BranchCache: Next Generation Branch Office Optimizations

Ravi Rao, Molly Brown
Microsoft Corporation
Agenda

- Problem background
- Solution framework
- Going Deep
  - Content Identification
  - SMB and HTTP protocol flows
  - Security framework
- Resources
Problem background

Thin, expensive WAN links between main office and branch offices

- High link utilization
- Poor application responsiveness
- Trend towards data centralization
Solution Tenets

Local

- **Distributed** – retrieve from other clients in the branch
- **Centralized** – retrieve from a “hosted cache” in the branch

Secured

- Client can only retrieve content locally if authorized by the content server
- All data transfers in the branch are encrypted

End to End

- Maintains protocol integrity
- Benefits from protocol optimizations
- Optimizes SSL, IPsec, SMB signing, HTTP, SMB
Distributed Cache

Main Office

Get

Branch Office

Get

Get

Get

Get

Data

Data
Hosted Cache

Main Office

Get

Get

Get

Branch Office

Search

Get

Offer

Search

Put

Storage Developer Conference 2009
© 2009 Insert Copyright information here. All rights reserved.
Overall framework

3rd Party Applications

Office  Copy File  Explorer  SharePoint  Silverlight  BITS  Flash  IE

SMB2  HTTP/1.1

BranchCache™
Deployment

Group Policy to enable clients

Install BranchCache™ feature R2 content servers

Optionally, install a hosted cache in your branch.
Content identifiers

**Hashes**
- Returned by server
- Segment hashes, Block hashes
- Up to 2000x data reduction on the WAN link
- Exact structure described in PCCRC

**Blocks**
- Unit of download

**Segments**
- Unit of discovery

**Content**
HTTP integration

IE

Open URL

Data

wininet

Hashlist

Data

Branch Cache

“BranchCache Capable”

Data

Get data

Hashlist

IIS

http.sys

Data

Branch Cache

Hashlist

Data

Hashlist

Data

Storage Developer Conference 2009
© 2009 Insert Copyright information here. All rights reserved.
### First Access – HTTP Request

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.435444</td>
<td>TCP</td>
<td>TCP: Flags=...A..., SrcPort=51811, DstPort=HTTP(80), PayloadLen=0, Seq=342927444</td>
</tr>
<tr>
<td>3.471576</td>
<td>HTTP</td>
<td>HTTP: Request, GET /welcome.png</td>
</tr>
<tr>
<td>3.472643</td>
<td>HTTP</td>
<td>HTTP: Response, HTTP/1.1, Status Code = 200, URL: /welcome.png</td>
</tr>
<tr>
<td>3.537146</td>
<td>PCCRDP</td>
<td>PCCRDP: SOAP Message: WSD message</td>
</tr>
<tr>
<td>3.541229</td>
<td>PCCRDP</td>
<td>PCCRDP: SOAP Message: WSD message</td>
</tr>
</tbody>
</table>

#### Frame Details

- **TCP**: Flags=...A..., SrcPort=51811, DstPort=HTTP(80), PayloadLen=0, Seq=342927444
- **HTTP**: Request, GET /welcome.png
  - **Command**: GET
  - **URI**: /welcome.png
  - **ProtocolVersion**: HTTP/1.1
  - **Accept**: image/jpeg, application/x-ms-application, image/gif, Accept-Language: en-US
  - **UserAgent**: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Accept-Encoding: gzip, deflate, peerdist
  - **Host**: anishd10
  - **Connection**: Keep-Alive
  - **Pccrtp**: Version=1.0
  - **XP2PPeerDist**: Version=1.0
  - **HeaderEnd**: CRLF
First Access – HTTP Response

PCCRTP

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.471526</td>
<td>HTTP</td>
<td>HTTP Request, GET /welcome.png</td>
</tr>
<tr>
<td>3.472643</td>
<td>HTTP</td>
<td>HTTP Response, HTTP/1.1, Status Code = 200, URL: /welcome.png</td>
</tr>
<tr>
<td>3.537146</td>
<td>PCCRD</td>
<td>PCCRD:SOAP Message: WSD message</td>
</tr>
</tbody>
</table>

