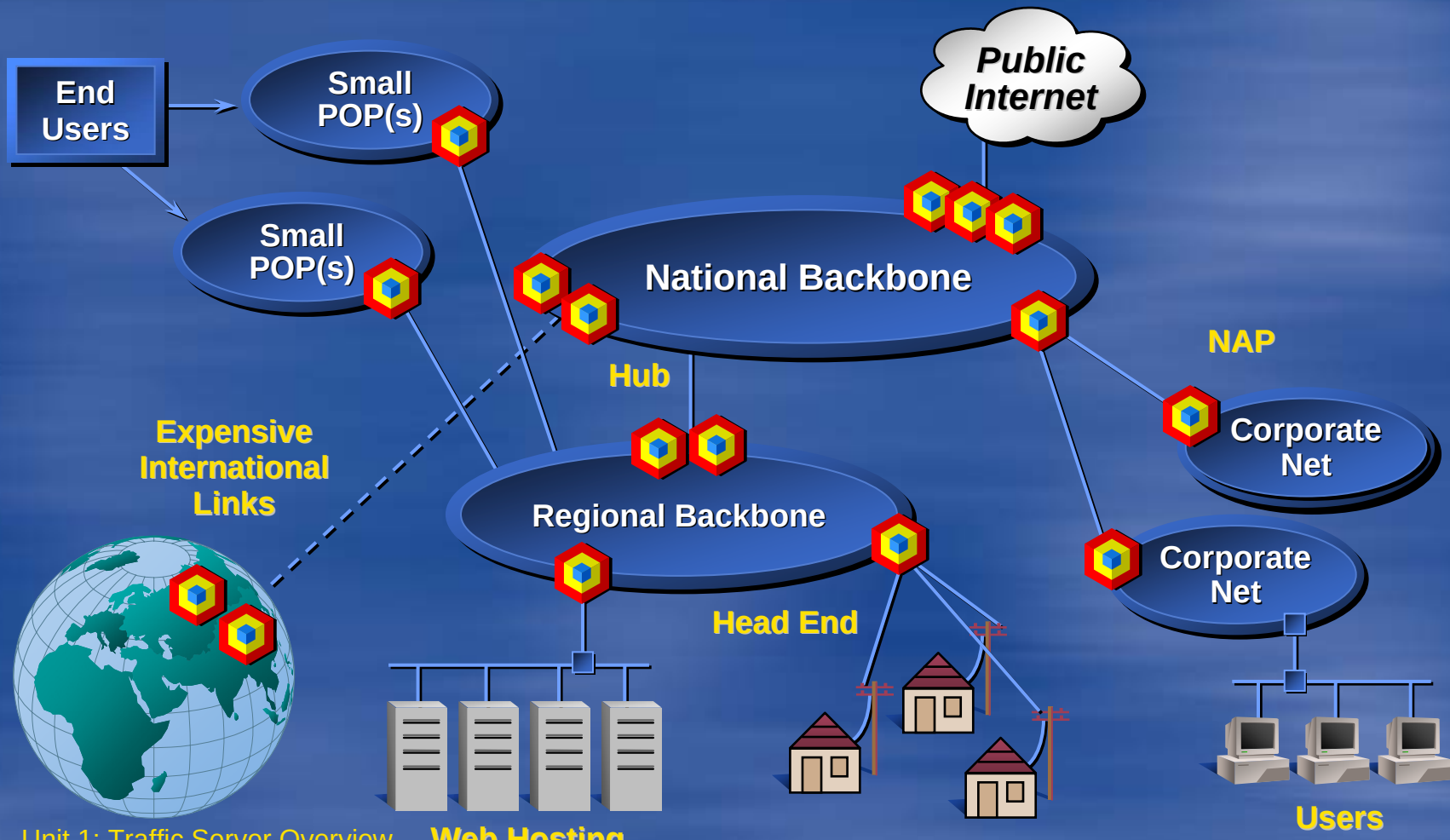
Manageability and Ease of Use

- Browser-Based Graphical UI
  - Full cluster with a single point administrator interface
  - Extensive real-time stats and graphical analysis
  - Configurable cluster-wide alerts and alarms
- Secure Command Line Interface
  - Application and script integration
- Improved Installation and Logging Facility
- Reverse Proxy for Web Hosting
- Simple Network Management Protocol (SNMP)

Performance and Control

- Performance
  - Increased cluster performance and RAM cache
- Cache is simpler, faster, better
- Access Control
  - What client IP's are allowed
  - Multiple administrative access levels
- SSL, SOCKS and HTTPS Support

Meets Any Network Need



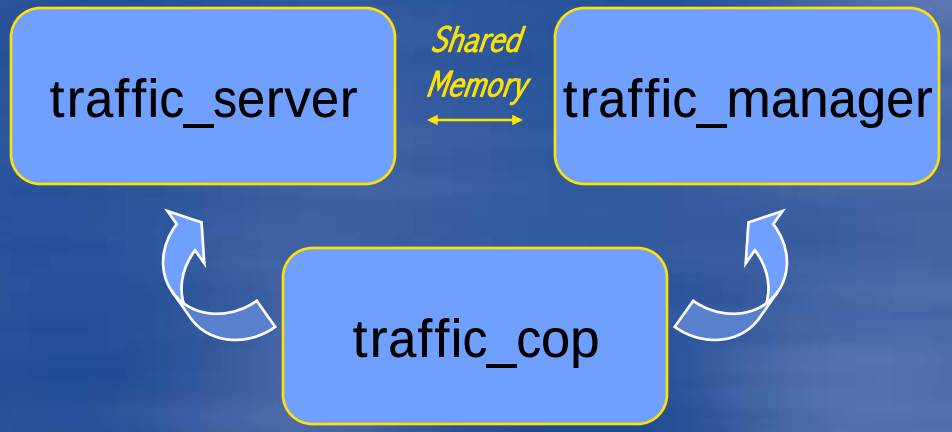
Traffic Server Innovations

- 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
  - Flexible logging and centralized administration

Traffic Server 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

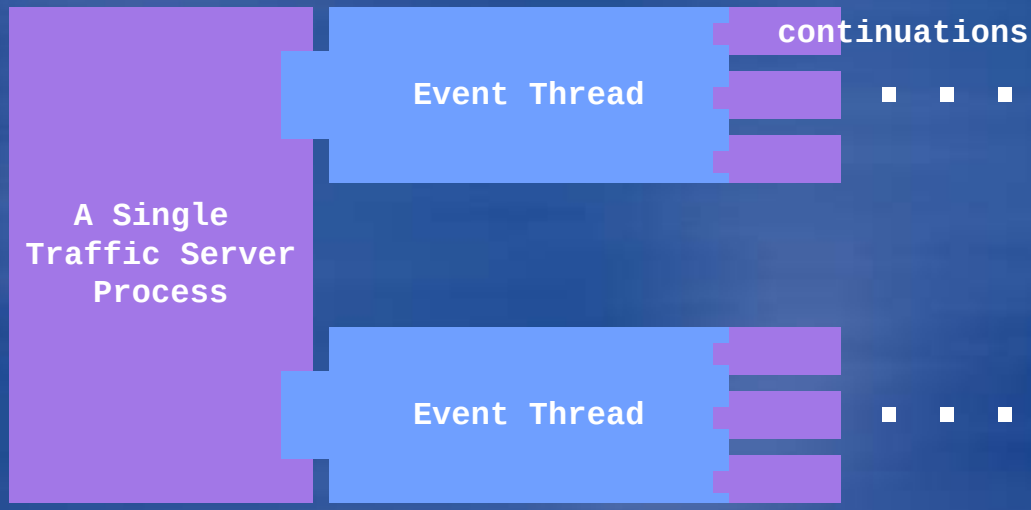
Very small - watches the Traffic Server and Traffic Manager to make sure they are running. Restarts as necessary.

Traffic Server Architecture

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

Remarkable Efficiency

- Engine built to support multi-threading
- Fast, light-weight processes break large transactions into small memory efficient tasks
- Thousands of concurrent tasks can run, so work continues efficiently even during peak periods



Thousands of Active State Machines

Each performing a limited task as a part of an event

Continuations are very small C++ objects that capture state, activation functions and currency controls

Traffic Manager Architecture

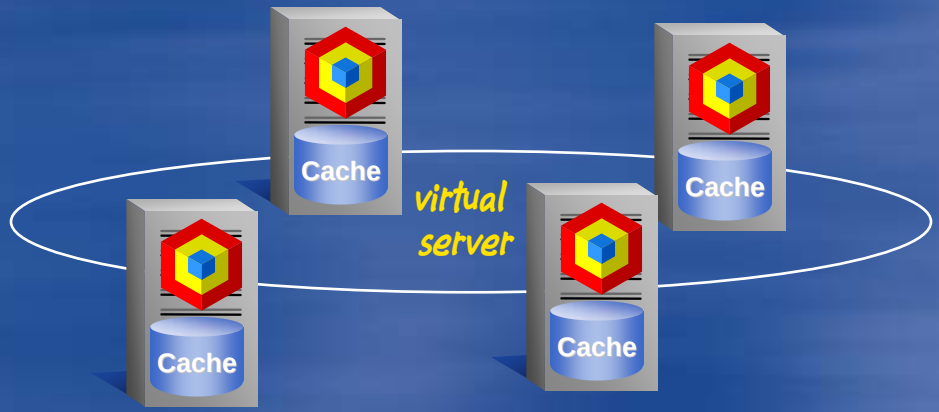
- Cluster Management System with
  - Automatic configuration distribution
  - Aggregation of statistics
  - Coupled Clustering creates virtual shared cache
    - High reliability thanks to virtual IP fail-over
    - Scalable for high throughput
- Outstanding user interface with single point administration
- Powerful management tools

Configuration Management

- Because the Traffic Server is composed of “look-alike” nodes, you can
  - Easily add one or more nodes
  - Add additional disks to a node
  - Bring nodes up and down for maintenance
  - Remove a node
- Configuration “snapshots” allow you to capture a set of configuration files
  - In less than a minute you could restore an old configuration
  - You can switch back and forth between configurations for “what if” tuning

Reliability and Scalability

- Coupled clustering provides high availability and easy scaling as needs grow
  - Nodes work together as a single unit
  - Automatically reconfigured within seconds
  - Traffic destined for failed node is intercepted by working nodes

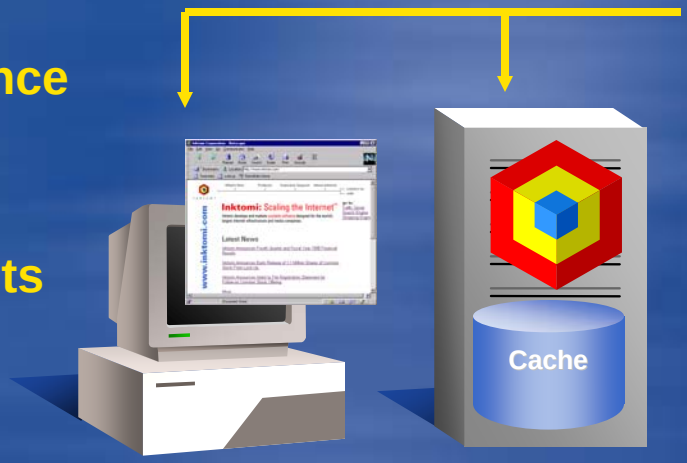


Install the number of nodes that meet your demands today
Add extra nodes as your needs change

Advanced Dataflow Engine

- 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



Host Database

- Includes a fast, asynchronous DNS resolver to streamline conversion of host names to IP addresses
- 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

Object Database

- Each node maintains a cache of popular objects in a custom flat-file database which includes:
  - The disks that are used to store data objects
  - An index for locating stored objects
- 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

Hierarchical Caching

- Hierarchical caching allows you to identify a “parent cache” to speed object retrieval
  - 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
  - The parent cache can be any other proxy server
- The Traffic Server stores alternative versions of the same document (different languages or browser formats) and serves the correct version to users based on browser settings

Web Server Acceleration

- Traffic Server can proxy for your web server (or a group of servers)
  - Assumes load that would normally fall directly on your web servers
    - Impersonates your web server
    - Much faster than most servers are capable of responding
  - Balances the load of your web servers
    - Shields them from load spikes
    - Simplifies content management
    - Adds another layer of security between your servers and the Internet

Flexible Logging

- Traffic Server provides a powerful logging system to meter and record network accesses:
  - Provides information about:
    - Every user request handled
    - All error conditions detected
  - You can set:
    - Amount of space allocated to log files
    - The format and content of log files (typically Netscape or Squid)
    - Guidelines on clearing the logs
- To analyze logs, use Netscape or Squid tools

