

NFS/RDMA

Tom Talpey
Network Appliance
tmt@netapp.com

RDMA

- ▶ **“Remote Direct Memory Access”**
- ▶ **Read and write of memory across network**
 - Hardware assisted
 - OS bypass
 - Application control
 - Secure
- ▶ **Examples:**
 - Infiniband
 - iWARP/RDDP
 - (Proprietary cluster interconnects)
 - (Virtual Interface Architecture (VI))

Benefits of RDMA

- ▶ **RDMA greatly reduces overhead via:**
 1. **Data copy avoidance**
 - **Especially in the receive path**
 - **Each data copy adds 2x line rate BW to memory bus**
 2. **Hardware offload**
 3. **OS bypass**
 - **Direct access to network from application**
- ▶ **If it hurts at 1Gb, it's deadly at 10Gb**
 - **And Moore's law won't fix it**
 - **Memory busses aren't scaling fast enough**

Relative benefits of RDMA

▶ High client benefits:

- Copy avoidance
- Data alignment
- Processing offload
- OS bypass (kernel, trap and interrupt avoidance)

▶ Substantial Server benefits:

- Data alignment
- Processing offload
- Interrupt avoidance

File protocol RDMA benefits

- ▶ **Separation of header and data**
- ▶ **Zero-copy enables 0-touch directio, or removes one copy in cache path**
- ▶ **Operations map to wire ops 1-1**
- ▶ **RDMA is perfect for files**
 - **And pretty durn good for others too**

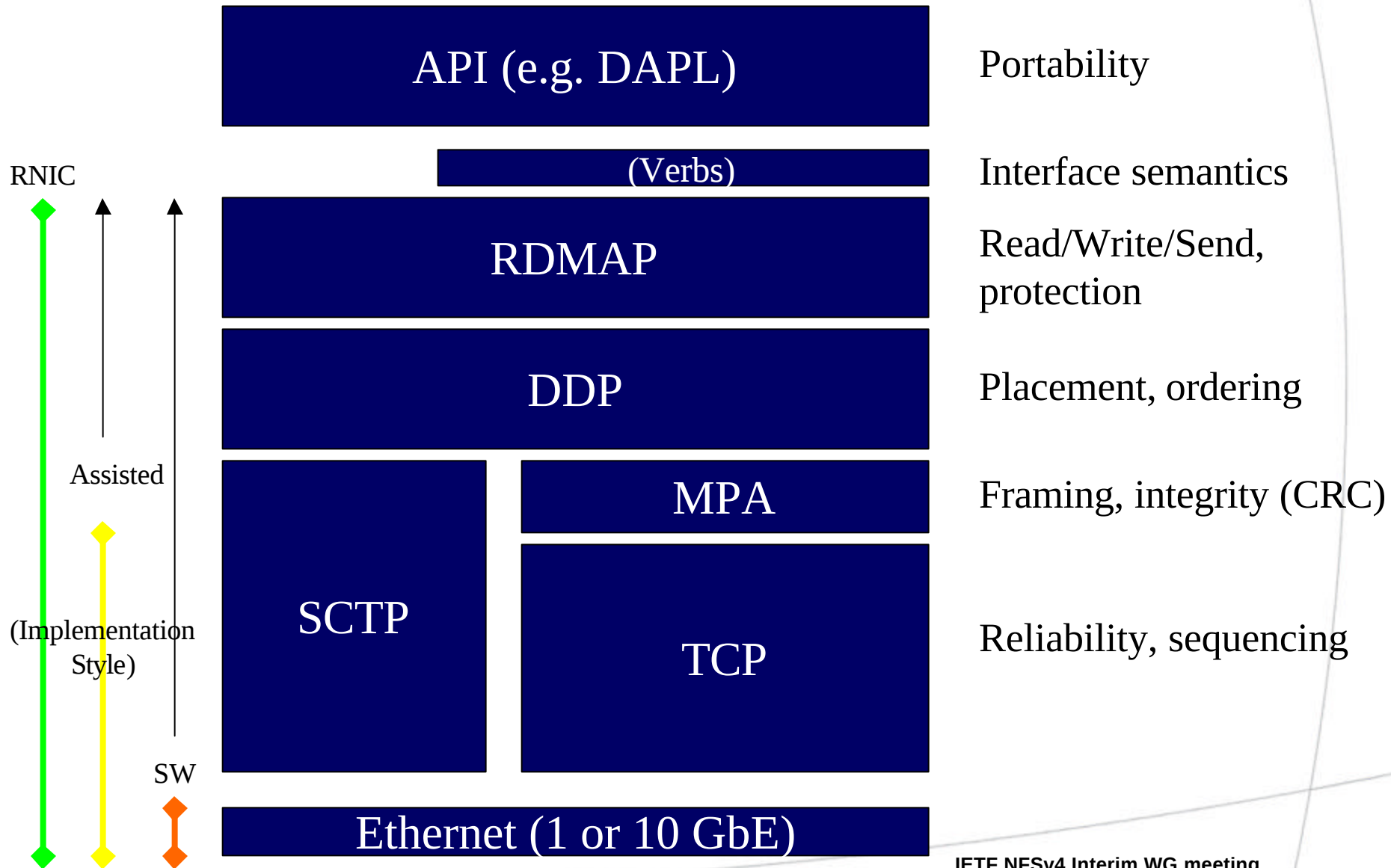
Why not just TOE?

