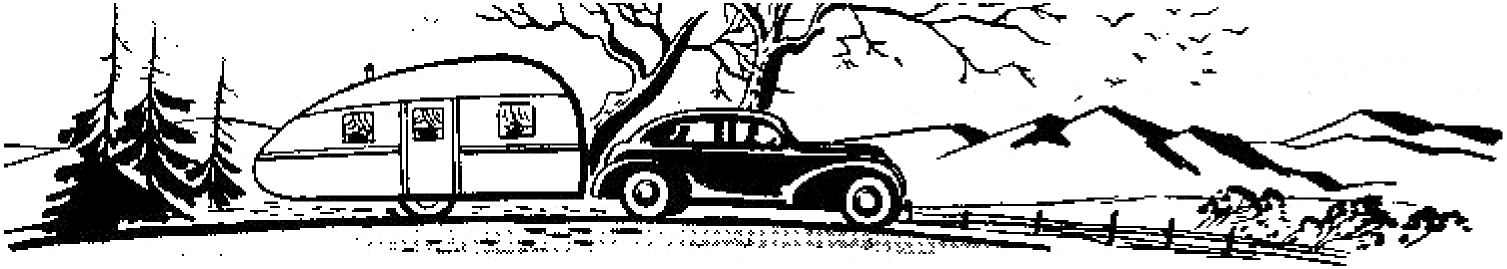
CIFS Geeks in Exile

– or –

What We Did on our Holiday



Christopher R. Hertel

Storage Architect, CIFS Geek

Founder and CTO

ubiqx Consulting, Inc.

SambaXP

May, 2011

Introductions

Me

Your Friendly Neighborhood CIFS Geek

- CIFS Author
- jCIFS project co-founder
- Samba Team member since 97/98
- Incurable Idealist
- Etc., etc., ad nauseam

A ruminant mammal (Geekus geekus) with long legs, humped shoulders, and broadly palmated antlers.