Built-In Recovery

- Since clustered nodes work as a single unit they can automatically cover each other if there is a problem
  - The database and all cached objects are periodically saved to protect against system crashes
    - In the event of a node failure, cache recovery is automatic
  - The Traffic Server automatically balances database contents across all of the nodes

Routing Features

- Sophisticated routing features allow you to establish:
  - HTTP Parent Caching
    - Traffic Server participates as a member of an HTTP cache hierarchy (can include other caching products)
    - Supports multiple parent caches and parent failover
  - Internet Caching (ICP) to allow the Traffic Server to query ICP hierarchy members (peers) for cache hits
  - Reverse proxying (Web Server Acceleration) allows Traffic Server to act as the proxy for a web server rather than a client
    - Specify document routing rules that translate client URL requests and redirect them to a Traffic Server
    - Reverse mappings rewrite location headers in origin server responses

Security Features

- The Traffic Server secures access to the Traffic Manager
  - Authentication on or off - provide ID and password
- The Traffic Manager secures access to cached objects
  - Supports SOCKS firewall protection
    - Client and web server communicate using SSL through a tunnel provided by the Traffic Server
    - Does not cache or examine encrypted data
- Traffic Server also provides for SSL connections to the manager port, so the Traffic Manager session can be secure
    - Requires an SSL certificate issued by Inktomi

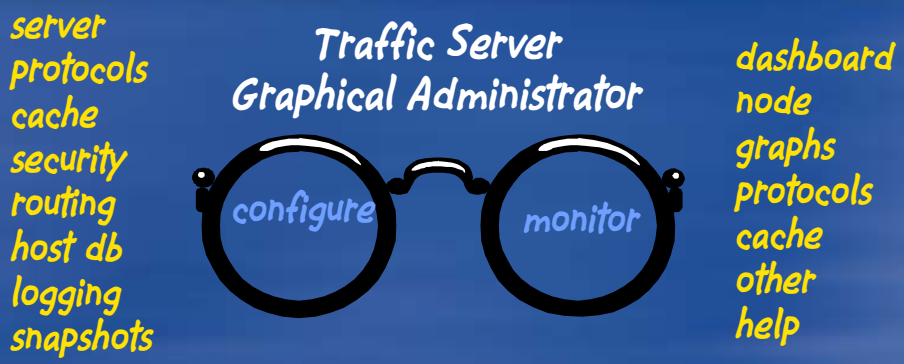
Invisible to Users

- Data goes directly to the user while caching is underway
- Users never notice the Traffic Server or its caches
  - Simple browser options activate the Traffic Server
  - Transparency can be set on the server side for automatic browser configuration
  - Users specify standard web addresses
  - The Traffic Server 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

Maintaining Current Information

- Sophisticated garbage collectors remove stale data
  - At installation disks are partitioned to allocate space for caches
  - The Traffic Server will automatically begin garbage collection when the cache fills to 90%

Graphical Administration



- Provides secure *single-point* administration for large clusters
  - Configure, monitor and tune all features and services
  - Encrypted remote administration
  - Powerful and centralized logging system

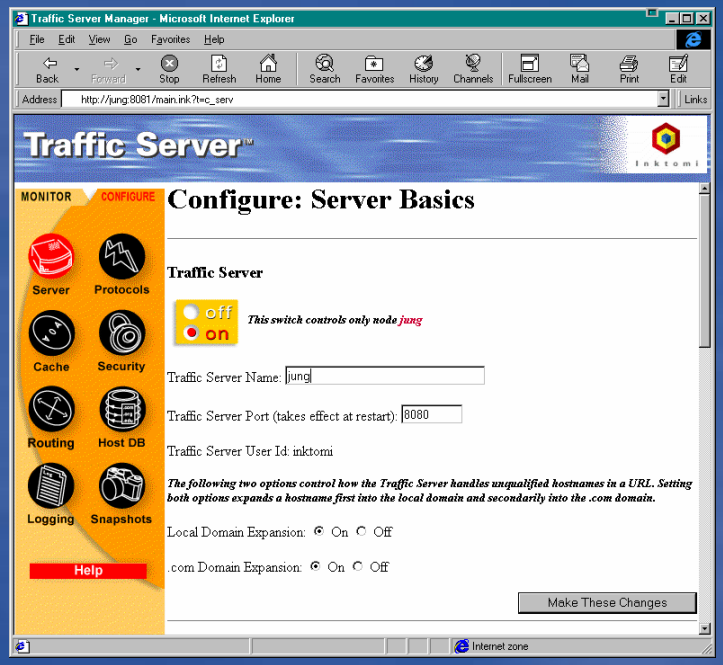
Monitoring Performance

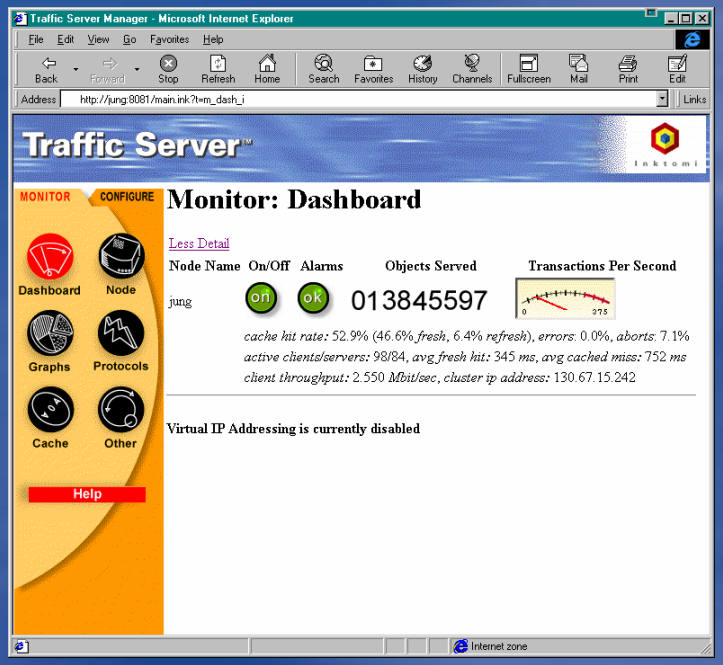
- You can view aggregate statistics for the entire cluster or zoom in on a specific node
  - Check if nodes are up on your Dashboard
    - Alarms notify you if there is a problem
  - Request graphs that depict time and performance averages
  - Compare the performance of a single node to the overall performance of its cluster
  - Monitor caching activities and caching size

investigate details of high-level information as needed!

Graphical Administration Tools

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





http://charlotte:8081
http://<node name>:<admin port>

alarms alert you about problems and logs track all requests to the server

Server Configuration Basics

- Default (recommended) configuration values are assigned during installation

Details are in your workbook:

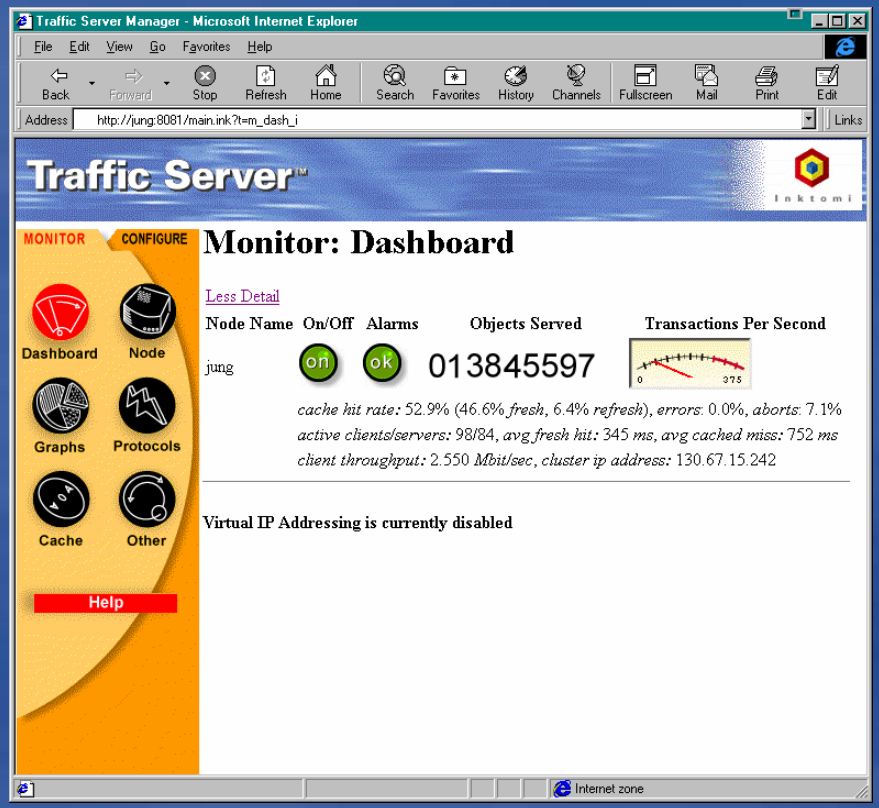
- Turn the server on or off
- Identify ports and processes
- Restart or reconfigure the Traffic Server
- Configure virtual IP addresses
- Auto-configure browsers to connect to the Traffic Server
- Configure maximum number of connections
- Turn SNMP on or off

Traffic Server Help

- New online HELP allows you to learn more about particular pages in the Traffic Manager

Server Monitoring Features

- The Traffic Manager process monitors all Traffic Server activities and reports performance through a series of browser pages



Details are in your workbook:

- Show all nodes in a cluster with alarms and alerts
- Review workload on the cluster or a selected node
- Request a graphical display of various statistics
- Show activities on HTTP, NNTP, ICP and FTP connections
- Show size and activities on cache space
- Evaluate DNS performance, SOCKS connections and remote logging activities

Practice Lab

- Please complete the Unit 1 lab detailed in your Student Workbook
  - Start the Traffic Manager
  - Review Configuration and Monitoring Options
  - Take the Unit 1 Spot Quiz

Installing the Traffic Server

- Key Installation Steps
- Configuring the Target Node
- Installing Traffic Server
- Verifying Your Installation
- Working with Special Class Tools

Preparing to Install

- Root privileges are required for installation
  - Facilitates creation of directories and files in areas that are restricted to root user
    - traffic_manager and vip_config executables are setuid root
- The installer automatically creates a user account for you during installation:
  - This account is a non-privileged “inktomi” user account
  - Used for the Traffic Server daemon, traffic_manager and traffic_cop processes
- Installation is a two-step process
  - Prepare the target node
  - Install the Traffic Server software on the node

Preparing the Target Node

- **Prior to installation:**
  - Verify your host system meets the minimum system requirements. *Multiple node clusters must be configured identically.*
  - Ensure you have a default backup partition that spans the disk (or re-partition using defaults)
  - Assign primary IP addresses for Traffic Server nodes
  - Select virtual IP addresses for dynamic assignment if desired
    - Virtual IP addresses cannot include primary IP addresses
    - Primary IP addresses need not be externally accessible if mapping Virtual IP

Minimum System Requirements

- **Traffic Server supports Solaris SPARC or Digital Alpha/OSF (Future release will include Silicon Graphics IRIX and NT)**

Computer server	*Sun Ultra SPARC with 256 MB RAM*	*Digital Alpha/OSF with 256 MB RAM*
Operating System	*Solaris 2.6, Solaris 2.6 Patch Cluster from SunSolve (Downloadable from Sun)*	*Digital UNIX 4.0D*
Minimum Disk Space	*6-8 disks formatted in raw disk partitions*	*6-8 disks formatted in raw disk partitions*
Network Interface	*100 MB Ethernet or FDDI*	*100 MB Ethernet or FDDI*
Additional Software		*Digital-supplied AdvFS patch (Downloadable from Digital)*

Formatting Disks for Cache

- The Traffic Server stores its cache on raw disk partitions
  - Provides optimum performance
  - Disks can be sized between 2 GB and 16 GB
  - Typically uses the default backup partition that spans the entire disk
    - You should not need to format your disks unless you have reformatted the disk
    - If you have reformatted, you will need to re-partition using the disk's default parameters.
  - Installation will only use a disk with no filesystem mounted and no swap partition

Enabling DNS

- Enable DNS on the Traffic Server
  - DNS is required for any Traffic Server activity
  - Add at least one valid nameserver entry to /etc/resolv.conf
- Traffic Server's Use of DNS Round Robin in Resolution and Caching
  - Traffic Server recognizes DNS round robin in its own DNS cache and resolver
  - It follows the the same round robin rotation as it serves successive requests to the same web server name
  - Traffic Server establishes client affinity to server addresses to avoid authentication errors