- ▶ **TOE reduces stack overhead**
 - But stack overhead is relatively small
- ▶ **TOE does not avoid receive data copies**
 - Unless TOE includes ULP processing such as NFS header cracking, SSL, etc.
- ▶ **TOE requires substantial reassembly buffer space**
- ▶ **No defined TOE API**
- ▶ **Savings from TOE are not general to all platforms**

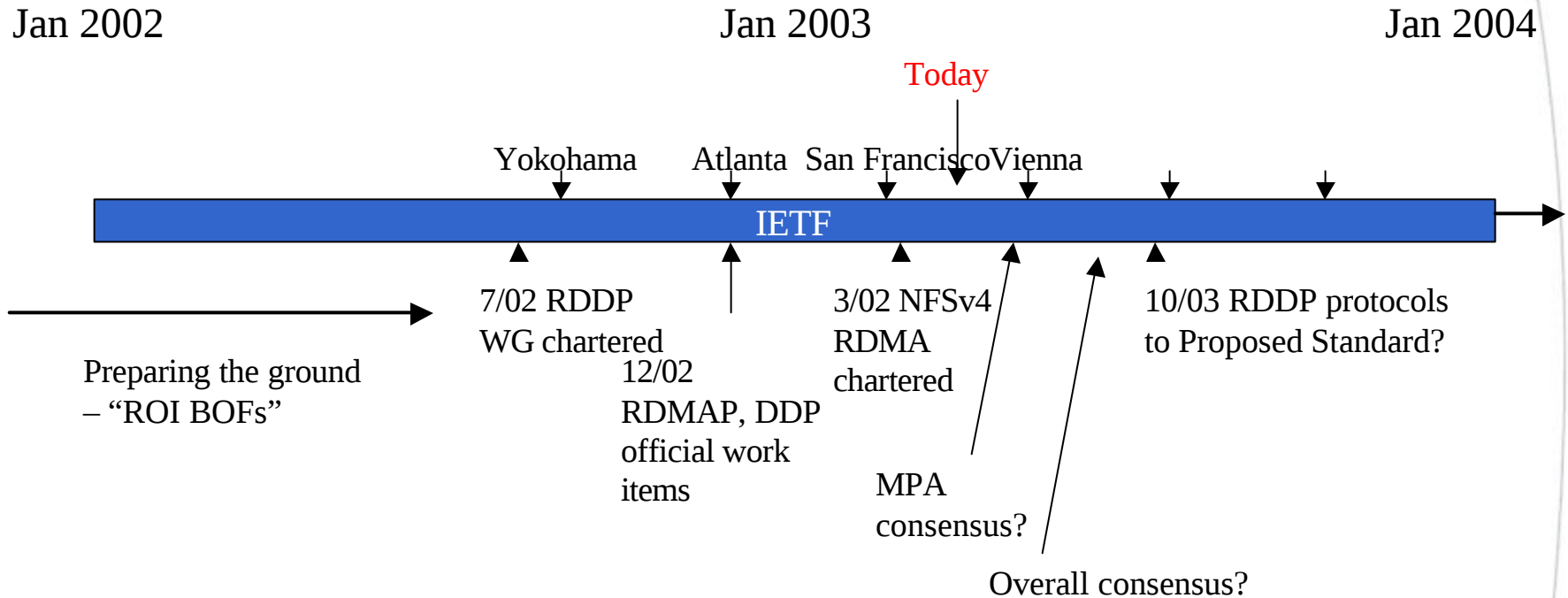
IETF RDDP Working Group

- ▶ **Specify RDMA over TCP, “iWARP”:**
 - **RDMAP (RDMA Protocol)**
 - **DDP (Direct Data Placement Protocol)**
 - **MPA (Markers with PDU Alignment – framing)**
- ▶ **Also consider RDMA over SCTP**

iWARP Components



IETF RDDP WG Timeline



NFS/RDMA Internet-Drafts

- ▶ **RDMA Transport for ONC RPC**
 - Basic ONC RPC transport definition for RDMA
 - Transparent, or nearly so, for all ONC ULPs
- ▶ **NFS Direct Data Placement**
 - Maps NFS v2, v3 and v4 to RDMA
- ▶ **NFSv4 RDMA and Session extensions**
 - Transport-independent Session model
 - Enables exactly-once semantics
 - Sharpens v4 over RDMA

ONC RPC over RDMA

- ▶ **Internet Draft, published May 16**
 - **draft-callaghan-rpcrdma-00**
 - **Brent Callaghan and Tom Talpey**
- ▶ **Defines RDMA RPC transport type**
- ▶ **Goal: Performance**
 - **Achieved through use of RDMA for copy avoidance**
 - **No semantic extensions**

NFS Direct Data Placement

- ▶ **Internet Draft, published May 16**
 - draft-callaghan-nfsdirect-00
 - Brent Callaghan and Tom Talpey
- ▶ **Defines NFSv2 and v3 operations mapped to RDMA**
 - READ and READLINK
- ▶ **Also defines NFSv4 COMPOUND**
 - READ and READLINK

