IETF 75
Stockholm

NFSv4 Working Group Meeting

July 29, 2009

Agenda
9:00 – 11:30 am

- 9:00 Intro/Blue Sheets/Note Well/Agenda Bash (Pawlowski)
- 9:05 Server Side Copy Offload - (Lentini)
  - draft-lentini-nfsv4-server-side-copy-02.txt
- 9:25 Federated FS - (Lentini)
  - "Using DNS SRV to Specify a Global File Name Space with NFS version 4"
    - <draft-ietf-nfsv4-federated-fs-dns-srv-namespace-01.txt>
  - "Administration Protocol for Federated Filesystems"
    - <draft-ietf-nfsv4-federated-fs-admin-01.txt>
  - "Requirements for Federated File Systems"
    - <draft-ietf-nfsv4-federated-fs-reqts-03.txt>
  - "NSDB Protocol for Federated Filesystems“
    - <draft-ietf-nfsv4-federated-fs-protocol-01.txt>
- 9:40 NFS operation over IPv4 and IPv6 (Alex RN)
  - <draft-alexrn-nfsv4-ipv6-00.txt>
- 10:10 NFSv4 Multi-Domain Access (Adamson)
  - <draft-adamson-nfsv4-multi-domain-access.txt>
- 10:25 Proposal for an NFSv4 extension to allow the use of NFS clients as pNFS data servers (Adamson)
  - <draft-myklebust-nfsv4-pnfs-backend-00.txt>
- 10:55 Access checks and pNFS (Sorin Faibish)
- 11:25 Wrapup (Pawlowski)
- 11:30 End

Note Well

From: http://www.ietf.org/about/note-well.html

Any submission to the IETF intended by the Contributor for publication as all or part of an IETF Internet-Draft or RFC and any statement made within the context of an IETF activity is considered an "IETF Contribution". Such statements include oral statements in IETF sessions, as well as written and electronic communications made at any time or place, which are addressed to:

- The IETF plenary session
- The IESG, or any member thereof on behalf of the IESG
- Any IETF mailing list, including the IETF list itself, any working group or design team list, or any other list functioning under IETF auspices
- Any IETF working group or portion thereof
- The IAB or any member thereof on behalf of the IAB
- The RFC Editor or the Internet-Drafts function

All IETF Contributions are subject to the rules of RFC 5378 and RFC 3979 (updated by RFC 4879).

Statements made outside of an IETF session, mailing list or other function, that are clearly not intended to be input to an IETF activity, group or function, are not IETF Contributions in the context of this notice.

Please consult RFC 5378 and RFC 3979 for details.

A participant in any IETF activity is deemed to accept all IETF rules of process, as documented in Best Current Practices RFCs and IESG Statements.

A participant in any IETF activity acknowledges that written, audio and video records of meetings may be made and may be available to the public.

Agenda
9:00 – 11:30 am

- 9:00 Intro/Blue Sheets/Note Well/Agenda Bash (Pawlowski)
- 9:05 Server Side Copy Offload - (Lentini)
  - draft-lentini-nfsv4-server-side-copy-02.txt
- 9:25 Federated FS - (Lentini)
  - "Using DNS SRV to Specify a Global File Name Space with NFS version 4"
    - <draft-ietf-nfsv4-federated-fs-dns-srv-namespace-01.txt>
  - "Administration Protocol for Federated Filesystems"
    - <draft-ietf-nfsv4-federated-fs-admin-01.txt>
  - "Requirements for Federated File Systems"
    - <draft-ietf-nfsv4-federated-fs-reqts-03.txt>
  - "NSDB Protocol for Federated Filesystems“
    - <draft-ietf-nfsv4-federated-fs-protocol-01.txt>
- 9:40 NFS operation over IPv4 and IPv6 (Alex RN)
  - <draft-alexrn-nfsv4-ipv6-00.txt>
- 10:10 NFSv4 Multi-Domain Access (Adamson)
  - <draft-adamson-nfsv4-multi-domain-access.txt>
- 10:25 Proposal for an NFSv4 extension to allow the use of NFS clients as pNFS data servers (Adamson)
  - <draft-myklebust-nfsv4-pnfs-backend-00.txt>
- 10:55 Access checks and pNFS (Sorin Faibish)
- 11:25 Wrapup (Pawlowski)
- 11:30 End

NFS Server-side Copy

James Lentini
jlentini@netapp.com

IETF 75 NFSv4 WG Meeting
July 29, 2009

Summary

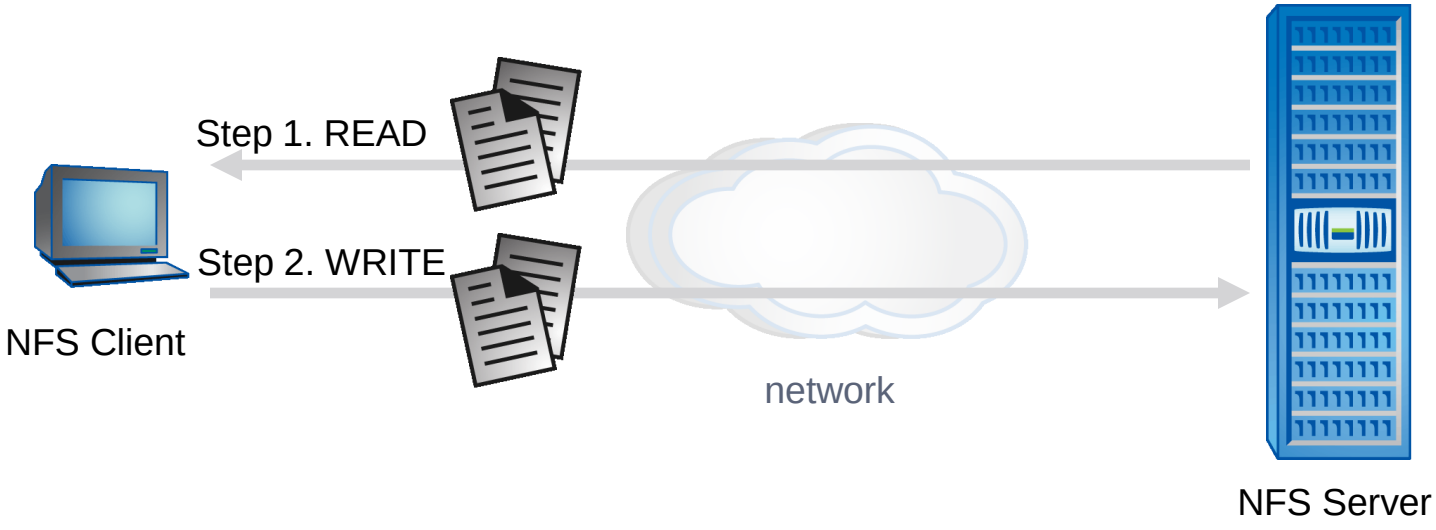
- The NFS server-side copy offload operations allow:
  - Copying a file on a single NFS server
  - Copying a file between two NFS servers.
- Server-side copy is a possible feature for NFSv4.2.

draft-lentini-nfsv4-server-side-copy

- IETF Individual I-D by
  - James Lentini
  - Mike Eisler
  - Rahul Iyer
  - Deepak Kenchamanna
  - Anshul Madan
- Extensive feedback and comments on the NFSv4 WG mailing list starting in April, 2009.

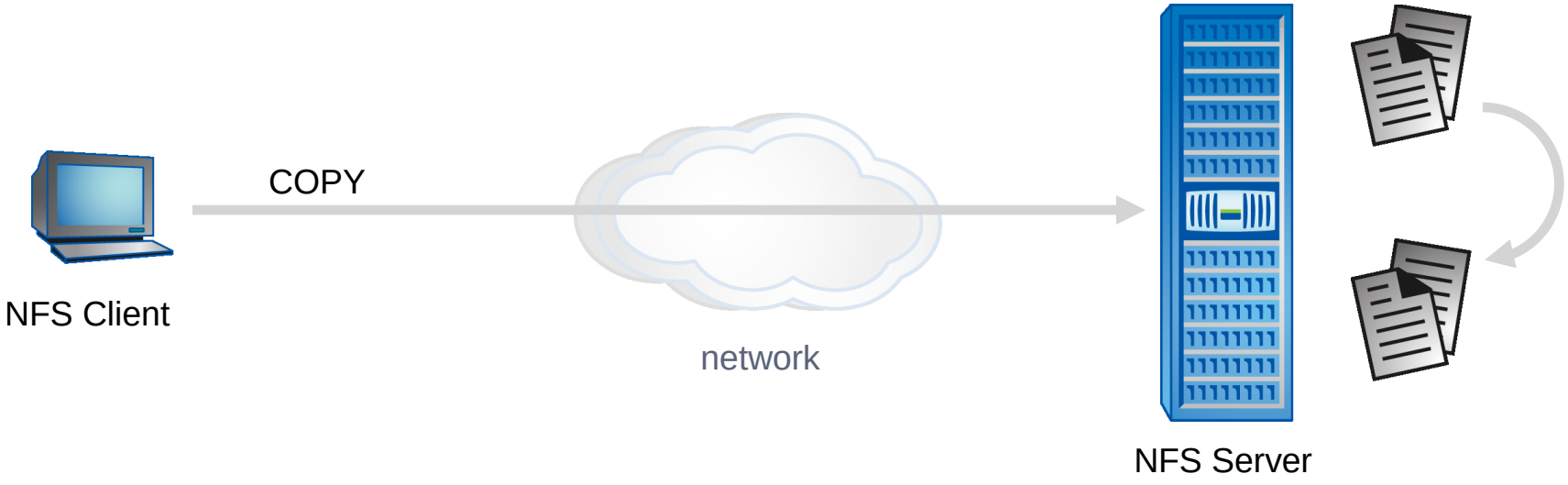
Copying with NFSv2/v3/v4[.1]

- The NFS client reads and writes the file over the network.
- Wastes client and network resources.



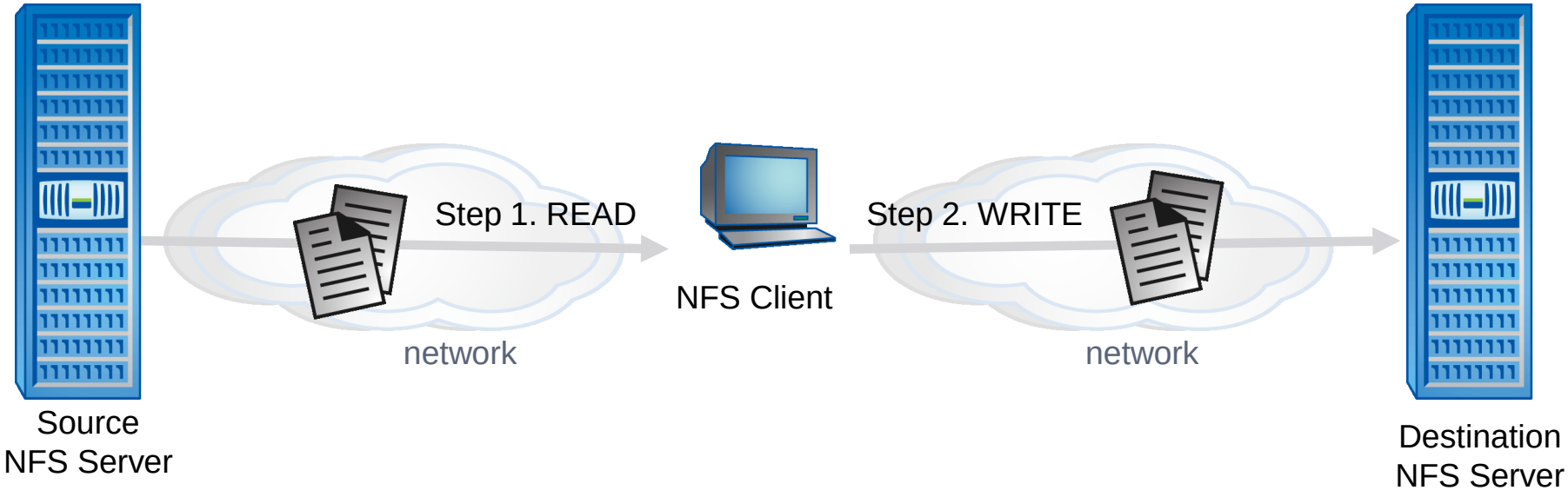
Copying with Server-side Offload

- The NFS client instructs the server to perform the copy.
- Saves client and network resources.



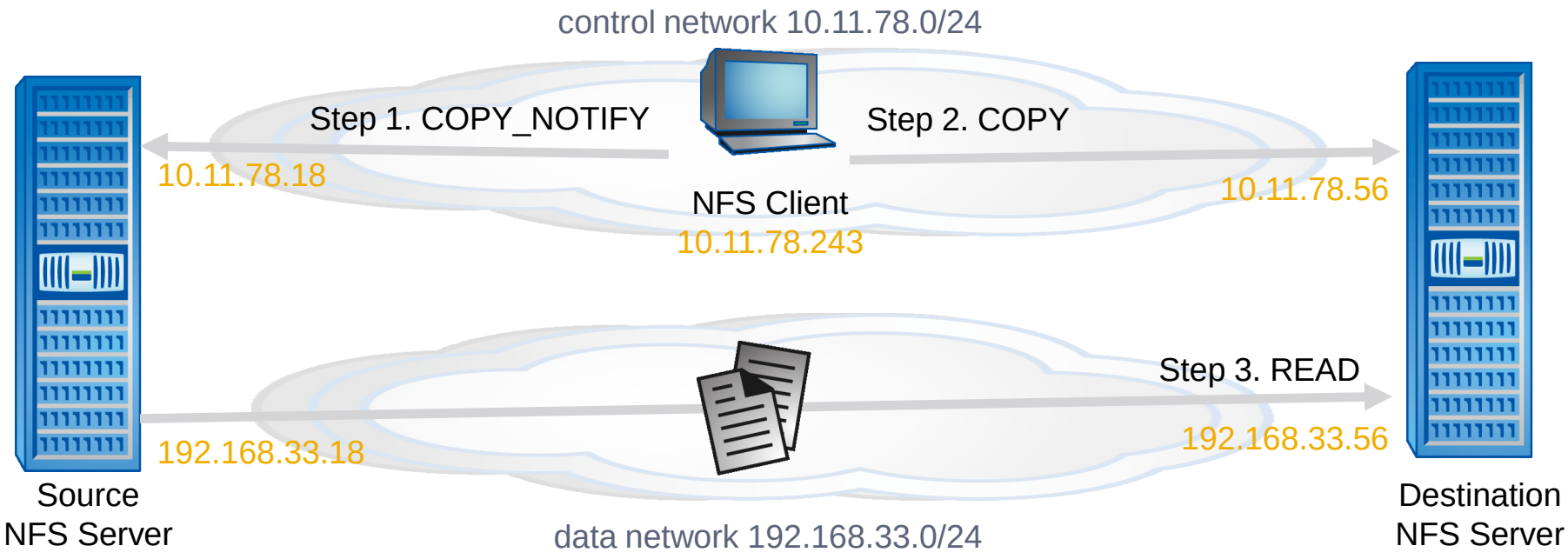
Copying between NFS Servers with NFSv2/v3/v4[.1]

- Client reads the file from the source server and writes the file to the destination server.
- Client is an extra network hop between source and destination.



