DECADE Survey

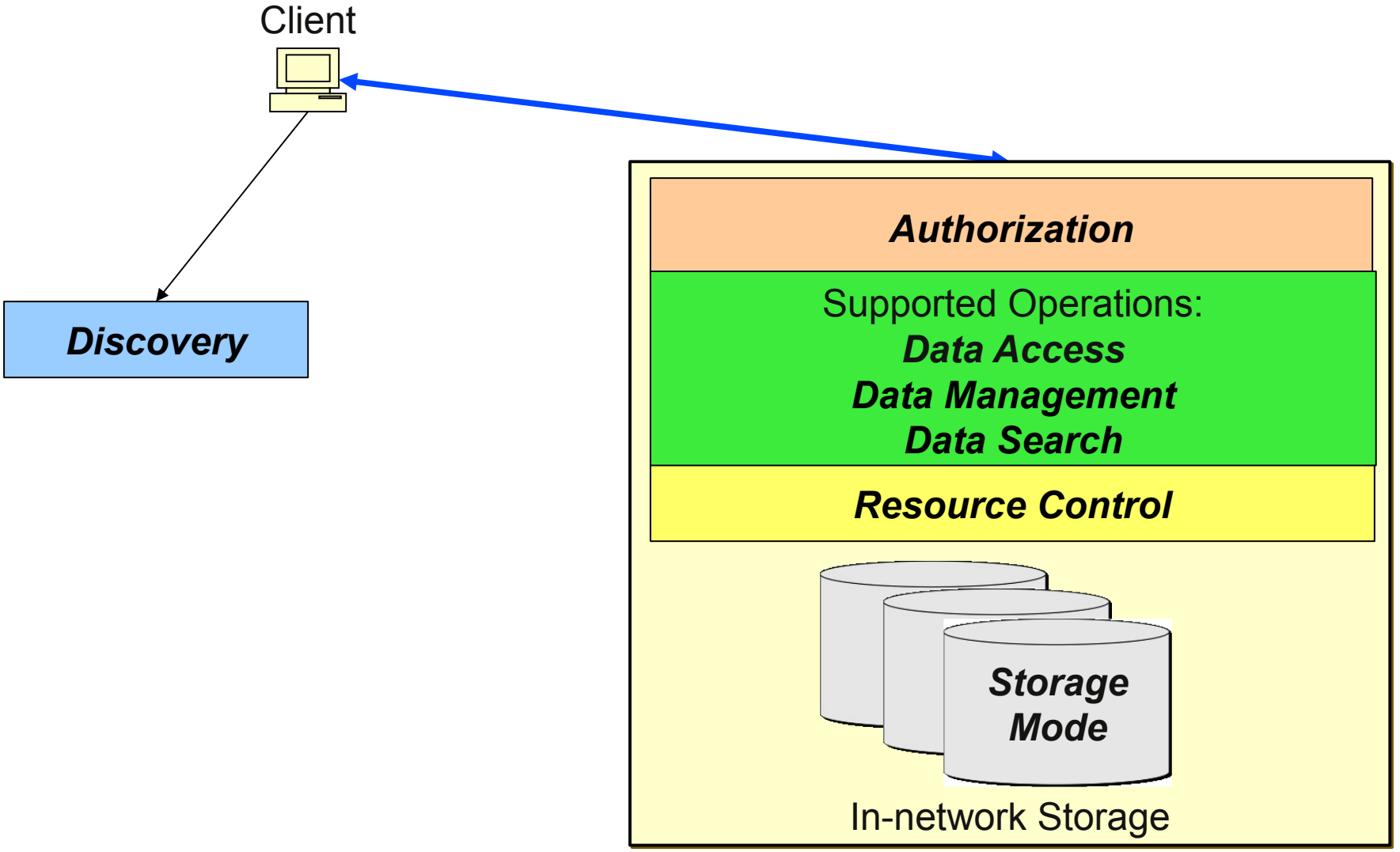
draft-song-decade-survey-03

Richard Alimi
ZhiHui Lu
Haibin Song
Y. Richard Yang

Survey Overview

- In-network storage used in many contexts
  - One common use is to increase efficiency of content distribution
- Existing systems have been useful in their own contexts
  - Systems' capabilities reflect their specific context
- Survey evaluates in context of DECADE
- ***Purpose***
  - ***Focus discussion of capabilities needed in DECADE beyond what is provided by existing protocols/systems***
  - ***Not to debate details or merits of existing storage solutions***