Frame Details

- Tcp: Flags=...AP..., SrcPort=HTTP(80), DstPort=51811, PayloadLength=198
- Http: Response, HTTP/1.1, Status Code = 200, URL: /welcome.png
  - ProtocolVersion: HTTP/1.1
  - StatusCode: 200, Ok
  - Reason: OK
  - ContentLength: 198
  - ContentType: image/png
  - ContentEncoding: peerdist
  - Last-Modified: Mon, 10 Aug 2009 20:17:23 GMT
  - Accept-Ranges: bytes
  - ETag: "6ab31d92f719ca1:0"
  - Server: Microsoft-IIS/7.5
  - Pccrtp: Version=1.0, ContentLength=184946
    - XP2PPeerDist: Version=1.0, ContentLength=184946
  - Date: Tue, 11 Aug 2009 00:05:02 GMT
  - HeaderEnd: CRLF
  - payload: HttpContentType = image/png
  - HTTPPayloadLine: PCCRC
First Access – Probe
PCCRD

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.472643</td>
<td>HTTP</td>
<td>HTTP:Response, HTTP/1.1, Status Code = 200, URL: /welcome.png</td>
</tr>
<tr>
<td>3.537146</td>
<td>PCCRD</td>
<td>PCCRD:SOAP Message: WSD message</td>
</tr>
<tr>
<td>3.541229</td>
<td>PCCRD</td>
<td>PCCRD:SOAP Message: WSD message</td>
</tr>
<tr>
<td>3.661625</td>
<td>PCCRD</td>
<td>PCCRD:SOAP Message: WSD message</td>
</tr>
<tr>
<td>3.755178</td>
<td>PCCRD</td>
<td>PCCRD:SOAP Message: WSD message</td>
</tr>
</tbody>
</table>

Frame Details

Frame: Number = 77, Captured Frame Length = 856, MediaType = ETHERNET

Ethernet: Etype = Internet IP (IPv4), DestinationAddress: [01-00-5E-7F-FF-FA]
Ipv4: Src = 157.59.24.118, Dest = 239.255.255.250, Next Protocol = UDP, Pcap
Udp: SrcPort = 56021, DstPort = 3702, Length = 822
Soap: xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:wsa="http:
Pccrd: SOAP Message: WSD message

ProbeMessage: XML: <wsd:Probe>
  + STag: <wsd:Probe>
  + TypesField: XML: <wsd:Types>
  + ScopesField: XML: <wsd:Scopes>
  + ETag: </wsd:Probe>

PeerDist:PeerDistData
  2A8A065AA9CB17A8FE4CB88AA3CF4FD1
First Access – Repeat HTTP Request for missing data (PCCRTP)

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.755178</td>
<td>PCCRD</td>
<td>PCCRD:SOAP Message: WSD message</td>
</tr>
<tr>
<td>3.768576</td>
<td>TCP</td>
<td>TCP:[ReTransmit #76]Flags=...AP..., SrcPort=HTTP(80), DstPort=51811, PayloadLen=</td>
</tr>
<tr>
<td>3.768772</td>
<td>TCP</td>
<td>TCP:Flags=...A..., SrcPort=51811, DstPort=HTTP(80), PayloadLen=0, Seq=3429274600</td>
</tr>
<tr>
<td>3.803702</td>
<td>HTTP</td>
<td>HTTP:Request, GET /welcome.png</td>
</tr>
<tr>
<td>3.819178</td>
<td>HTTP</td>
<td>HTTP:Response, HTTP/1.1, Status Code = 200, URL: /welcome.png</td>
</tr>
<tr>
<td>4.020530</td>
<td>TCP</td>
<td>TCP:Flags=...A..., SrcPort=51811, DstPort=HTTP(80), PayloadLen=0, Seq=3429275200</td>
</tr>
</tbody>
</table>