Key Installation Steps

- Choose the installation target directory and specify logging path
- Set the Traffic Server name (use same name for all nodes if clustering)
- Configure Traffic Server to take advantage of network interfaces (cluster only)
- Set a multicast group address (cluster only) for simultaneous transmissions to Traffic Server nodes
- Decide if you will set up Transparency (automatic redirection)
- Decide if you will set up Traffic Server as a web server accelerator (reverse proxy)
- Assign ports for Traffic Server communications
- Specify an email address for the administrator
- Set username and password for Traffic Manager
- Configure Traffic Server Cache

Installing the Traffic Server

- You will need at least 100 MB of free space for the installation and another 100 MB of free space for the logging system
- You must be root to run the setup utility: `./install.sh`
- The setup script prompts you for target directories and configuration settings for networking and security
  - Disk location for files and logs
  - Port mappings
  - Email address, username & password for UI administrator
  - Cache disk drive information

Installing a Cluster

- Install one host at a time - configuring each node identically
- Enter the Traffic Server proxy name for the cluster (must be the same for all nodes)
- Properly configured nodes (same name and port settings) cluster automatically
- With Virtual IP enabled, available IP addresses will be divided among cluster nodes automatically
  - Use virtual IP addresses rather than actual physical addresses (define on Virtual IP page under Configure --> Server)
  - The vip_config program performs the VIP assignments
  - When nodes enter or leave the cluster, the IP addresses are redistributed

Installing Transparency

Inktomi

- Transparency (covered in our Solutions Workshop) makes it possible to automatically route user traffic directly to your Solaris Traffic Server
  - 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 (check with Tech Support for the latest list of compatible vendors)
    - Software solution requires the use of external software packages (included on your Traffic Server CD)

Exploring Installation Files

- The install script calls appropriate choices based on the operating system

```
# ls
alpha.tar      gated403.tar  setup.gd      suninst.tar   traffic.tar
decinst.tar    install.sh    solaris.tar   tools         transp.tar
```

Each tar file has all the information it needs to create Traffic Server for each platform

```
# tar tf solaris.tar
bin/
bin/mib2agt
bin/snmpdm
bin/traffic_server
bin/optimize/
bin/optimize/traffic_server
bin/optimize/traffic_cop
bin/optimize/traffic_manager
bin/optimize/traffic_line
bin/optimize/shmem_clean
bin/optimize/vip_config
bin/optimize/nntp_auth
bin/debug/
bin/debug/traffic_server
bin/debug/traffic_cop
bin/debug/traffic_manager
bin/debug/traffic_line
bin/debug/shmem_clean
bin/debug/vip_config
bin/debug/nntp_auth
bin/traffic_cop
bin/traffic_manager
bin/traffic_line
bin/shmem_clean
bin/vip_config
bin/nntp_auth
bin/traffic_cop
bin/traffic_manager
bin/traffic_line
bin/shmem_clean
bin/vip_config
bin/nntp_auth
bin/example_alarm_bin.sh
bin/example_prep.sh
bin/killnode
bin/start_traffic_server
bin/stop_traffic_server
bin/traffic_mom.tab
config/
config/snmpinfo.dat
config/mibs/
config/mibs/inktomi-ts-mib.my
config/mibs/inktomi-ts-mib.v1.my
config/mibs/inktomi-global-reg.v1.my
config/mibs/inktomi-global-reg.my
config/snmpd.cnf
config/mgr.cnf
ui/
ui/InkChart.class
ui/InkChart.jar
ui/WaitForParams.class
ui/dial.class
```

The Installation Script

Setting the Framework

```
# ./install.sh


################################################################################
#
#        Traffic Server 2.0 Installation
#        This script will install the Traffic Server cache
#          on system wolverine.
#
################################################################################


Please enter an account name for the Traffic Server: [inktomi]

Using account inktomi for Traffic Server install.

Enter the full path of the destination directory in which to
install Traffic Server: [/export/home/inktomi/TS_2.0]
>/export/home/inktomi/inktomi
/export/home/inktomi/inktomi does not already exist. Create it? y
Enter the full path of the directory in which to store
Traffic Server log files: [/export/home/inktomi/inktomi/logs]
>
/export/home/inktomi/inktomi/logs does not already exist. Create it? y
Is this installation part of a multi-machine Traffic Server cluster? n

Traffic Server port configuration
---------------------------------


 Will this server perform Reverse Proxy? [y/n]? n
```

```
Traffic Server makes use of 10 ports on your server.
Please enter the starting port number: [8080] 9000

The following port selections were made
---------------------------------------
1.  Traffic Server Proxy Port   9000
2.  Web  Administration port    9001
3.  Dynamic graphing port       9002
4.  Auto config port            9003
5.  Process manager port        9004
6.  Logging server port         9005
7.  Clustering port             9006
8.  Secondary clustering port   9007
9.  Reliable service port       9008
10. Multicast port              9009

***Verifying port assignment conflicts***

The port assignment check has found no conflicts

Enter the port assignment you would like to change (1-10)
"0" for no changes, "h" for help
> 0
```

Setting Ports

more...

Configuring Transparency & Cache

Inktomi

```
The Traffic Server transparency option enables Traffic Server
to recognize and respond to user HTTP traffic, redirecting user
web requests transparently through the cache without requiring
the users to reconfigure their browser settings for a proxy
configuration. Transparency uses an IP-Filter driver package.
Fully installing and enabling transparent proxying requires proper
network configuration.  Please see the Traffic Server Transparency
Configuration Guide for a detailed description of the transparency
options and configuration requirements.

If you are unsure whether you need transparency, do *not* install
the IP-Filter package.

Would you like to enable transparency and install the IP-Filter package? n

Traffic manager administration information:
Please enter an e-mail address for Traffic Server alarm notification: [inktomi]
>student@inktomi.com
Using notification email address student@inktomi.com

Please enter the Traffic Server admin user name. This name is not
a Unix user account name, and is only for the Traffic Manager web-based
administration program: [admin]
>
Traffic Manager administrator name admin
Please enter the Traffic Server admin password:
>
Please enter the Traffic Server admin password again
>
```

Transparency not installed

Choose from disks available

```
Checking available space for cache
Any disk that includes a mounted file system or swap partition is not
available for use as cache storage. Only disk drives not used for any
other purpose will be listed for cache selection. The system vendor's
"backup" partition normally spans the entire disk drive, and will
be used to identify drives for cache storage.
Ready to configure disk space to be used for Traffic Server cache.
Select which of your available disk resources should be used for cache.
Remember that space used for cache cannot be shared with any other use.

Here is the list of available disk drives :
(1) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@2,0:c,raw
(2) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@3,0:c,raw
(3) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@4,0:c,raw
(4) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@5,0:c,raw
(5) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@8,0:c,raw
(6) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@9,0:c,raw

Please choose one of the following options:
(l) LIST   LIST current cache storage selections.
(a) ADD    ADD a cache storage selection.
(r) REMOVE REMOVE a cache storage selection.
(s) SELECT SELECT ALL cache storage selections.
(d) DONE   DONE with selection, continue Traffic Server installation.
(q) QUIT   QUIT from Traffic Server installation now.

OPTION: a

[ ] (1) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@2,0:c,raw
[ ] (2) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@3,0:c,raw
[ ] (3) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@4,0:c,raw
[ ] (4) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@5,0:c,raw
[ ] (5) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@8,0:c,raw
[ ] (6) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@9,0:c,raw
```

Selecting Disk Drives for Cache

```
CHOICE to add: 1

Here is an updated list of your choice of disk drives:

[X] (1) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@2,0:c,raw
[ ] (2) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@3,0:c,raw
[ ] (3) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@4,0:c,raw
[ ] (4) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@5,0:c,raw
[ ] (5) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@8,0:c,raw
[ ] (6) /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@9,0:c,raw

Please choose one of the following options:
(l) LIST   LIST current cache storage selections.
(a) ADD    ADD a cache storage selection.
(r) REMOVE REMOVE a cache storage selection.
(s) SELECT SELECT ALL cache storage selections.
(d) DONE   DONE with selection, continue Traffic Server installation.
(q) QUIT   QUIT from Traffic Server installation now.

OPTION: d
You are quitting from disk drive selection.

Here is the final choices of disk drives for your cache storage configuration:
/devices/sbus@1f,0/SUNW,fas@e,8800000/sd@2,0:c,raw

Configuration for cache storage is done.
```

```
Installing Traffic Server 2.0 files to /export/home/inktomi/inktomi

/devices/sbus@1f,0/SUNW,fas@e,8800000/sd@2,0:c,raw cache partition
ln: /etc/rc3.d/S25snmpd and /etc/init.d/S25snmpd are identical
Configuring Traffic Server cache. This may take a few minutes.
Do not interrupt cache configuration or you will have an unusable cache.
CLEAR

Clearing Configuration
Clearing Host Database
Clearing Cache

RECONFIGURE, succeeded
Traffic Server 2.0 installation complete.
Please reboot this system before starting Traffic Server.
To start Traffic Server, login as inktomi and enter the command
        start_traffic_server
A log file of this installation process has been written to
/export/home/inktomi/TSinstall.log
Please consult the Traffic Server User's Guide for full operating information.
#
```

- Completes a standard install
- Must reboot the system
- **`TSInstall.Log`** captures responses

After Installation

Inktomi

Bin holds executable programs

Config holds site configuration files

UI holds HTML documents and images for the Traffic Server

```
$ pwd
/export/home/inktomi/inktomi
$ ls
bin         config      diags.log   logs        ui
$ ls bin
config                  mib2agt                 start_traffic_server   traffic_mom.tab
debug                   nntp_auth               stop_traffic_server    traffic_server
example_alarm_bin.sh    optimize                traffic_cop            vip_config
example_prep.sh         shmem_clean             traffic_line
killnode                snmpdm                  traffic_manager
$ ls config
cache.config            ip_allow.config         mibs                   public_key.der         socks.config
cluster.config          lm.config               nntp_access.config     records.config         storage.config
filter.config           logs.config             nntp_servers.config    remap.config           vaddrs.config
icp.config              mgmt_allow.config       parent.config          snmpd.cnf
internal                mgr.cnf                 proxy.pac              snmpinfo.dat
$ ls logs
$ ls ui
InkChart.class          c_routing_on.gif        logging.config.ink     pac_missing.html
InkChart.jar            c_security_off.gif      logging.files.ink      protocols.config.ink
WaitForParams.class     c_security_on.gif       logo.html              protocols.stats.ink
about.jpeg              c_servbasics_off.gif    m_cache_off.gif        remap.files.ink
about_ts.ink            c_servbasics_on.gif     m_cache_on.gif         restart.gif
alarm_off.gif           c_snapshot_off.gif      m_dash_off.gif         restart_msg.html
alarm_on.gif            c_snapshot_on.gif       m_dash_on.gif          routing.config.ink
alarm_warning.gif       cache.config.ink        m_graphs_off.gif       rsa_logo.gif
alerts.gif              cache.stats.ink         m_graphs_on.gif        security.config.ink
autoconf_add.html       cache_results.pie.ink   m_help_off.gif         socks.conf.ink
back.gif                change_passwd.html      m_help_on.gif          stop.gif
background.gif          change_passwd_g.html    m_node_off.gif         storage.files.ink
ball.gif                configure_on.gif        m_node_on.gif          stripChart.start
bb                      dashboard.stats.ink     m_os_off.gif           switch_off.gif
blank.html              dial.class              m_os_on.gif            switch_on.gif
c_cache_off.gif         forward.gif             m_other_off.gif        system.config.ink
c_cache_on.gif          gc_results.pie.ink      m_other_on.gif         traffic.gif
c_help_off.gif          graphGen.html           m_protocols_off.gif    traffic_bg.jpeg
c_help_on.gif           graphs_select.html      m_protocols_on.gif     trate.pie.ink
c_hostdb_off.gif        help                    main.ink               ts-index-bkg.gif
c_hostdb_on.gif         hostdb.config.ink       misc.stats.ink         vmap.files.ink
c_logging_off.gif       icp.conf.ink            monitor_on.gif         vu.gif
c_logging_on.gif        images                  netcharts              warning.gif
c_protocols_off.gif     index.ink               object_size.pie.ink    warning_big.gif
c_protocols_on.gif      ink_logo.gif            off_button.gif
c_routing_off.gif       ink_logo_trans.gif      on_button.gif
```