Copying between NFS Servers with Server-side Offload

- Client sets up the copy between the servers.
- Removes client hop and (optionally) allows a high performance server data network to be used.



Uses Cases

- In general, this feature is useful whenever data is copied from one location to another.
- **File Restore**: It is useful when copying the contents of a snapshot.
- **Virtualized Environment:** Copy offload allows a hypervisor to efficiently:
  - Snapshot a VM
  - Clone a VM
  - Migrate a VM's storage

Design Choices (1)

- File versus Directory copies: proposal is to only support regular file copies.
  - Simplifies the protocol
  - Directory copies can by synthesized using multiple file copies and directory creates.
- Allow for asynchronous copies
  - Server decides if the copy will be asynchronous
- Support for
  - partial file copies
  - space reservations
  - guarded copies: fail if the destination file exists
  - metadata copy: duplicate all NFS attributes

Design Choices (2)

- Support intra- and inter- server copies
  - intra-server copy: source and destination on the same fileserver
  - inter-server copy: source and destination on different fileservers
    - Server-to-server protocol is NOT specified. A standard or proprietary protocol can be used.
    - Use a pull rather than push model

Server-to-Server Copy Protocol

- The proposal doesn't require a particular server-to-server copy protocol. The reply to COPY_NOTIFY contains URLs for the protocols the source server supports.
- NFSv4.1 is a good candidate for heterogeneous environments.
  - Standard protocols (FTP, HTTP, …) in addition to NFS are also supported.
- Proprietary protocols are possible in homogeneous environments:
  - source and destination may be using a clustered file system, no data may actually need to be copied or may have the same file system format allowing physical block-level replication.

Security

- Requirements:
  - flexible enough to allow for different server-to-server copy protocols.
  - compatible with using NFSv4.x as the server-to-server copy protocol.
  - no pre-configuration between the source and destination.
  - support mutual authentication between the participants (client, source server, and destination server).
- Two options:
  - RPCSEC_GSSv3 (work in progress) for strong security
  - host-based security (e.g. AUTH_SYS)

Suggested Next Steps

- Complete the RPCSEC_GSSv3 I-D.
- Consider making the copy-offload I-D a WG work item.
  - Is it within the scope of the current charter?
- Include copy-offload as a feature in NFSv4.2.

Questions? Comments?

Additional Information

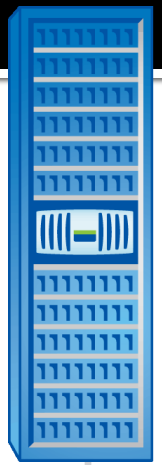
Protocol Diagrams

Operations

- **COPY_NOTIFY**: For inter-server copies, the client sends this operation to the source server to notify it of a future file copy from a given destination server for the given user.
- **COPY_REVOKE**: Also for inter-server copies, the client sends this operation to the source server to revoke permission to copy a file for the given user.
- **COPY**: Used by the client to request a file copy.
- **COPY_ABORT**: Used by the client to abort an asynchronous file copy.
- **COPY_STATUS**: Used by the client to poll the status of an asynchronous file copy.
- **CB_COPY**: Used by the destination server to report the results of an asynchronous file copy to the client.

Synchronous Intra-server Copy

NFS Client



NFS Server

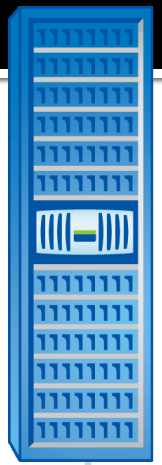
ONC RPC Call - COMPOUND { COPY }

ONC RPC Reply - COMPOUND { COPY }

Server performs copy and then replies to client

Asynchronous Intra-server Copy

NFS Client



NFS Server

ONC RPC Call - COMPOUND { COPY }

ONC RPC Reply - COMPOUND { COPY }

Server starts the copy and replies to client.

ONC RPC Call - COMPOUND { COPY_STAUS }

ONC RPC Reply - COMPOUND { COPY_STATUS }

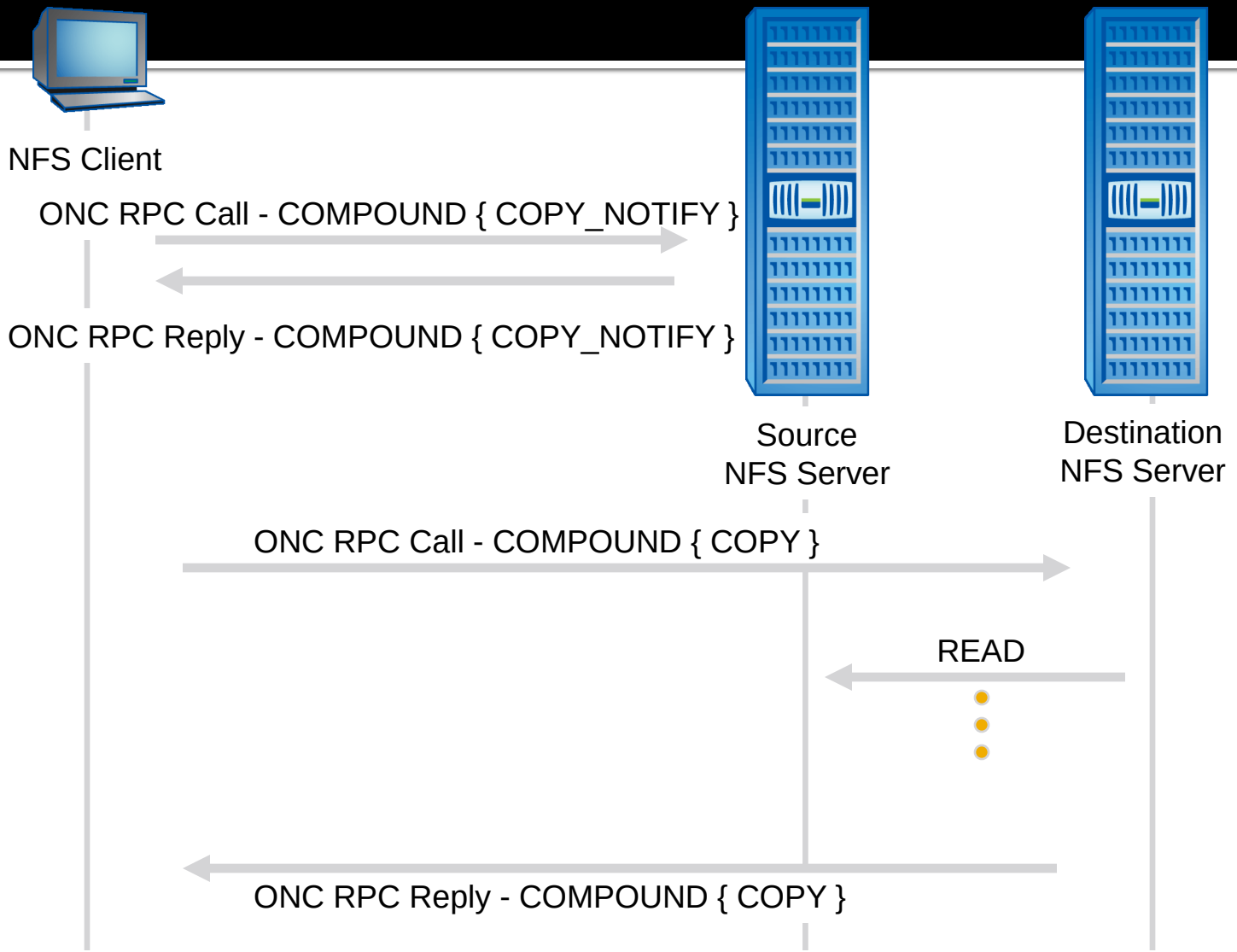
Client polls for status. Client can also cancel copy with COPY_ABORT.

ONC RPC Call – CB_COMPOUND { CB_COPY }

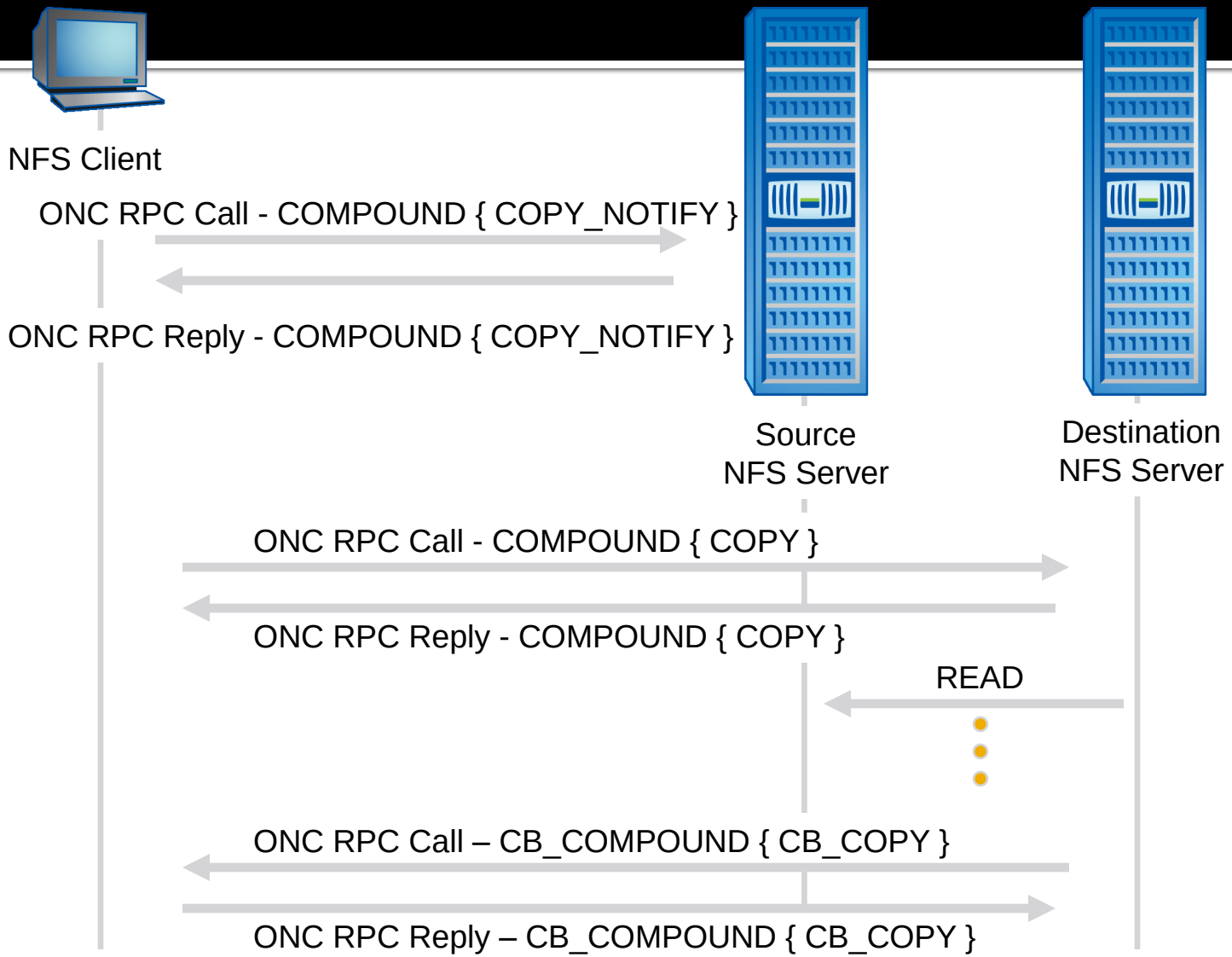
ONC RPC Reply – CB_COMPOUND { CB_COPY }

Server issues callback with final results of the copy

Synchronous Inter-server Copy



Asynchronous Inter-server Copy



Additional Information

Security

RPCSEC_GSSv3 Security (1)

- We propose 3 new RPCSEC_GSSv3 privileges:
  - copy_from_auth_priv: established by the client on the source server to allow a copy operation from the specified destination server on behalf of the given user.
  - copy_to_auth_priv: established by the client on the destination server to allow a copy operation from the specified source server on behalf of the given user.
  - copy_confirm_auth_priv: for ONC RPC server-to-server copy protocols, established by the destination server on the source server to allow a copy operation on behalf of the given user.

RPCSEC_GSSv3 Security (2)

- Client establishes copy_from_auth_priv, source server creates <"copy_from_auth", user id, destination> record. Client sends COPY_NOTIFY using the copy_from_auth RPCSEC_GSSv3 handle. Source server annotates record with source filehandle.
- Client establishes copy_to_auth_priv, destination server creates <"copy_to_auth", user id, source> record. Client sends a COPY using the copy_to_auth RPCSEC_GSSv3 handle.
- The destination establishes a copy_confirm_auth_priv on the source. Subsequent ONC RPC requests from the destination of the source use the copy_confirm_auth_priv handle.

Host-based Security

- Without real security, only a minimal level of protection is possible.
- Unique URLs used to encode the destination's copy privilege and identify a specific copy.
- Source server returns URLs in COPY_NOTIFY reply:

  nfs://10.11.78.18//_COPY/10.11.78.56/_FH/0x12345

  nfs://192.168.33.18//_COPY/10.11.78.56/_FH/0x12345
- Destination server will identify itself by performing these operations:

  COMPOUND { PUTROOTFH, LOOKUP "_COPY" ; LOOKUP "10.11.78.56"; LOOKUP "_FH" ; OPEN "0x12345" ; GETFH }

Additional Information

Miscellaneous

Copy Offload Stateids

- Copy Offload Stateids: a new type of stateid to identify asynchronous copies.
- Valid until either:
  - the client or server restart.
  - the client replies to a CB_COPY operation.
- A copy offload stateid's seqid MUST NOT be 0 (which would indicate the most recent offloaded copy). No real use case for this.

NFS Client Support

- When does an NFS client use the server-side copy offload operations?
  - Some NFS clients may require modifications to use these operations. Changes may be needed to the OS's user/kernel interface.
    - In Linux, reflink(2) (work in progress) looks promising. reflink(2) being proposed by OCFS2 developers for use by Oracle VM, see http://blogs.oracle.com/wim/2009/05/ocfs2_reflink.html
  - Some NFS clients may be ready to take advantage of these operations right away (e.g. a hypervisor).
  - These considerations are not (and should not be) part of the IETF proposal.