Frame Details

- **Tcp**: Flags=...AP..., SrcPort=51811, DstPort=HTTP(80), PayloadLen= 
- **Http**: Request, GET /welcome.png

Command: GET

- **URI**: /welcome.png
- **ProtocolVersion**: HTTP/1.1
- **Accept-Encoding**: gzip, deflate

**Pccrtp**: Version=1.0, MissingDataRequest=true

**XP2PPeerDist**: Version=1.0, MissingDataRequest=true

**UserAgent**: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1, 
- **Host**: anishd10
- **Connection**: Keep-Alive
- **HeaderEnd**: CRLF
Second Access – HTTP
PCCRTP, PCCRD

- The HTTP request and response are identical to the first access case
  - Request indicates BranchCache capability
  - Response includes the content identifiers encoded using PCCRC
- The Probe on the network is also identical to the first access case
  - PCCRD is used to look up the segment ID

- What is different is that a response is received this time from a peer
### Second Access – Probe Match

**PCCRD**

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.625723</td>
<td>PCCRD</td>
<td>PCCRD:SOAP Message: WSD message</td>
</tr>
<tr>
<td>2.673355</td>
<td>PCCRD</td>
<td>PCCRD:SOAP Message: WSD Probe-Match message</td>
</tr>
</tbody>
</table>

#### Frame Details

```xml
  <soap:Header>
    <ns2:ProbeMatchRequest xmlns:ns2="urn:ebxml:nas:ProbeMatch:1.0"/>
  </soap:Header>
  <soap:Body>
    <ns2:ProbeMatchResponse xmlns:ns2="urn:ebxml:nas:ProbeMatch:1.0"> 
      <ns2:ProbeMatches>
        <ns2:ProbeMatch>
          <ns2:PeerDist>
            <ns2:PeerDistData/>
          </ns2:PeerDist>
          <ns2:Scopes>
            <ns2:Scope>2A8A065AA9CB17A8FE4CB88AA</ns2:Scope>
          </ns2:Scopes>
          <ns2:XAddr>157.59.25.43:80</ns2:XAddr>
          <ns2:CustomXML>1</ns2:CustomXML>
        </ns2:ProbeMatch>
      </ns2:ProbeMatches>
    </ns2:ProbeMatchResponse>
  </soap:Body>
</soap:envelope>
```
Second Access – Block request

PCCRR

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.789555</td>
<td>HTTP</td>
<td>HTTP: Request, POST /116B50EB-ECE2-41ac-8429-9F9E963361B7/</td>
</tr>
<tr>
<td>2.789801</td>
<td>PCCRR</td>
<td>PCCRR: Message Size is 68, Body message is GETBLKS</td>
</tr>
<tr>
<td>2.790038</td>
<td>TCP</td>
<td>TCP: Flags=...A,..., SrcPort=HTTP(80), DstPort=51800, PayloadLen=0, Seq=3734051367, Ack=12963</td>
</tr>
</tbody>
</table>

**Frame Details**

- **Http**: HTTP Payload, URL: /116B50EB-ECE2-41ac-8429-9F9E963361B7/
- **PCCRR**: Message Size is 68, Body message is GETBLKS

**Request Message**

- ProtVer: Version is 1.0
- MsgType: Body message is GETBLKS
- MsgSize: 68 (0x44)
- **CryptoAlgoId**: AES 128-bit encryption

**Body**: Body message is GETBLKS

- GetBlocks: Request block range count 1
  - SizeOfSegmentID: 32 (0x20)
  - **SegmentID**: 0x2a8a065aa9cb17a8fe4cb88aa3cf4fdd51b8ded17ef6d0a0e7bed
  - ZeroPad: 0 Bytes
  - ReqBlockRangeCount: 1 (0x1)
  - ReqBlockRanges: The first block is 0, This range contains 1 blocks
    - **Index**: 0 (0x0)
    - **Count**: 1 (0x1)
    - **SizeOfDataForVrfBlock**: 0 (0x0)
Second Access – Block Response