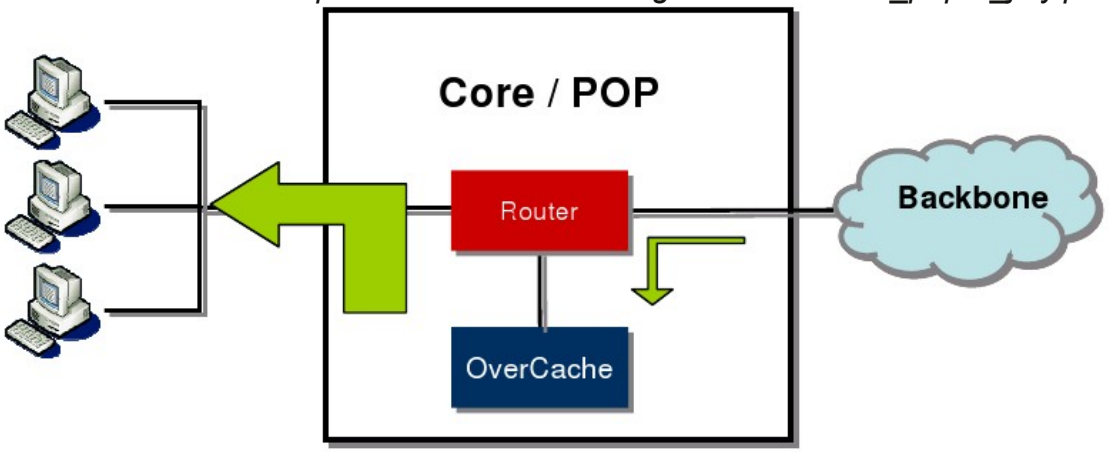
In-network Storage System Components



Transparent P2P Cache

- Cache P2P content and serve locally
- Implements P2P applicatior protocols to avoid changes to P2P clients
- Uses DPI to avoid explicit discovery by P2P clients
  - Acts as intermediary in session with remote peer

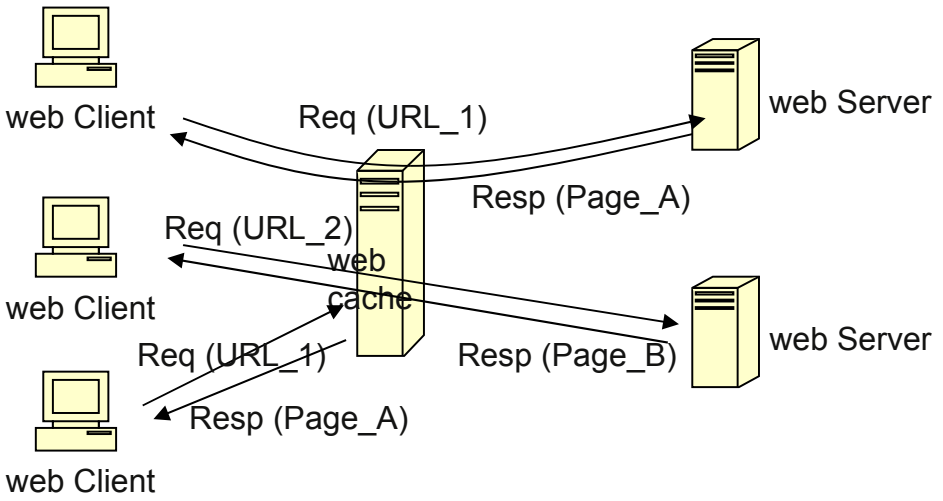
Source: http://www.oversi.com/images/stories/white_paper_july.pdf



Discovery	DPI (transparent to client)
Authorization	Not provided
Data Access	Read/write (transparent to client) according to caching/ISP policy
Data Mgmt	Not provided
Data Search	Not provided
Resource Ctrl	Not provided
Storage Mode	Object-based (chunks of content stored)