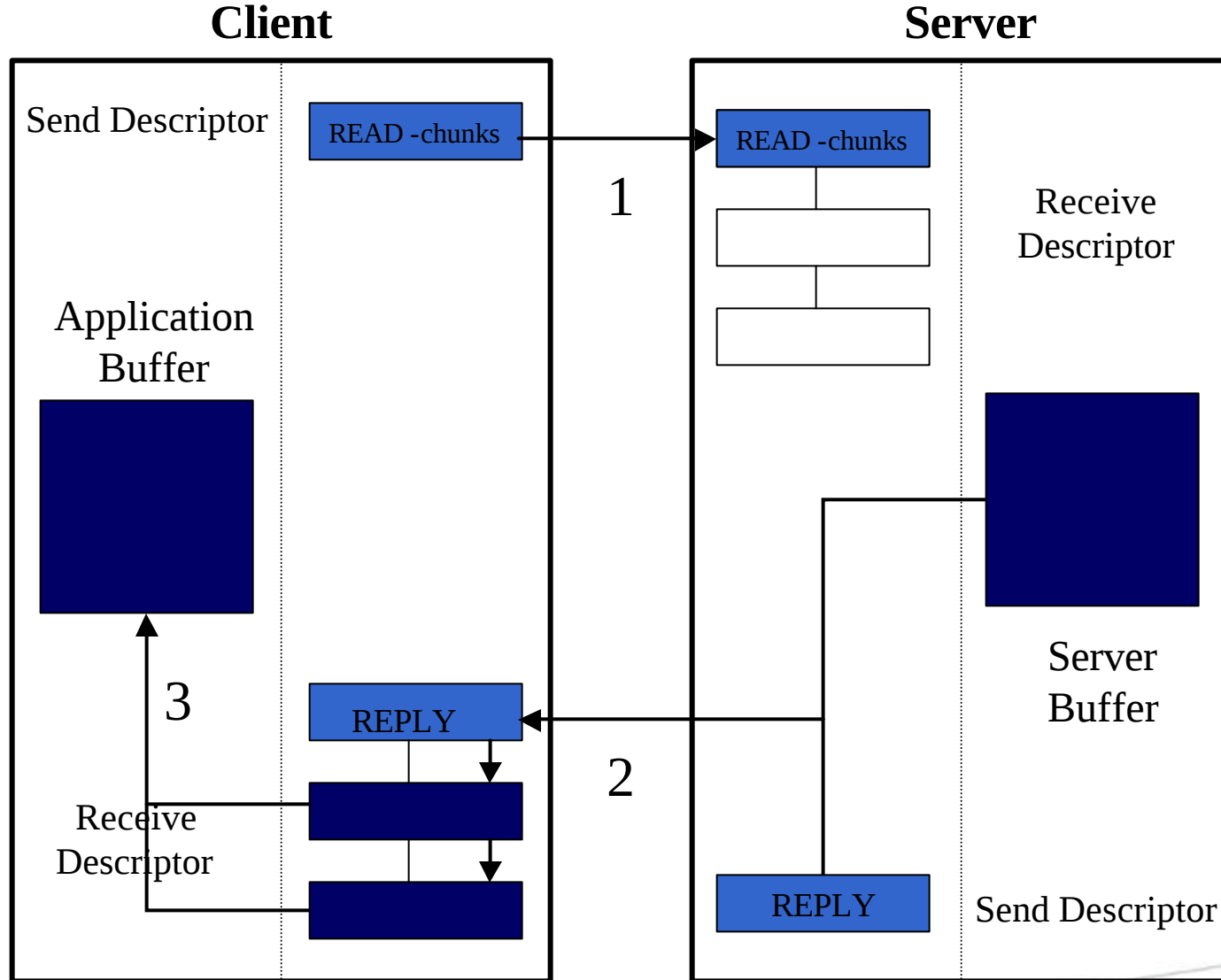
NFSv4 RDMA and Session extensions

- ▶ **References ONC RPC RDMA document**
- ▶ **Internet Draft, published May 16**
 - draft-talpey-nfsv4-rdma-sess-00
 - Tom Talpey and Spencer Shepler
- ▶ **Goal: enable best use of Transport by NFSv4**
 - Size negotiations
 - Channel management
 - Connection model (supports TCP, IB and iWARP)
- ▶ **Also**
 - Sessions
 - Exactly-once semantics