PCCRR

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.793739</td>
<td>HTTP</td>
<td>HTTP:Response, HTTP/1.1, Status Code = 200, URL: /116b50eb-ece2-41ac-8429-9f9e63361b7/</td>
</tr>
<tr>
<td>2.794109</td>
<td>PCCRR</td>
<td>PCCRR: Message Size is 65640, Body message is BLK, Need reassemble</td>
</tr>
<tr>
<td>2.794280</td>
<td>TCP</td>
<td>TCP: Flags=....A...., SrcPort=51800, DstPort=HTTP(80), PayloadLen=0, Seq=12963813</td>
</tr>
</tbody>
</table>

**Frame Details**

- **HTTP Payload, URL:** /116b50eb-ece2-41ac-8429-9f9e63361b7/
- **PCCRR:** Message Size is 65640, Body message is BLK, Need reassemble
  - **ResponseMessage:** Message Size is 65640, Body message is **BLK**, Need
  - **TranspotResponseHeader:** Total message Size is 65640
  - **MessageHeader:** Message Size is 65640, Body message is BLK, Need
    - **ProtVer:** Version is 1.0
    - **MsgType:** Body message is BLK
    - **MsgSize:** 65640 (0x10068)
    - **CryptoAlgoId:** AES 128-bit encryption
  - **Body:** Body message is BLK
  - **Blk:** Block Index 0, Next Block Index 1
    - **SizeOfSegmentID:** 32 (0x20)
    - **SegmentID:** 0x2a8a065aa9cb17a8fe4cb88aa3cf4fd51b8ded17ef6
    - **Pad:** 0 Bytes
    - **BlockIndex:** 0 (0x0)
    - **NextBlockIndex:** 1 (0x1)
    - **SizeOfBlock:** 65552 (0x10010)
Since block size is 64KB and content size is ~185KB, there are three blocks that need to be retrieved

Below are the final two block requests

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol...</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.823735</td>
<td>HTTP</td>
<td>HTTP:Request, POST /116B50EB-ECE2-41ac-8429-9F9E963361B7/</td>
</tr>
<tr>
<td>2.824192</td>
<td>PCCRR</td>
<td>PCCRR:Message Size is 68, Body message is GETBLKS</td>
</tr>
<tr>
<td>2.824372</td>
<td>TCP</td>
<td>TCP:Flags=....A...., SrcPort=HTTP(80), DstPort=51800, PayloadLen=0</td>
</tr>
<tr>
<td>2.825401</td>
<td>HTTP</td>
<td>HTTP:Response, HTTP/1.1, Status Code = 200, URL: /116B50EB-ECE2-41ac-8429-9F9E963361B7/</td>
</tr>
<tr>
<td>2.827390</td>
<td>PCCRR</td>
<td>PCCRR:Message Size is 65640, Body message is BLK, Need reassemble</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol...</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.924460</td>
<td>HTTP</td>
<td>HTTP:Request, POST /116B50EB-ECE2-41ac-8429-9F9E963361B7/</td>
</tr>
<tr>
<td>2.925026</td>
<td>PCCRR</td>
<td>PCCRR:Message Size is 68, Body message is GETBLKS</td>
</tr>
<tr>
<td>2.925345</td>
<td>TCP</td>
<td>TCP:Flags=....A...., SrcPort=HTTP(80), DstPort=51800, PayloadLen=0,</td>
</tr>
<tr>
<td>2.930422</td>
<td>HTTP</td>
<td>HTTP:Response, HTTP/1.1, Status Code = 200, URL: /116B50EB-ECE2-41ac-8429-9F9E963361B7/</td>
</tr>
<tr>
<td>2.932160</td>
<td>PCCRR</td>
<td>PCCRR:Message Size is 53976, Body message is BLK, Need reassemble</td>
</tr>
</tbody>
</table>
SMB Integration