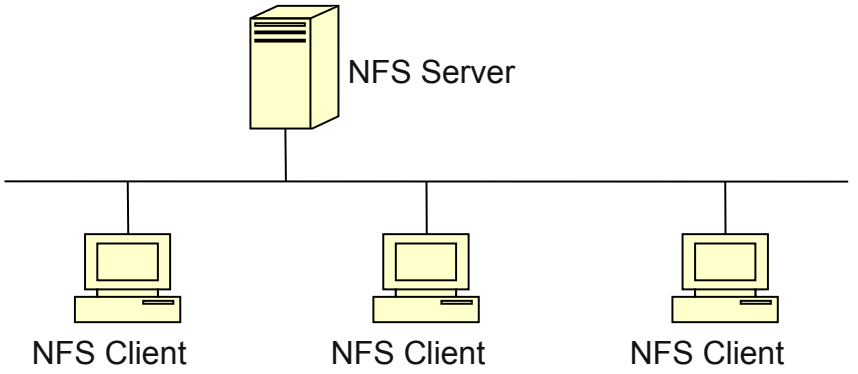
Web Cache

- Cache web content and serve locally
  - HTML pages, images, etc
- Server indicates cachability, clients indicate if cached response is acceptable
- HPTP: Extension for P2P
  - Proposes to share P2P content using HTTP
  - Aims to use existing web caches



Discovery	Manual configuration, DNS, or transparent (DPI)
Authorization	Not provided
Data Access	Read/write according to caching/ISP policy
Data Mgmt	Not provided
Data Search	Not provided
Resource Ctrl	Not provided
Storage Mode	Object-based (keyed by HTTP request fields)

NFSv4, NFSv4.1



- Allow client to access network storage in manner similar to local storage
- Major features
  - Authentication mechanisms
  - Delegation to clients
  - Locking
  - Split metadata and data (pNFS)
  - Access control supports ACLs and modes
  - Named attributes

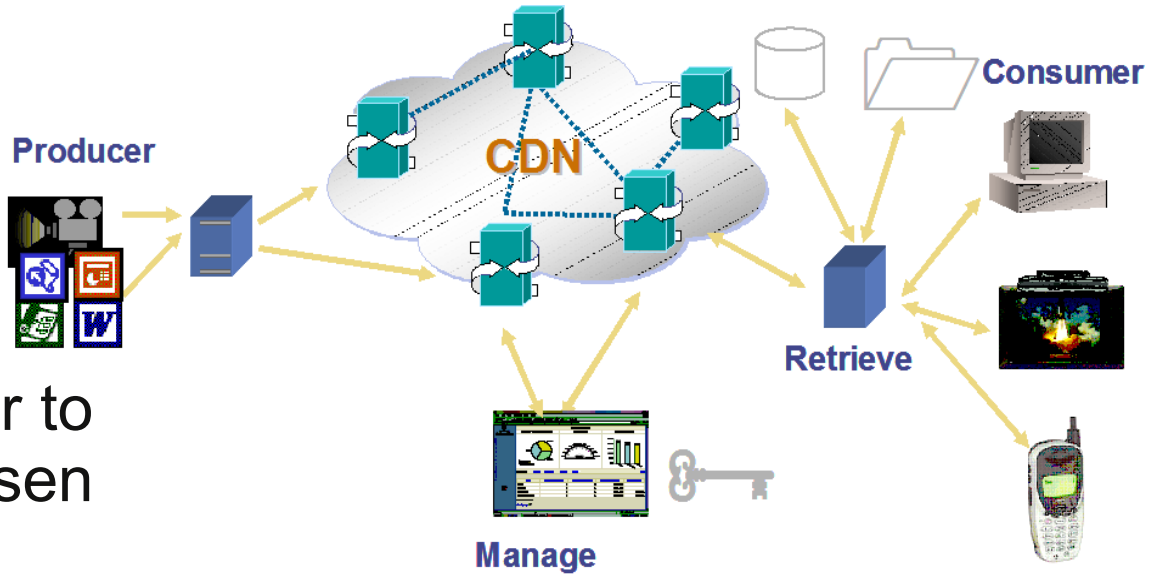
Discovery	Manual (IP address or via DNS lookup of well-known hostname)
Authorization	User-based; processes using ACL
Data Access	Traditional FS operations (e.g., open/close, read/write, remove)
Data Mgmt	Traditional FS operations (e.g., rename, link, getattr/setattr)
Data Search	Enumerate directory to find desired file (e.g., readdir, lookup)
Resource Ctrl	User-based storage quota
Storage Mode	File-based

WebDAV

- Distributed authoring for web resources
  - And various other uses
- Major features
  - Properties, Locking
- Extensions
  - Versioning (RFC3253)
  - SEARCH (RFC5323)
  - ACL (RFC3744)
  - Tickets for authorization (draft-ito-dav-ticket-00)
  - Quotas (RFC4331)