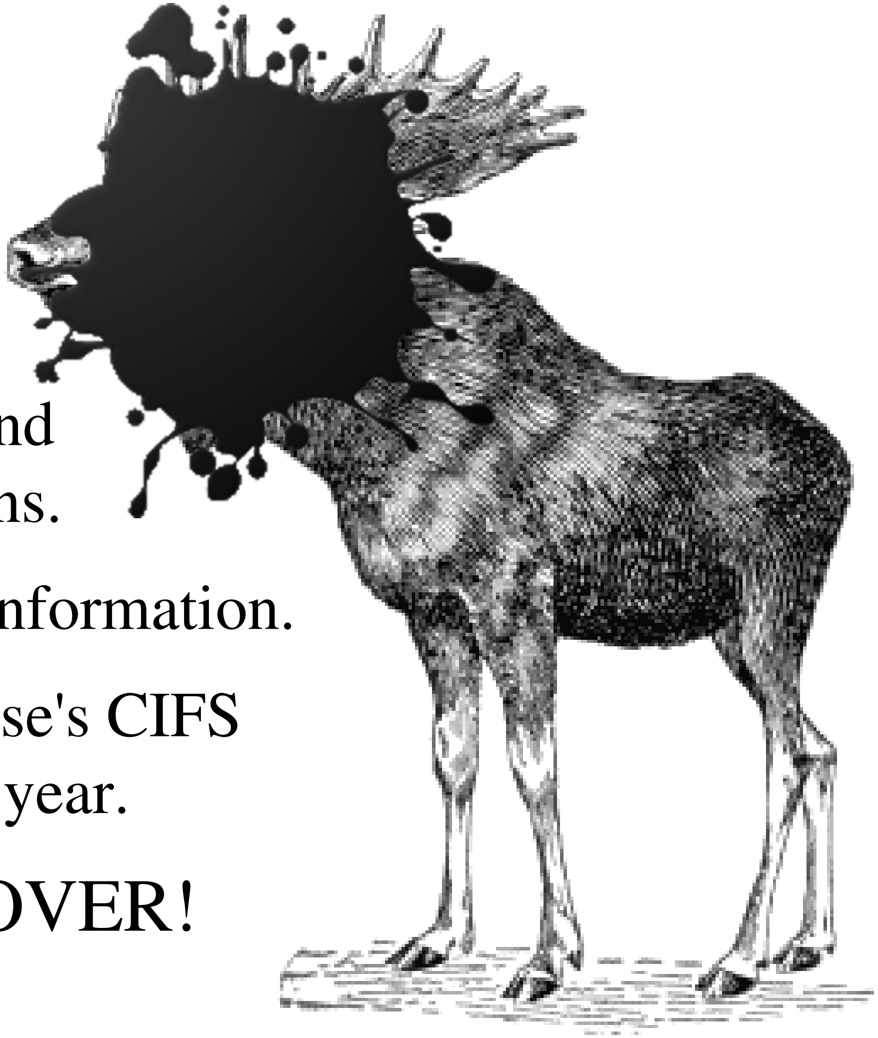
Me

Your Friendly Neighborhood CIFS Geek

Tainted!

- Lead author of the Microsoft [MS-CIFS] and [MS-SMB] specifications.
- Access to MS Internal Information.
- Mustn't touch anyone else's CIFS implementation for one year.

That year is now OVER!



A ruminant mammal (Geekus geekus) with long legs, humped shoulders, and broadly palmated antlers.

What We Did on our Holiday

What We Did on our Holiday

This is my report on what we did on our CIFS holiday.

- **Linux Clusters**
  Worked On GFS2 “virtual clusters”.
- **BITS Protocol**
  Created a BITS client toolkit.
- **MS BranchCache™**
  Studied Microsoft's BranchCache™ system.

Linux Clusters

Linux Clusters with GFS2

Why GFS2?

- In-kernel cluster file system
- Red Hat Cluster Suite
  - Supported in Fedora
- Local (to me)
  - Originally a U of MN project
  - I know these geeks
  - Easy to interact
- Good “community” choice

...but some Samba Team members have reported difficulties configuring and running GFS2-based clusters.

Linux Clusters with GFS2

There are several other cluster FS options:

Ceph	– work in progress
GlusterFS	– cache consistency issues
MooseFS	– untested (to my knowledge)
OCFS	– similar to GFS

See Wikipedia for a longer list.

Linux Clusters with GFS2

Short Term Goal:

- Virtual “Cluster in a Box”
- Single server testing cluster
  - Fedora-14
  - KVM/QEMU

Status:

- The `cbox` **cluster-in-a-box** script works
- Virtual GFS2 clusters on KVM do not
  - I/O stress causes FS hang
  - A fix is in the works

Linux Clusters with GFS2

Long Term Goal:

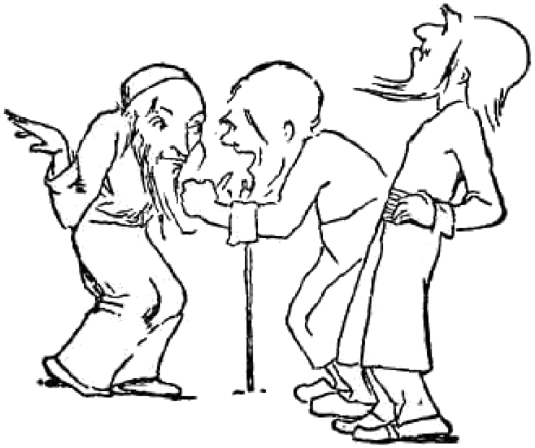
- Samba/CTDB/GFS2 HowTo
  - Do-it-yourself virtual clusters
  - "Real" hardware clusters
- Production clusters running Samba and NFS

Status:

- 3 HowTos, need to be combined into one
- RedHat has built working Samba clusters
  - ...but has not yet performed extensive testing
  - Focus is on `cbox` clusters

Linux Clusters with GFS2

Why Clusters?



