NFS/RDMA Standards and Implementation Update

Tom Talpey

Technical Director

Network Appliance, Inc.

tmt@netapp.com

Outline

- Take a simplified view
- NFS/RDMA Goals
- NFS/RDMA Components
- A walk up the stack
- A look at the implementations

NFS/RDMA Goals

Motivation

- Reduce NFS overhead
- Eliminate data copies
- Use new high-performance transports
- Overcome traditional implementation bottlenecks

Why NFS/RDMA?

- **Reduced client-side overhead**
  - Achieved through elimination of data copies
  - RPC (NFS) data handled in network buffers
    - Arbitrarily aligned in packet
    - Kernel address space (buffer cache, mbuf)
    - ► data copies

Why NFS/RDMA?

Additional benefits

- Reduced latency
- Access to bandwidth
  - CPU overhead not the limiting factor
- Improve client throughput (ops)
- NFS performance ~ local FS

What it *doesn't* provide

- Does not, by itself, increase bandwidth
  - NFS limits are not due to the wire
  - (Bandwidth from server cache is better)
- Does not increase server performance
  - Unless the server is out of CPU (a different problem)

Typical NFS/RDMA Client Improvement*

- **NFS per-operation overhead similar to best Fibre Channel**
- Bandwidth approaching full server spindle/wire/bus capacity, at very low client CPU/MB/sec
- Low latencies improve metadata ops (attribute checks)

* (Stock Linux 2.4 NFSv3 kernel VFS, over 4xIB)

NFS/RDMA Components

Protocol components

- An RDMA fabric
  - e.g. iWARP and/or Infiniband, with kDAPL
- RPC/RDMA
- NFS v3
- NFS v4
- NFS v4.1/Sessions