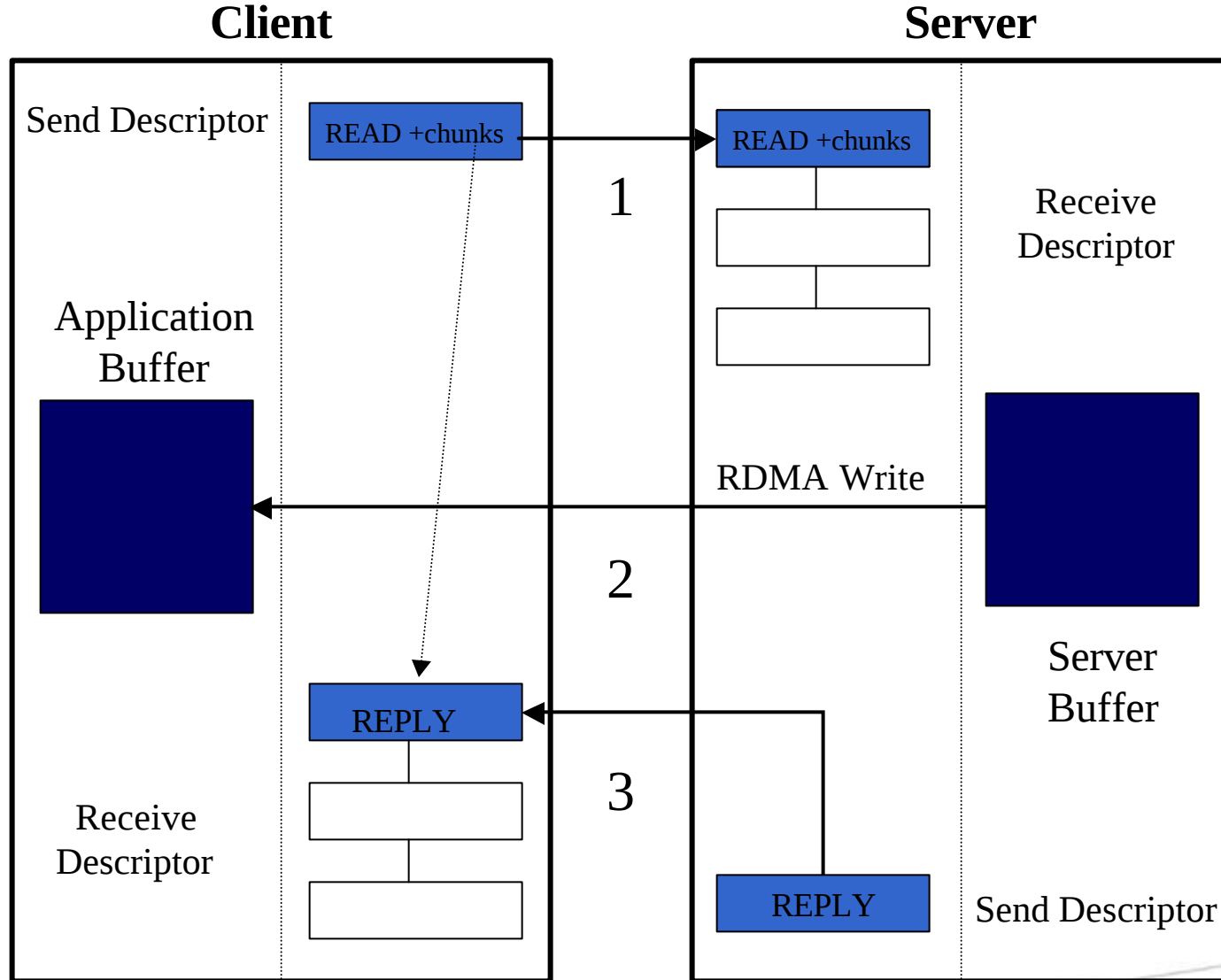
DAT – Direct Access Transport

- ▶ **Common requirements and an abstraction of services for RDMA - Remote Direct Memory Access**
 - Portable, high-performance transport underpinning for DAFS and applications
 - Defines communications endpoints, transfer semantics, memory description, signalling, etc.
- ▶ **Transfer models:**
 - Send (like traditional network flow)
 - RDMA Write (write directly to advertised peer memory)
 - RDMA Read (read from advertised peer memory)
- ▶ **Transport independent**
 - 1 Gb/s VI/IP, 10 Gb/s InfiniBand, future RDMA over IP
- ▶ **<http://www.datcollaborative.org>**