- Failover
  - SMB does not handle disconnect/reconnect very well
  - ...but SMB2 does
- Active/Active load balancing
  - SMB/CIFS/SMB2 is stateful
  - CTDB provides shared state
- Scalability

Are there other, better ways to approach these goals?

BITS

01000010 01100001
01100011 01101011
01100111 01110010
01101111 01110101
01101110 01100100
00100000 01001001
01101110 01110100
01100101 01101100
01101100 01101001
01100111 01100101
01101110 01110100

BITS: Background “Intelligent” Transfer Service

“BITS is Earth’s most widely used file transfer service, with more than 600 million unique users across the planet.”

– Vipul Bansal, Microsoft WMI Blog, Jan 2009.

BITS: Background “Intelligent” Transfer Service

“BITS is Earth’s most widely used file transfer service, with more than 600 million unique users across the planet.”

– Vipul Bansal, Microsoft WMI Blog, Jan 2009.

Note Well: *nobody cares.*

BITS: Background "Intelligent" Transfer Service

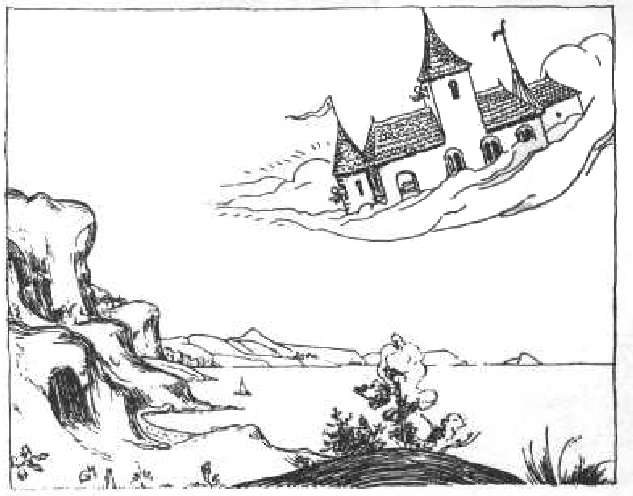
"BITS is Earth's most widely used file transfer service, with more than 600 million unique users across the planet."
– Vipul Bansal, Microsoft WMI Blog, Jan 2009.

What does that mean anyway?

- It doesn't say "protocol", it says "file transfer service".
- BITS is the Windows system service used by Windows Update to download patches.
- Most users don't even know it's there.

BITS: Background “Intelligent” Transfer Service

BITS Features



- Built into Windows
- Restartable Transfers
  - ...but only linearly; does not “patch”.
- Both Download and Upload
  - ...and “Upload Reply”.
- Job priority levels
- Senses network traffic to manage impact

BITS: Background “Intelligent” Transfer Service

BITS Download Jobs

- The overwhelming majority of BITS jobs are probably Windows Update downloads.
- BITS Downloads use HTTP/HTTPS.
- Sort of like `uucp`?
  `wget` + `batch` + `nice` + `diffserv`?

The “special sauce” is the use of network traffic monitoring to limit BITS data transfer rates.

BITS: Background “Intelligent” Transfer Service

BITS *Up*load Jobs

- Much less common.
- Proprietary extensions to HTTP/HTTPS.
- Only between Windows BITS clients and Windows HTTP[S] servers.

BITS: Background “Intelligent” Transfer Service

BITS *Up*load Jobs

- Much less common.
- Proprietary extensions to HTTP/HTTPS.
- Only between Windows BITS clients and Windows HTTP[S] servers – Until now!

BITS: Background "Intelligent" Transfer Service

STiB means:

- **S**low **T**ransfer **in** **B**ackground?
- **S**illy **T**echnology **is** **B**oring?
- **S**ipping **T**ea **in** **B**elgium?
- BITS spelled sdrawkcab with a small '*i*'?

STiB: It Is what It Is.

- ...a toolkit for testing BITS Uploads.
- ...example code for others to read / use.

A CGI script could be written to
accept BITS Uploads.

