A CIFS Geek in Exile

– or –

What I Did on My Holiday

ubiqx Consulting, Inc.

Christopher R. Hertel

Storage Architect, CIFS Geek

Founder and CTO

SNIA SDC 2011

Septemberly, 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.

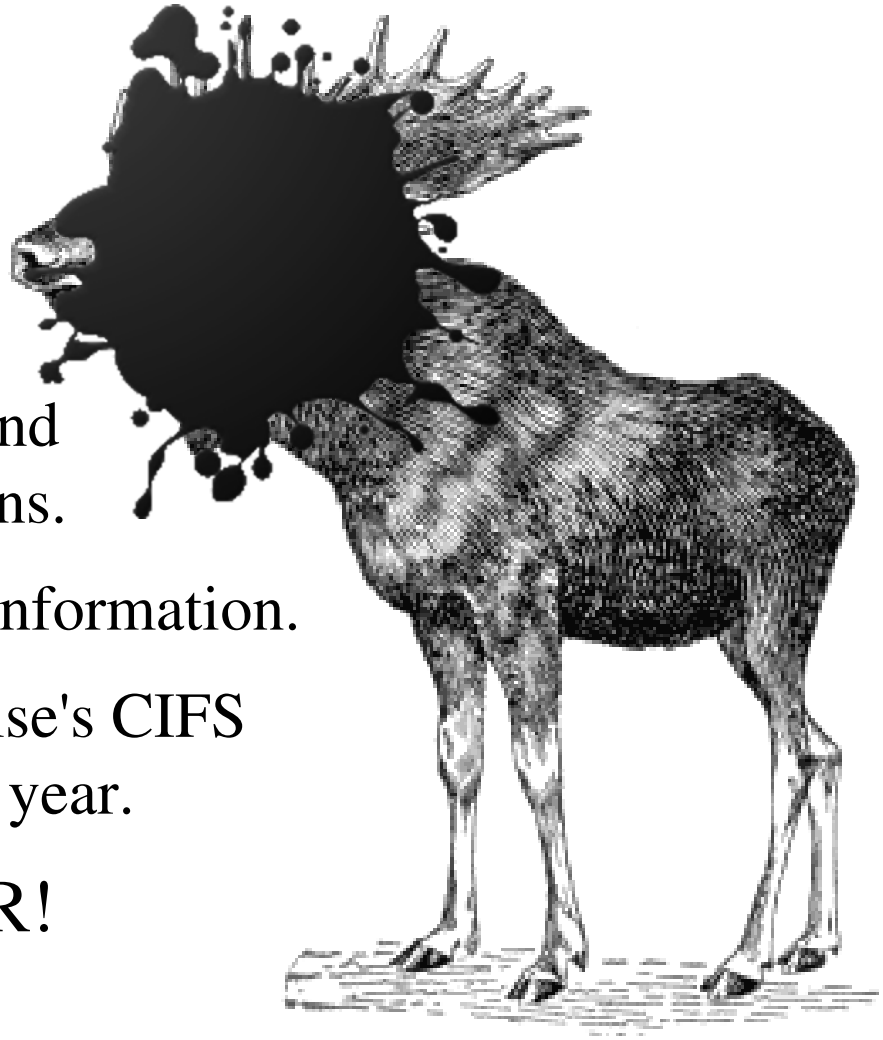
Me

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 OVER!



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

What I Did on my Holiday

What I Did on my Holiday

This is my report on what I did on my CIFS holiday.

I worked on implementing *other* Windows protocols:

- **BITS Protocol**

  Created STiB: a BITS client toolkit.

- **MS BranchCache™**

  Started implementing PeerDist (BranchCache™).

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:

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."

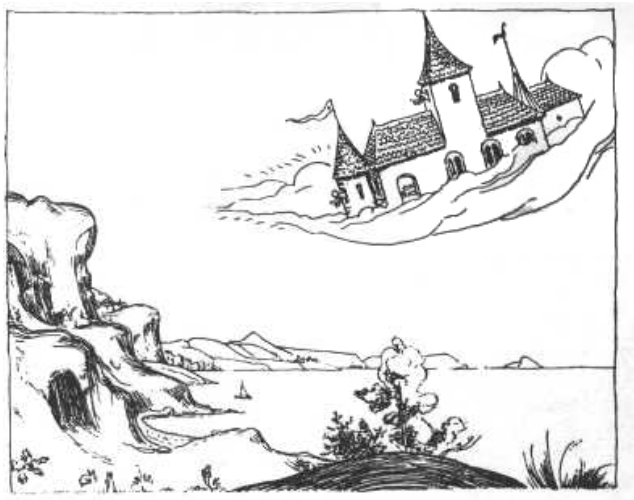
– Vipul Bansal, Microsoft WMI Blog, Jan 2009.