Agenda
9:00 – 11:30 am

- 9:00 Intro/Blue Sheets/Note Well/Agenda Bash (Pawlowski)
- 9:05 Server Side Copy Offload - (Lentini)
  - draft-lentini-nfsv4-server-side-copy-02.txt
- 9:25 Federated FS - (Lentini)
  - "Using DNS SRV to Specify a Global File Name Space with NFS version 4"
    - <draft-ietf-nfsv4-federated-fs-dns-srv-namespace-01.txt>
  - "Administration Protocol for Federated Filesystems"
    - <draft-ietf-nfsv4-federated-fs-admin-01.txt>
  - "Requirements for Federated File Systems"
    - <draft-ietf-nfsv4-federated-fs-reqts-03.txt>
  - "NSDB Protocol for Federated Filesystems“
    - <draft-ietf-nfsv4-federated-fs-protocol-01.txt>
- 9:40 NFS operation over IPv4 and IPv6 (Alex RN)
  - <draft-alexrn-nfsv4-ipv6-00.txt>
- 10:10 NFSv4 Multi-Domain Access (Adamson)
  - <draft-adamson-nfsv4-multi-domain-access.txt>
- 10:25 Proposal for an NFSv4 extension to allow the use of NFS clients as pNFS data servers (Adamson)
  - <draft-myklebust-nfsv4-pnfs-backend-00.txt>
- 10:55 Access checks and pNFS (Sorin Faibish)
- 11:25 Wrapup (Pawlowski)
- 11:30 End

FedFS

James Lentini
jlentini@netapp.com

IETF 75 NFSv4 WG Meeting
July 29, 2009

Summary

The FedFS protocol drafts are on track for WG Last Call in October, 2009.

Drafts

Four drafts published as working group documents:

- Requirements
- Namespace Root Discovery
- NSDB Protocol
- Admin Protocol

Future extensions possible (Root Fileset, FSL type for SMB, etc.).

Requirements
draft-ietf-nfsv4-federated-fs-reqts-03

Summary: Requirements for a federated filesystem.
Proposed Category: Informational

Status:
- Passed WG Last Call in May, 2009.

Next Steps:
- Spencer Shepler shepherding to IESG.

Namespace Root Discovery

draft-ietf-nfsv4-federated-fs-dns-srv-namespace-01

Summary: Defines a DNS record format for publishing the namespace's root location to clients.

Proposed Category: Standards Track

Status:

- No outstanding issues.

Next Steps:

- WG Last Call scheduled for October, 2009.

NSDB Protocol

draft-ietf-nfsv4-federated-fs-protocol-02

Summary: Defines NSDB LDAP types and operations.
Proposed Category: Standards Track

Status:

- LDAP chosen as the NSDB protocol.
  - LDAP schema allows for future extension (e.g. SMB).
  - LDAP Expert Review process initiated.
- Procedure for FSL Caching defined.

Next Steps:

- Complete LDAP Expert Review process.
- Review Security Considerations section.
- WG Last Call scheduled for October, 2009.

Admin Protocol
draft-ietf-nfsv4-federated-fs-admin-02

Summary: Describes ONC RPC protocol to create/delete/query a junction on a fileserver.
Proposed Category: Standards Track

Status:

- No outstanding issues.

Next Steps:

- WG Last Call scheduled for October, 2009.

Acknowledgements

Many people have contributed!

- George Amvrosiadis (University of Ioannina)
- Andy Adamson (NetApp)
- Dan Ellard (BBN Technologies)
- Craig Everhart (NetApp)
- Paul Lemahieu (EMC)
- James Lentini (NetApp)
- Pavan Mettu (Sun)
- Manoj Naik (IBM)
- Chris Stacey (EMC)
- Renu Tewari (IBM)
- Robert Thurlow (Sun)
- Mario Würzl (EMC)

…and several others have attended one or two meetings.

Related Work

Referrals in NFSv4

draft-ietf-nfsv4-referrals-00.txt

- Expired draft that defines how to use NFSv4 (RFC3530) fs_locations for referrals.
- Ideas incorporated into NFSv4.1 draft.
- Suggest resurrecting draft and
  - including in RFC3530bis (preferred) or
  - publishing as a standalone standards track RFC
- Resolution is not required to move forward with FedFS. FedFS is independent of the details of the referral mechanism.

NFSv4 Multi-Domain Access

draft-adamson-nfsv4-multi-domain-access-00

- See today's presentation from Andy Adamson.
- Complements FedFS by addressing issues of identity mapping in multi-domain environments. As with referrals, FedFS is independent of the details of the identity mapping.

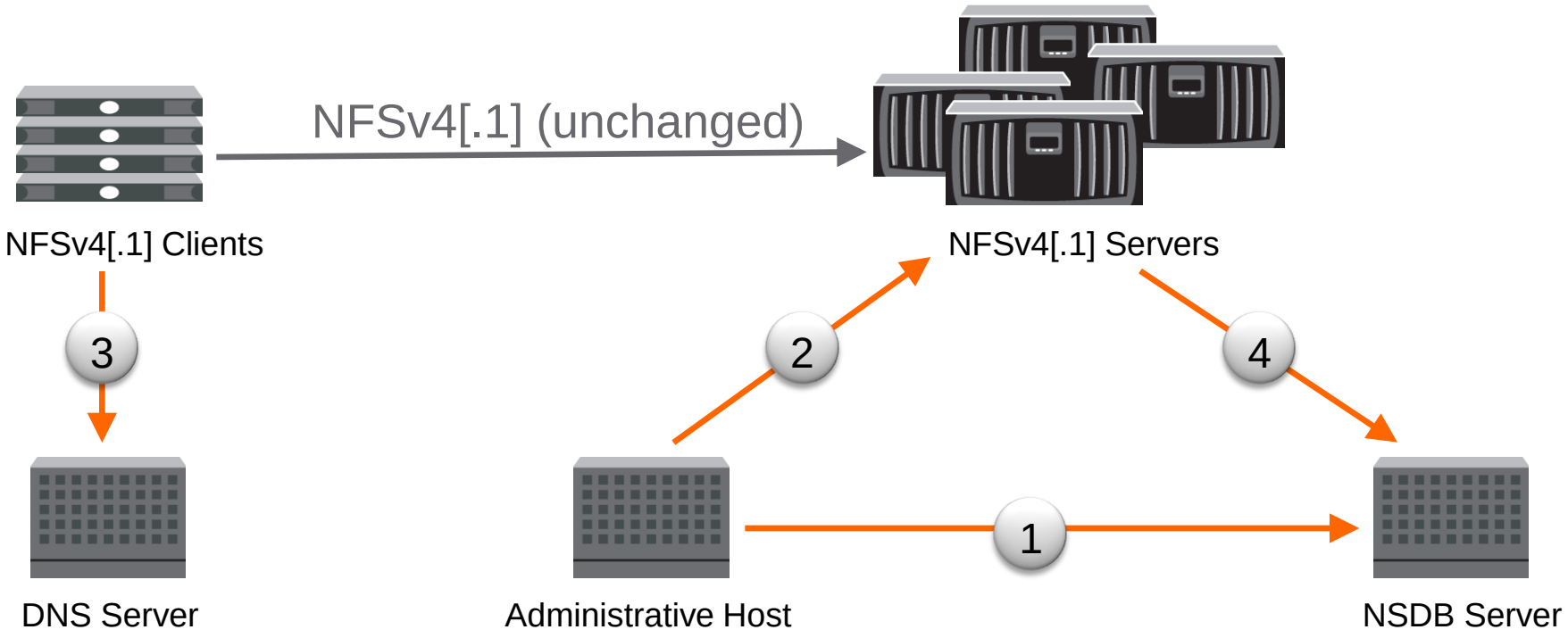
Background Information

What is FedFS?

- FedFS is a set of open protocols that permit the construction of a scalable, cross-platform federated file system namespace accessible to unmodified NFSv4[.1] clients.
- Key points:
  - Unmodified clients
  - Open: cross-platform, multi-vendor
  - Federated: participants retain control of their systems
  - Scalable: supports large namespaces with many clients and servers in different geographies

FedFS Protocols

Namespace Management	Namespace Navigation
1 NSDB Management (LDAP)	3 Client root discovery (DNS)
2 Junction Management (ONC RPC)	4 Junction resolution (LDAP)

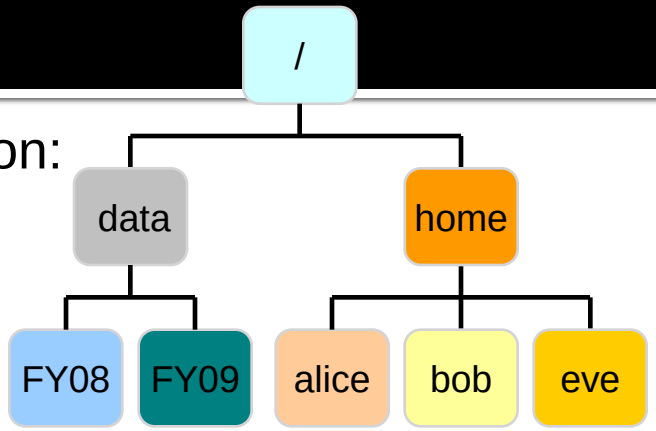


What are the benefits?

- Simplified management: eliminates complicated software such as the automounter.
- Separates logical and physical data location: allows data movement for cost/performance tiering, worker mobility, and application mobility.
- Enhances:
  - Data Replication: for load balancing or high availability
  - Data Migration: for moving data closer to compute or decommissioning systems
  - Cloud Storage: for the dynamic data center, enterprise clouds, or private internet clouds.

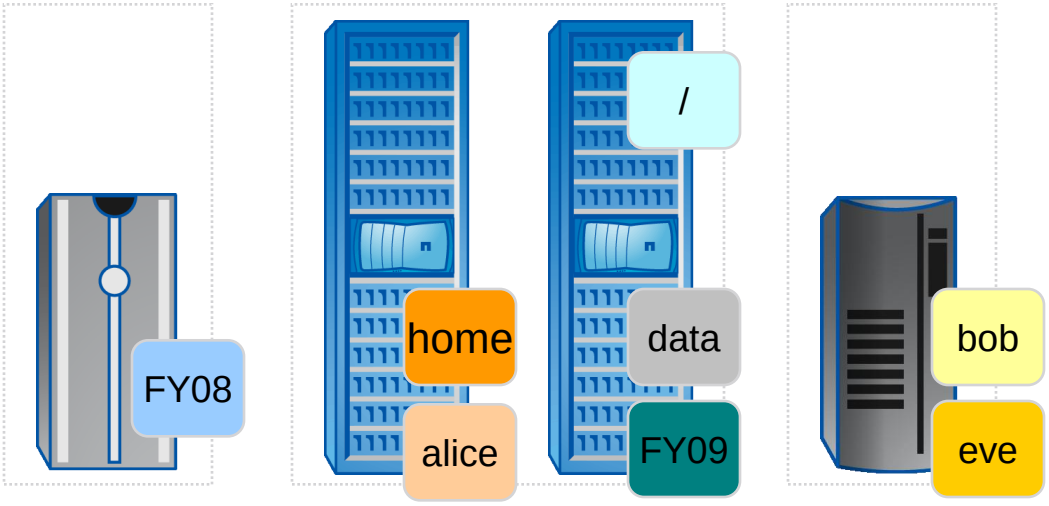
Federated Namespace Example

The illusion:



- The user and application software see a simple, hierarchical namespace.

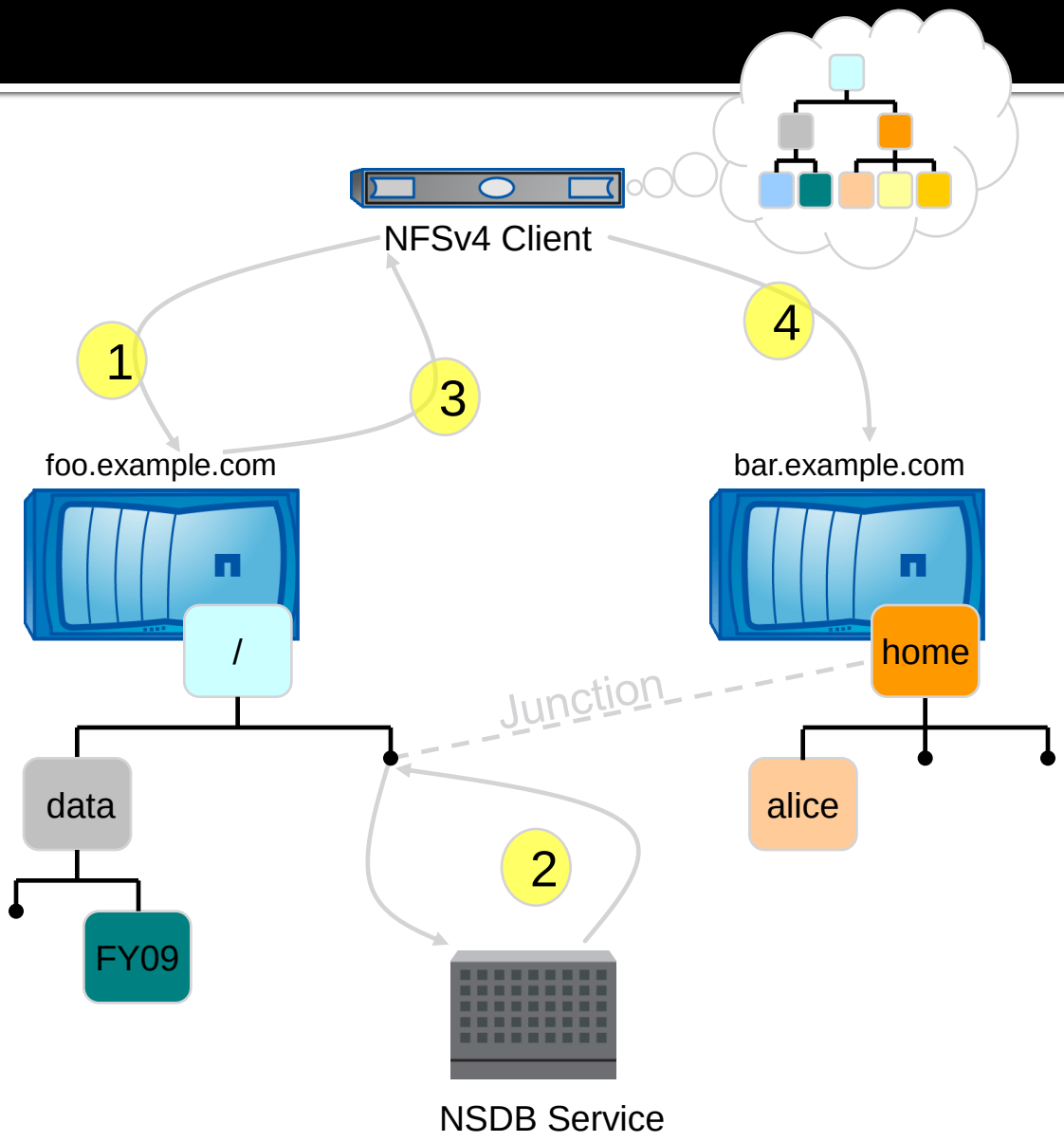
The reality:



L.A. New York Boston

- Behind the scenes, simple management operations allow data mobility for high performance, high reliability, and high availability.

FedFS in Action



The user requests */home/alice*:

1. The client attempts to access */home/alice* on server foo.
2. Server foo discovers that *home* is a namespace junction and determines its location using the FedFS NSDB service.
3. Server foo returns an NFSv4 referral to the client directing it to server bar.
4. The client accesses *home/alice* on server bar.