Discovery	Manual (IP address or via DNS lookup of well-known hostname)
Authorization	User-based; permissions include read, write, etc
Data Access	Traditional filesystem operations (e.g., read, write); no update
Data Mgmt	Traditional filesystem operations (e.g., move, delete)
Data Search	Enumeration, or list by user-supplied criteria
Resource Ctrl	User- or collection-based storage quota
Storage Mode	File-based (organized by collections)

CDNs



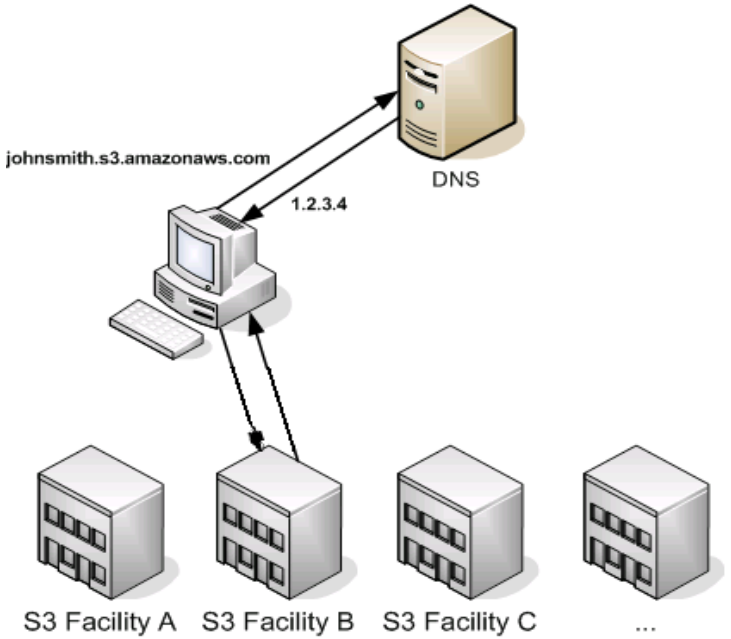
- Distribute content to cache/edge servers closer to users; direct users to chosen servers
- Content owner has management frontend
- Typically have extensive infrastructure
  - Distribution amongst CDN nodes, cache management, request routing, etc

Discovery	DNS or other redirection
Authorization	Typically not provided
Data Access	Read-only for clients Writable for content provider
Data Mgmt	Only to content provider
Data Search	Only to content provider
Resource Ctrl	Not provided
Storage Mode	File-based

Adapted From: http://docs.amazonwebservices.com/AmazonS3/2006-03-01/dev/

Amazon S3

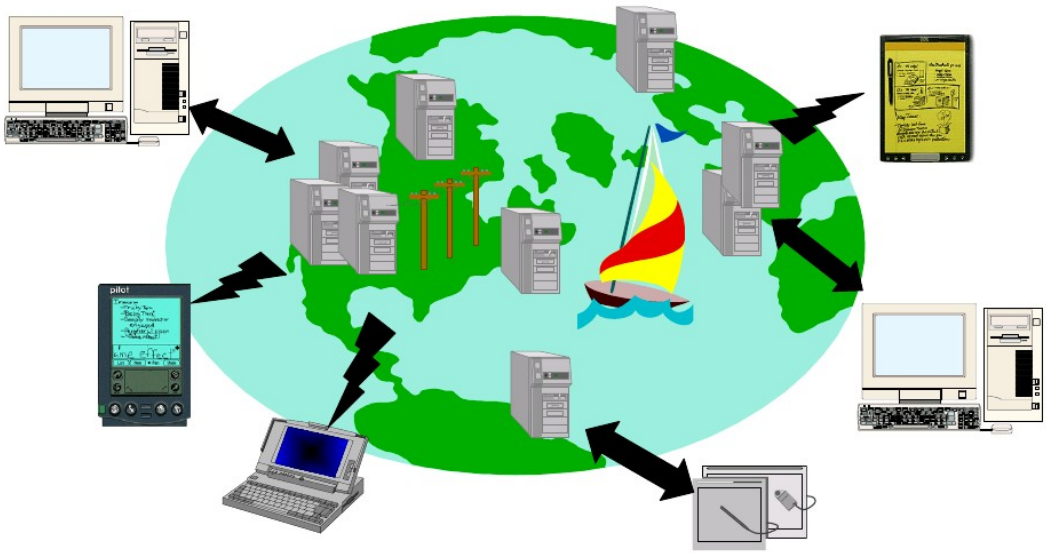
- Online storage service for end users
- Storage organized into buckets containing data objects
- Popular backend storage for other applications
- Related services
  - Windows Azure Blob service