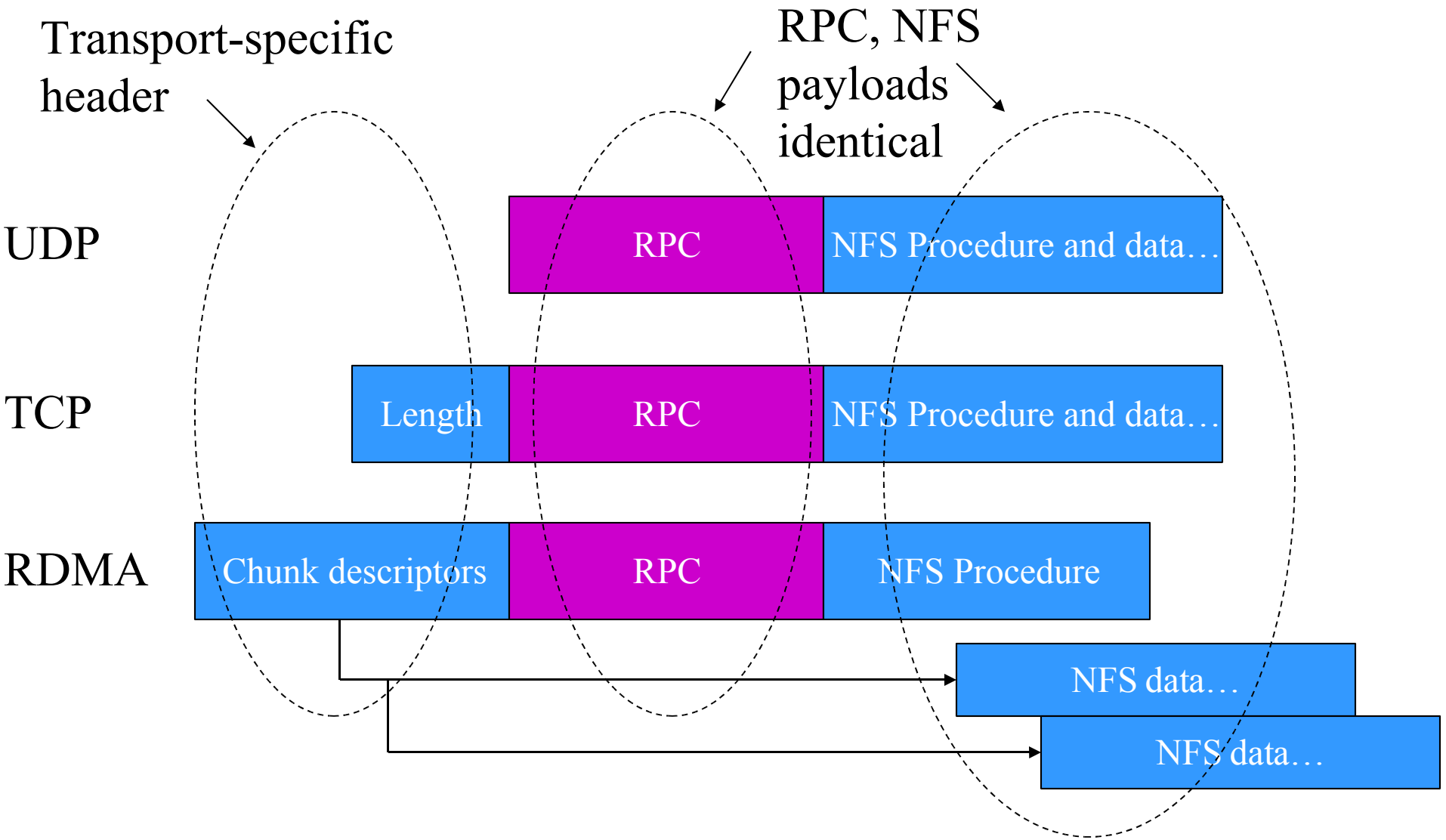
Implementation components

- RDMA API
  - e.g. kDAPL
- RPC transport switch
- RPC/RDMA support
- NFS client and server
- **Optional** RDMA-aware NFS tweaks