Starting the Traffic Server

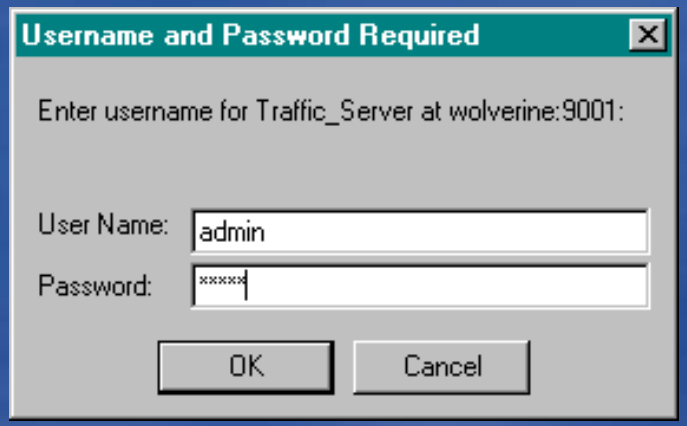
- When you have successfully installed the Traffic Server software on all the nodes in your cluster and have rebooted, you are ready to start the server
- There is a startup script in the bin directory which directs a Traffic Cop process to initiate the chain of interdependent Traffic Server processes that start and run the system

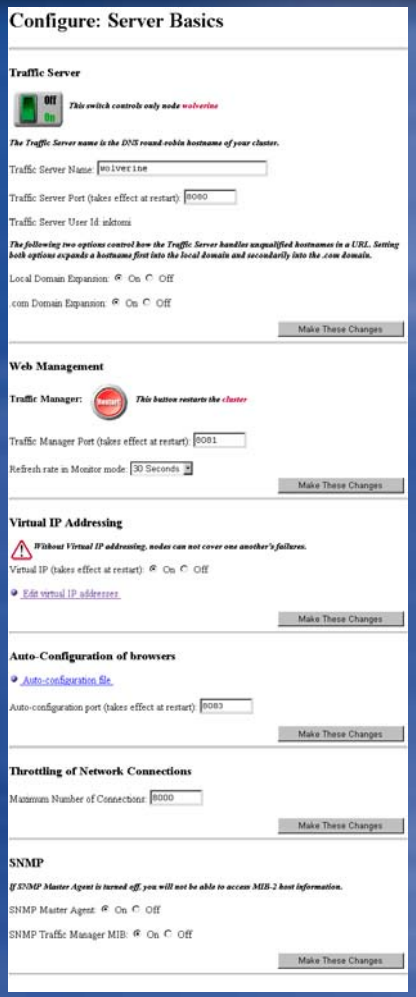
```
$ cd /etc
$ more traffic_server
/export/home/inktomi/inktomi
$ pwd
/etc
$ cd /export/home/inktomi/inktomi/bin
$ start_traffic_server
Started Traffic Server
```

```
$ ps -ef | grep traff
  inktomi 18568     1  0 12:04:39 pts/12    0:00 ./traffic_cop
  inktomi 18570 18568  0 12:04:40 pts/12    0:00 bin/traffic_manager
  inktomi 18577 18570  0 12:04:43 pts/12    0:01 bin/traffic_server -M -A8:X
$
```

Verifying Installation

- The best test of a successful installation is to point at the Traffic Manager port to review configuration and monitor results:
  - http://news.inktomi.com:8081
  - https://news.inktomi.com:8081 (secure)





Class Tools: Populating the Cache

- Special class tools are available for populating the cache from the command line to allow you to:
  - Test your Traffic Server
  - See impact on cache as content increases
  - Monitor and analyze logs
- You are welcome to take these tools with you for use at your site

Practice Lab

- Please complete the Unit 2 lab detailed in your Student Workbook
  - Review the Pre-Installation Worksheet
  - Install the Traffic Server
  - Review the Application Environment
  - Use Class Tools to Populate the Cache
    - Monitor Activities
    - Analyze Log Files

Configuring the Traffic Server

- The Traffic Server Processes
- Exploring Configuration Options
  - Server Basics
  - Protocols
  - Security
  - Routing
  - Host Database
  - Logging
  - Snapshots
- Practice Lab

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 (cached or from 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 Traffic Server'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

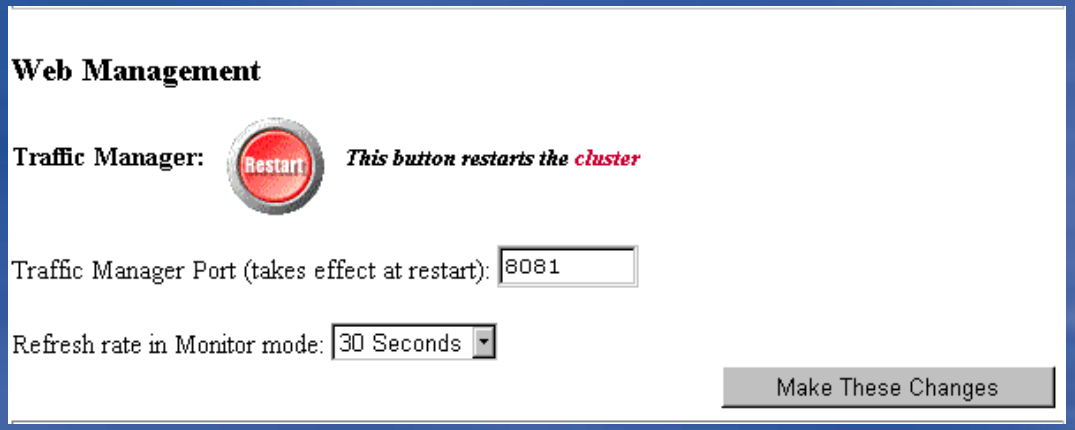
```
Oct 25 03:30:00 wolverine traffic_cop[1166]:  Cop Starting - Version:
traffic_cop 2.0.0e - (build # 92219 on Oct 22 1998 at 19:19:47)
Oct 25 03:30:00 wolverine traffic_cop[1166]:  Periodic heartbeat
successful, another cop still on duty
```

Server Basics: The General Options

- Shutting down the server stops all caching and proxying services on a specific node
- Server name is the proxy name
- Proxy port must be dedicated to Traffic Server (default is 8080)
- User ID is for the Traffic Server's proxy process (default is inktomi)
- Turn on auto-expansion to have the Traffic Server automatically preface host names with www. and suffix them with .com

Web Management Options

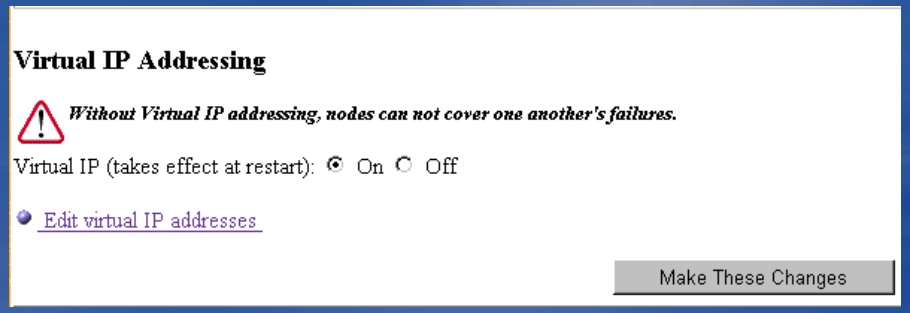
- Restart is used to effect changes to port numbers and virtual IP address numbers
  - Takes about 15 seconds,
  - Disables all caching and proxying across the entire cluster
- Traffic Manager Port is for the Administrator's browser connection (default is 8081)
- Refresh rate governs how often displays of graphics and statistics will be updated for you to monitor performance



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
- You can edit your virtual IP list from this page

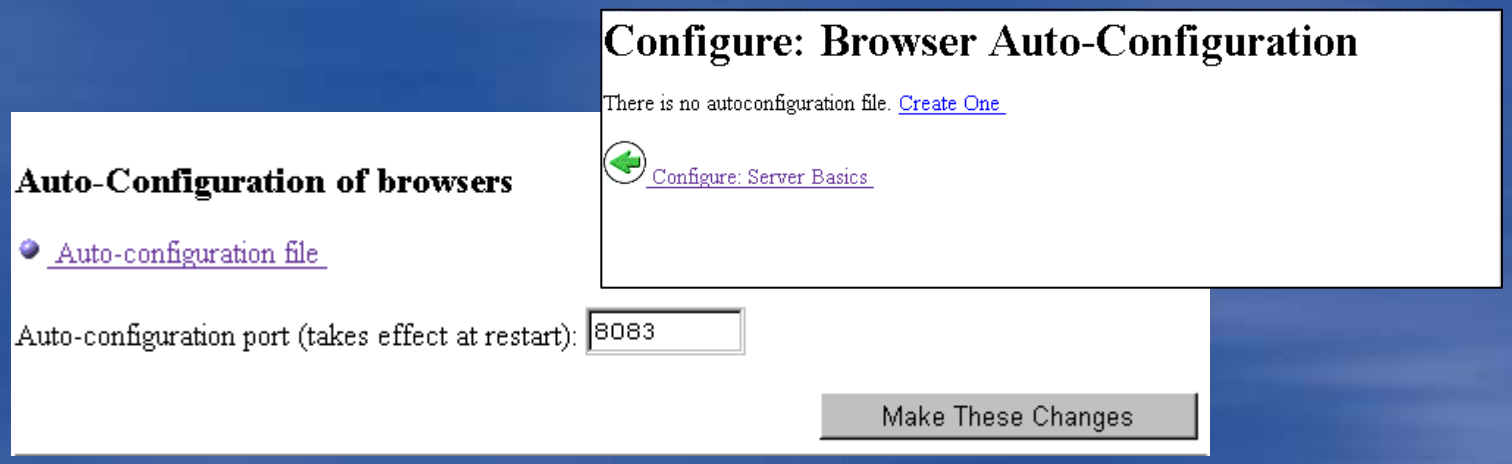
caution: incorrect IP addressing can effectively disable your system.



be sure you understand how these IP addresses work before changing them!

Setting 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
- Users will need to set their browsers to connect to your Traffic Server auto-configuration file as appropriate for each browser

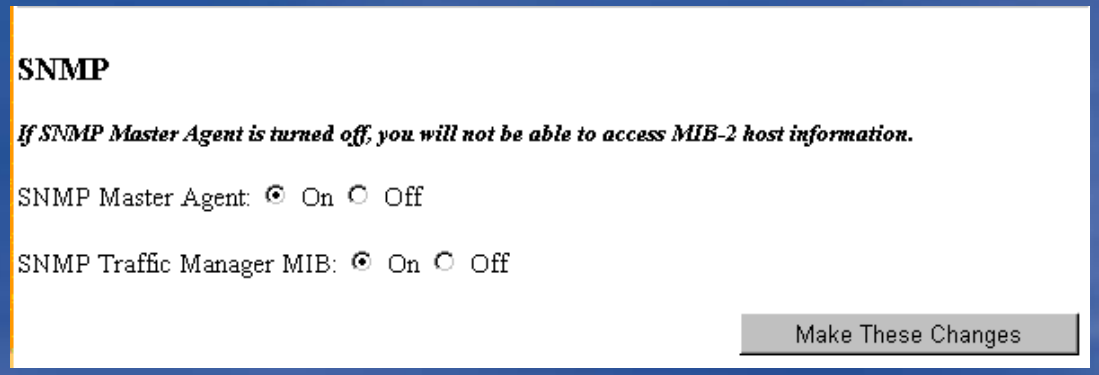


Throttling Network Connections

- The Traffic Server can restrict the number of network connections it will accept to prevent system overload if a traffic bottleneck develops