BITS: Background “Intelligent” Transfer Service

BITS Upload Extensions:

- HTTP Extension Method: BITS_POST
- BITS Packet Types
  - Ping
  - Create-Session
  - Fragment
  - Cancel-Session
  - Close-Session
  - Ack

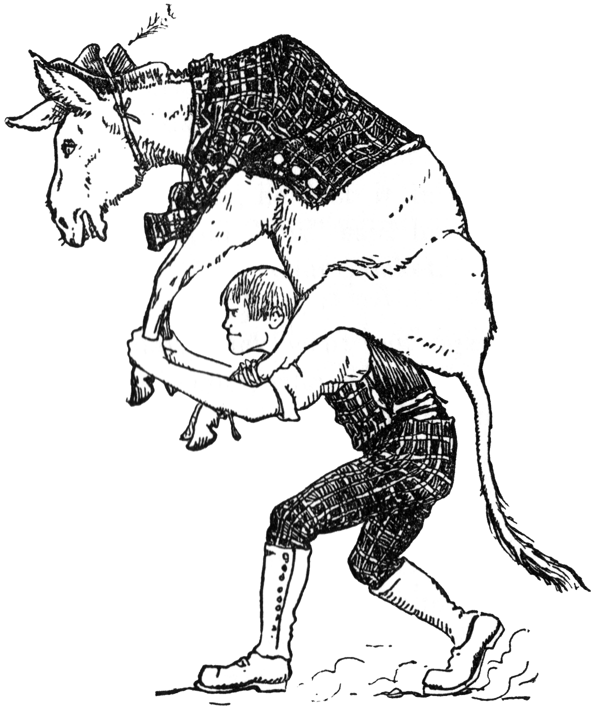
BITS Documentation:

- MSDN: BITS Upload Protocol
- WSPP: [MC-BUP]

BITS: Background “Intelligent” Transfer Service

BITS Upload Extensions:

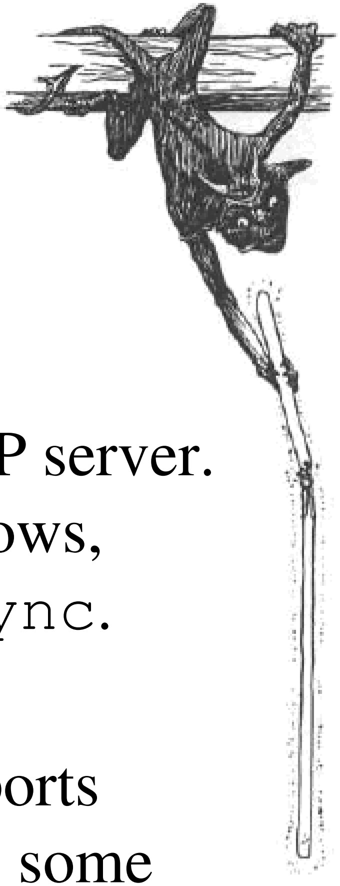
- HTTP Extension Method: BITS_POST
- BITS Packet Types
  - Ping
  - Create-Session
  - Fragment
  - Cancel-Session
  - Close-Session
  - Ack



BITS Documentation:

- MSDN: BITS Upload Protocol
- WSPP: [MC-BUP]

BITS: Background “Intelligent” Transfer Service



Do we care?

Yet Another Windows Protocol

- BITS Upload is supported in IIS,
  - and in Microsoft's “lightweight” HTTP server.
- It's convenient when working with Windows,
  - but not nearly as powerful as, eg., `rsync`.

MS-BITS, however, also supports BranchCache™, which suggests some very useful testing scenarios.

- Add “Get” support to STiB,
- Include the modified header,
- See what happens!

Prequel

Pay Attention!

This is where it finally gets interesting.

Prequel

What the heck is *Prequel*?

Prequel

Prequel: A project to build an
Open Source Implementation
of Microsoft's BranchCache™.

So what the heck is BranchCache™?

Prequel

Prequel: A project to build an
Open Source Implementation
of Microsoft's BranchCache™.