What does that mean anyway?

- It does not 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 **i**n **B**ackground?
- **S**illy **T**echnology **i**s **B**oring?
- **S**ipping **T**ea **i**n **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]

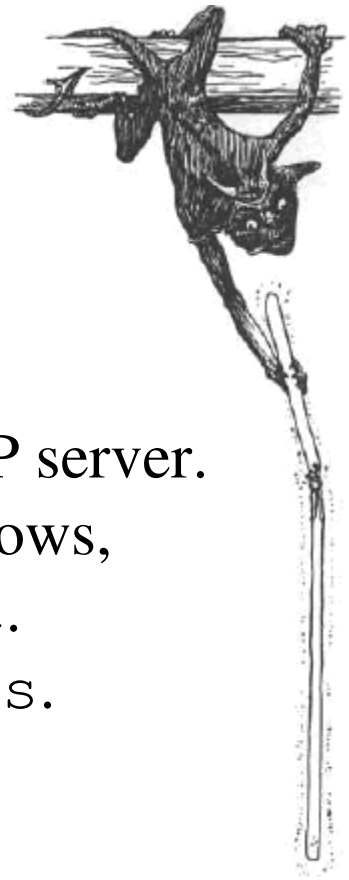
BITS: Background "Intelligent" Transfer Service

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]

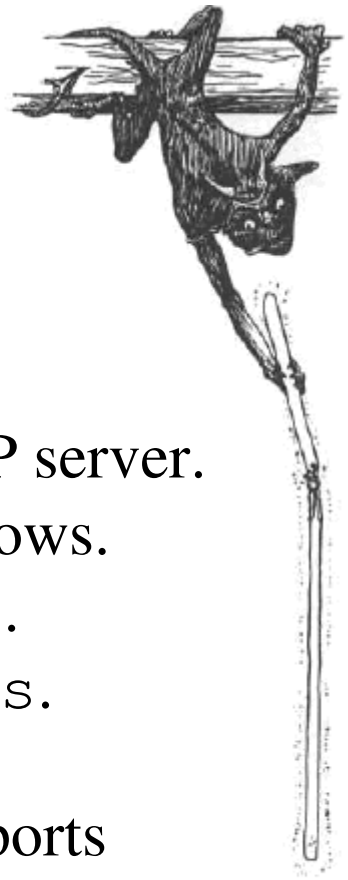


Do we care?

YAWP (Yet Another Windows Protocol)

- BITS Upload is supported in IIS,
  - and in Microsoft's “lightweight” HTTP server.
- It's convenient when working with Windows,
  - Not nearly as powerful as, eg., `rsync`.
  - Not as secure as `sftp`, `scp`, or `sshfs`.

BITS: Background "Intelligent" Transfer Service



Do we care?

YAWP (Yet Another Windows Protocol)

- BITS Upload is supported in IIS,
  - and in Microsoft's "lightweight" HTTP server.
- It's convenient when working with Windows.
  - Not nearly as powerful as, eg., `rsync`.
  - Not as secure as `sftp`, `scp`, or `sshfs`.

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

- `GET` support added to STiB,
- PeerDist included in the header,
- It works!

BITS: Background "Intelligent" Transfer Service

http://www.ubiqx.org/proj/STiB/

STiB is at version 0.2

It contains **`stibtest`**, which can:

- Send files using MS-BUP protocol,
- Get files using HTTP1.1,
  - Get a subrange of a requested file,
- Specify "peerdist" encoding when requesting all or part of a file.

Please download and test it.
Send patches.

Prequel

Pay Attention!

This is where it 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

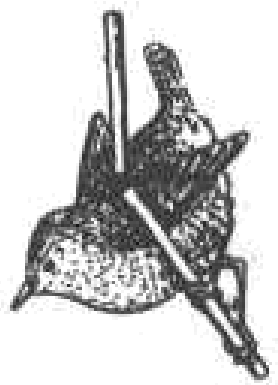
- supported in W2K8r2 servers,
- and Windows7 clients.