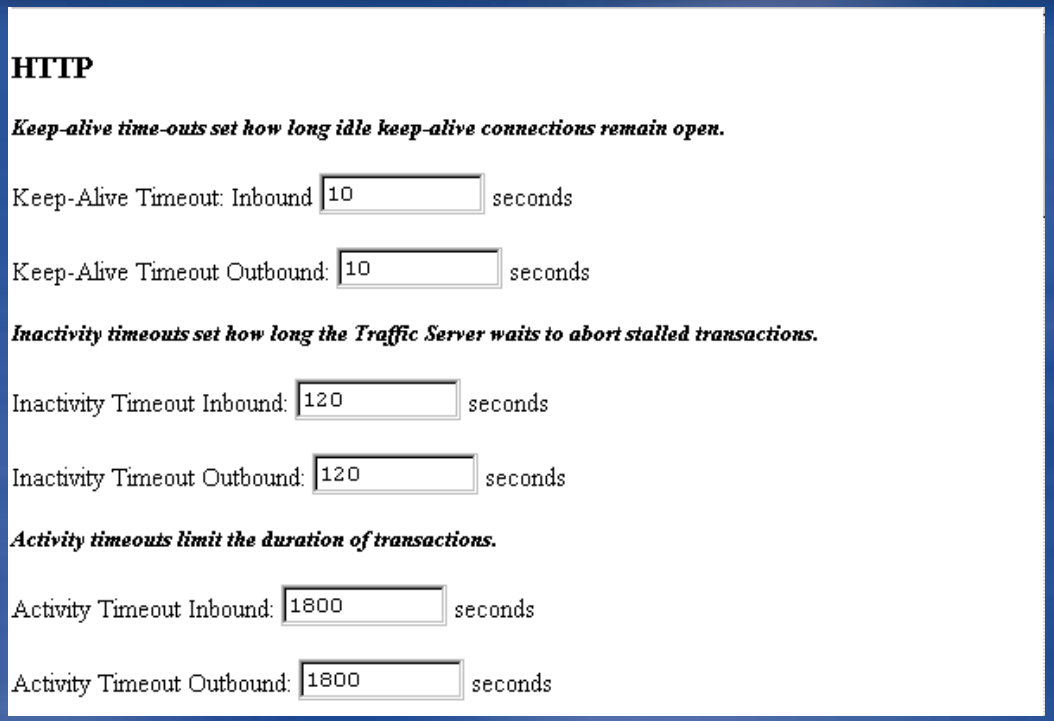
Configure SNMP

- Traffic Server supports SNMP
  - View performance information about the Traffic Server
  - Warning messages (SNMP traps) to SNMP monitoring stations
    - Two management information bases (MIBs)
      - Master Agent is MIB-2 (standard MIB)
      - Inktomi Traffic Server MIB (contains node-specific and cluster-wide information)



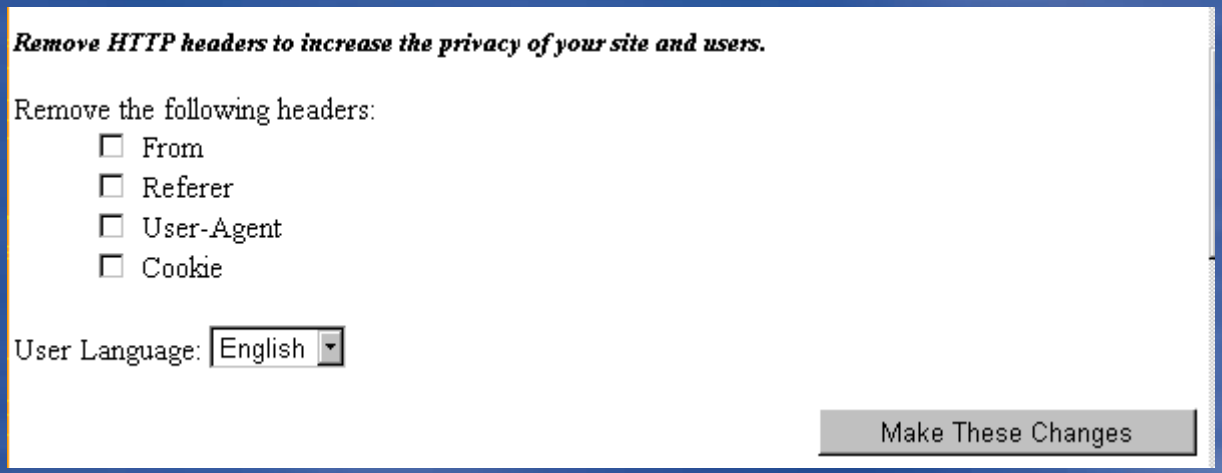
The Protocols Page

- This page allows you to tune HTTP and FTP timeouts and set user privacy features
  - Keep-alive timeouts (holding a connection open for a subsequent request)
  - Inactivity timeouts (holding connections open if a transaction stalls)
  - Inbound (connections to users)
  - Outbound (connections to servers)



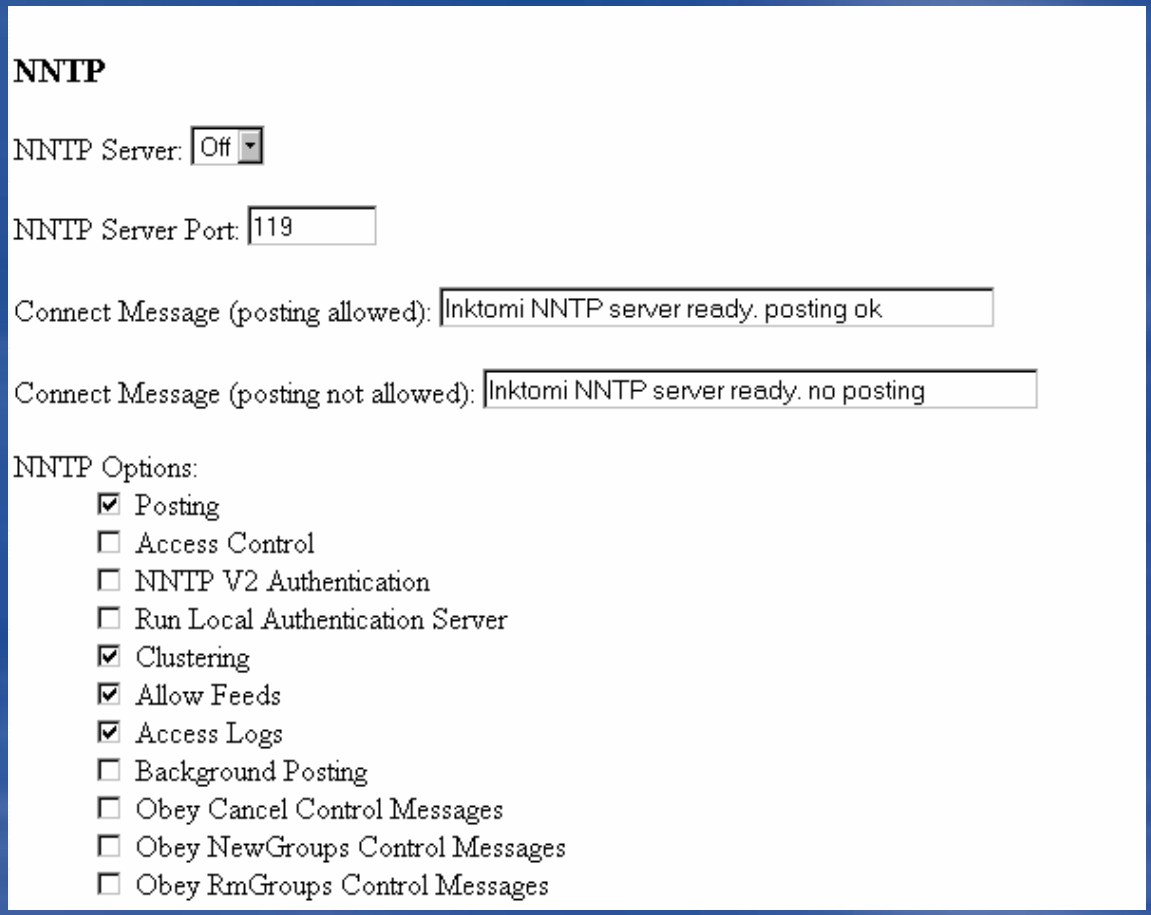
Configuring Privacy Options

- 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)



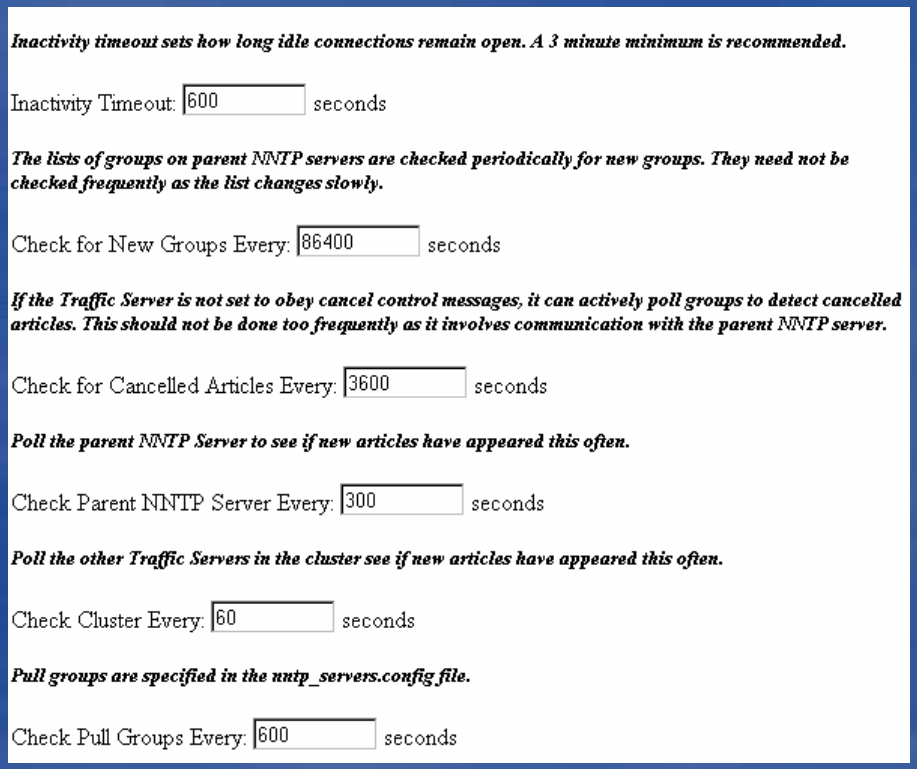
Configuring NNTP

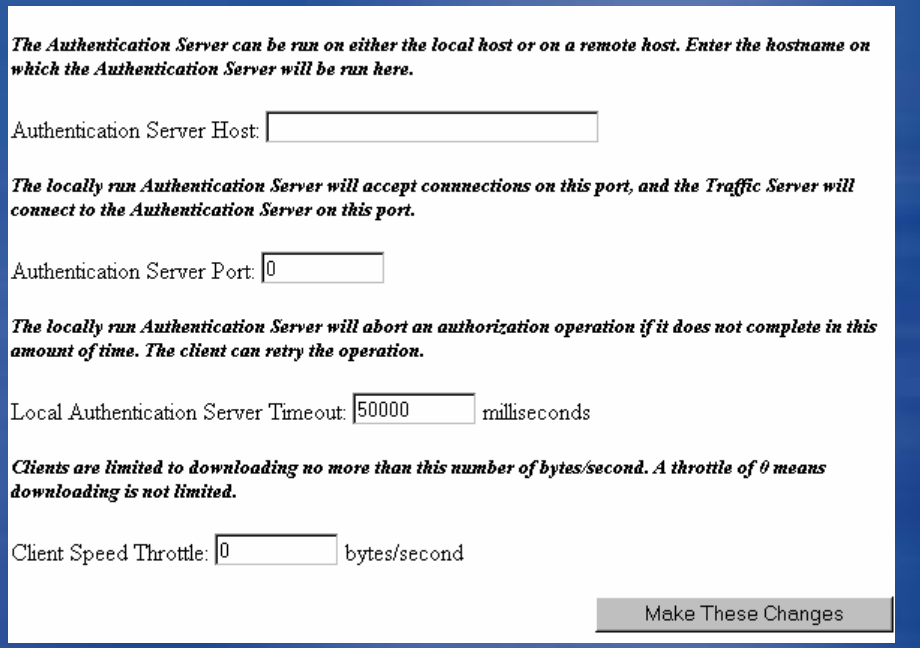
- Enable Traffic Server to cache and serve news articles by turning NNTP server on or off
- Caution: you must click the Restart button to affect this change
- This page allows you to configure basic NNTP options



Configuring NNTP Polling

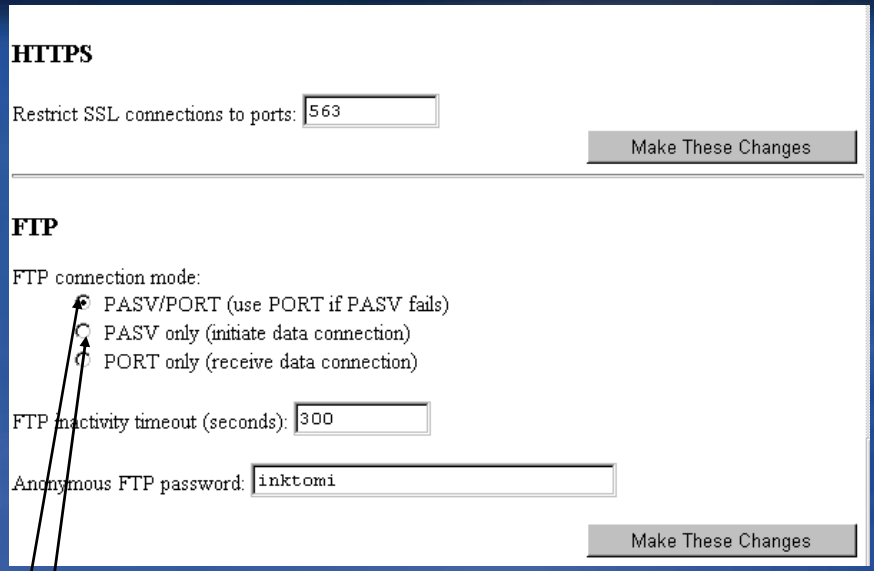
- This page allows you to configure other NNTP options, like inactivity timeout, polling and authentication server