Discovery	Manual (via DNS lookup of well-known hostname)
Authorization	Typically not provided
Data Access	Read, write
Data Mgmt	Delete
Data Search	User may enumerate bucket contents to find desired file
Resource Ctrl	Not provided
Storage Mode	Object-based (organized into buckets)

OceanStore

Source: http://oceanstore.cs.berkeley.edu/publications/talks/StanfordOceanStore.pdf



- Research storage system from UC Berkeley
- Aim is to provide globally-distributed storage
- Multiple storage providers pool resources together
- Focus on
  - Resiliency
  - Self-organization
  - Self-maintenance

Discovery	Manual (via DNS lookup of well-known hostname)
Authorization	Provided (specifics unclear from published paper)
Data Access	Read, write
Data Mgmt	Allows update of existing objects; multiple versions may be retained
Data Search	Not provided
Resource Ctrl	Not provided
Storage Mode	Object-based (though, NFS and HTTP interfaces built on top of it)

OAuth

- NOT a storage protocol
  - Included here due to its authentication model
- “client” vs. “resource owner”
  - OAuth separates them
  - Resource owner can provide limited access to a client
- Features of credentials
  - Expiration time
  - Allow revocation by owner

Discovery	N/A
Authorization	Client creates delegation request; approved by resource owner
Data Access	N/A
Data Mgmt	N/A
Data Search	N/A
Resource Ctrl	N/A
Storage Mode	N/A

Comments and questions?

See draft for additional information

Non-Transparent P2P Cache

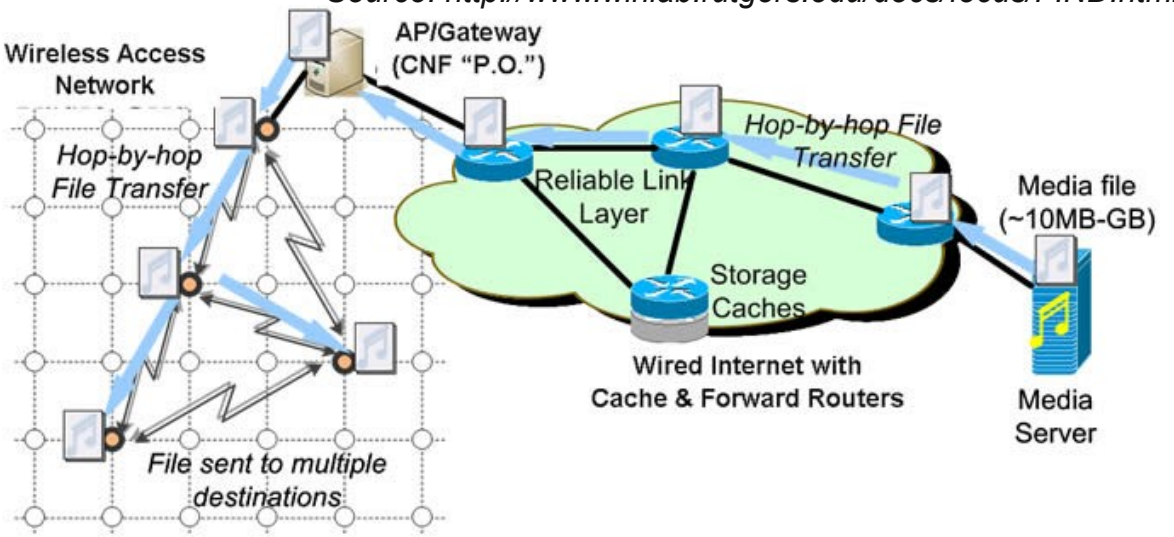
- Cache frequently-used P2P content and serve locally
- Implements P2P application protocols to avoid changes to P2P clients
- Explicitly peers with a client

Discovery	Normal discovery in P2P overlay (tracker, DHT, PEX, etc.)
Authorization	Not provided
Data Access	Read/write Write is according to caching policy
Data Mgmt	Not provided
Data Search	Not provided
Resource Ctrl	Not provided
Storage Mode	Object-based (chunks of content stored)

Cache-and-Forward Architecture

- Proposal for content delivery in future Internet
- Storage placed at some nodes within network
  - At or nearby routers
- Store-and-forward
  - Disconnected mobile users
  - In-network caching
- Focus on large data files