Client Support for Referrals

NFSv4 clients supporting referrals are available on many platforms. For example:

- **AIX**: referrals and replication (including failover) supported since 5.3 (released August, 2004)
- **HPUX**: referrals supported in HP-UX 11iv3 with ONCplus B.11.31.03 (released May, 2008)
- **Linux**: referrals supported since 2.6.18 (released September, 2006)
  - Migration/replication support under development

Past Milestones

- Prototype of NSDB protocols demonstrated at the summer WG meeting in Dublin (Summer 2008)
- Four drafts published as NFSv4 WG documents (Fall 2008).
- Federated namespace added to the NFSv4 WG charter (Spring 2009)
- Requirements draft passed WG last call (May 2009)

Meetings

Open meetings are held each week to resolve issues and review proposals.

- Thursdays, 1:30 – 2:30 PM Eastern
  (10:30 - 11:30 AM Pacific)
- Conference Number: 1-888-765-3653
- Conference ID: 2354843

Agenda
9:00 – 11:30 am

- 9:00 Intro/Blue Sheets/Note Well/Agenda Bash (Pawlowski)
- 9:05 Server Side Copy Offload - (Lentini)
  - draft-lentini-nfsv4-server-side-copy-02.txt
- 9:25 Federated FS - (Lentini)
  - "Using DNS SRV to Specify a Global File Name Space with NFS version 4"
    - <draft-ietf-nfsv4-federated-fs-dns-srv-namespace-01.txt>
  - "Administration Protocol for Federated Filesystems"
    - <draft-ietf-nfsv4-federated-fs-admin-01.txt>
  - "Requirements for Federated File Systems"
    - <draft-ietf-nfsv4-federated-fs-reqts-03.txt>
  - "NSDB Protocol for Federated Filesystems“
    - <draft-ietf-nfsv4-federated-fs-protocol-01.txt>
- 9:40 NFS operation over IPv4 and IPv6 (Alex RN)
  - <draft-alexrn-nfsv4-ipv6-00.txt>
- 10:10 NFSv4 Multi-Domain Access (Adamson)
  - <draft-adamson-nfsv4-multi-domain-access.txt>
- 10:25 Proposal for an NFSv4 extension to allow the use of NFS clients as pNFS data servers (Adamson)
  - <draft-myklebust-nfsv4-pnfs-backend-00.txt>
- 10:55 Access checks and pNFS (Sorin Faibish)
- 11:25 Wrapup (Pawlowski)
- 11:30 End

NetApp™

Go further, faster™

NFS operation over IPv4 and IPv6

R N Alex

rnalex@netapp.com

Rev: July 29, 2009

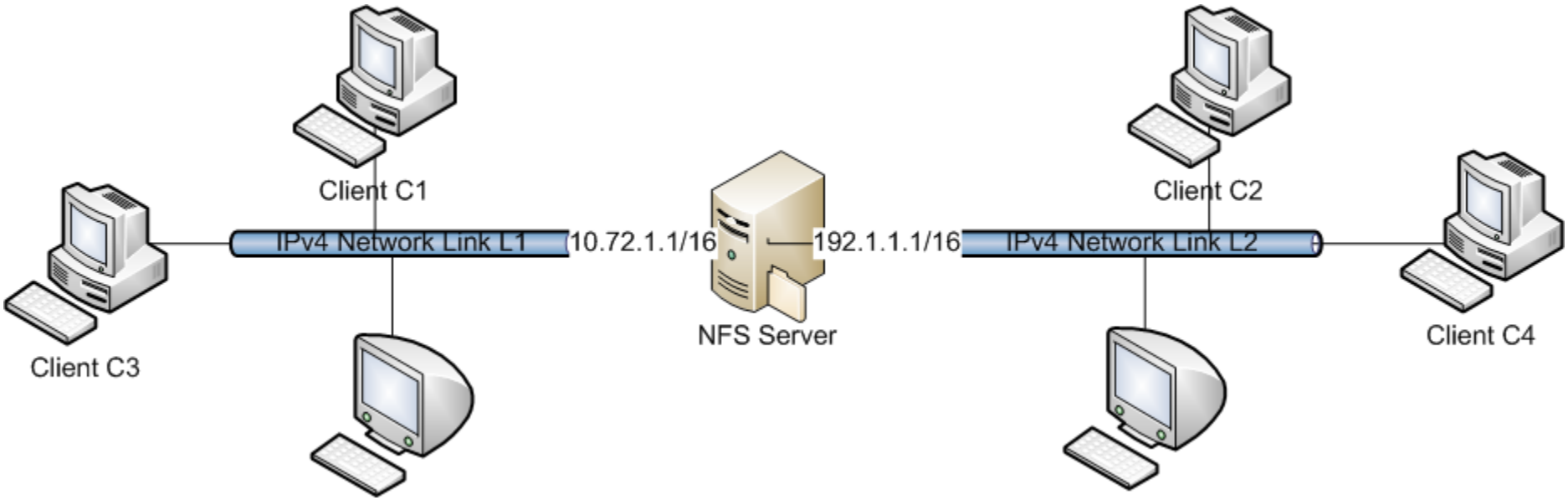
Agenda

- Purpose
  - 'The WG will also **update** the ONC RPC specification for compatibility with IPv6.'
- Problem Classes
  - Private addressing issues
    - Multi-homing
    - RPCBind
  - No single client id across Address Families
    - NSM/NLM
    - Nfsv4 Client Identification
    - Reply cache problem for Nfsv4
  - Abrupt Address Family disruption
    - Dual stack to single stack transition
- Summary and conclusion

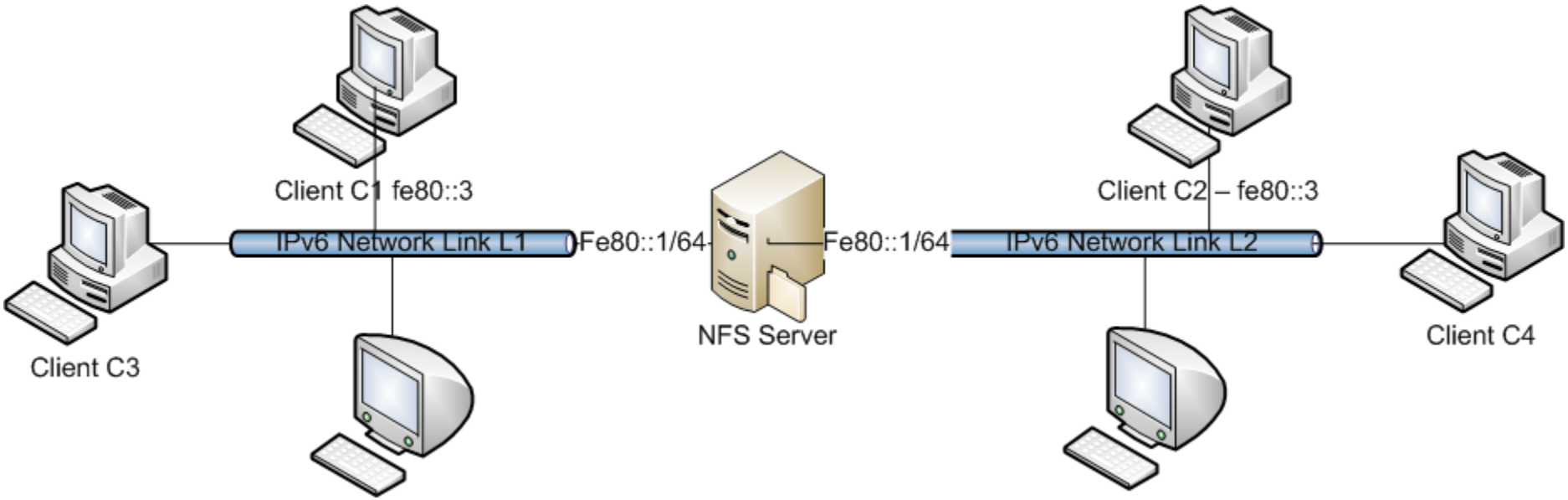
Multi-homing - Problem

- Administrative address boundary differences between IPv4 and IPv6.
- Private Addressing Boundaries
  - Should Link local addresses be supported
    - Needed in NFS boot scenarios
- Outbound Communication
  - Proper scope needs to be specified
    - Server – callbacks , NSM notify - client Rpcbind
- Information boundaries
  - Do not propagate Address Family information like embedded addresses across scopes and AF

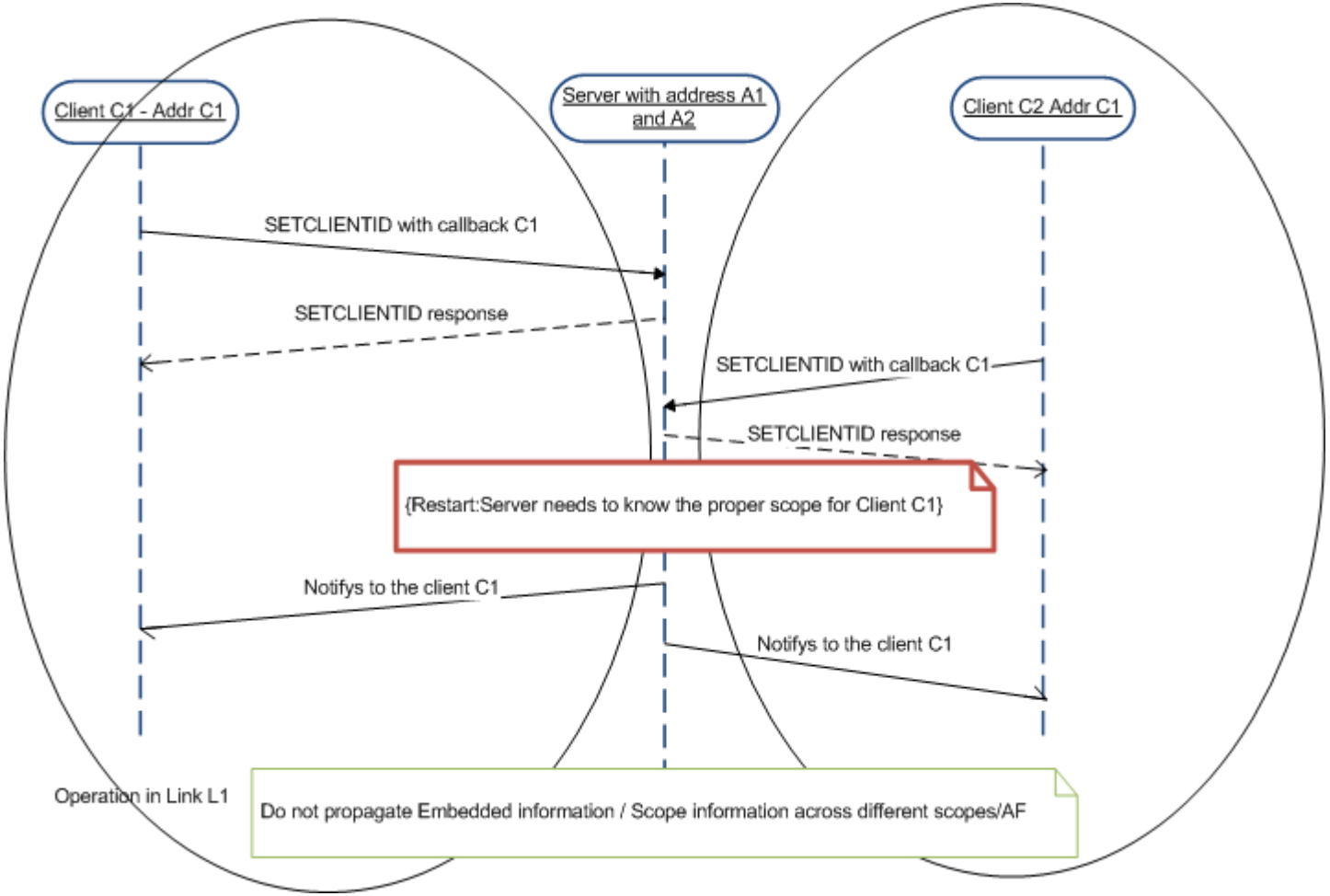
Multi-homing – Problem
Administrative difference: Different subnet different subnet ids



Multi-homing – Problem
Administrative difference: Potentially same subnet ids for private addresses



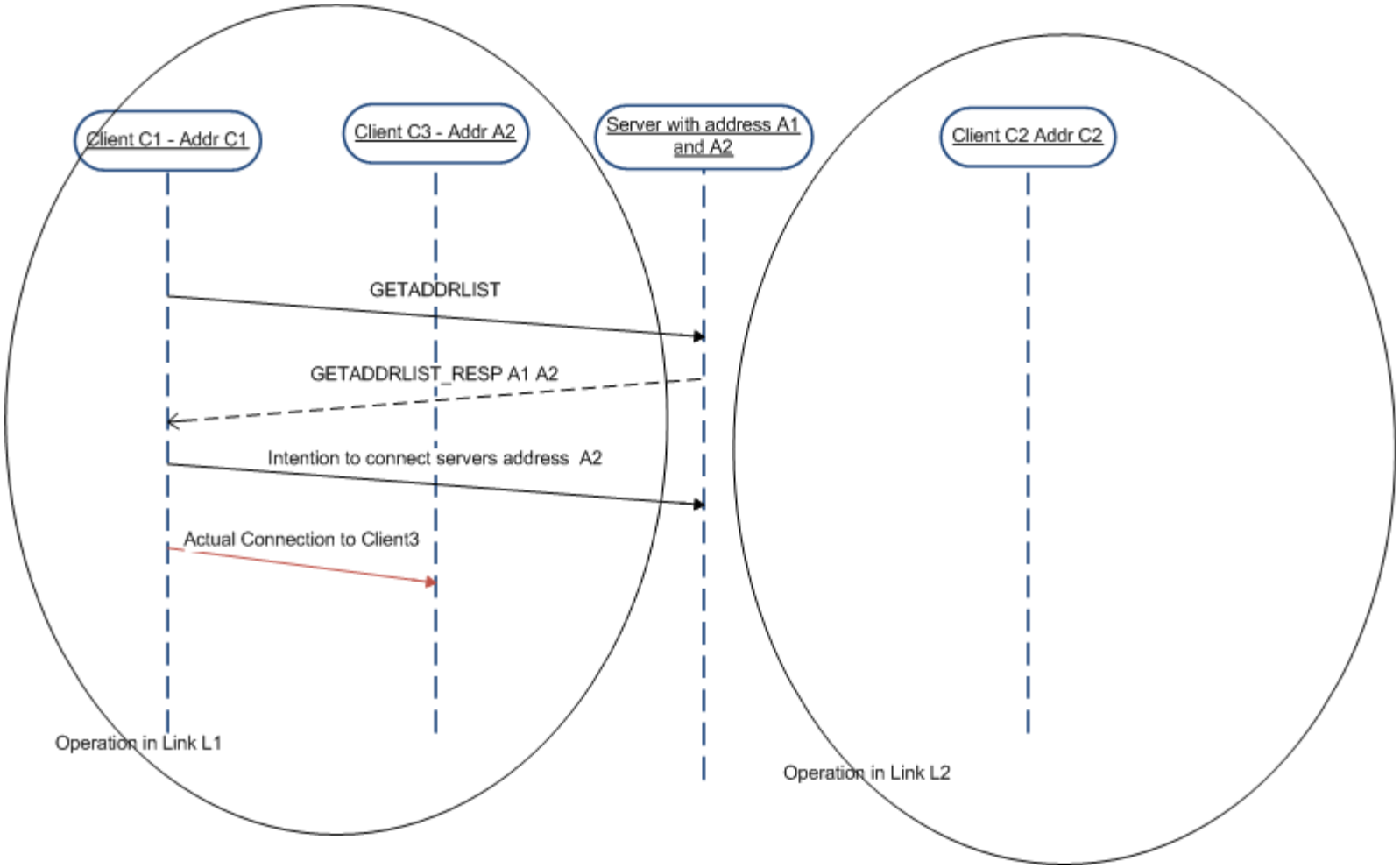
Multi-homing: Problem Server scope ambiguity