Configuring HTTPS and FTP

- Use the HTTPS setting to restrict SSL connections to certain ports
- FTP requires two connections
  - A control connection informs the FTP server of the request (always initiated by the Traffic Server)
  - A data connection sends the data (can be initiated by Traffic Server or FTP based on your settings)
    - PASV/PORT indicates try Traffic Server (firewall friendly) but allow FTP to initiate if not supported

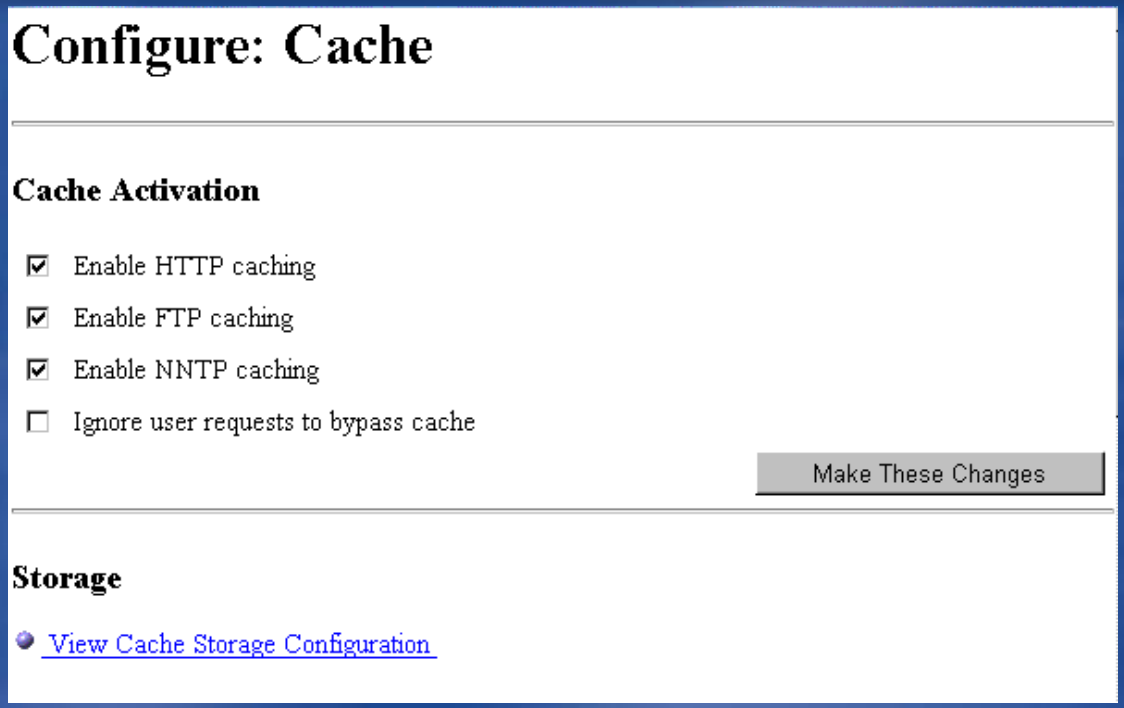


PASV only: Traffic Server initiates and FTP accepts it

PORT only: FTP initiates and the Traffic Server accepts it

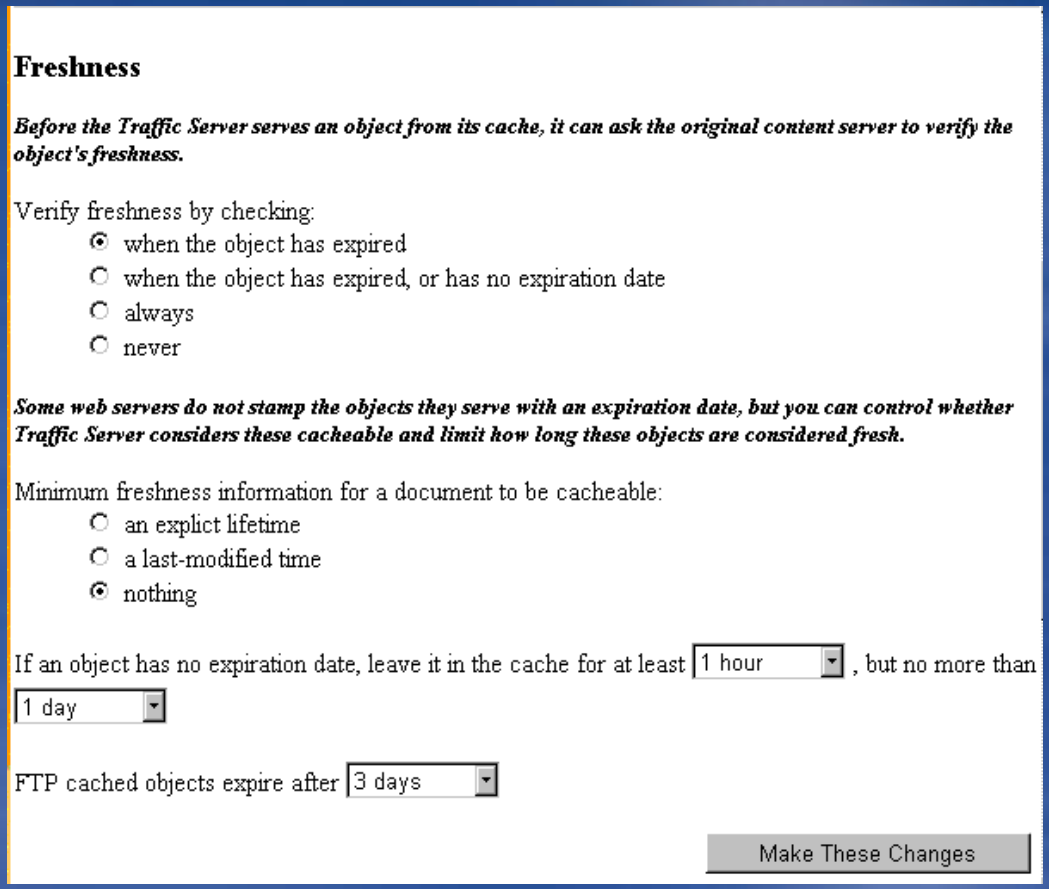
Using the Cache Page

- This page allows you to configure how caching will be handled:
  - Cache activation
    - What you will cache
    - What to do when users want to bypass using the cache



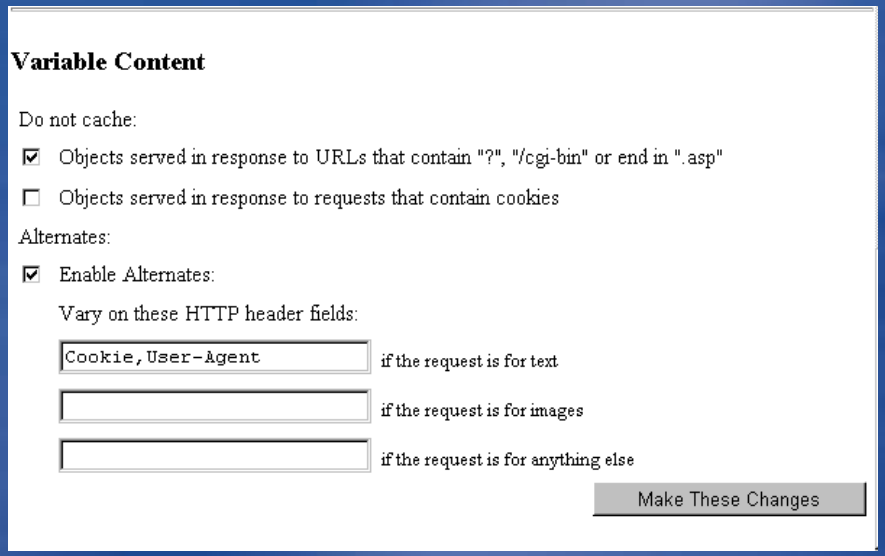
Configuring Object Freshness

- "Freshness" settings tell the Traffic Server how to handle verification with the origin server
- Set minimum freshness for objects with no expiration (from 15 minutes to 2 weeks)
- Set expiration on FTP objects (which carry no time stamp or date information)



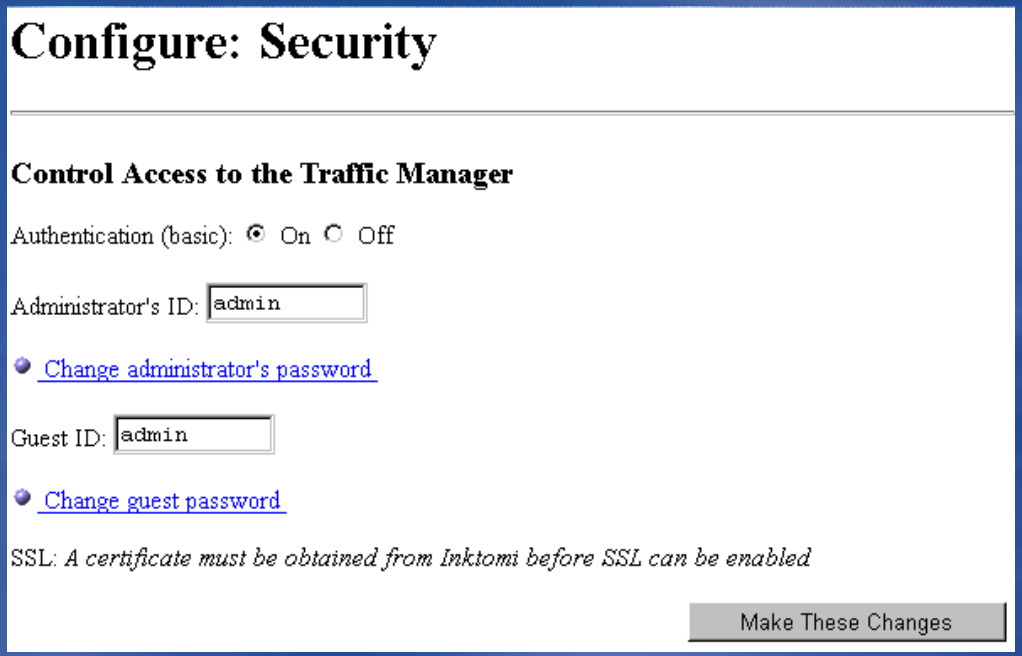
Handling Variable Content

- Web servers may answer requests to the same URL with a variety of objects
  - Different languages
  - Different browsers with different presentation styles
  - Variable content at different times of the day
- You can set options for preventing caching of:
  - Objects containing ? or /cgi-bin
  - Objects that contain cookies



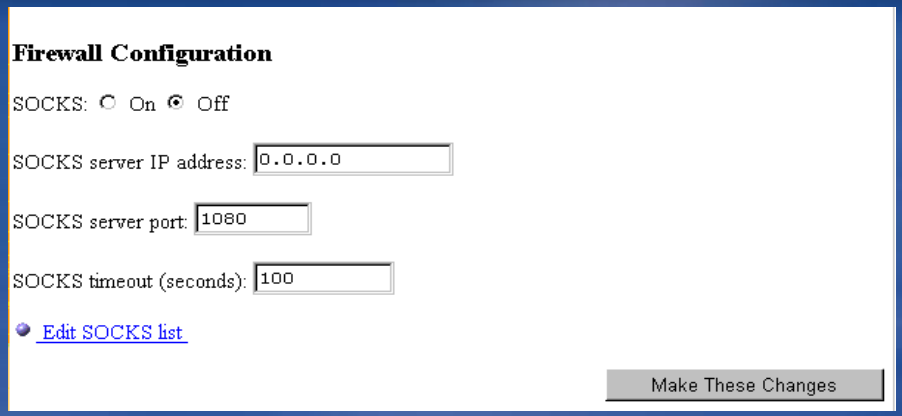
Traffic Manager Access

- The security page allows you to
  - Control access to the Traffic Manager
    - Authentication
    - Administrator ID
    - Administrator Password
  - Allows for a “guest” ID (static for all guests)
    - Monitor-only access