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 (peers)

- Request & receive content from content servers.

The Cache

- A copy of the original content, divided into segments and blocks; accessed via hash tags.

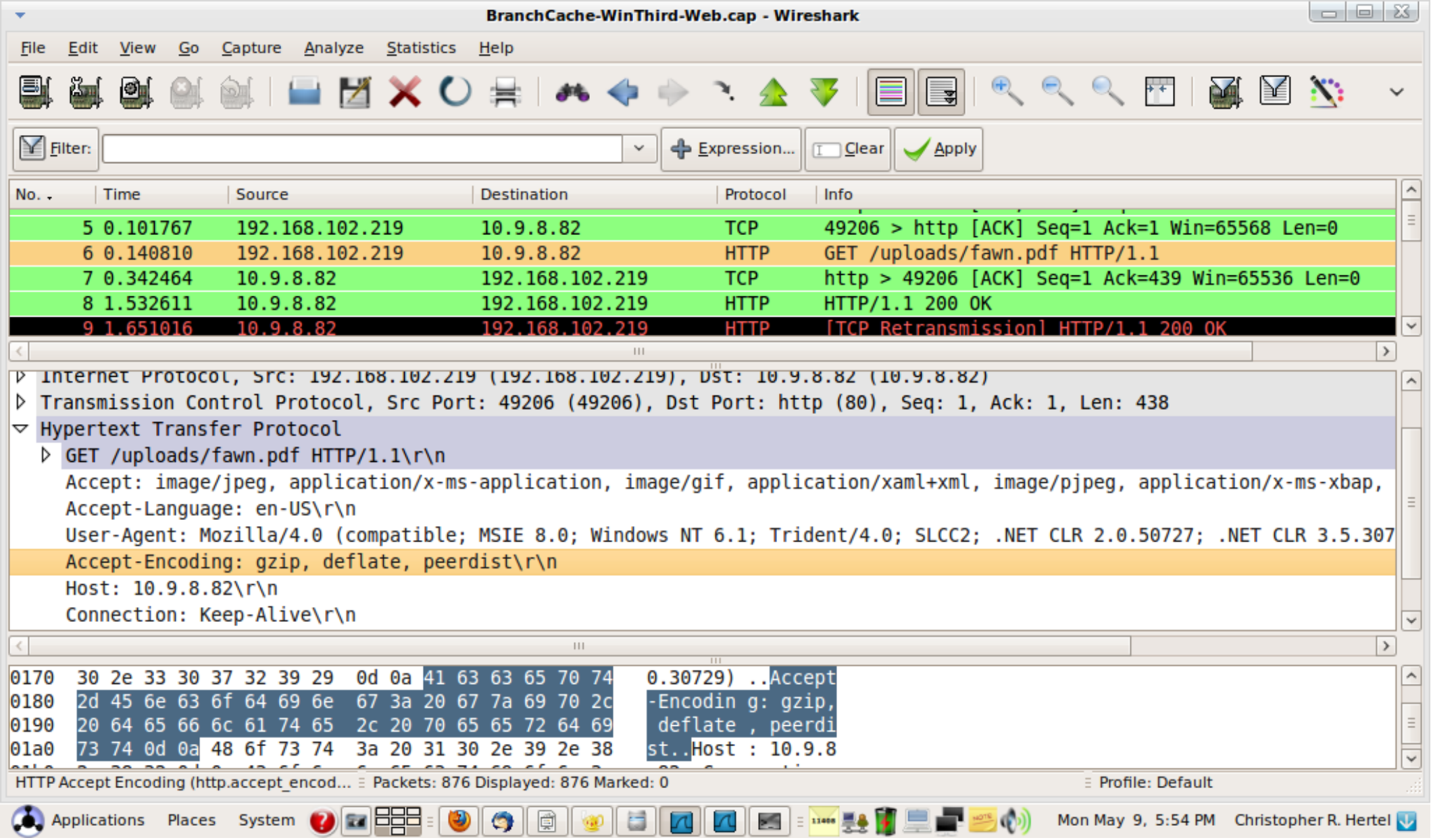
Prequel

Content Servers:

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

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

- If the tags are already calculated, they are returned by the BranchCache™-enabled server.
- Otherwise the content is returned, and the server (W2K8r2) calculates the tags for next time.