RPCBind : Problem/Solutions

- RPCBIND preferred over PORTMAP.
  - The path to the service is explicitly specified through netid and universal address
  - Useful for debugging too
- Discovering a Service
  - Propagation of information across domains
  - Use appropriate order of calls to get information

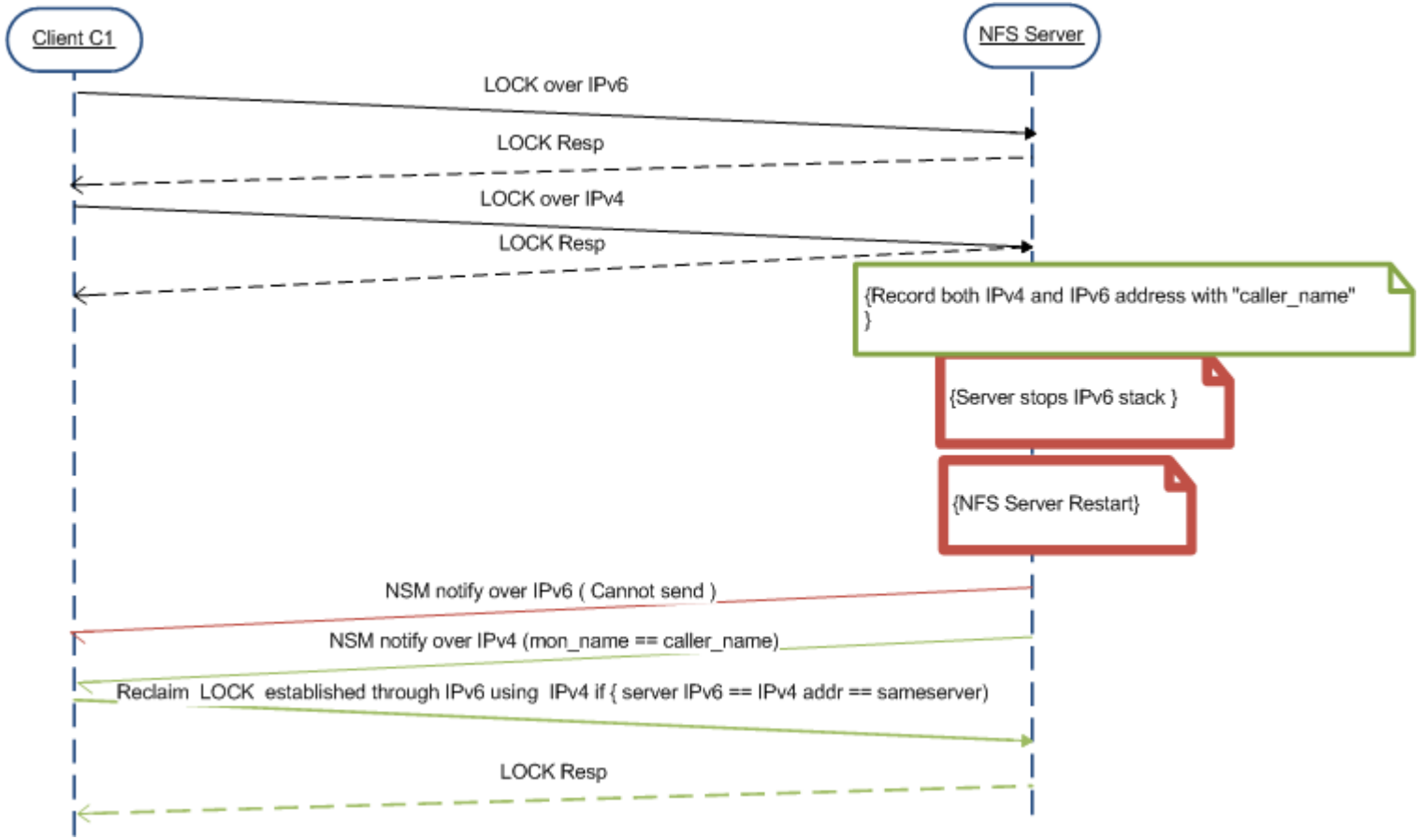
RPCBind : Problem
Advertising non local information



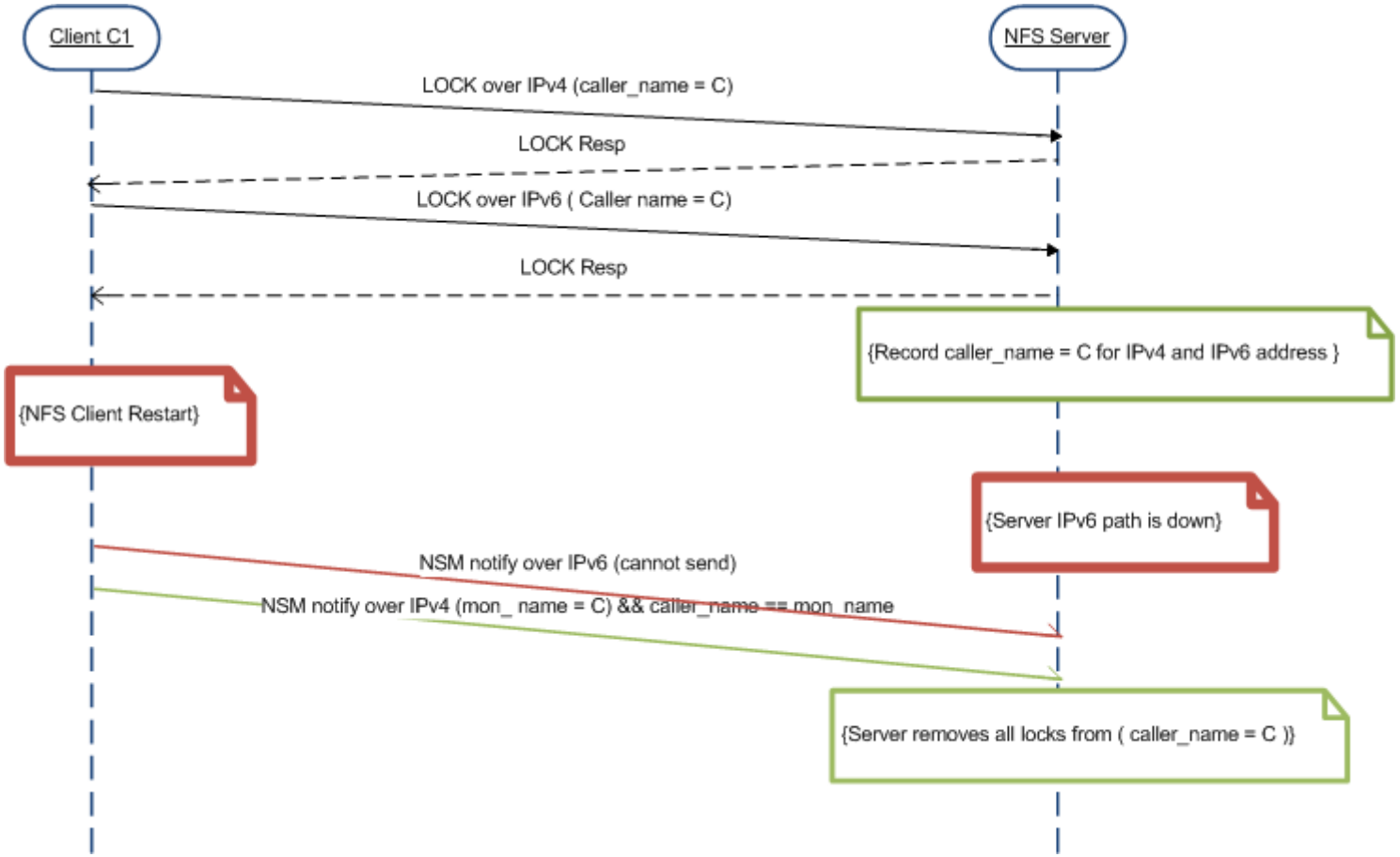
NLM/NSM: Problem/Solution

- Problem
  - Locks could be stuck if one of Address Family path goes down.
  - Disruption when partial service reboot
  - Ambiguous client identifier which owns state
- Solution
  - Use the caller_name and mon_name field as a client identifier
  - Record both IPv4 and IPv6 addresses corresponding to caller_name / mon_name field

NLM/NSM : Scenario problem at server



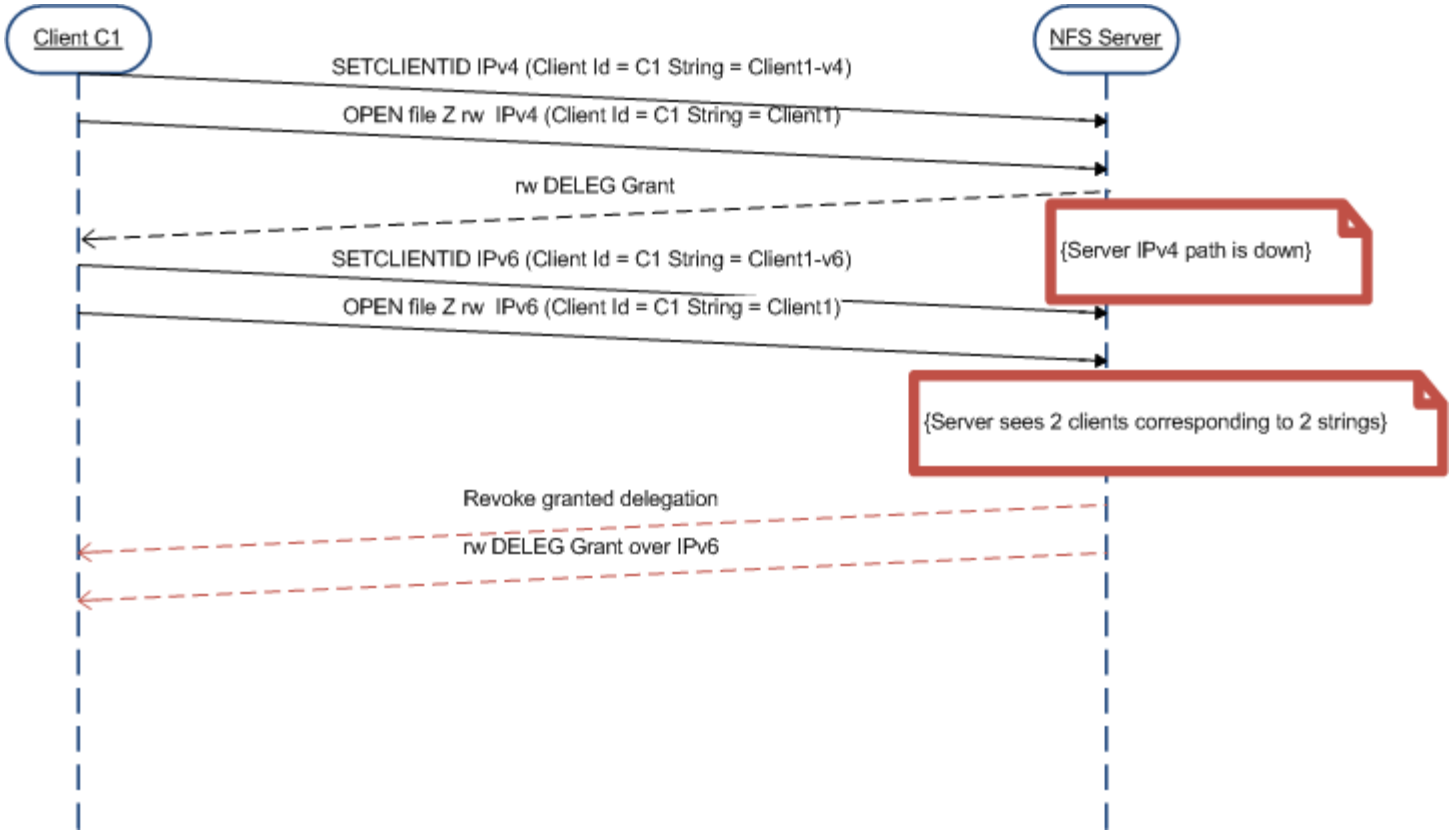
NLM/NSM : Scenario problem at client



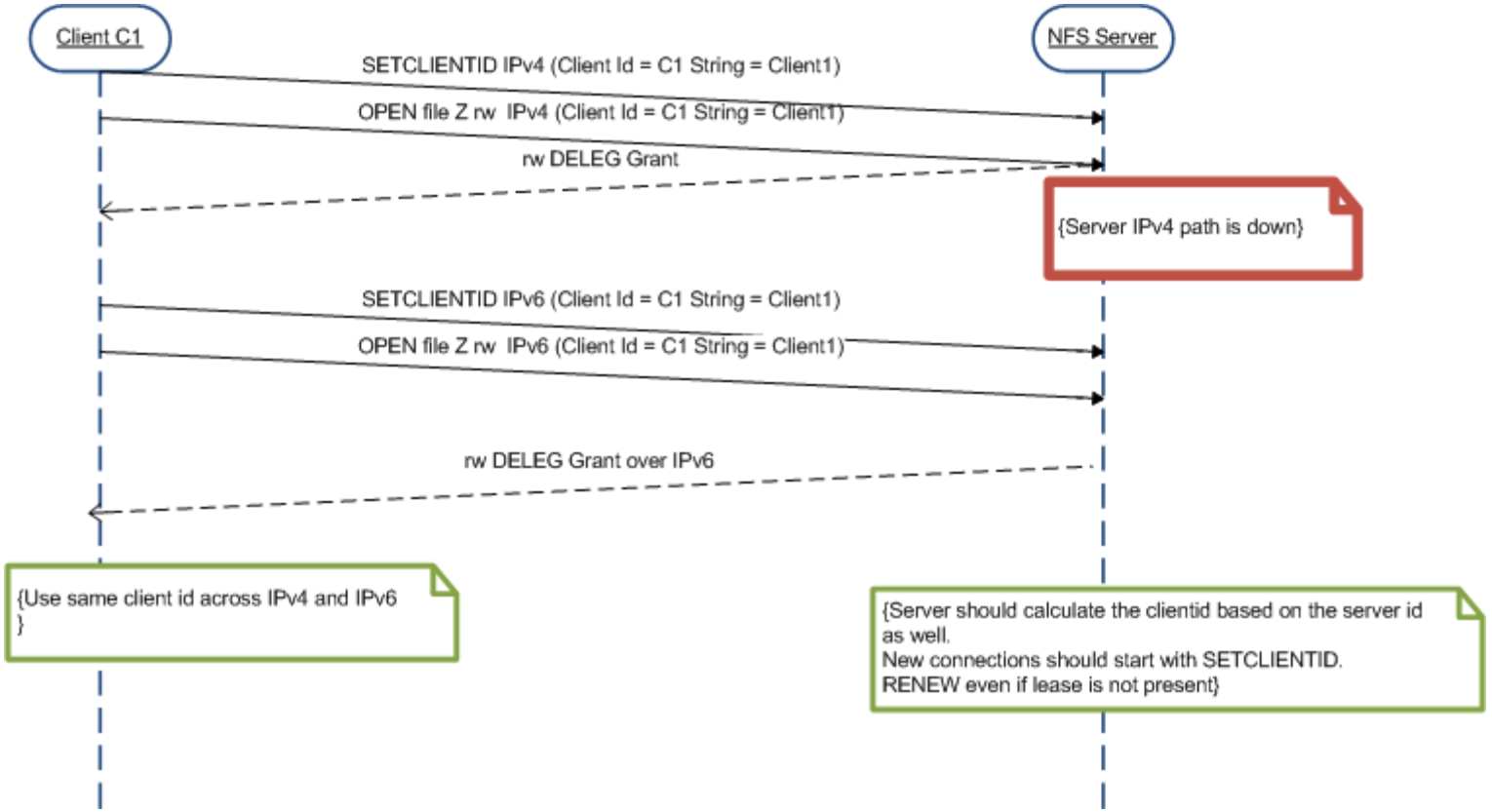
Nfsv4.0 Client Identification

- Problem
  - Same client trying to get a delegation through 2 address families
  - Delegations may be revoked
- Solution
  - Use the same client string across AF
  - Calculate clientid based on server id as well
  - Use setclientid as first operation on a connection