Application

CSC Driver

SMB Client Driver

Branch Cache

CSC Service

SMB Hash Generation Service

HashGen Utility

Hashlist

Data

Data

Request Hashes

Request Hashes

Hashlist

File Data

Hashlist

Data

Data

Hashlist

ReadFile

Prefetch File

Save hashes

Access hashes

Generate or update hash

Generate or update hash

Storage Developer Conference 2009
© 2009 Insert Copyright information here. All rights reserved.
Tree Connect Response

SMB 2.1

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.607090</td>
<td>SMB2</td>
<td>SMB2:C TREE CONNECT (0x3), Path=\paswan_test1\F</td>
</tr>
<tr>
<td>7.607128</td>
<td>SMB2</td>
<td>SMB2:R TREE CONNECT (0x3), TID=0x5</td>
</tr>
<tr>
<td>7.607713</td>
<td>SMB2</td>
<td>SMB2:C CREATE (0x5), Context=DhNQ, Context=MxAc, Context=QFid</td>
</tr>
</tbody>
</table>

Frame Details:

- Frame: Number = 311, Captured Frame Length = 158, MediaType = ETHERNET
- Ethernet: EthType = IPv6, DestinationAddress:[00-0D-56-4B-00-D0], SourceAddress:[00-11-06-DF-00-0F]
- IPv6: Next Protocol = TCP, Payload Length = 104
- TCP: Flags=...AF..., SrcPort=Microsoft-DS(445), DstPort=58584, PayloadLen=84, Seq=214
- SMBOverTCP: Length = 80
- SMB2: R TREE CONNECT (0x3), TID=0x5
  - SMBHeader: R TREE CONNECT (0x3)
  - RTreeConnect: 0x1
    - StructureSize: 16 (0x10)
    - ShareType: Disk (0x01)
    - Reserved: 0 (0x00)
  - ShareFlags: 0000 0000 0000 0000 0132 (0x2000)
    - SHI1000_FLAGS_DFS: 0000 0000 0000 0000 0000 0000 0000 0000 (0x00)
    - SHI1000_FLAGS_DFS_ROOT: 0000 0000 0000 0000 0000 0000 0000 0000 (0x00)
    - Reserved_bits3_4: 0000 0000 0000 0000 0000 0000 0000 0000 (0x00)
    - SHAREFLAGS_CACHING_TYPE: 0000 0000 0000 0000 0000 0000 0000 0000 (0x00)
    - SHI1000_FLAGS_RESTRICT_EXCLUSIVE_opens: 0000 0000 0000 0000 0000 0000 0000 0000 (0x00)
    - SHI1000_FLAGS_FORCE_SHARED_DELETE: 0000 0000 0000 0000 0000 0000 0000 0000 (0x00)
    - SHI1000_FLAGS_ALLOW_NAMESPACE_CACHING: 0000 0000 0000 0000 0000 0000 0000 0000 (0x00)
    - SHI1000_FLAGS_ACCESS_BASED_DIRECTORY_ENUM: 0000 0000 0000 0000 0000 0000 0000 0000 (0x00)
    - SHI1000_FLAGS_FORCE_LEVEL1_UNLOCK: 0000 0000 0000 0000 0000 0000 0000 0000 (0x00)
    - SHI1000_FLAGS_ENABLE_HASH: 0000 0000 0000 0000 0000 0000 0000 0001 (0x01)
    - Reserved_bits15_32: 0000 0000 0000 0000 0000 0000 0000 0000 (0x00)

Share Supports Hash Retrieval
Hash Request from Client
SMB 2.1

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.759353</td>
<td>SMB2</td>
<td>SMB2:R QUERY INFORMATION (0x10), FileName = Apps\vxutil.pdb@#352</td>
</tr>
<tr>
<td>7.763549</td>
<td>SMB2</td>
<td>SMB2:C IOCTL (0xb), FID=0xFFFFF000000019</td>
</tr>
<tr>
<td>7.807476</td>
<td>SMB2</td>
<td>SMB2:R - NT Status: System - Error, Code = (41217) Unhandled Error code IOCTL (0xb)</td>
</tr>
</tbody>
</table>