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

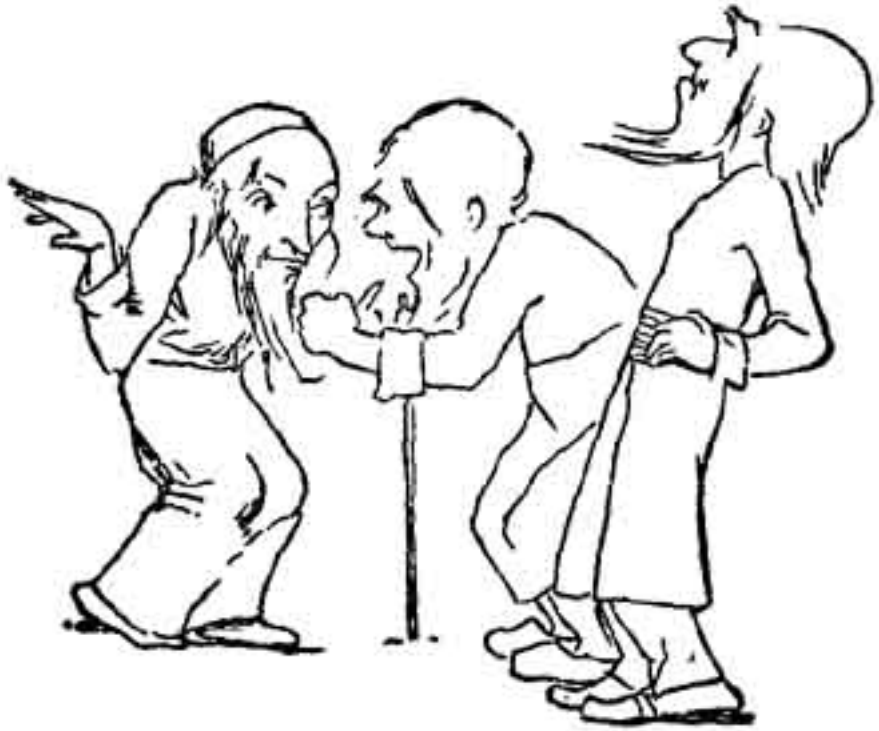
```
Accept-Encoding: gzip, deflate, peerdist\r\n
```