Inline Read



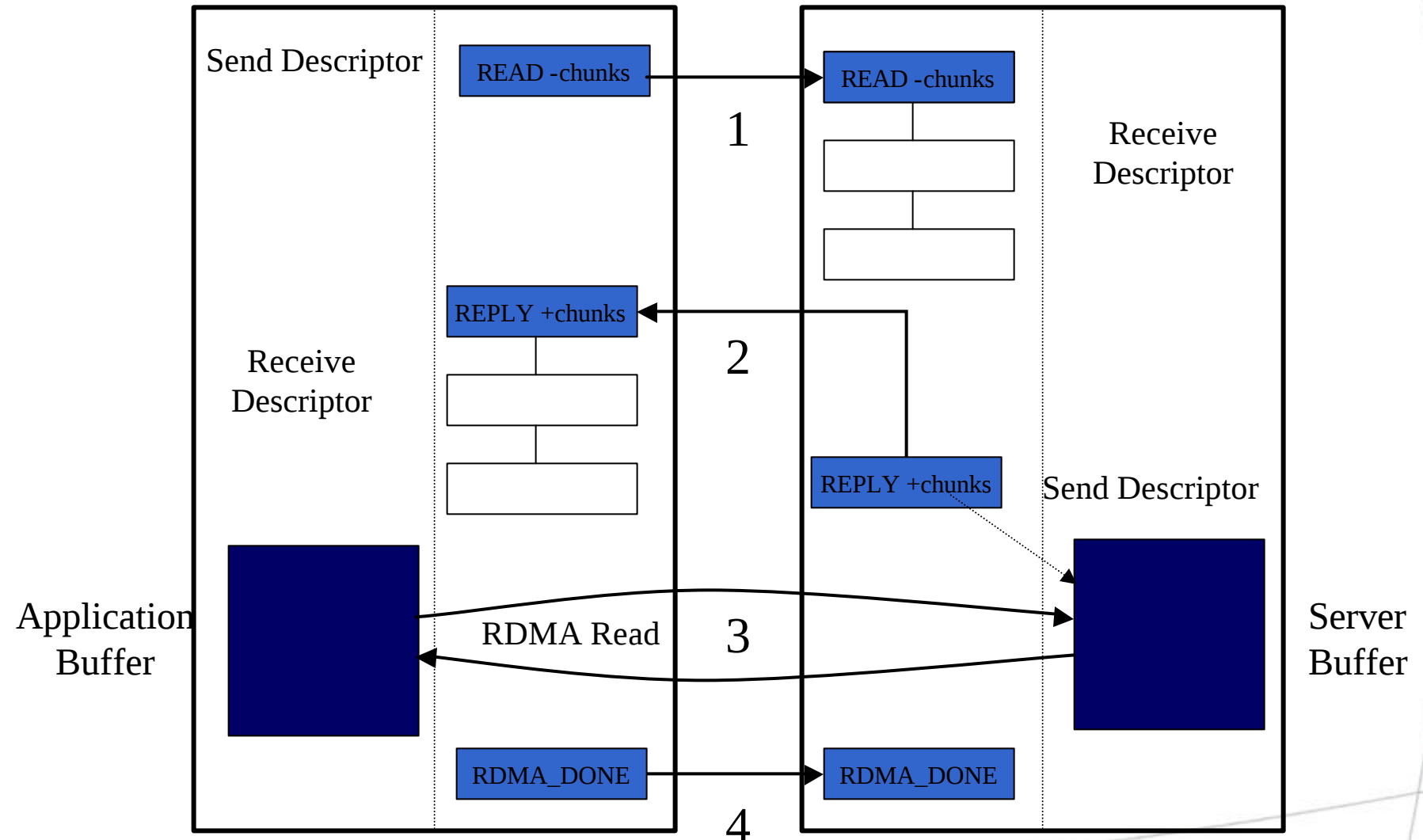
Direct Read (write chunks)