Frame Details

- Frame: Number = 363, Captured Frame Length = 222, MediaType = ETHERNET
- Ethernet: Etype = IPv6, DestinationAddress=00:11-D8-BB-BA-CC, SourceAddress=00:0D-56-C
- IPv6: Next Protocol = TCP, Payload Length = 168
- TCP: Flags=...AP..., SrcPort=58584, DstPort=Microsoft-DS(445), PayloadLen=148, Seq=3188
- SMBOverTCP: Length = 144
- SMB2: C IOCTL (0xb), FID=0xFFFFF000000019
  - SMBIdentifier: SMB
  - SMB2Header: C IOCTL (0xb)
    - CIOctl:
      - StructureSize: 57 (0x39)
      - Reserved: 0 (0x0)
      - Code: 0x001441BB - FSCTL_SRV_READ_HASH - Issues an implementation-specific file
  - FileId: Persistent: 0x230000000000, Volatile: 0xFFFFF000000019
    - InputOffset: 120 (0x78)
    - InputCount: 24 (0x18)
    - MaxInputResponse: 0 (0x0)
    - OutputOffset: 120 (0x78)
    - OutputCount: 0 (0x0)
    - MaxOutputResponse: 64512 (0xF000)
    - Flags: (00000000000000000000000000000001) FSCTL request
      - Reserved2: 0 (0x0)
  - SMB2ICtlSrvReadHashReq:
    - padding: Binary Large Object (4 Bytes)
### Hash Response from Server

**SMB 2.1**

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.755649</td>
<td>SMB2</td>
<td>SMB2.C IOCTL (0xb), FID=0x000000019</td>
</tr>
<tr>
<td>7.807476</td>
<td>SMB2</td>
<td>SMB2.R - NT Status: System - Error, Code = (41217) Unhandled Error code IOCTL (0xb)</td>
</tr>
</tbody>
</table>

**Frame Details**

- Frame: Number = 364, Captured Frame Length = 151, MediaType = ETHERNET
- SMB2: R - NT Status: System - Error, Code = (41217) Unhandled Error code IOCTL (0xb)
  - SMBIdentifier: SMB
  - SMB2Header: R IOCTL (0xb)
    - StructureSize: 64 (0x40)
    - Epoch: 1 (0x1)
    - Status: 0xC002A101, Facility = FACILITY_SYSTEM, Severity = STATUS_SEVERITY_ERROR,

<table>
<thead>
<tr>
<th>Time Offset</th>
<th>Protocol Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.766528</td>
<td>SMB2</td>
<td>SMB2.C IOCTL (0xb), FID=0x0000000000000000</td>
</tr>
<tr>
<td>8.777703</td>
<td>SMB2</td>
<td>SMB2.R IOCTL (0xb), FileName= Apps\vxutil\pdb@#352</td>
</tr>
<tr>
<td>8.807465</td>
<td>SMB2</td>
<td>SMB2.C_READ (0x8), FID=0xffffffff00000000 (Apps\vxutil\pdb@#352), 0x00 bytes from offset 1372</td>
</tr>
</tbody>
</table>

**Frame Details**

- Frame: Number = 2051, Captured Frame Length = 1214, MediaType = ETHERNET
- SMB2: R IOCTL (0xb), FileName= Apps\vxutil\pdb@#352
  - SMBIdentifier: SMB
  - SMB2Header: R IOCTL (0xb)
    - StructureSize: 64 (0x40)
    - Epoch: 1 (0x1)
    - Status: 0x0, Facility = FACILITY_SYSTEM, Severity = STATUS_SEVERITY_SUCCESS,
How is SSL Optimized?