Prequel

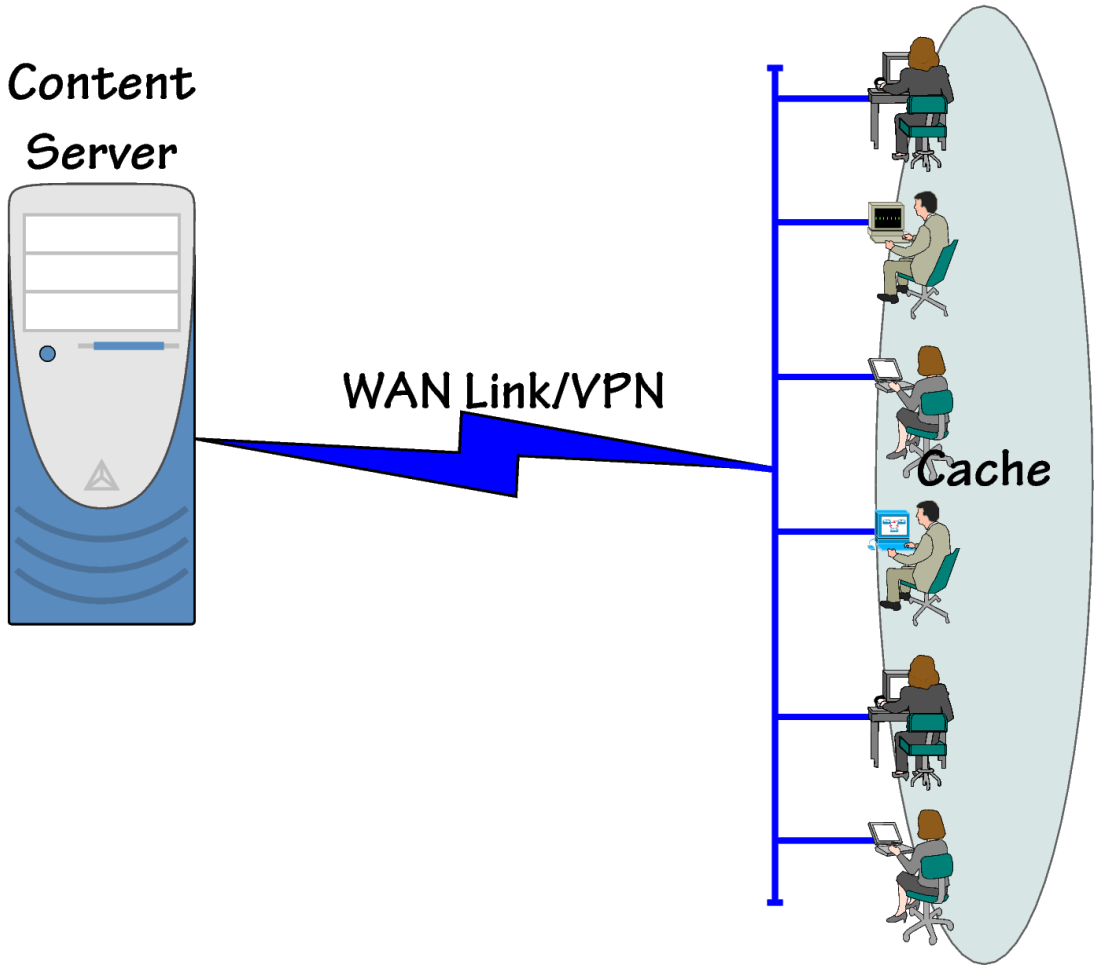
Client-side PeerDist Caching

There are two modes of operation:

- Distributed Mode
- Hosted Mode



Distributed Mode



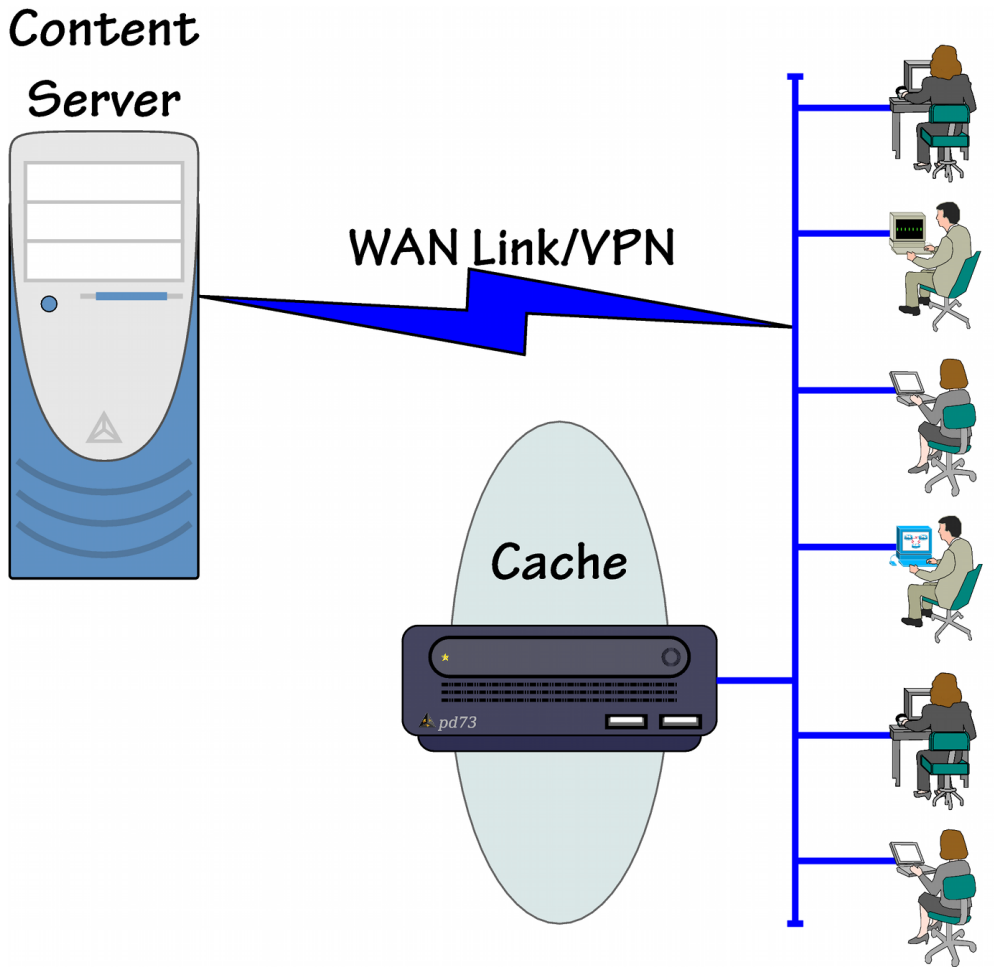
Prequel

Distributed Mode

- Each client keeps a local cache.
- A client requests PeerDist 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 CIFS Browse Service.