Nfsv4.0 Client Identification - Problem



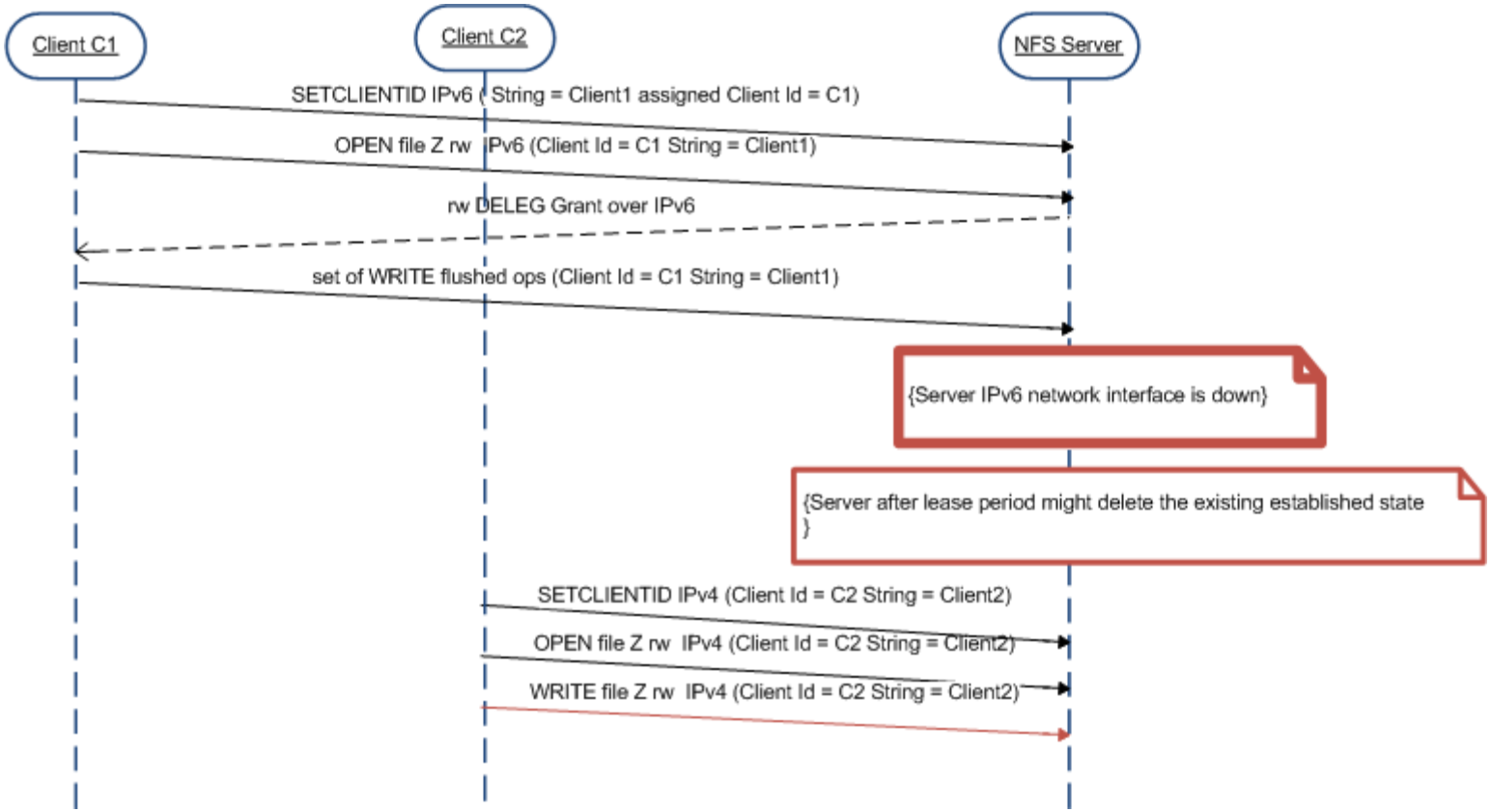
Nfsv4.0 Client Identification - Solution



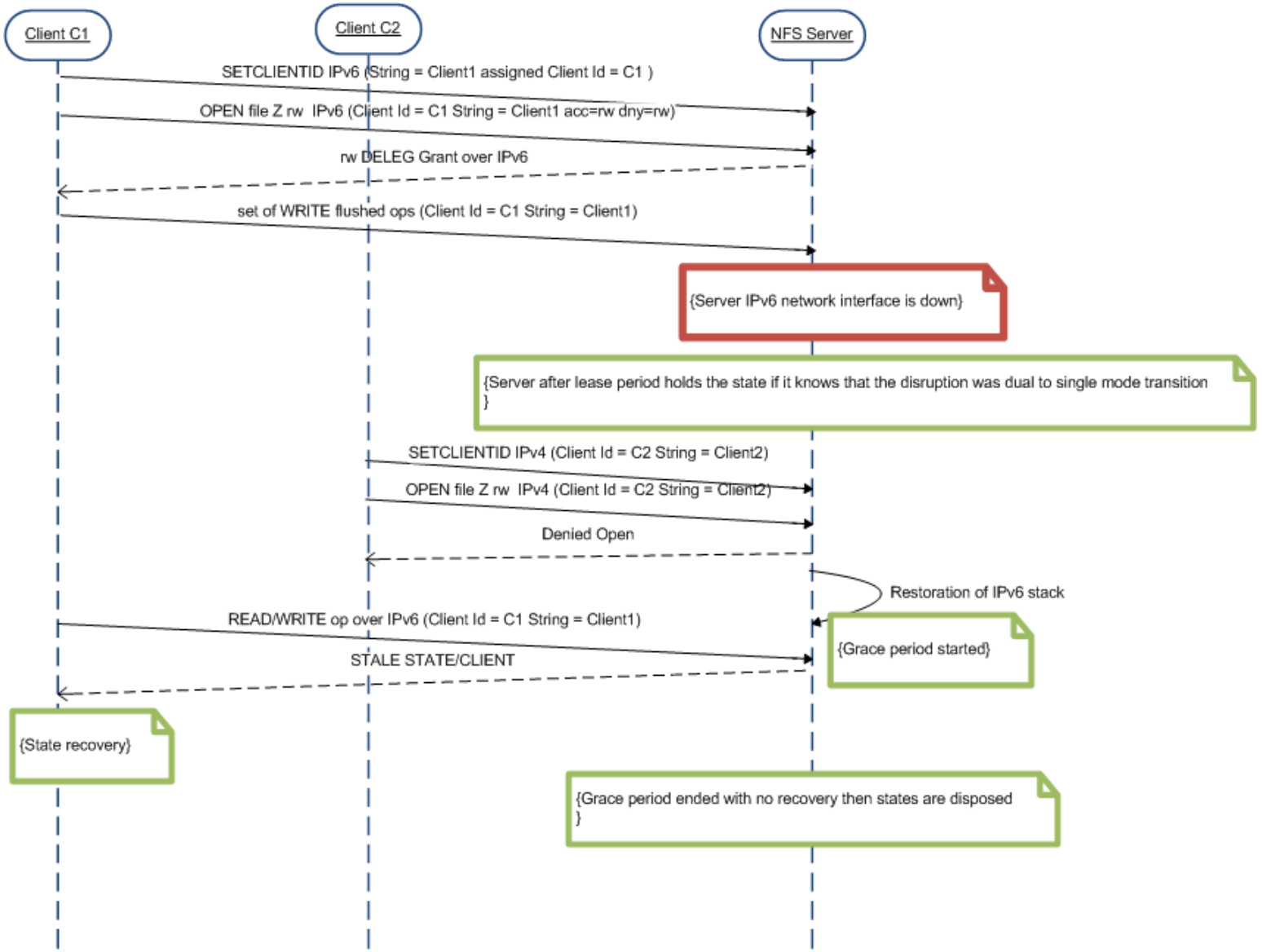
Reply-cache – Problem/Solution

- Problem: EOS for Nfsv2/3/4.0
  - Currently implementations use xid and src+dst info as keys, re-xmit may be from different src,dst
  - Need a unique key to identify op across connection
- Solution Suggestion for Nfsv4.0
  - Identify the client as the first op with setclientid
  - If retransmit then use a setclientid as first call
  - For retransmits use the same callback with same client string and verifier.
  - Focus on using clientid+xid as unique key
  - Nfsv4.1 EOS limited to a session.

Dual to Single Transition - Problem



Dual to Single Transition - Solution



Summary and conclusion

- The charter statement should be expanded to include implementation advice for NFSv2, v3, v4.0, and v4.1 over IPv6
- The update to the ONC RPC specs should include both
  - Standards updates (i.e. MUST use support RPCBIND on ONC RPC client and server)
  - Generic implementation advice for dealing with AF switching issues

Where Can I find more information

- Internet-ID draft reference
  - http://tools.ietf.org/search/draft-alexrn-nfsv4-ipv6-00
- Contact Information
  - RNALEX@netapp.com

Agenda
9:00 – 11:30 am

- 9:00 Intro/Blue Sheets/Note Well/Agenda Bash (Pawlowski)
- 9:05 Server Side Copy Offload - (Lentini)
  - draft-lentini-nfsv4-server-side-copy-02.txt
- 9:25 Federated FS - (Lentini)
  - "Using DNS SRV to Specify a Global File Name Space with NFS version 4"
    - <draft-ietf-nfsv4-federated-fs-dns-srv-namespace-01.txt>
  - "Administration Protocol for Federated Filesystems"
    - <draft-ietf-nfsv4-federated-fs-admin-01.txt>
  - "Requirements for Federated File Systems"
    - <draft-ietf-nfsv4-federated-fs-reqts-03.txt>
  - "NSDB Protocol for Federated Filesystems“
    - <draft-ietf-nfsv4-federated-fs-protocol-01.txt>
- 9:40 NFS operation over IPv4 and IPv6 (Alex RN)
  - <draft-alexrn-nfsv4-ipv6-00.txt>
- 10:10 NFSv4 Multi-Domain Access (Adamson)
  - <draft-adamson-nfsv4-multi-domain-access.txt>
- 10:25 Proposal for an NFSv4 extension to allow the use of NFS clients as pNFS data servers (Adamson)
  - <draft-myklebust-nfsv4-pnfs-backend-00.txt>
- 10:55 Access checks and pNFS (Sorin Faibish)
- 11:25 Wrapup (Pawlowski)
- 11:30 End

NFSv4 Multi Domain Access

Andy Adamson
andros@netapp.com
IETF 75 NFSv4 Working Group Meeting
July 29, 2009

Motivation

- The NFSv4 protocol can join servers which use separate name translation and separate security services into a common file system
  - Federated File System is an example
- NFSv4 ACL names and security identities need to be translated across administrative boundaries before users can traverse such a name space
- Current name translation service schema are unable to support the required translations

The First Draft

- Two new name service attributes are introduced to enable users to traverse and access files in a secure multi-domain NFSv4 name space
  - Addresses LDAP uidNumber translations
    - ACL's for foreign users are enabled
  - gidNumber translations are not discussed
    - Future work
- Some administrative choices WRT security and name construction are proposed
- NFSv4 protocol means NFSv4.0 and NFSv4.1, they both share this issue

NFSv4 Multiple Names

- Authentication occurs at the RPC level
  - RPC credential presents the security flavor user identity to the NFSv4 server for translation into a local representation
  - Multiple authentication methods supported
- NFSv4 ACL attribute name used for setting and getting file object access
  - user@dns_domain syntax provides a level of indirection so that the client and server can translate the local representation into a common syntax
  - Few restrictions on local user representation translation to user@dns_domain ACL name

Name Translation Service

- Network Information Service (NIS), Lightweight Directory service (LDAP) and Active Directory (AD) are the three widely used client-server directory service protocols that provide name translation.
- LDAP or AD are used instead of NIS in environments where scale and security are issues.
- AD uses LDAP for name translation, so LDAP is used as a name service in all examples and solutions.
- For this presentation, a name service exports a unique uidNumber space.

