

# Local and Wide-area Server Selection: Techniques and Challenges

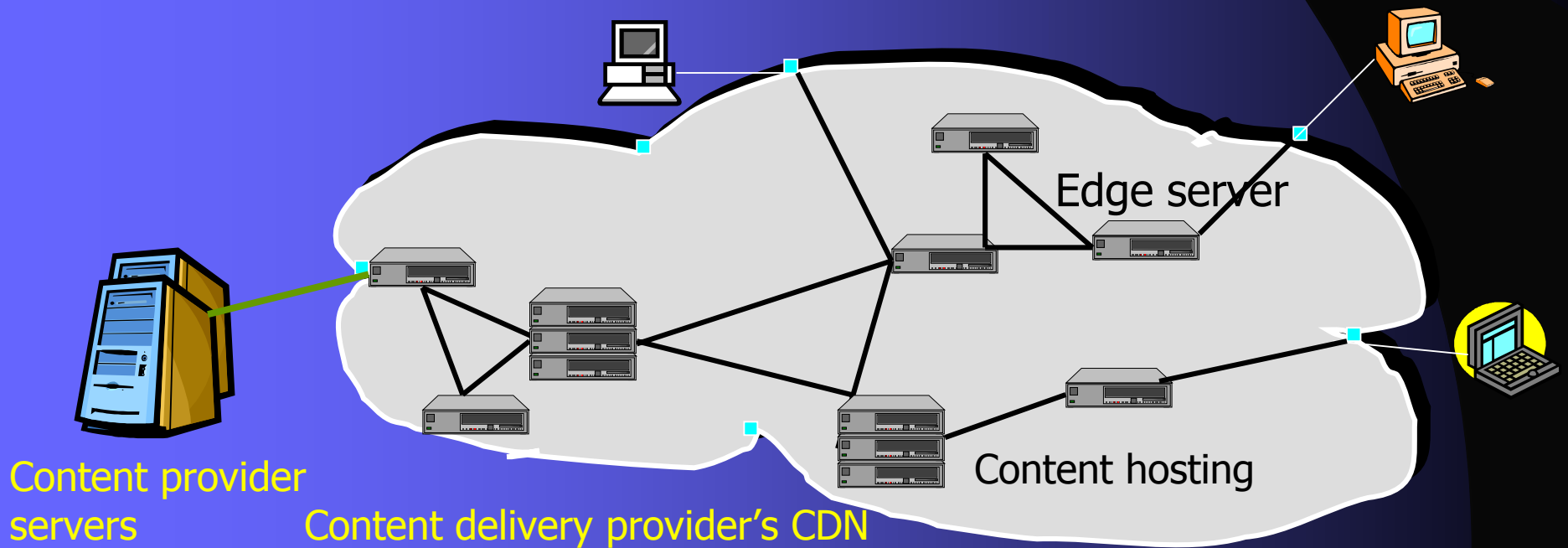
Arup Acharya, Anees Shaikh, Renu Tewari

*(arup, aashaikh, tewarir)@watson.ibm.com*

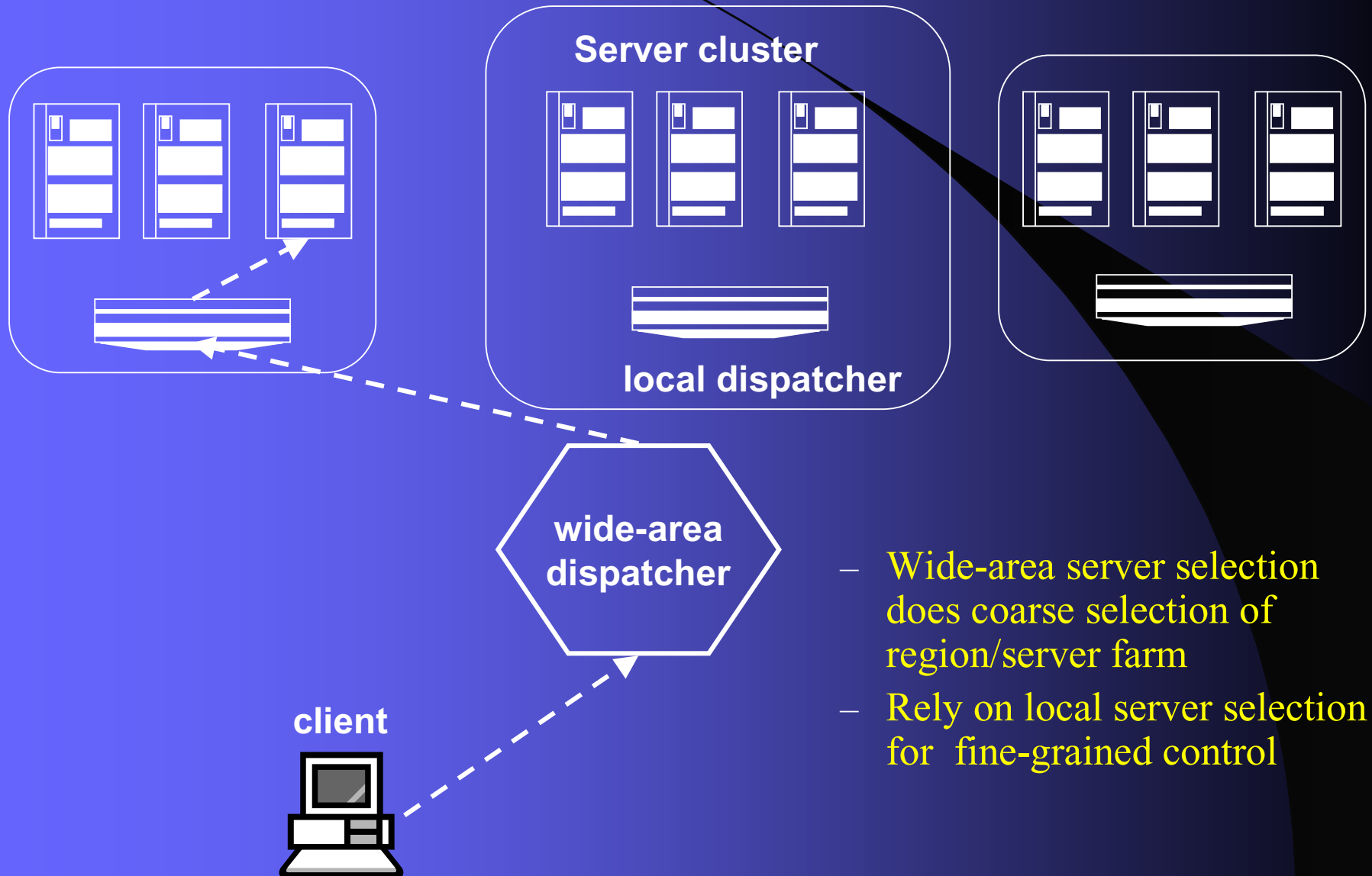
*IBM T.J. Watson Research Center*

# Trends

- 1 Content distribution providers distribute content to the network edges for better scalability, performance, QoS
- 1 Server clusters at large sites and web hosting service providers for better scalability and consolidation
- 1 Content everywhere but not a good way to find it
- 1 *Need to direct client to "best" content location*



# Two-level Server Selection



- Wide-area server selection does coarse selection of region/server farm
- Rely on local server selection for fine-grained control

# Talk Overview

- 1 Wide-area server selection
  - Techniques and metrics used
  - Issues in DNS-based server selection
  - Evaluation of DNS-based techniques
- 1 Local server selection
  - Overview and new challenges
  - MPLS-based local dispatching (a preview)
- 1 Conclusions