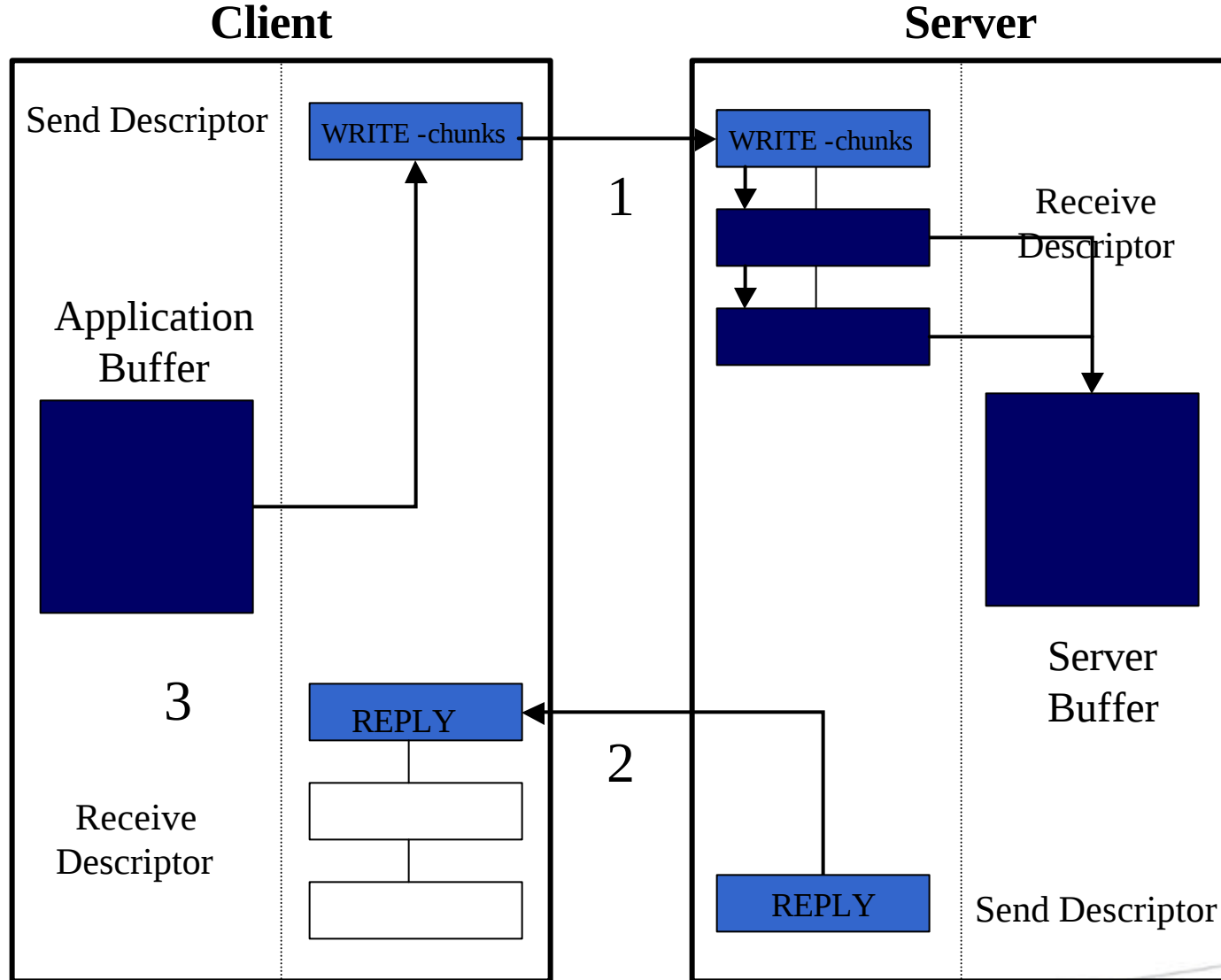
Direct Read (read chunks) – Rarely used

Client

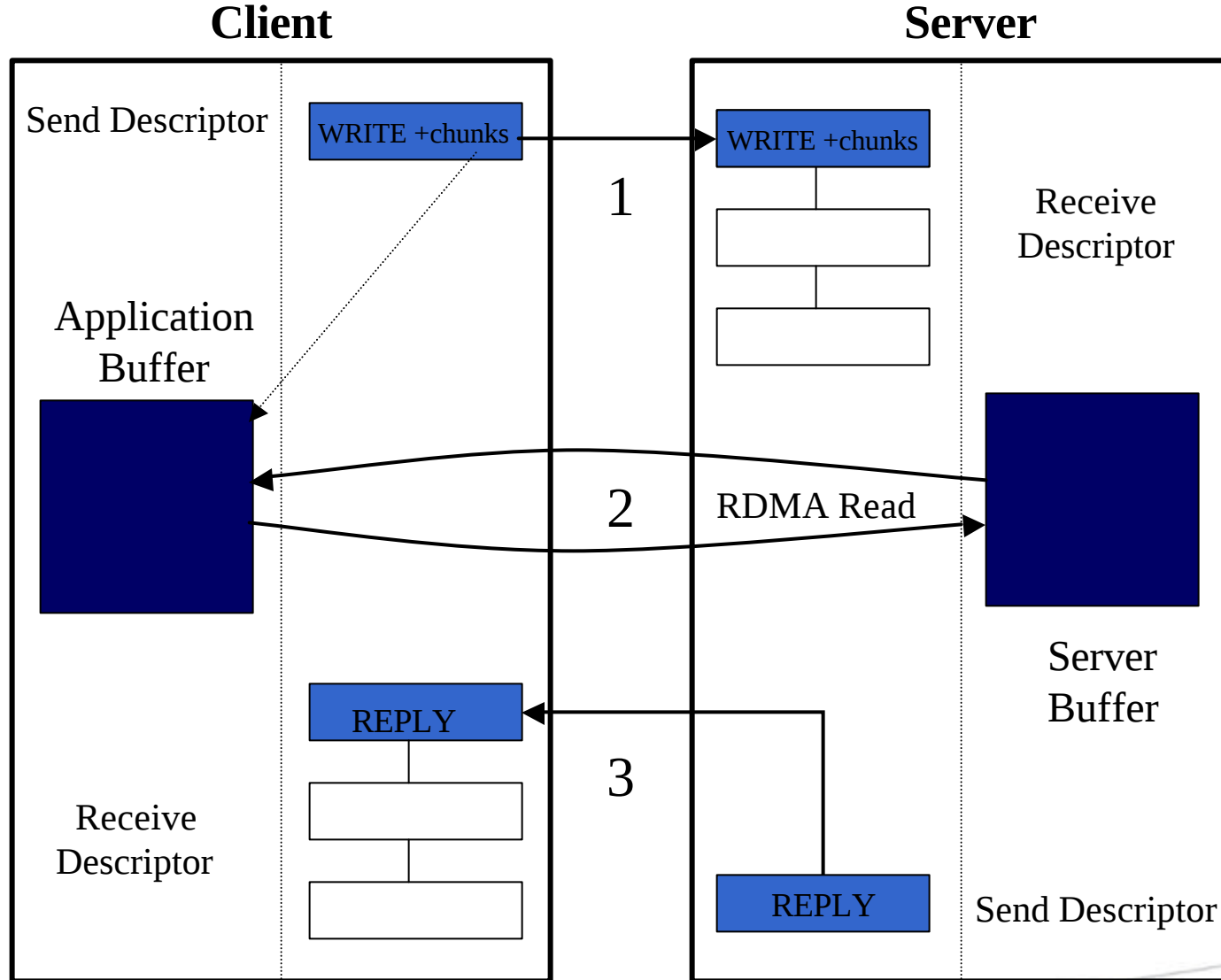
Server



Inline Write



Direct Write (read chunks)



NFSv4 RDMA and Session Extensions

Tom Talpey
Network Appliance
tmt@netapp.com

The Proposal

- ▶ **Add a session to NFSv4**
- ▶ **Enable operation on single connection**
 - Firewall-friendly
- ▶ **Enable multiple connections for trunking, multipathing**
- ▶ **Enable RDMA accounting (credits, etc)**
- ▶ ***Provide Exactly-Once semantics***
- ▶ **Transport-independent**