BranchCache™ is a
distributed content caching system

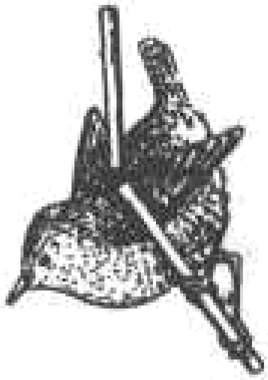
- supported in W2K8 servers,
- and Windows7 clients.

Cheap, effective WAN
acceleration for SMB2,
HTTP, and BITS.

Prequel

BranchCache Architecture

A quick overview



Content Servers

- Have content to share with multiple clients.

Clients

- Request & receive content from content servers.

The Cache

- A copy of the original content, cryptographically tagged and divided into segments and blocks.

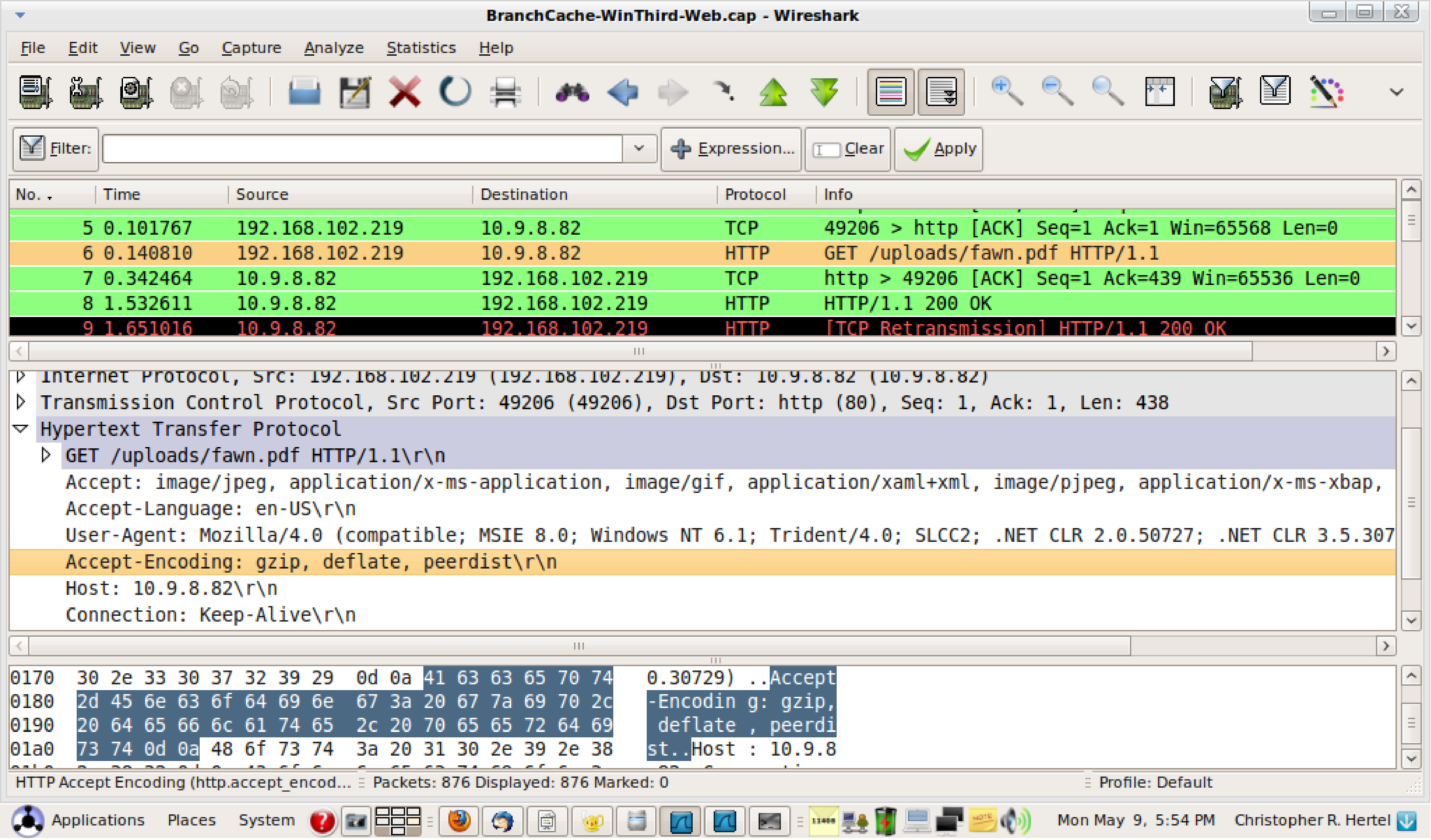
Prequel

Content Servers:

- Web Servers (HTTP, BITS)
- File Servers (SMB2)

The client must know to ask for *content tags* instead of actually content.

- If the tags are already calculated, they are returned by the BranchCache™-enabled server.
- Otherwise, the actual content is returned, and the server calculates the tags for next time.



```
No.  Time       Source            Destination       Protocol  Info
  5 0.101767   192.168.102.219   10.9.8.82         TCP       49206 > http [ACK] Seq=1 Ack=1 Win=65568 Len=0
  6 0.140810   192.168.102.219   10.9.8.82         HTTP      GET /uploads/fawn.pdf HTTP/1.1
  7 0.342464   10.9.8.82         192.168.102.219   TCP       http > 49206 [ACK] Seq=1 Ack=439 Win=65536 Len=0
  8 1.532611   10.9.8.82         192.168.102.219   HTTP      HTTP/1.1 200 OK
  9 1.651016   10.9.8.82         192.168.102.219   HTTP      [TCP Retransmission] HTTP/1.1 200 OK
```

```
▷ Internet Protocol, Src: 192.168.102.219 (192.168.102.219), Dst: 10.9.8.82 (10.9.8.82)
▷ Transmission Control Protocol, Src Port: 49206 (49206), Dst Port: http (80), Seq: 1, Ack: 1, Len: 438
▽ Hypertext Transfer Protocol
  ▷ GET /uploads/fawn.pdf HTTP/1.1\r\n
    Accept: image/jpeg, application/x-ms-application, image/gif, application/xaml+xml, image/pjpeg, application/x-ms-xbap,
    Accept-Language: en-US\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.307
    Accept-Encoding: gzip, deflate, peerdist\r\n
    Host: 10.9.8.82\r\n
    Connection: Keep-Alive\r\n
```