# Wide-area Server Selection

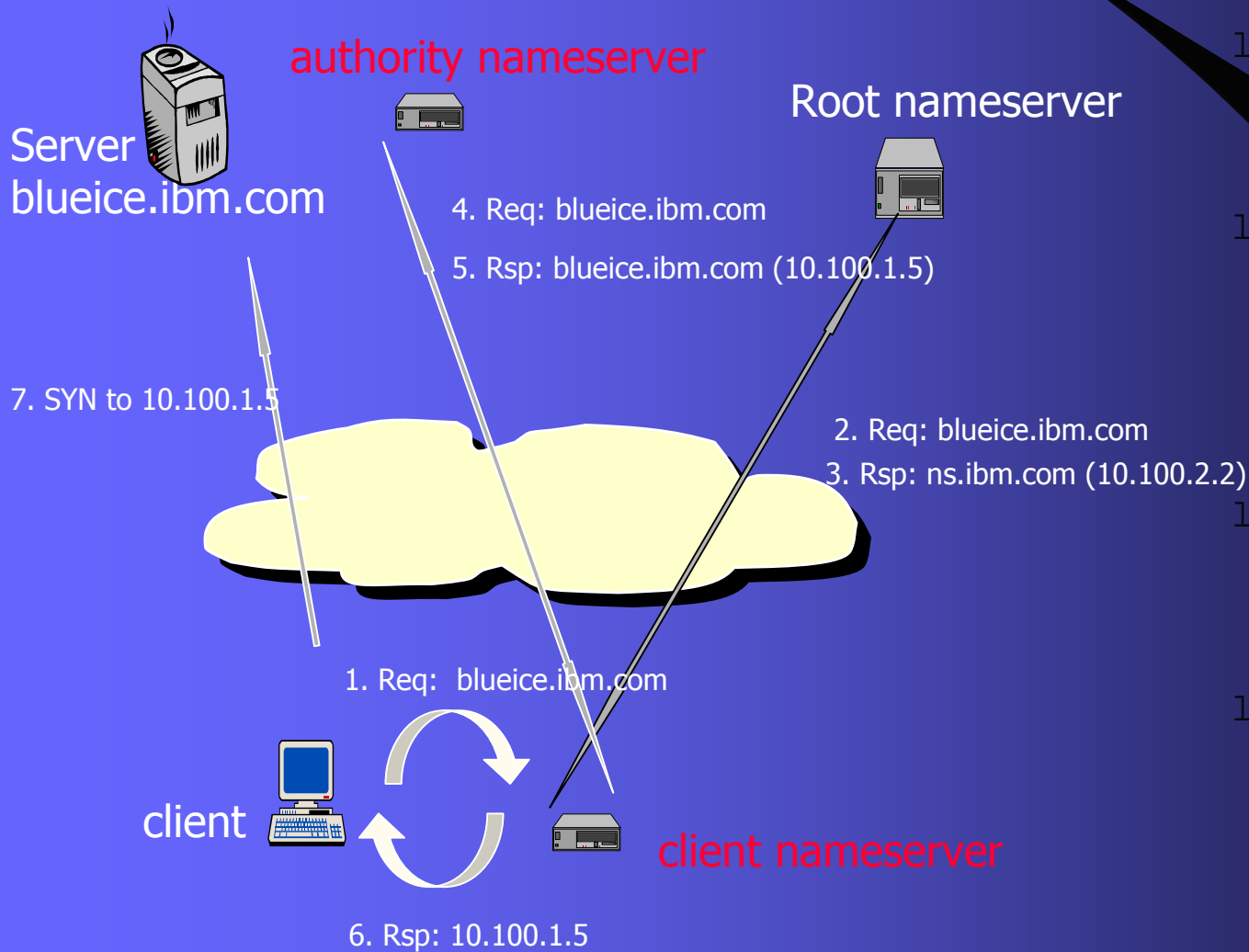
## 1 Metrics

- server load, network conditions
- client location
- QoS specification, content requested (images etc.)

## 1 Approaches

- **DNS based**
  - 1 nameserver dynamically maps names to addresses
  - 1 transparent, general, widely used
  - 1 Akamai, Cisco, F5 3DNS, Alteon, Digital Island ....
- HTTP redirect
- BGP-based (e.g., MSIPR)
- Application/IP layer anycast

# DNS Operation: Overview



- 1 Map names to IP addresses
- 1 Nameserver address configured statically or dynamically (DHCP, PPP)
- 1 Mapping is cached for TTL period
- 1 Remote DNS sees Client NS' Address

# Requirements and Challenges

- 1 Dynamic server selection
  - limit client-side DNS caching with low TTL values
  - effects of limiting DNS caching
    - End-user performance decreases (latency increase ~24%)
    - Scalability decreases (nameserver load and network load higher)
- 1 Location (or proximity) based server selection
  - need to identify the client
  - is the client nameserver representative of client location?
    - 1 client's local DNS mis-configured
    - 1 few nameservers across a large ISP
    - 1 nameserver in different AS domain
    - 1 clients and nameservers median cluster size ~8 hops