5 new ops

- ▶ **SESSION_CREATE**
- ▶ **SESSION_BIND**
- ▶ **SESSION_DISCONNECT**
- ▶ **OPERATION_CONTROL**
- ▶ **CB_CREDITRECALL**

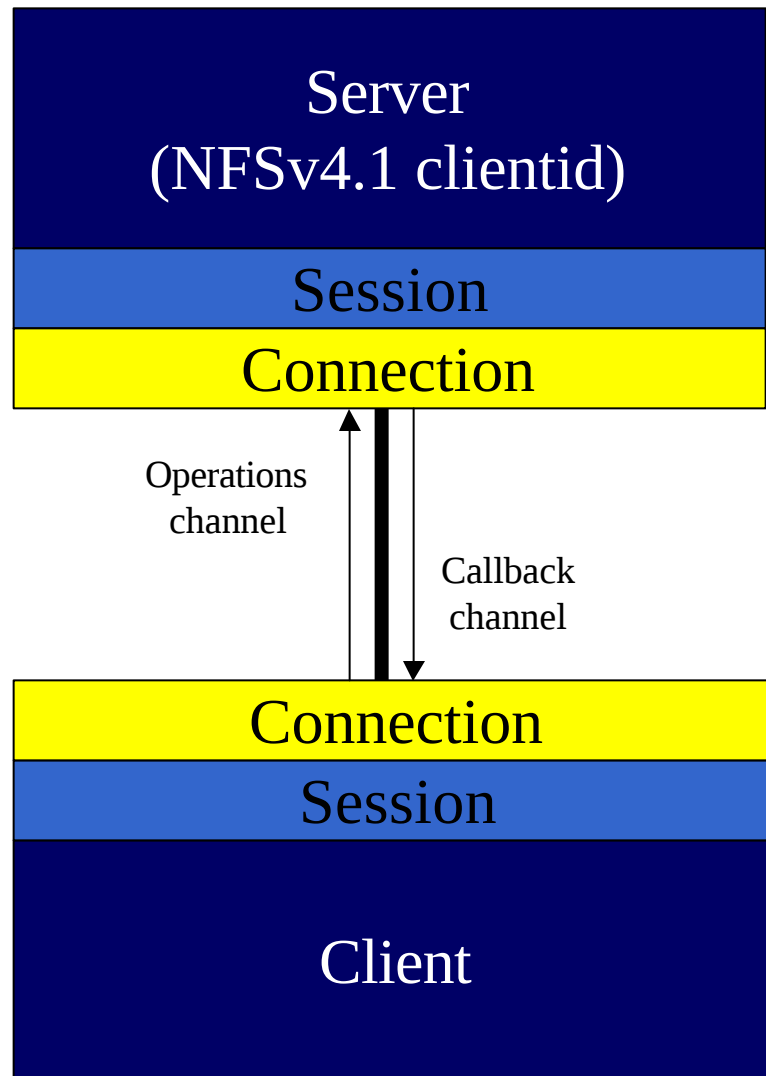
Channels versus Connections

- ▶ **Channel: a connection bound to a specific purpose:**
 - Operations (1 or more connections)
 - Callbacks (typically 1 connection)
- ▶ **Multiple connections per client, multiple channels per connection**
 - Many-to-many relationship
- ▶ **All operations require a channelid**
 - Encoded into COMPOUND