Name Translation

- RFC2307 defines the LDAP posixAccount object class
  - resolves account information such as user IDs to login names.
- Requires a one-to-one correspondence between the user login name (uid attribute) and the user integer identification number (uidNumber attribute)
- Can be used in some NFSv4 environments with restrictions
  - Login name == user portion of user@dns_domain
    - Strip / add @dns_domain portion during translation
  - AUTH_SYS places uidNumbers on the wire
  - Kerberos Realm(s) with login name == Kerberos principal
    - Strip @REALM portion of Kerberos principal during translation
- Not sufficient for multi-domain use

posixAccount

- The posixAccount translates between the uid ‘bob’ and the uidNumber ‘2975’
- NFSv4 ACL name translation – strip the @dns_domain
  - bob@business.com -> bob –> uidNumber 2975
- Kerberos Realm translation – strip the @REALM
  - bob@BUSINESS.REALM -> bob -> uidNumber 2975
  - bob@BUSINESS_ENG.REALM -> bob -> uidNumber 2975

Namespace Scope

- Many ways to administer an NFSv4 environment
- Stand alone sites choose which options serve their needs
  - Single enterprise with a firewalled network can use low security protocols and employ a common identity for users across all the file servers in the name space.
- Joining stand alone sites from different administrative domains into a multi-domain namespace requires agreement on a subset of the available administrative options

NFSv4 Domain

- The NFSv4 Domian is the administrative unit for a multi-domain NFSv4 namespace
  - Roughly equivalent to an AFS Cell
  - Gathers the agreed upon NFSv4 protocol adminstrative choices
- It is the collection of administrative services used to build a multi-domain NFSv4 (federated) file system including:
  - A name service exporting a unique uidNumber/gidNumber space
    - The name service MUST service only one NFSv4 Domain.
  - One or more security services and one or more DNS domains

Security Flavors and Multi-Domain Access

- AUTH_NONE is useful in a multi-domain NFSv4 name space to grant universal access to public data
- AUTH_SYS can not be used for authenticated traversal of a multi-domain NFSv4 name space
  - uidNumber is passed in RPC credential with no name service identifier so no way to avoid uidNumber collisions
- AUTH_SYS uses a host-based authentication model where the server authenticates the client, and trusts the client to authenticate all users
  - Could be configured at the server to default to AUTH_NONE if client authentication fails
  - Policy for AUTH_SYS use is needed

Security Flavors and Multi-Domain Access

- RPCSEC_GSS & Kerberos security mechanism can be used in a multi-domian NFSv4 namespace
  - Kerberos principal in can be translated to uidNumber
  - Multiple Kerberos Realms per local NFSv4 environment
  - Same Kerberos Realm can serve multiple NFSv4 environments
- RPCSEC_GSS with an X.509 based security mechanism can also be used

DNS and Multi-Domain Access

- Multiple DNS domains are allowed per local NFSv4 environment
  - An NFSv4 client can mount servers using the same name service and in different DNS domains
- The dns_domain portion of the NFSv4 ACL name user@dns_domain assignment is not constrained by the NFSv4 protocol
  - It's possible (not practical !) to have servers return different user@dns_domian names for the same user

NFSv4 Domain Name

- The NFSv4 domain administrator MUST choose one of the DNS domains servicing the NFSv4 file servers and client machines to use as the NFSv4 domain name
- The NFSv4 domain name MUST be unique among all NFSv4 Domains.
- All the NFSv4 clients and servers MUST be configured to use the NFSv4 domain name as the "dns_domain" portion of the user@dns_domain NFSv4 ACL name
- Thus each user in the mulit-domain name space has a unique user@dns_domain name

Multiple NFSv4 Domain Translation

- Striping the @dns_domain portion of an NFSv4 ACL name or the @REALM portion of a Kerberos principal does not work in a multiple NFSv4 domain translation due to cross domain login name collisions
- NFSv4 ACL name translation – strip the @dns_domain
  - bob@business.com -> bob –> uidNumber 2975 (ok)
  - bob@university.edu ->bob -> uidNumber 2975 (oops!, wrong bob)

NFSv4Name Attribute

The NFSv4Name attribute provides a one-to-one correspondence between the unique NFSv4 Domain user@dns_domain NFSv4 ACL name and the uidNumber.

attributetype (1.3.6.1.4.1.250.10.5

NAME ('NFSv4Name')

DESC 'NFS version 4 Name'

EQUALITY caseIgnoreIA5Match

SYNTAX 1.3.6.1.4.1.1466.115.121.1.26

SINGLE-VALUE)

GSSAuthName Attribute

The GSSAuthName attribute provides a many-to-one correspondence between each GSS export name and the uidNumber.

attributetype (1.3.6.1.4.1.250.10.6

NAME ('GSSAuthName')

DESC 'RPCSEC GSS authenticated user name'

EQUALITY caseIgnoreIA5Match

SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)

GSSAuthName

- A Kerberos GSSAuthName would hold the principal@REALM
- An X.509 GSSAuthName would hold the Distinguished Name “/C= /ST= /O= /OU= /CN= /USERID=/Email=“.

NFSv4Person Object Class

The NFSv4Person class holds the minimal information required for NFSv4 access.

objectclass (1.3.6.1.4.1.250.10.7 NAME 'NFSv4Person'

DESC 'NFS version4 person from remote NFSv4 Domain'

SUP top AUXILIARY

MUST (uidNumber $ gidNumber $ NFSv4Name)

MAY (cn $ GSSAuthName $ description))

Multiple NFSv4 Domain Translation

- By adding the NFSv4Person class to the LDAP schema, the NFSv4Name and GSSAuthName attributes become available
- Foreign users are assigned a uidNumber
  - Use the posixAccount for users who need local machine accesss
  - Use the NFSv4Person for users who only need NFSv4 access.
- NFSv4 ACL name translation with NFSv4Name attribute
  - bob@business.com –> uidNumber 2975 [via posixAccount]
  - bob@university.edu -> uidNumber 3001 [via NFSv4Person]

Multiple NFSv4 Domain Translation

- GSS export name translation with GSSAuthName Attribute
  - bob@BUSINESS.REALM -> uidNumber 2975
  - bbar@BUSINESS_ENG.REALM -> uidNumber 2975

  - bob@UNIVERSITY.REALM -> uidNumber 3001
  - /C=USA /ST=MI /O=University of Michigan /OU=Engineering /CN=Bob Bar /USERID=bob /Email=bob@university.edu -> uidNumber 3001

Summary

- For local users the NFSv4 Name Attribute removes the need to strip / add the @dns_domain portion of the NFSv4 ACL name.
- For local users, the GSSAuthName attribute removes the requirement to synchronize some portion of the GSS export name with the posixAccount uid (login name)
  - Kerberos @REALM no longer needs to be stripped.

Summary

- For foreign users the NFSv4 Name Attribute enables the translation of an NFSv4 ACL name into a local uidNumber
- For foreign users, the GSSAuthName attribute enables the translation of a GSS export name into a local uidNumber.
- For all users, the removal of the NFSv4Name attribute and GSSAuthName attributes from an LDAP entry removes access to the NFSv4 Domain.
  - For users with a posixAccount, local machine access will not be affected.

Possible To Do's

- Translate foreign groups
  - NFSv4Name Attribute can hold group names
  - NFSv4Person already has the gidNumber
- Describe best practices concerning
  - Anonymous access across the federated file system
    - AUTH_NULL, AUTH_SYS->nobody, use of ANONYMOUS who
  - Authenticated access across the federated file system
    - Use of the AUTHENTICATED who

Possible To Do's

- Automation of uidNumber assignment when Kerberos cross-realm trust is established between two NFSv4 Domains.
- Protocol to inform a foreign NFSv4 Domain that a user is no longer valid, and their NFSv4Name and GSSAuthName attributes should be invalidated.

Agenda
9:00 – 11:30 am

- 9:00 Intro/Blue Sheets/Note Well/Agenda Bash (Pawlowski)
- 9:05 Server Side Copy Offload - (Lentini)
  - draft-lentini-nfsv4-server-side-copy-02.txt
- 9:25 Federated FS - (Lentini)
  - "Using DNS SRV to Specify a Global File Name Space with NFS version 4"
    - <draft-ietf-nfsv4-federated-fs-dns-srv-namespace-01.txt>
  - "Administration Protocol for Federated Filesystems"
    - <draft-ietf-nfsv4-federated-fs-admin-01.txt>
  - "Requirements for Federated File Systems"
    - <draft-ietf-nfsv4-federated-fs-reqts-03.txt>
  - "NSDB Protocol for Federated Filesystems“
    - <draft-ietf-nfsv4-federated-fs-protocol-01.txt>
- 9:40 NFS operation over IPv4 and IPv6 (Alex RN)
  - <draft-alexrn-nfsv4-ipv6-00.txt>
- 10:10 NFSv4 Multi-Domain Access (Adamson)
  - <draft-adamson-nfsv4-multi-domain-access.txt>
- 10:25 Proposal for an NFSv4 extension to allow the use of NFS clients as pNFS data servers (Adamson)
  - <draft-myklebust-nfsv4-pnfs-backend-00.txt>
- 10:55 Access checks and pNFS (Sorin Faibish)
- 11:25 Wrapup (Pawlowski)
- 11:30 End

NetApp™

Go further, faster™

Using NFS clients as data servers

Trond Myklebust (trond@netapp.com)
Presented by: Andy Adamson

Outline

- Motivation
- Workload characteristics
- Outline of the I-D proposal
- Brief summary of the protocol extensions
- Security considerations
- Prototyping status
- Other considerations

Motivation

- Targeting “library” workloads, where typically, several clients are accessing the same read-only data at more or less the same time
  - Examples include use of shared boot image files, program image files (/bin, /usr/bin), dynamically linked libraries (/lib, /usr/lib), static image libraries, etc.

Workload characteristics

- NFS traffic often spikes at cluster boot time, when all nodes are reading the same data, then trails off once the data is cached by the clients.
- pNFS striping does not entirely solve the problem. The problem of all clients accessing the same NFS server in the same patterns becomes the problem of all clients accessing the MDS and DSes in the same patterns.
- Data replication either on the server side or the client side (cachefs) is a potential cure, but expensive
  - Requires additional storage
  - Requires management of the data.
- Peer to peer systems?
  - They do scale as the number of peers.
  - However, security models are poorly understood.

Outline of the I-D proposal

- Proposal is based on work done by NetApp intern Yamini Allu and Trond Myklebust in the summer of 2008
- A trusted NFS client is allowed to share the contents of its cache by acting as a pNFS Data Server for one or more files.
  - Limited peer-to-peer model, but reuses the pNFS security model
  - Requires a protocol extension to allow the client both to offer to act as a DS, and to rescind that offer.
  - Requires the addition of a minimal control protocol to allow the DS to determine layout stateid validity and authorisation information.
- The server can then hand out pNFS file layouts that reference this DS as long as the latter holds a read delegation for that file.
  - If a client is not caching the data, then there is no benefit to using it as a DS.
  - The delegation requirement also allow the DS to cache authorisation information (acl and mode bits cannot change).
- Protocol extension documented in the following internet draft: http://www.ietf.org/id/draft-myklebust-nfsv4-pnfs-backend-protocol-00.txt
- Our goal is to add this protocol extension to NFSv4 minor version 2.

Brief summary of the protocol extensions

- REGISTER_DS
  - Offer to act as a DS for all filesystems, specific filesystems, or specific files.
  - The client also specifies a 64-bit cookie to be returned as the first 64-bits of all data server filehandles, so that the client can identify from which MDS they originated.
- UNREGISTER_DS
  - Revokes the offer to act as a DS.
- PROXY_OPEN
  - Checks whether a layout stateid that was presented by a pNFS client is valid.
  - Checks whether or not the pNFS client is authorised to access the file using a parameter that describes the authentication that was presented to the DS.
  - Translates the data server filehandle into a real/metadata server filehandle.
- CB_PROXY_REVOKE
  - Callback that revokes a layout stateid that was authorised via PROXY_OPEN

Security considerations

- A client acting as a Data Server could act as a vector for man-in-the-middle attacks
  - Implies that the current proposal is mainly useful inside the data center, or in situations where the administrator can designate specific clients as being fully trustworthy.
  - Would data integrity checksums be of help in allowing clients to decide whether or not to trust a DS?

Prototyping status

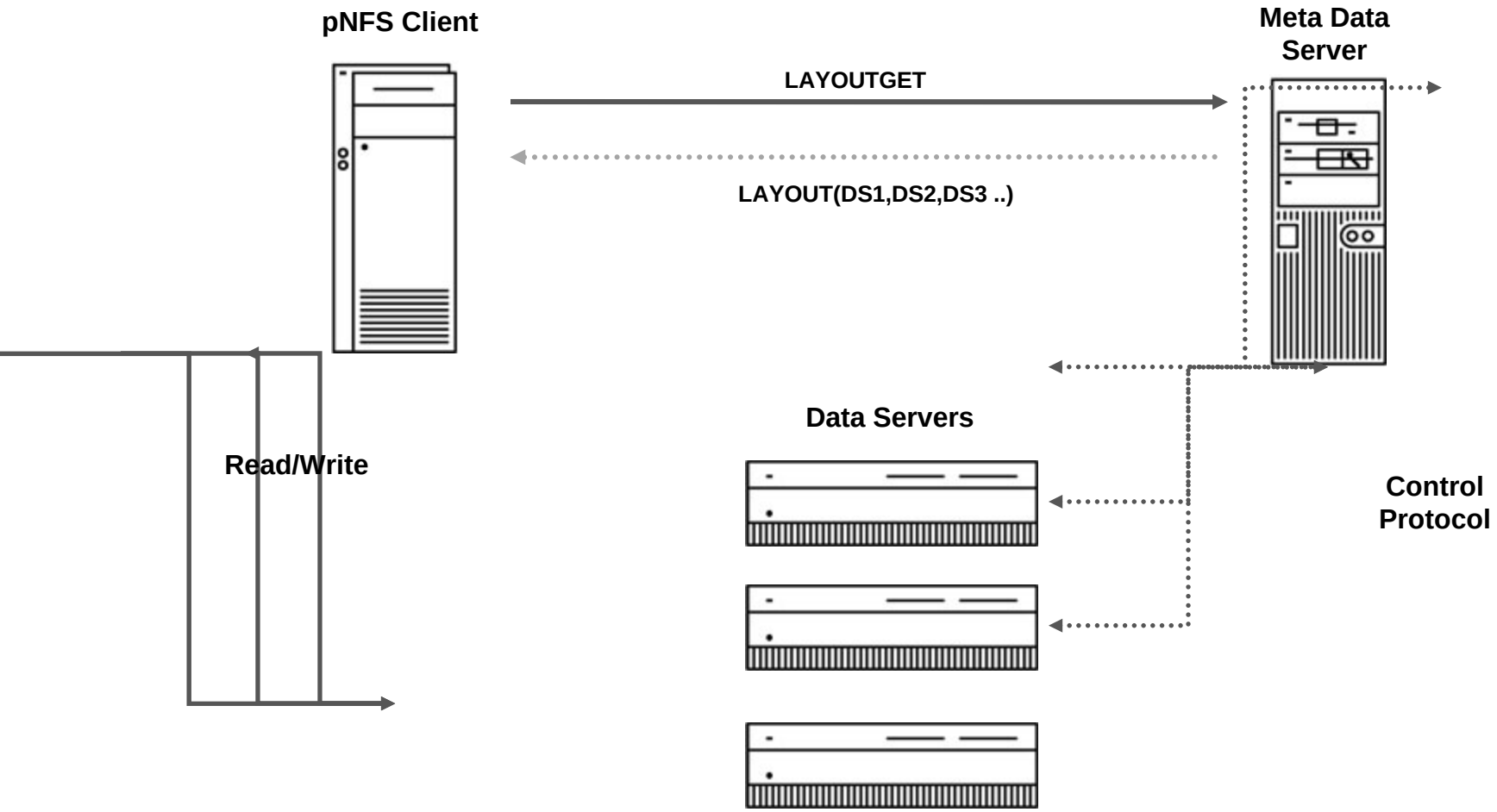
- The protocol was prototyped and tested by Yamini Allu during her internship in the summer of 2008.
- She used a modified Linux pNFS client and server system with read-only workloads (mainly iozone).
- The resulting system was shown to scale correctly as the number of clients increased beyond 2.
- Unfortunately, testing was limited by the quality of the prototype Linux pNFS code (which should be substantially improved now). The system has not yet been demonstrated for > 4-5 clients.