# Factors in End-user Latency

- 1 Name resolution latency
  - varies with level of address caching
    - 1 No cache, nameserver address, server address
- 1 Number of resolutions required
  - number of embedded objects (e.g., images)
  - location of objects (co-located)
  - HTTP version (1.1 with keep-alive)
- 1 Page size and transfer time

# Name Resolution Time

DNS Cache level	Median Resolution Time
No local DNS cache	200 ms
75 percentile (popular sites)	3 sec
85 percentile (popular sites)	5 sec
Cached authority nameserver IP	60 ms
Cached server IP	2.3 ms

## 1 Data sets (for hostnames)

- Proxy logs: medium-sized ISP, single pop
- Popular sites (Media Metrix Top 50)

## 1 Measured name resolution time from multiple sites

- Massachusetts, NY, Michigan, California

# End-user Latency

Page Statistics	Mean Value
Page download time (popular sites)	6.3 secs
Total Page size	30.9 KB
# of objects per page	35 (median 25)

Cache Level	Resolution Overhead
No caching (25 objects 200 ms. each)	5 sec.
NS address cached (25 objects. 60 ms)	1.5 sec.

## *Summary*

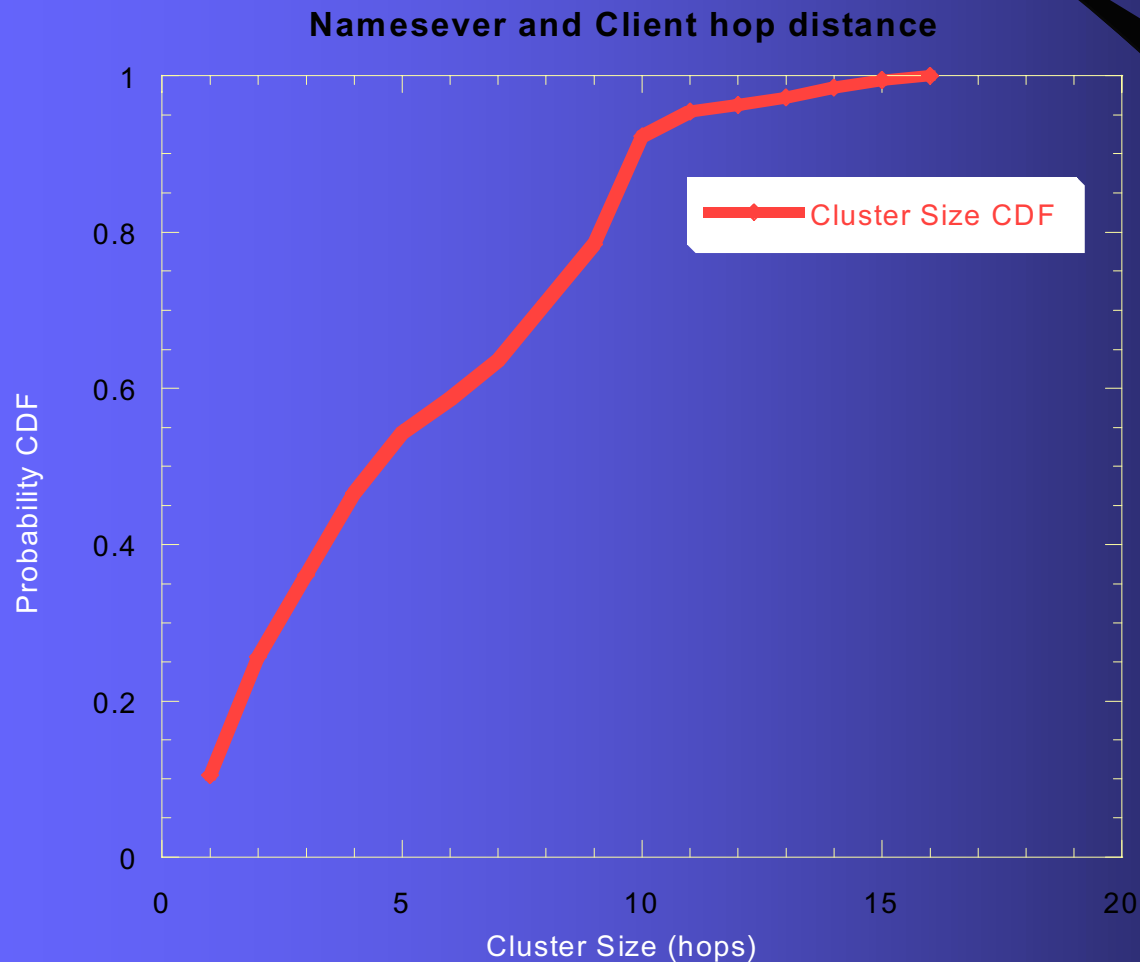
Complete lookup per object costly

Higher TTL preferred

Embedded objects located on same server