Firewall Integration

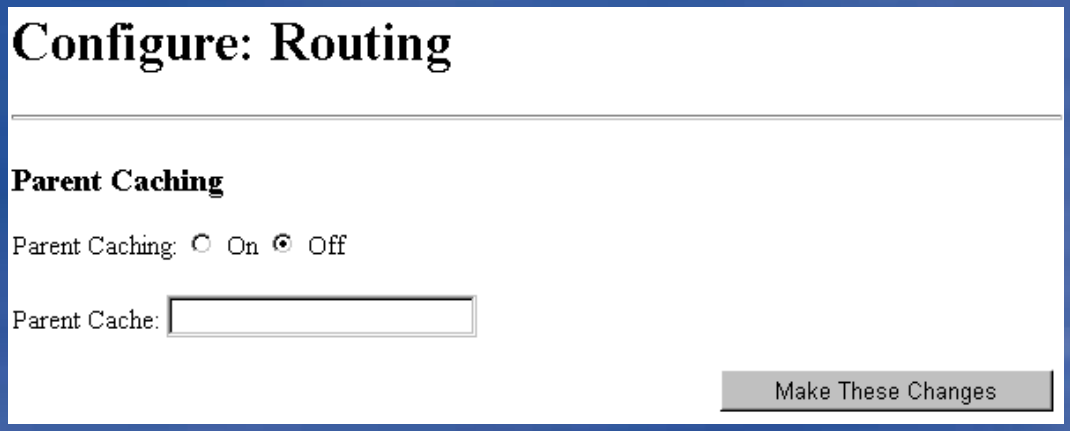
- If the Traffic Server is outside of your firewall leave the SOCKS flag off (default)
- If your Traffic Server is inside the firewall turn the SOCKS flag on and provide:
  - IP address of SOCKS server
  - The port for Traffic Server to connect to the SOCKS server
  - Edit your SOCKS list for modifying IP addresses



Unidentified machines are assumed to be outside the firewall

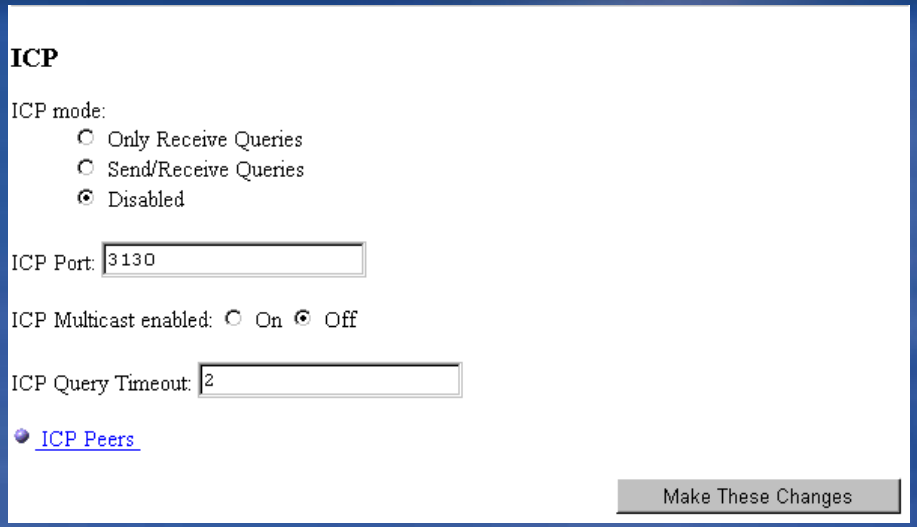
Enabling Parent Caching

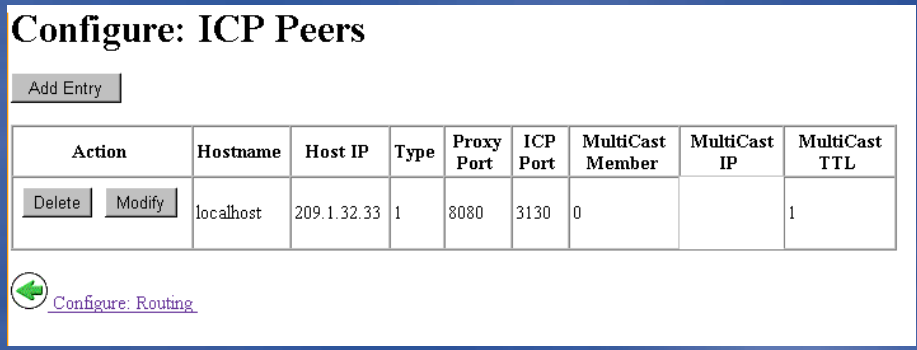
- You can point your Traffic Server at another Traffic Server (or a different caching product) to form a hierarchy to search for requested objects
- If the object is not found in the local cache, the next check is against the parent cache



Enabling ICP Caching

- Traffic Server supports Internet Cache Protocol (ICP)
  - Allows specific proxy caches to exchange information about their content (replies “hit” or “miss”
  - Specify ICP peers
- Checks (in order)
  - Traffic Server cache
  - Sibling ICP caches
  - Parent ICP caches
  - Parent Traffic Server caches
  - Origin server

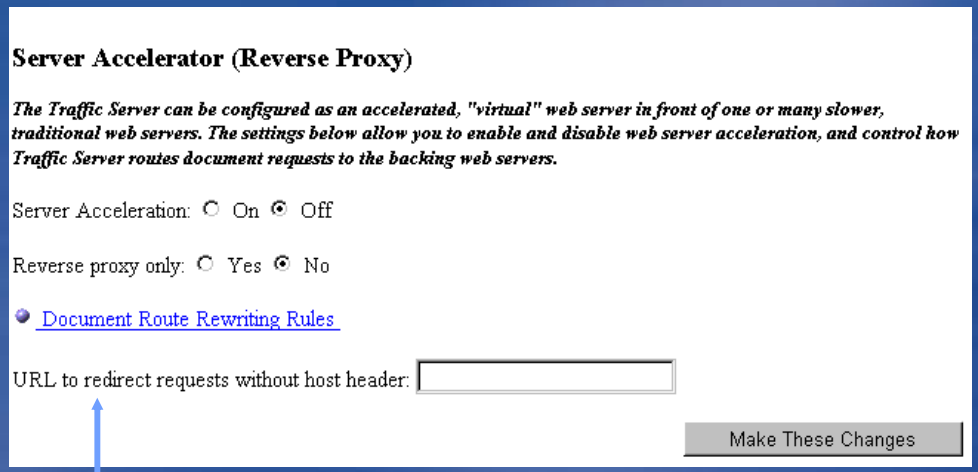




Server Accelerator Options

- Reverse proxy allows Traffic Server to proxy for your web server (become your web server)
  - Much faster than most web servers can respond
  - Balances load of web servers
  - Centralizes administration
- Traffic Server intercepts server requests from clients (DNS for origin server resolves to Traffic Server)
  - Path only
  - Routing rules clarify where to look

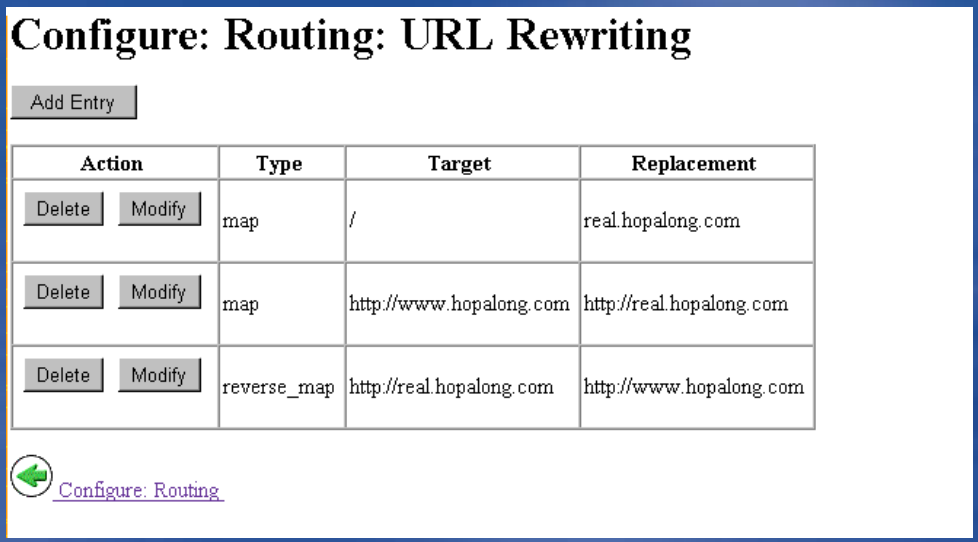
You define routing rules for Traffic Server to refer full paths



Traffic Server cannot route URLs from older browsers that do not use a Host: header. Routing rules can define default mapping rules to handle this. If no default rule is provided, this URL would simply explain the situation and request user upgrade their browser.

Server Accelerator Options

- Routing rules include:
  - Rule type (map for user requests or reverse map for origin responses)
  - Target URL (“from” URL)
  - Replacement URL
- Client’s URL requests are compared against the target URLs in map rewriting rules
  - Hosts must be the same
  - Ports must be the same
  - Path of the target URL must be the same as prefix of requesting URL



Configure: Routing: URL Rewriting

Add Entry

Action	Type	Target	Replacement
Delete Modify	map	/	real.hopalong.com
Delete Modify	map	http://www.hopalong.com	http://real.hopalong.com
Delete Modify	reverse_map	http://real.hopalong.com	http://www.hopalong.com

Configure: Routing

Web Server Redirects

- Traffic Server uses reverse mappings to prevent redirects from origin servers to cause clients to bypass the Traffic Server
- 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*

Transparent Proxy Status

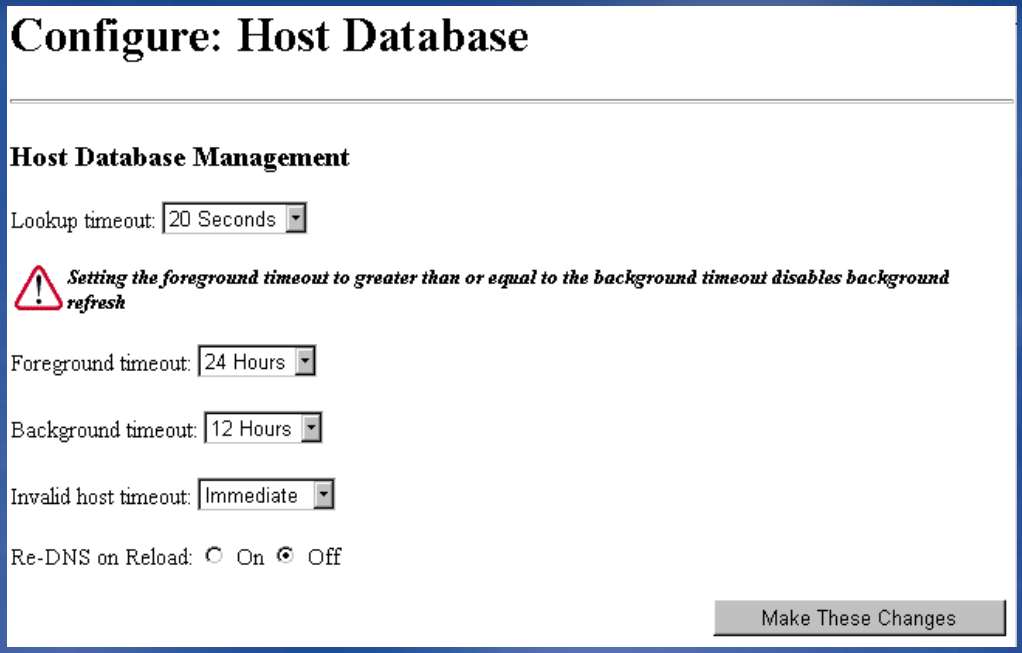
- Transparency allows Traffic Server to intercept and respond to port 80 requests without the user having to configure their browser (NNTP on 119)
  - Destination IP address is changed to Traffic Server (80 to 8080)
  - If in cache, serves request, changing IP back to origin server port
- There are three routing solutions
  - Layer 4-aware switch (most rapid switching)
  - Policy-based routing (router between Traffic Server and clients)
  - Software routing (uses Traffic Server as the router)
- Transparency is setup during installation, this simply shows status

Transparent Proxy

The Transparency option is installed. Redirected users will be served transparently.