Hosted Mode



Prequel

Hosted Mode

- A client request tags from the content server
- The client then asks the local 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 the content server or cache server)
- Are 64K
  (or less, for the last block in a file only)

The block tag is an SHA$\{^{256}_{384}_{512}\}$ hash of the block.

Segments

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

Segments are identified by a hash of the 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. Peer Cache
- Implement a stand-alone peer caching client.
- Maybe a FUSE file system on top.

III. Cache Server
- Implement a Hosted Cache server.

Prequel

http://www.ubiqx.org/proj/Prequel/

Prequel does not have a release number yet.

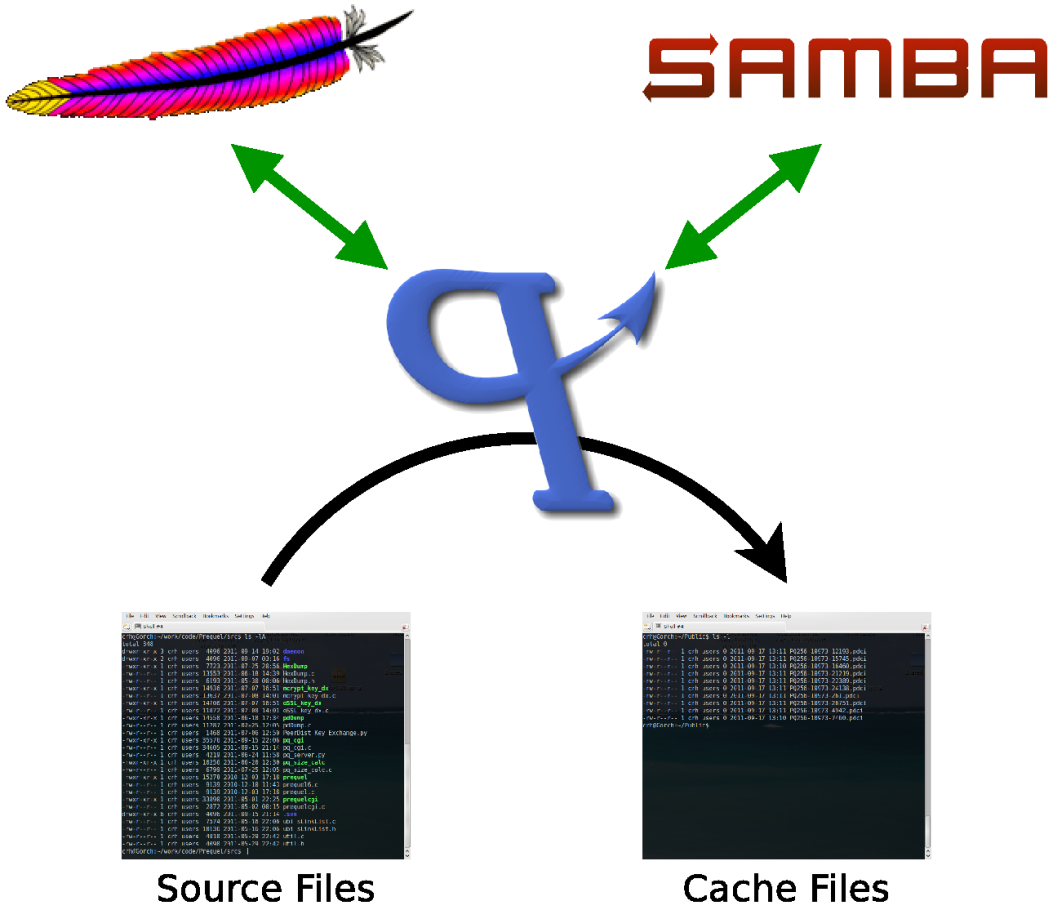
`pq_cgi` – CGI program to generate PeerDist Content Information.
- Tested with Apache.

`pdDump` – Pretty-print Content Information.

`*_key_dx` – Extract W2K8r2 Server Passphrase and Server Secret

Prequel Dæmon

Conceptual Overview



Prequel

Demo?

Other Stuff

CIFS.ORG

The End