HTTP Keep-Alive

# Client Proximity Mismatch



IGS DNS and HTTP logs

Cluster Size: Distance from first common ancestor

Average: 5.7

Max: 16

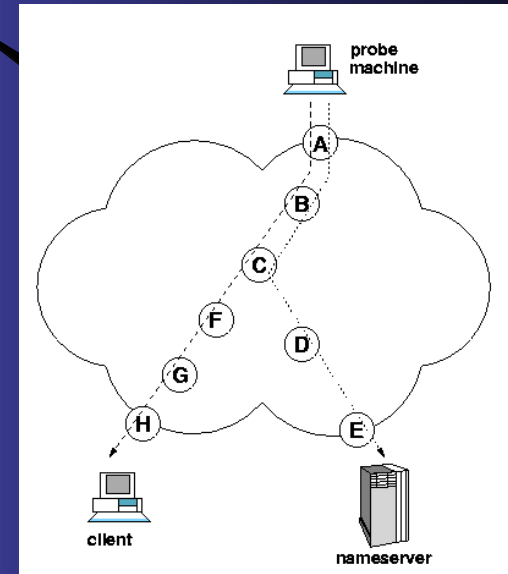
Median: 5 hops

65 percentile: 8 hops

Optimistic since removed mismatches

# Dial-Up Proximity Mismatch

- 1 Direct distance:
  - Mean 7.6 hops; median 8 hops
  - Avg RTT: 234 ms (first hop 188 ms)
- 1 Cluster sizes
  - Median: 8 hops (NY probe),
- 1 Common/disjoint path ratio
  - High ratio  $\Rightarrow$  long common path
  - Median ratio: 0.25 (NY probe)



ISP accounts	9 national retail; 2 free
Unique nameserver addr	54
Nameserver addr per ISP	2-15; avg 7.4

# Summary of DNS Based Server Selection

## 1 Limiting caching

- Increases resolution latency by two orders of magnitude
- Increase end-user latency 24%
  - 1 Larger TTL better TTL
  - 1 Embedded objects co-located on same server
  - 1 HTTP Keep-Alive

## 1 Client/local nameserver proximity

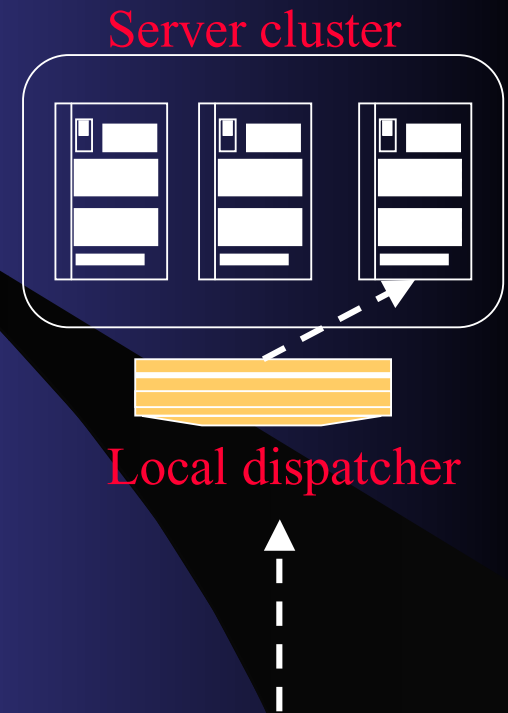
- Clients often far from their nameservers (8 hops or more)
- Correlation between delay to client and local nameserver: positive, but small
- Proposal: include client address in DNS query