Using the Host Database Page

- The host database stores DNS entries of servers that the Traffic Server contacts to fulfill user requests
- Settings determine how long DNS entries remain in the database (before they are flagged as stale and refreshed)
- You can set entries to refresh in background so they can be refreshed after they are served, rather than before

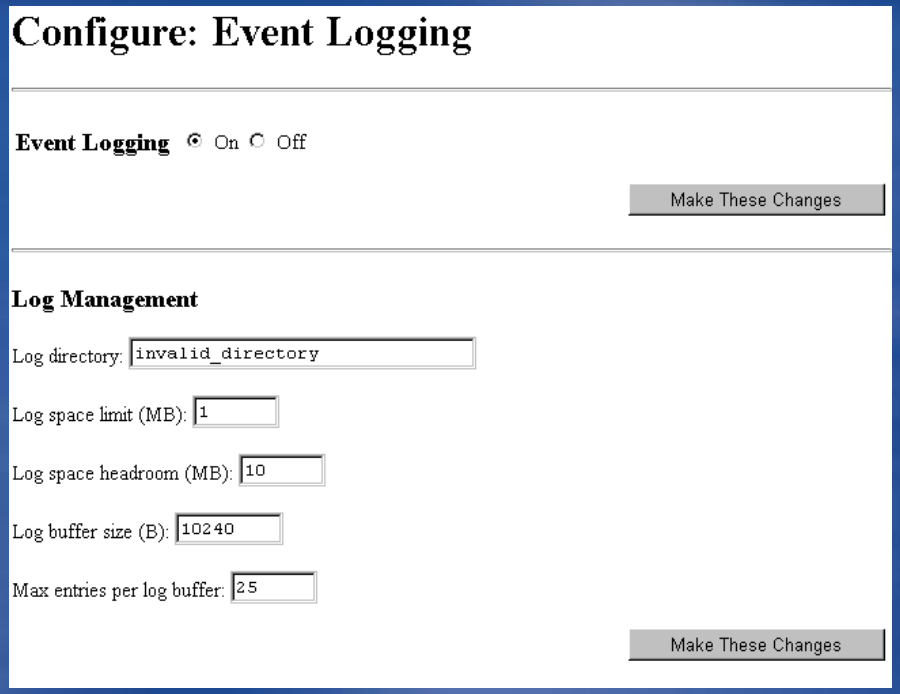


Configuring DNS

- To provide DNS services, the Traffic Server uses a list of DNS servers obtained from the DNS table in your resolv.conf file
  - Always tries to connect to the first server on this list
  - If unsuccessful, it moves to the next entry
- Specify how long the Traffic Server should wait for the DNS server to respond with an IP address
  - If user gives up the response will still be cached for subsequent use -- if it arrives within the time limit you set
- Specify how many times the Traffic Server should allow a look-up before it sends back an “invalid host name” message

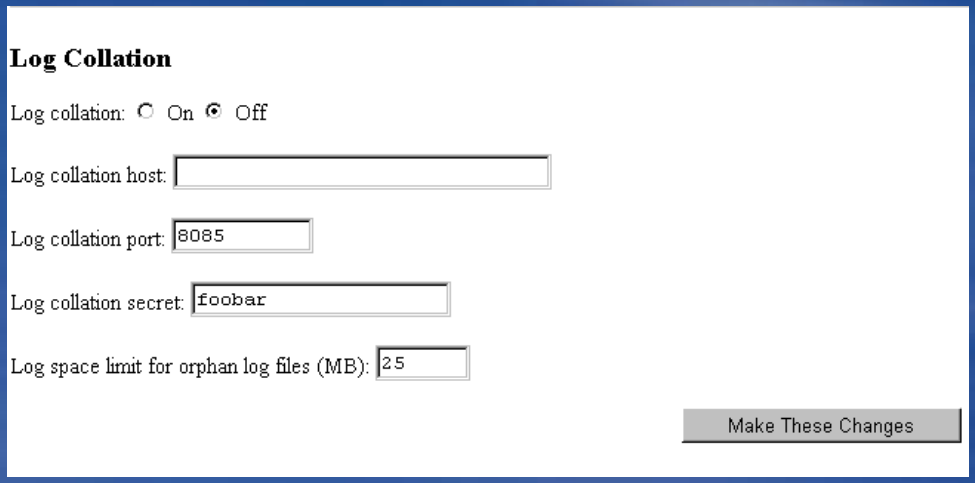
Using the Logging Page

- Choose a central location for storing and collating logs
  - How much disk space to allow for log files (make sure it's smaller than actual space available)
    - Default is 10 MB
    - Recommend more like 1 GB (per node)
  - Headroom is minimum space remaining to kick of deletion of oldest log files



Configuring Log Collation

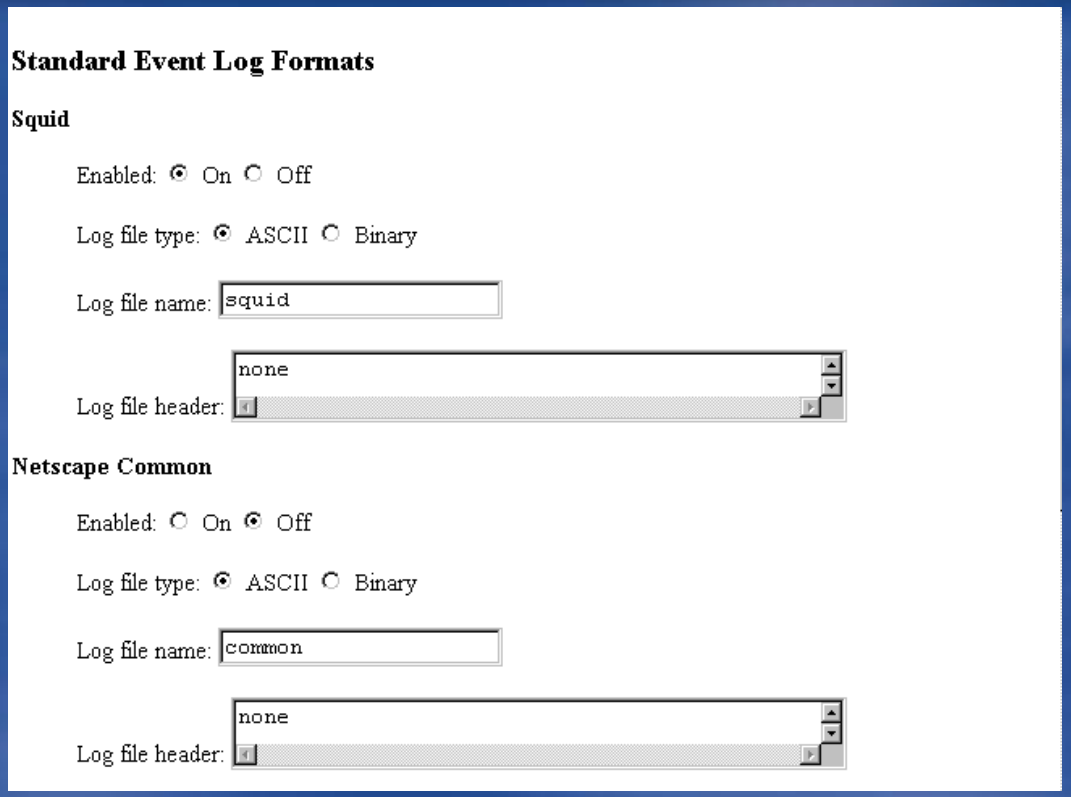
- Heavy activity consumes cluster bandwidth
- Brings all logs together when you establish a log collation server and port
  - If it can't connect for some reason, it writes individual "orphan" log files to local disks
  - You provide a name and port for this server (default is 8085)
- Specify a secret code (Log Secret) to prevent any process other than the Traffic Server from writing to the log directory



Must be a Traffic Server. Separate collation server is planned, but not yet in the product.

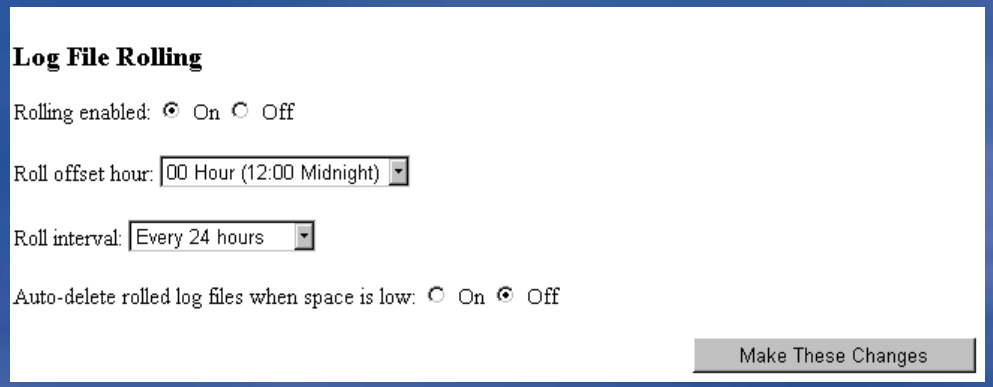
Log File Formats

- Choose the format and name log files
  - Squid Format
  - Netscape Common or Extended Format
  - Custom Format
- Samples are included in the workbook



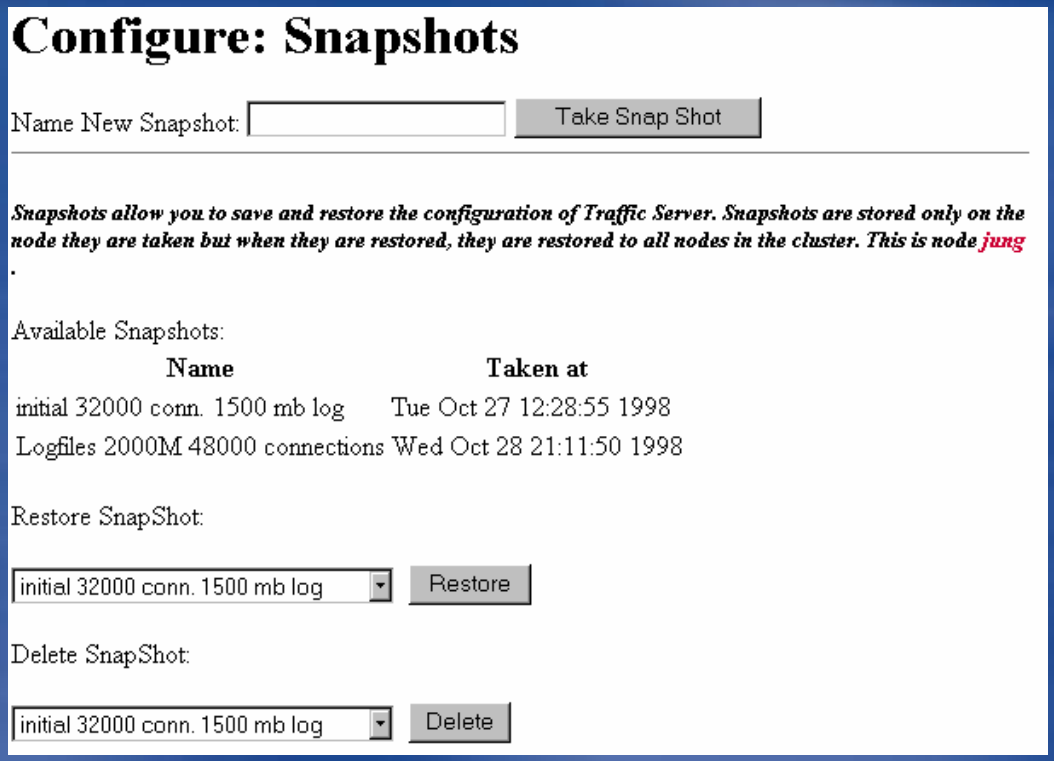
Configuring Log Rolling

- Set guidelines for rolling your log files
  - Roll interval indicates how often to roll or clear log files (default is 6 hours starting at midnight)
  - Set roll interval several times a day to ensure no single file becomes too large
  - Log files roll automatically on a server restart
- Auto-delete eliminates the oldest files when disk space is less than specified headroom



Using the Snapshots Page

- Snapshots represent the sum of all configuration settings at a particular place in time
  - Options allow you to name and take a Snapshot
  - Create a Snapshot before you make any changes or do system maintenance



Practice Lab

- Complete the Unit 3 Practice Lab
  - Review and re-configure the Traffic Server
    - Web Management
    - Transparency and Server Acceleration
    - Protocols and Security
    - Logging