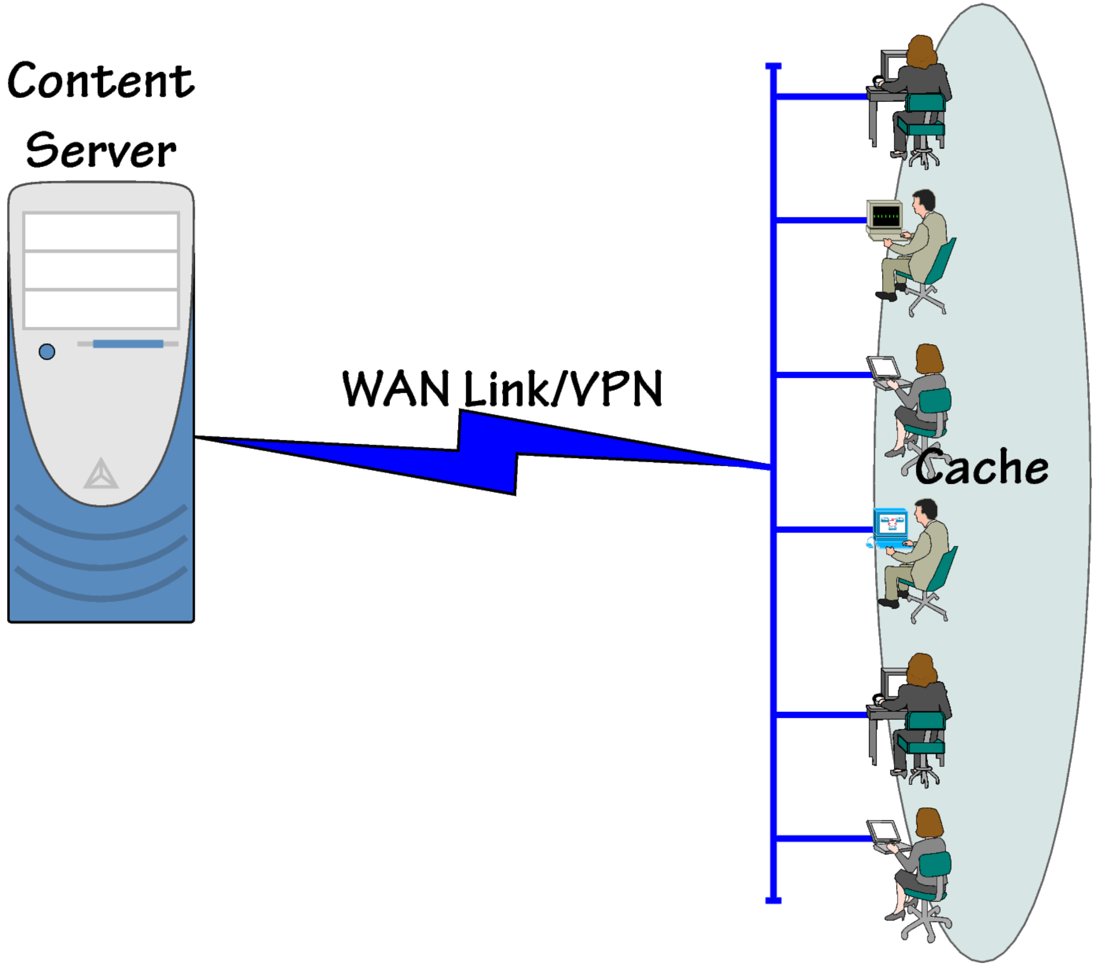
```
0170  30 2e 33 30 37 32 39 29  0d 0a 41 63 63 65 70 74   0.30729) ..Accept
0180  2d 45 6e 63 6f 64 69 6e  67 3a 20 67 7a 69 70 2c   -Encodin g: gzip,
0190  20 64 65 66 6c 61 74 65  2c 20 70 65 65 72 64 69    deflate , peerdi
01a0  73 74 0d 0a 48 6f 73 74  3a 20 31 30 2e 39 2e 38   st..Host : 10.9.8
```

```
HTTP Accept Encoding (http.accept_encod...   Packets: 876 Displayed: 876 Marked: 0   Profile: Default
```

IE 8 indicates support for BranchCache™ by listing “peerdist” as an acceptable encoding.