# Talk Overview

- 1 Wide-area server selection
  - Techniques and metrics used
  - Issues in DNS-based server selection
  - Evaluation of DNS-based techniques
- 1 Local server selection
  - Overview and new challenges
  - MPLS-based local dispatching (a preview)
- 1 Conclusions

# Local Server Selection

- 1 Local dispatcher for a cluster of servers
  - Arrowpoint (Cisco), Nortel, F5, Alteon, Foundry, IBM
- 1 L4 switches
  - use TCP/IP header for simple client based dispatch
  - provide session persistence, QoS
  - use server load information for load balancing
- 1 L5-L7 switches
  - TCP termination for content based routing
  - dispatching based on HTTP headers, SSL id, cookies, tags
    - 1 connection splicing (LD overhead per connection)
    - 1 connection handoff (modify server kernel)
- 1 Our goals
  - a common solution for all web-switching functions
  - avoid the TCP termination bottleneck
  - use “commodity” hardware



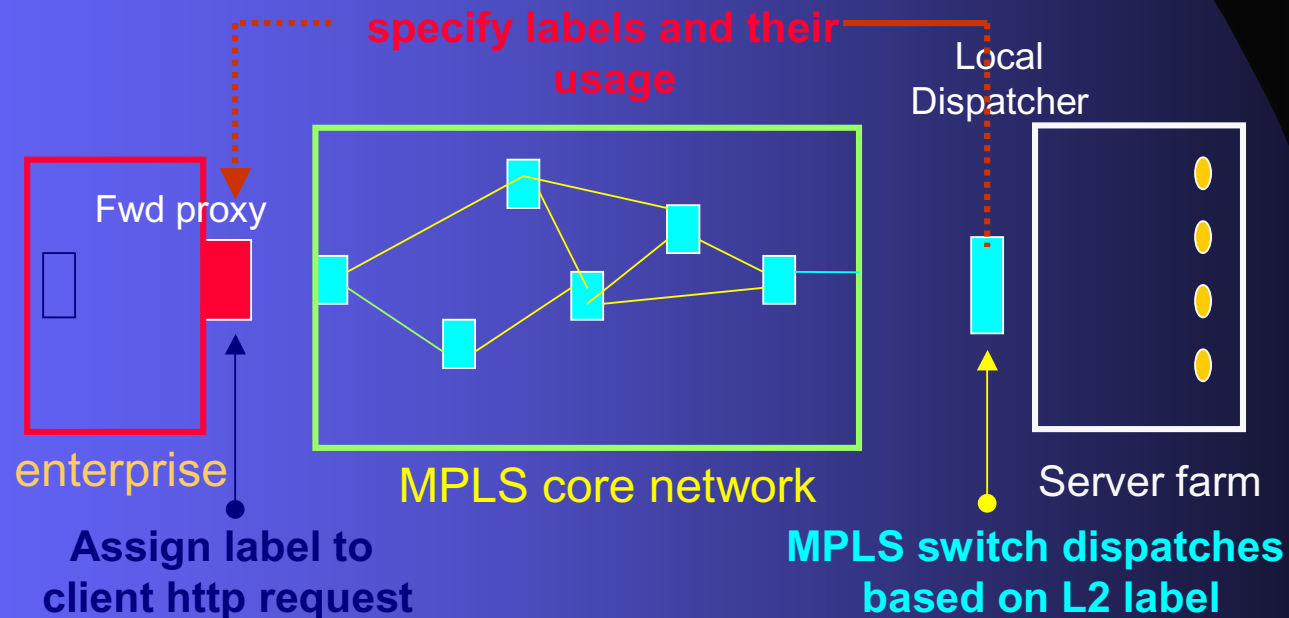
# Open Questions

- 1 Can local dispatching be done at layer 2 switching speeds?
  - replace with MPLS switch
- 1 Can we replace a web switch (L7) with commodity h/w?
  - conjecture: MPLS switches (L2/ L3) will be “commodity”
- 1 Can we provide the same functionality (content routing, load balancing, session affinity, QoS) ?
  - application layer information encoded in layer 2/3/4 headers
  - L3/routing semantics applied to L2 labels