Client

- IE
- HTTP
- SSL
- Sockets
- IPsec

Data encrypted

Server

- IIS
- HTTP
- SSL
- Sockets
- IPsec

Data encrypted

Branch Cache

Data in clear

Data in clear

Data encrypted

Data encrypted

Data encrypted

Data encrypted

Data encrypted

Data encrypted
Security

Encryption key
\[ Ke = Kp \]

Segment Id (HoHoDk)
\[ \text{HMAC}(Kp, HoD+K) \]

Segment Secret (Kp)
\[ \text{HMAC}(Ks, HoD) \]

Segment hash of data
\[ HoD = \text{Hash} \text{ (Blockhashes)} \]

Block hashes
\[ \text{Hash} \text{ (block)} \]

Blocks

Client

Server secret
\[ Ks \]
Flow – a Security View

- Client requests data from the server, and indicates BranchCache capability
  - Server authorizes the client
  - Server retrieves content identifiers (PCCRC)
    - Block hashes, segment HoD, Kp
  - Server sends content identifiers on same channel as data
- Client computes a segment ID
  - Broadcasts on the local network
Flow, Continued

- Serving clients receive the broadcast
  - Decrypt the segment HoD from the segment ID
  - Respond with data availability
- Client requests blocks from the serving client
  - Serving client computes encryption key from Kp
  - Serving client encrypts each block with the encryption key
- Client receives the data
  - Decrypts the data
  - Validates block data against the block hash
  - If valid, returns to application
Security of Data at Rest

- **Clients**
  - Cache only contains content requested by the client
  - Data in cache ACL’d so that it is only accessible if authorized by the server
  - If data leakage is a concern, then use BitLocker or EFS

- **Hosted Cache**
  - Cache contains content requested by all branch clients
  - Use BitLocker or EFS to encrypt cache as necessary

- All data can be purged from the cache using netsh
To Summarize

- BranchCache™ reduces WAN bandwidth consumed by end users for intranet based HTTP and SMB traffic and improves end user experience.

- BranchCache™ accelerates delivery of encrypted and signed content such as when using HTTPS, IPsec, SMB signing and at the same time ensures authorization of users by the server at the central office.

- BranchCache™ doesn’t require additional equipment in the branch offices and can be easily managed using existing systems management technology such as group policy.

- BranchCache has a vibrant and growing ecosystem giving customers the choice to pick a solution that works best for their needs.
Resources

- Protocols
  - Content Identification (PCCRC)
  - Discovery (PCCRD)
  - Retrieval (PCCRR)
  - Hosted Cache Offer (PCHC)
  - HTTP extensions for BranchCache (PCCRTP)
  - SMB extensions for BranchCache (SMB2.1)

- Website
  - http://www.branchcache.com

- Collateral
  - BranchCache Technical Overview Whitepaper
  - BranchCache Executive Overview
  - BranchCache Early Adopter’s Guide

- Case studies
  - Sporton International
  - Convergent Computing

- Email
  - branch@microsoft.com
“We are improving the efficiency of our branch offices and saving bandwidth by using BranchCache in Windows Server 2008 R2 and Windows 7,” said Lukas Kucera, IT services manager of Lukoil CEEB, one of the largest integrated oil and gas companies in the world. “Some of our smaller facilities, such as the office in Slovakia and the storage terminal in Belgium, have just five to 10 users, so it’s not efficient to deploy a file server on-site, but it consumes bandwidth to have them continually accessing files from the main servers. BranchCache is the perfect solution.”

“Taking advantage of the BranchCache feature in Windows Server 2008 R2, we can spend $20,000 rather than $50,000 per year on bandwidth by postponing our expansion schedule.”
David Feng, IT Director, Sporton International

Convergent Computing (CCO) wanted to improve remote network access for its mobile users. Using the DirectAccess and BranchCache™ features in Windows Server® 2008 R2 and Windows 7, CCO has simplified remote connection to its network and sped the downloading of important files. It has cut costs by eliminating its virtual private network and has seen a 43 percent savings in wide area network (WAN) bandwidth.