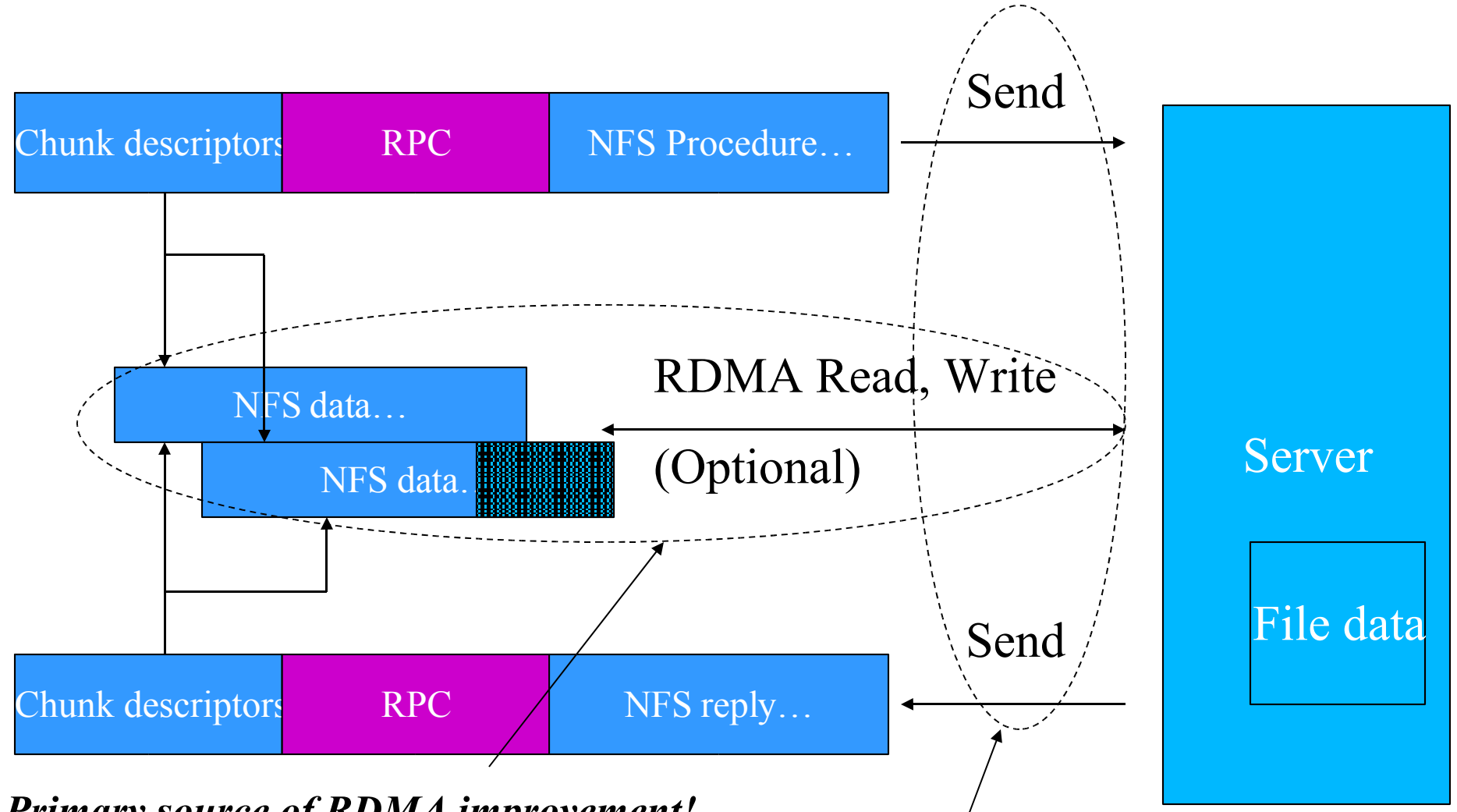
RPC/RDMA

- Direct placement chunks
- Other functions
  - Control operations
  - **Credits**
  - Input processing efficiency (XID)
  - Alignment, padding, etc.
  - Versioning, errors

Transport RPC format



RPC/RDMA Transfer



Primary source of RDMA improvement!

Contributor to RDMA improvement

NFS Direct

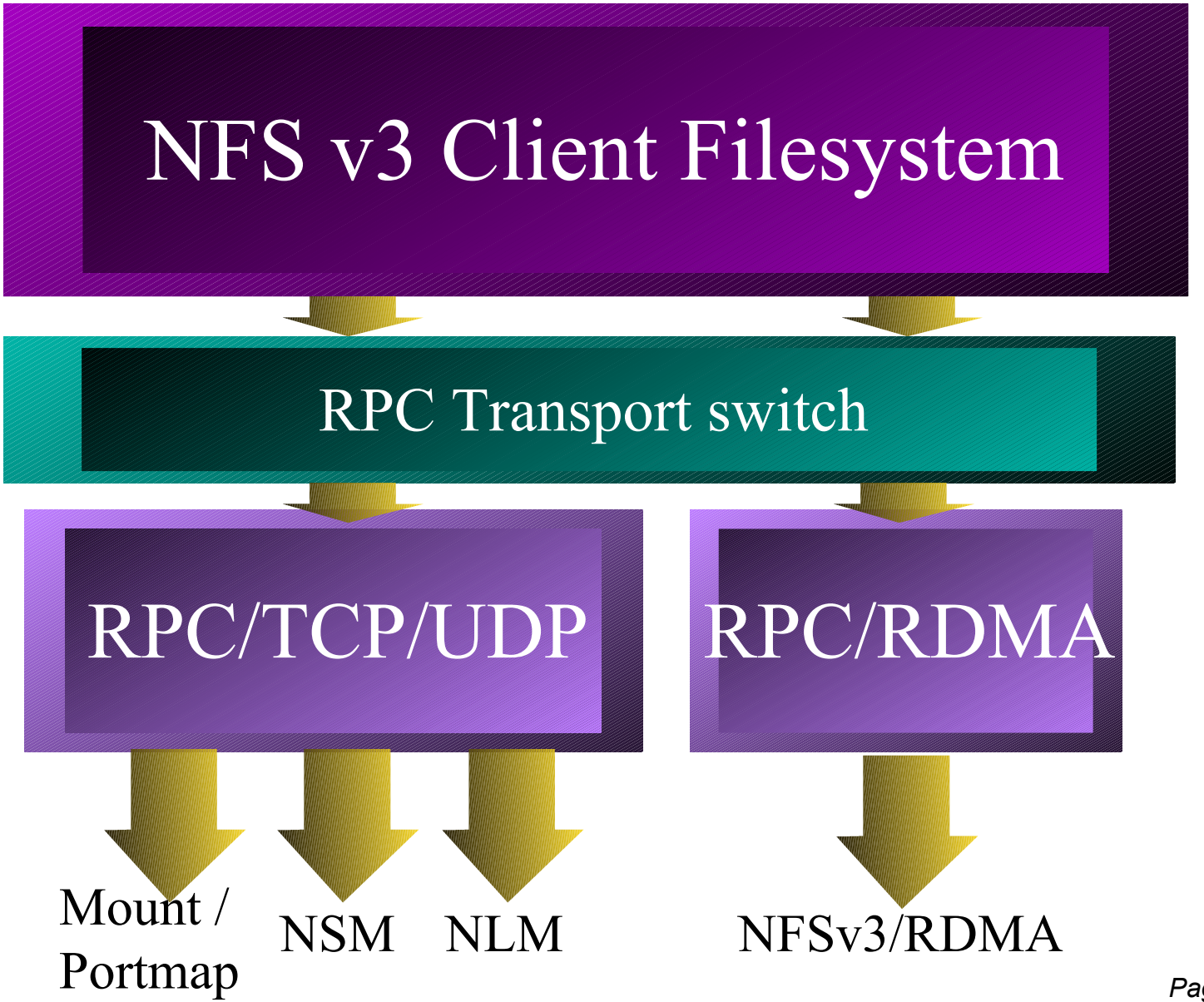
- RDMA used for read/write/readlink
- Not for RPC headers themselves
- Not used in metadata ops
- Remote placement (RDMA) improves data transfer overhead
- RDMA hardware, fabric attributes contribute to performance