*Yes with MPLS?*

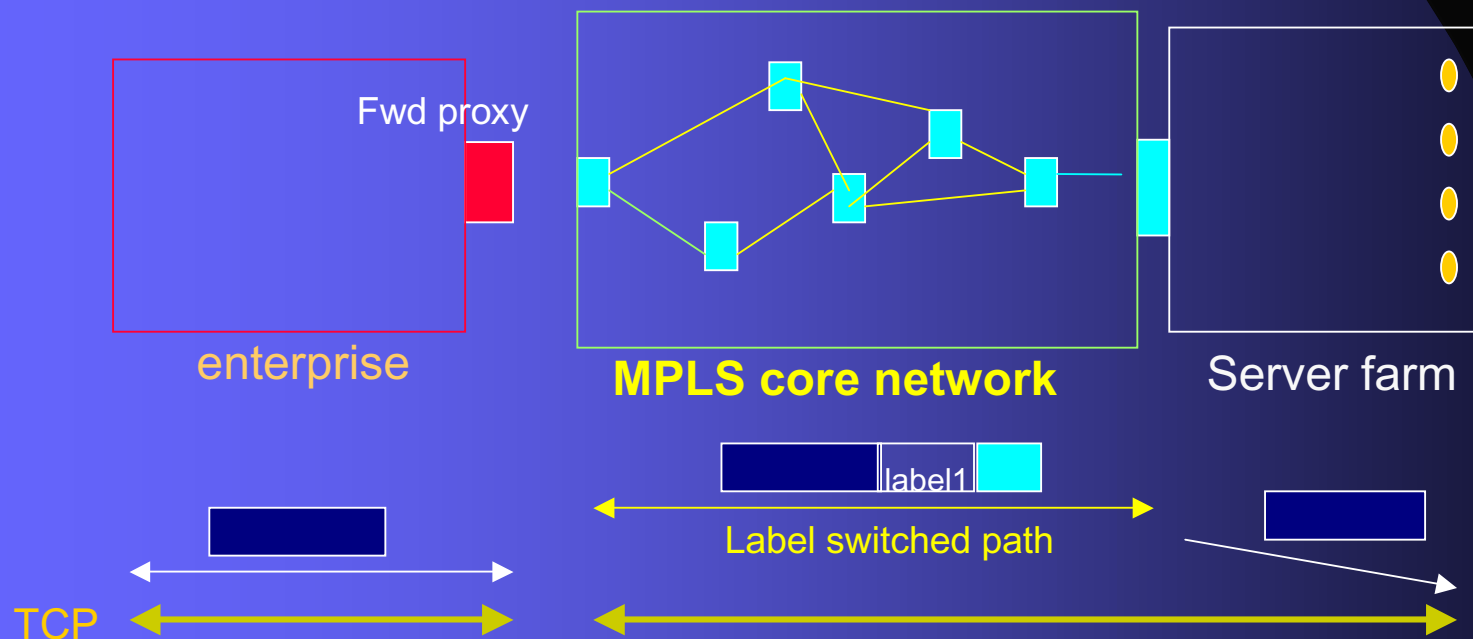
# Proposed Approach

- Use MPLS label stacking feature
- Encode L4-L7 semantics onto MPLS inner label
  - 1 Outer label used for routing
- No TCP termination at dispatcher for content routing, load balancing, affinity
  - 1 Out of path return allowed
- Maintain control connection for label distribution with forward proxies