Source: http://www.winlab.rutgers.edu/docs/focus/FIND.html

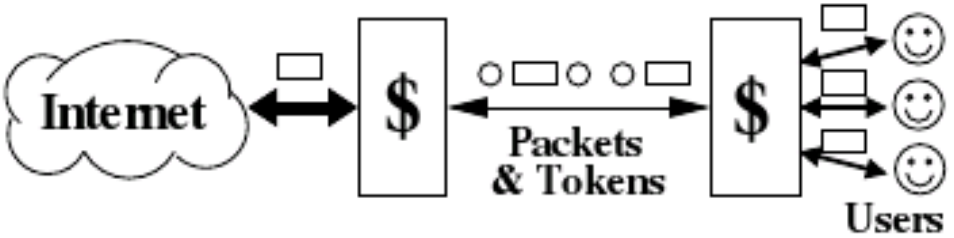


Discovery	Lookup cache-and-forward node via location-independent content ID
Authorization	Not provided
Data Access	Read/write (transparent to client) Write is according to caching policy
Data Mgmt	Not provided
Data Search	Not provided
Resource Ctrl	Not provided
Storage Mode	Object-based (with objects representing individual files)

Traffic Redundancy Elimination (RE)

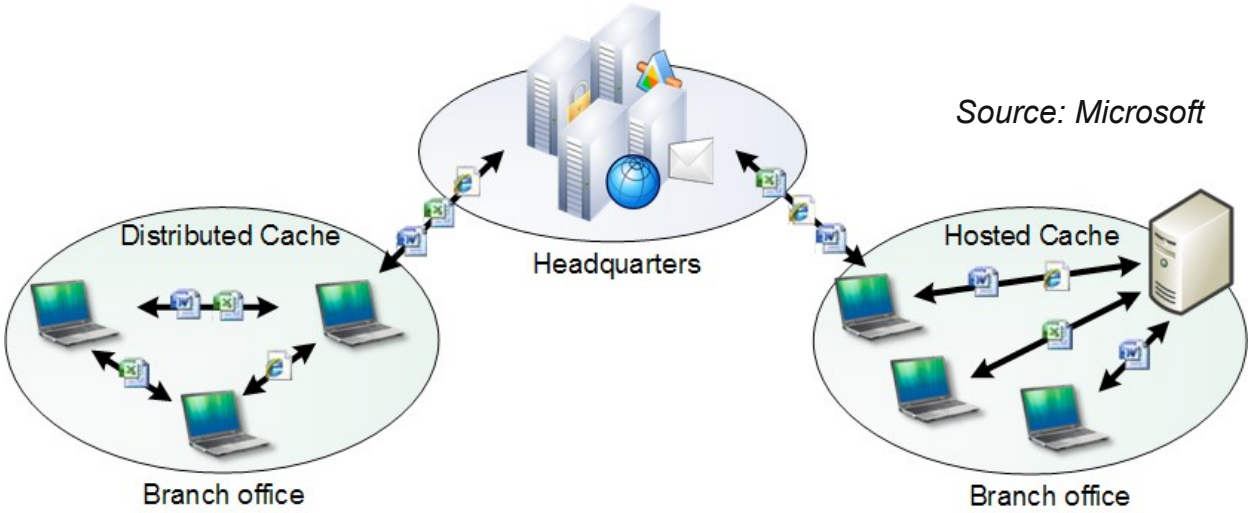
Source: N. Spring, D. Wetherall. "A protocol-independent technique for eliminating redundant network traffic", SIGCOMM 2000.

- Identify and remove repeated content in network transfers
- Packet-level RE
  - Forwarding elements equipped with storage
  - Cache data from forwarded packets
  - Upstream routers can replace previously-forwarded data with fingerprint



Discovery	Not necessary; implemented entirely within network elements
Authorization	Preserves endpoint control
Data Access	Read/write (transparent to user)
Data Mgmt	Not provided
Data Search	Not provided
Resource Ctrl	Content provider still moderates packet sending rate
Storage Mode	Object-based (with objects being data from transferred packets)

BranchCache



Source: Microsoft

- Caches and shares content within branch office
  - Reduce WAN link utilization
  - Improve application responsiveness
- Transparent to end-user
  - Instrument networking stack
- Hosted Cache and Distributed modes
- Maintains end-to-end security

Discovery	Distributed: multicast Hosted: provisioning or manual
Authorization	Keys derived from content server; data decryptable by auth'd clients
Data Access	Read/write (transparent to client) Write is according to caching policy
Data Mgmt	Not provided to end user
Data Search	Not provided to end user
Resource Ctrl	Hosted: admin-controled policy Distributed: backoff and throttling
Storage Mode	Object-based