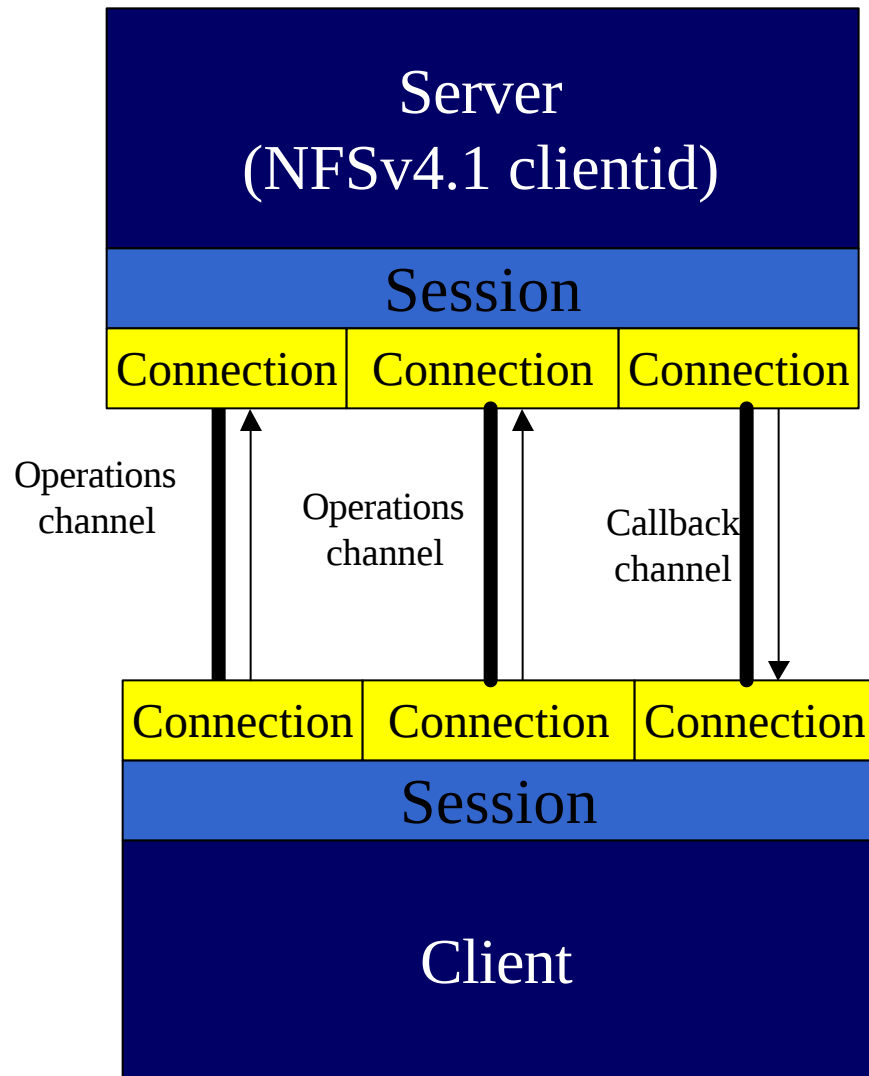
Session Connection Model

- ▶ **Client connects to server**
- ▶ **First time only:**
 - New session via **SESSION_CREATE**
- ▶ **Initialize channel:**
 - Bind “channel” via **SESSION_BIND**
 - May bind operations, callback to same connection
 - May connect additional times
 - Trunking, multipathing, failover, etc.
- ▶ **CCM fits perfectly here**
- ▶ **If connection lost, may reconnect to existing session**
- ▶ **When done:**
 - Destroy session context via **SESSION_DISCONNECT**

Example Session – single connection



Example Session – multiple connections



Example Session – single connection

- ▶ **Resource-friendly**
- ▶ **Firewall-friendly**
- ▶ **No performance impact**
- ▶ **Isn't this the way callbacks should have been spec'ed?**

Exactly-Once Semantics

- ▶ **Highly desirable, but never achievable**
- ▶ **Need flow control (N) , operation sizing (M) in order to support RDMA**
- ▶ **Flow control provides an “ack window”**
 - **Use this to retire response cache entries**
- ▶ **$N * M =$ response cache size**
- ▶ **Session provides accounting and storage**
- ▶ **Done!**

Streamid

- ▶ **A per-operation identifier in the range 0..N-1 of server's current flow control**
 - In effect, an index into an array of legal in-progress ops
- ▶ **Highly efficient processing – no lookup**
- ▶ **Used in conjunction with RPC transaction id to maintain duplicate request cache**

Chaining

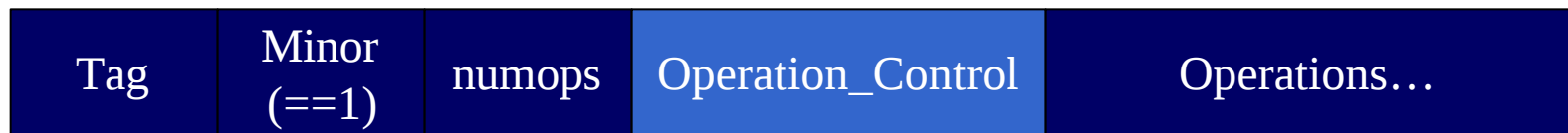
- ▶ **Problem: COMPOUND restricted in length at session negotiation**
- ▶ **Chaining provides strict sequencing of requests**
- ▶ **Start, middle, end flags (and none)**
- ▶ **Maintains current and saved filehandles like COMPOUND**

Connection model and negotiation

- ▶ **Simplest form – no session at all**
- ▶ **Session creation enables use of RDMA**
 - Per-channel (connection) RDMA mode too
 - Mix TCP and RDMA channels per-client!
- ▶ **TCP mode if either RDMA mode is off**
- ▶ **Dynamic enabling of RDMA at session binding**
 - After RDMA mode, sizes, credits, etc exchanged
- ▶ **Statically enabled RDMA (e.g. Infiniband) also supported**
 - Requires preposted buffer

V4 Protocol integration

- ▶ Piggyback on existing COMPOUND
- ▶ New OPERATION_CONTROL first in each session COMPOUND request and reply
- ▶ Conveys channelid, streamid, and chaining



V4 efficiencies

- ▶ **No need for sequenceid**
 - Field will stay, but ignored under a session
- ▶ **No need for clientid per-op**
 - Clientid may be provided as zero
- ▶ **Each request within session renews leases**
- ▶ **OPEN_CONFIRM not needed**
- ▶ **CCM is enabled**

Summary

- ▶ **This is a v4 proposal, not just RDMA**
- ▶ **Sessions are a substantial simplification**
 - Clients associated with connections
 - Recoverable
 - Firewall-friendly
- ▶ **Exactly-once semantics are enabled**

RDMA Requirements

- ▶ **Can make simple statement:**
 - RDMA concepts map well to RPC and file protocols
 - These concepts benefit all transports and server implementations
 - The “RDMA changes” are in fact a fundamental, beneficial alignment
- ☞ **These are transport requirements, general to RDMA and TCP.**
- ▶ **Much text exists already in the documents**