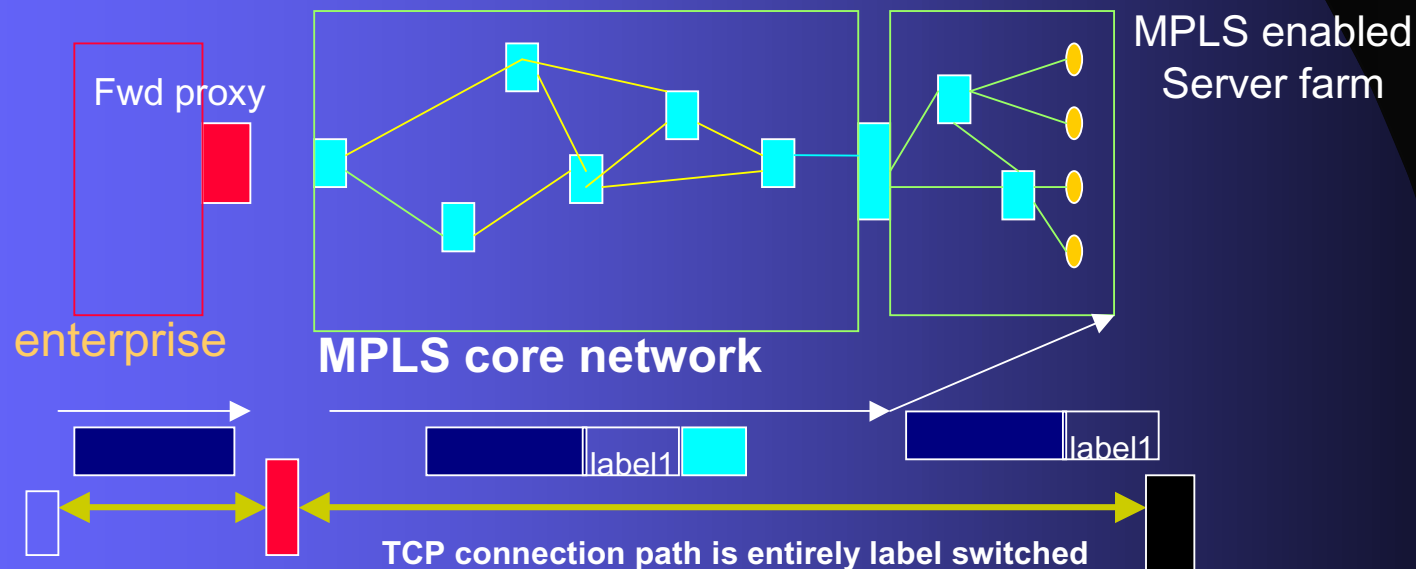
# MPLS-based Content Routing

- 1 Local dispatcher communicates semantics of (inner) label to fwd proxy
  - Label1 → [www.cnn.com/headlinenews/](http://www.cnn.com/headlinenews/)
  - Label2 → [www.cnn.com/fn/](http://www.cnn.com/fn/)
- 1 Fwd proxy assigns inner label based on URL
  - Outer label assigned based on route within the core
  - Dispatcher routes to the right server based on inner label



# MPLS-based Load Balancing

- 1 Local dispatcher communicates set of labels to fwd proxies
  - label1, label2, label3
  - assign weights to labels (w1, w2, w3)
  - different label sets to different proxies
- 1 Fwd proxy assigns inner label per connection based on weights
- 1 Dispatcher can re-map labels for temporary load imbalance
- 1 Redistribute labels/weights for long term changes
- 1 If the server farm network MPLS enabled, then entire path is label switched



# Label Distribution Scenarios

- 1 Control connection between dispatcher and proxies
  - Content based routing
    - 1 table of URL⇒label mappings
  - Load balancing
    - 1 set of labels and weights, proxy round-robins
  - Affinity
    - 1 set of labels, proxy assigns same label to all client requests for a session
  - Service differentiation
    - 1 sets of labels, one set per class (gold/.../..)
    - 1 assumes pre-defined service agreement
- 1 Dispatcher populates label/server mapping table at layer2

# Deployment Issues

- 1 What is the incentive for proxies to participate?
  - Benefits to dispatcher is better performance
  - Proxies could belong to same Web hosting/ISP providers (mutual benefit)
  - Profit sharing between ISP proxies and the content hosting companies
- 1 How many proxies need to participate?
  - #enterprises/ISP proxies accessing a given server is very small (~250) even for a large client access base (~60 million)
  - limited participation adequate to derive large performance gains (?)
- 1 If proxy and dispatcher are in different MPLS domains, how label stacking will work needs to be resolved

# Conclusions

- 1 DNS-based wide-area server selection promising but flawed
  - Need larger TTL for scalability and client latency
  - Need to solve proximity mismatch issues
- 1 Label encoding techniques for local server selection
  - MPLS inner label encodes higher layer information
  - Layer 7 switching at price/performance of layer 2
- 1 Open Research Issue:
  - Can label encoding work in wide-area server selection too?