Other considerations

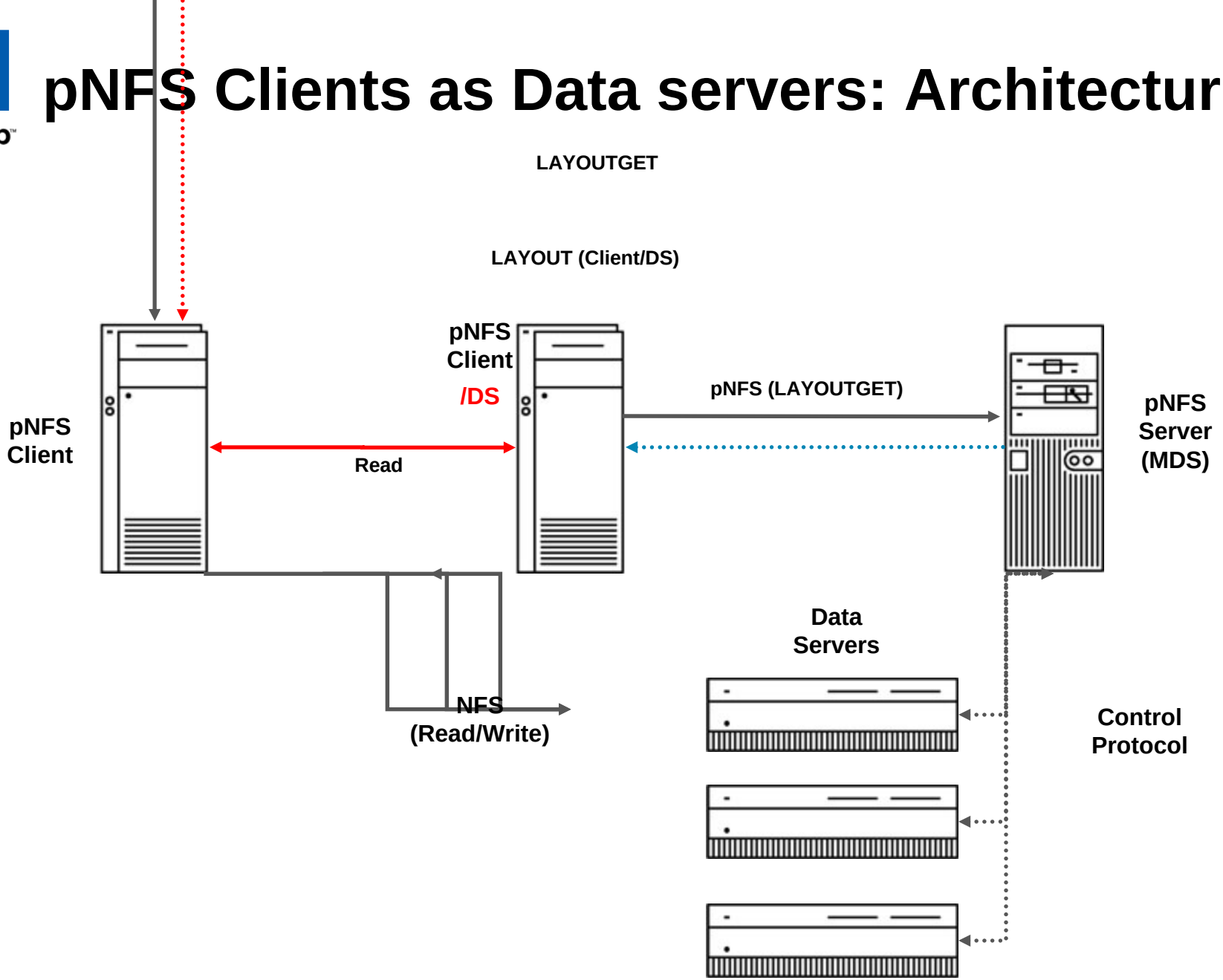
- It might also be possible to allow an NFS client that possesses a write delegation to act as a DS to read-only clients.
  - Would allow a client to keep the write delegation while ensuring that the pNFS clients can still read the data as it get written back to the client's cache.
  - Haven't pursued this due to lack of credible use cases.

Questions?

pNFS Architecture



pNFS Clients as Data servers: Architecture



Agenda
9:00 – 11:30 am

- 9:00 Intro/Blue Sheets/Note Well/Agenda Bash (Pawlowski)
- 9:05 Server Side Copy Offload - (Lentini)
  - draft-lentini-nfsv4-server-side-copy-02.txt
- 9:25 Federated FS - (Lentini)
  - "Using DNS SRV to Specify a Global File Name Space with NFS version 4"
    - <draft-ietf-nfsv4-federated-fs-dns-srv-namespace-01.txt>
  - "Administration Protocol for Federated Filesystems"
    - <draft-ietf-nfsv4-federated-fs-admin-01.txt>
  - "Requirements for Federated File Systems"
    - <draft-ietf-nfsv4-federated-fs-reqts-03.txt>
  - "NSDB Protocol for Federated Filesystems“
    - <draft-ietf-nfsv4-federated-fs-protocol-01.txt>
- 9:40 NFS operation over IPv4 and IPv6 (Alex RN)
  - <draft-alexrn-nfsv4-ipv6-00.txt>
- 10:10 NFSv4 Multi-Domain Access (Adamson)
  - <draft-adamson-nfsv4-multi-domain-access.txt>
- 10:25 Proposal for an NFSv4 extension to allow the use of NFS clients as pNFS data servers (Adamson)
  - <draft-myklebust-nfsv4-pnfs-backend-00.txt>
- 10:55 Access checks and pNFS (Sorin Faibish)
- 11:25 Wrapup (Pawlowski)
- 11:30 End

Access checks and pNFS

IETF 75 NFSv4 WG Meeting
July 29, 2009

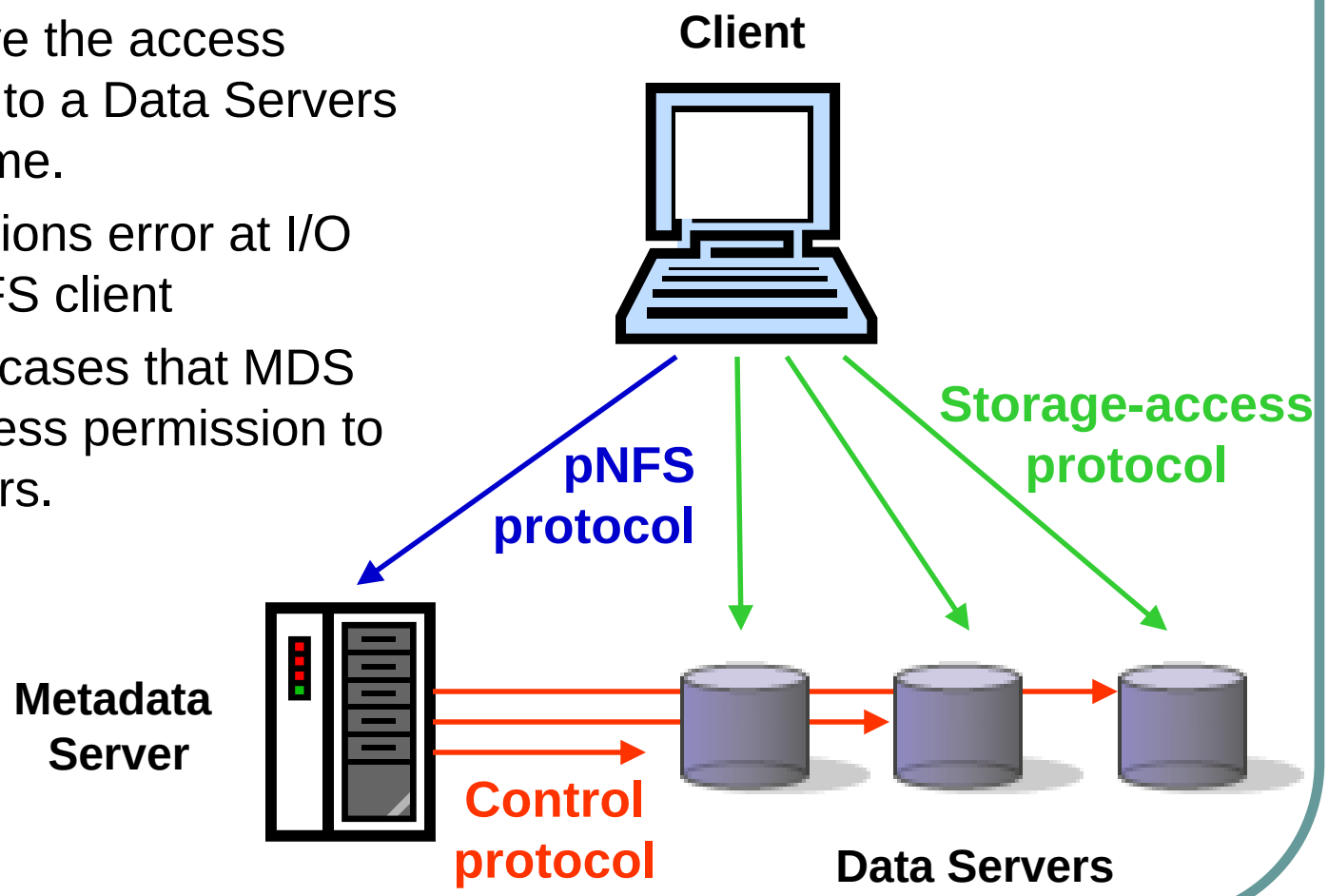
Sorin Faibish (Discussed with Mike Eisler and David Black?)
sfaibish@emc.com

Outline

- Problem Statement
- Different Layout examples
- Protocol Gaps
- Proposed Remedies
- Implementation Ideas
- Questions

Problem Statement

- No error in cases that a client doesn't have the access permission to a Data Servers at mount time.
- No permissions error at I/O time of pNFS client
- No error in cases that MDS has no access permission to Data Servers.



pNFS block layout

- Client checks access to storage servers and mounts as NFS if lack access
- MDS cannot mount and export a share if it has no access to storage – error logged
- In I/O phase client fallback to NFS on lack of access to storage and return to pNFS when access is re-established
- No errors are logged in neither case by client
- Admin can take correction measures
- Access via MDS is scalability limitation

pNFS file/object layout

- Client doesn’t check access to Data Servers at mount time
- MDS exports shares assuming that it has access to all Data Servers but doesn’t explicitly check access
- In I/O phase client fallback to MDS on lack of access to data server for which it has valid layout – no difference in access type
- No errors are logged in neither case by client or MDS
- Admin cannot take correction measures and access via MDS is never detected
- Access via MDS is scalability limitation

Protocol gaps

1. Client doesn't communicate to the MDS access denial due to a permission issue to DS
2. MDS deliver valid layout to clients that have no permission to a DS
3. The permission problem is not reported at mount time (/ is pNFS mounted) and may have a performance penalty during I/O
4. No guarantees that fallback to MDS will be able to deliver the I/O when access is denied to allow redirection
5. pNFS specification does not address the protocol between the MDS and DS (should it?)

Proposed Remedies (protocol)

1. Add permission checks of the clients to access all the Data Servers (using a list sent by MDS) at mount time.
2. Add new client error case when client cannot access a Data Server at mount time and propagate to MDS
3. Add permission check of MDS to DS after a client permission access error report to that DS
4. Add new I/O error when a pNFS client cannot access a DS that was accessible at mount time and then ask for the re-direct

Proposed Remedies (recommend)

5. The pNFS server that granted a layout to the client, should check that the client has access to the storage devices (files, luns, or objects).
6. pNFS client should add a new mount switch –pNFS to inform the pNFS server of client's pNFS access intention and log on both (client/server) in case of failure
7. pNFS MDS should check that it can perform normal I/Os to any device it hands out in a pNFS layout

Implementation Ideas

- Add an error case into LAYOUTRETURN or LAYOUTCOMMIT.
- Add a new layout return type that is "FSID with prejudice", i.e., return all layouts for this FSID and tell the server that the reason for the return is a connectivity issue
- Add periodic access permission checks retries and return layout only after several retries
- Add a new mount switch –pNFS and a possible error on pNFS optimization that didn't work and carries on using plain NFS (not pNFS) to the MDS

Questions

- Should we leave this entire issue as an implementation detail?
- Should we include protocol changes to address the scalability limitation to pNFS “scalable” protocol?
- If we answer yes to protocol changes should we introduce a new layout command or modify LAYOUTGET, LAYOUTCOMMIT?
- Should we amend/enhance NFSv4.1 or leave it for v4.2?

Agenda
9:00 – 11:30 am

- 9:00 Intro/Blue Sheets/Note Well/Agenda Bash (Pawlowski)
- 9:05 Server Side Copy Offload - (Lentini)
  - draft-lentini-nfsv4-server-side-copy-02.txt
- 9:25 Federated FS - (Lentini)
  - "Using DNS SRV to Specify a Global File Name Space with NFS version 4"
    - <draft-ietf-nfsv4-federated-fs-dns-srv-namespace-01.txt>
  - "Administration Protocol for Federated Filesystems"
    - <draft-ietf-nfsv4-federated-fs-admin-01.txt>
  - "Requirements for Federated File Systems"
    - <draft-ietf-nfsv4-federated-fs-reqts-03.txt>
  - "NSDB Protocol for Federated Filesystems“
    - <draft-ietf-nfsv4-federated-fs-protocol-01.txt>
- 9:40 NFS operation over IPv4 and IPv6 (Alex RN)
  - <draft-alexrn-nfsv4-ipv6-00.txt>
- 10:10 NFSv4 Multi-Domain Access (Adamson)
  - <draft-adamson-nfsv4-multi-domain-access.txt>
- 10:25 Proposal for an NFSv4 extension to allow the use of NFS clients as pNFS data servers (Adamson)
  - <draft-myklebust-nfsv4-pnfs-backend-00.txt>
- 10:55 Access checks and pNFS (Sorin Faibish)
- 11:25 Wrapup (Pawlowski)
- 11:30 End

Agenda
9:00 – 11:30 am

- 9:00 Intro/Blue Sheets/Note Well/Agenda Bash (Pawlowski)
- 9:05 Server Side Copy Offload - (Lentini)
  - draft-lentini-nfsv4-server-side-copy-02.txt
- 9:25 Federated FS - (Lentini)
  - "Using DNS SRV to Specify a Global File Name Space with NFS version 4"
    - <draft-ietf-nfsv4-federated-fs-dns-srv-namespace-01.txt>
  - "Administration Protocol for Federated Filesystems"
    - <draft-ietf-nfsv4-federated-fs-admin-01.txt>
  - "Requirements for Federated File Systems"
    - <draft-ietf-nfsv4-federated-fs-reqts-03.txt>
  - "NSDB Protocol for Federated Filesystems“
    - <draft-ietf-nfsv4-federated-fs-protocol-01.txt>
- 9:40 NFS operation over IPv4 and IPv6 (Alex RN)
  - <draft-alexrn-nfsv4-ipv6-00.txt>
- 10:10 NFSv4 Multi-Domain Access (Adamson)
  - <draft-adamson-nfsv4-multi-domain-access.txt>
- 10:25 Proposal for an NFSv4 extension to allow the use of NFS clients as pNFS data servers (Adamson)
  - <draft-myklebust-nfsv4-pnfs-backend-00.txt>
- 10:55 Access checks and pNFS (Sorin Faibish)
- 11:25 Wrapup (Pawlowski)
- 11:30 End

Tack
(Thanks)

広島でお会いしましょう。

(See you in Hiroshima)