Prequel

Distributed Mode



Prequel

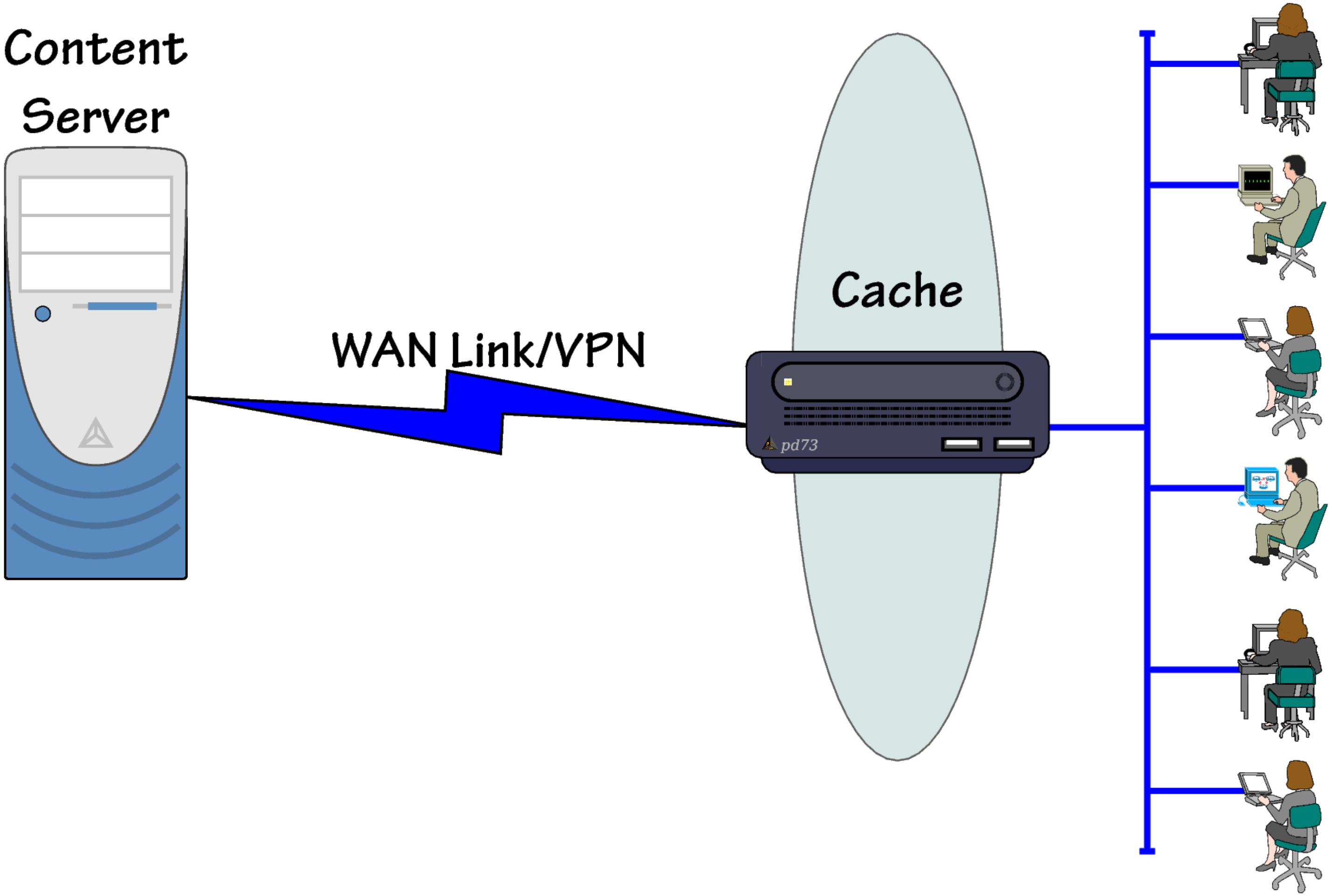
Distributed Mode

- Each client keeps a local cache.
- A client requests tags from the server, then broadcasts to find the cached content.
- If the content is not cached,
  - The client requests the content from the content server,
  - The client stores both content and tags in its own cache.

Reminiscent of the Browse Service.

Prequel

Hosted Mode



Prequel

Hosted Mode

- A client request tags from the content server
- The client then asks the cache server for the content
- If the content is not cached, the client requests content from the content server
- The client sends both content and tags to the cache server
- Content can now be retrieved from the cache server using only tags

Prequel

Content Tags

Blocks

- Are a unit of download
  (from either content server or cache server)
- Are 64K
  (or less, for the last block in a file only)

The block tag is an SHA hash of the block.

Segments

- Are a unit of discovery
- Are 32M == 512 blocks
  (or less, if the last block is short)

The segment has is an SHA of the included block hashes.

Prequel

Prequel Goals

I. Content Server

- CGI script for Apache that generates correct tags.
- Server-side code to provide a starting point for Samba implementation.

II. Cache Server

- Implement a Hosted Cache server.

III. Peer Cache

- Implement a stand-alone peer caching client.

Other Stuff

CIFS.ORG

The End