NFS/RDMA Stacks

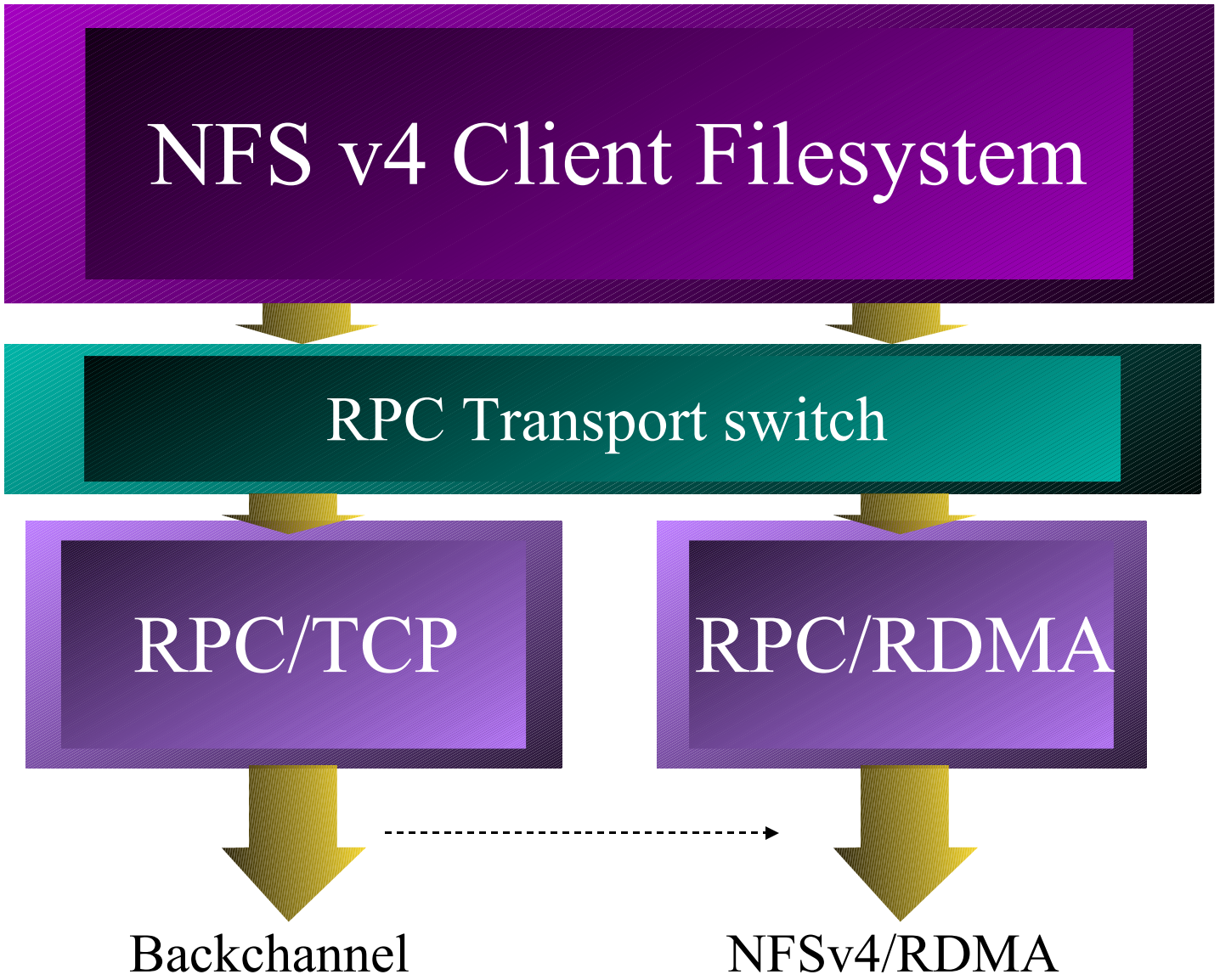
NFSv3/v4/v4.1 Stacks

- Each NFS case is different!
  - Side protocols in v3
  - Backchannel in v4
  - Session in v4.1
- RDMA used similarly for each

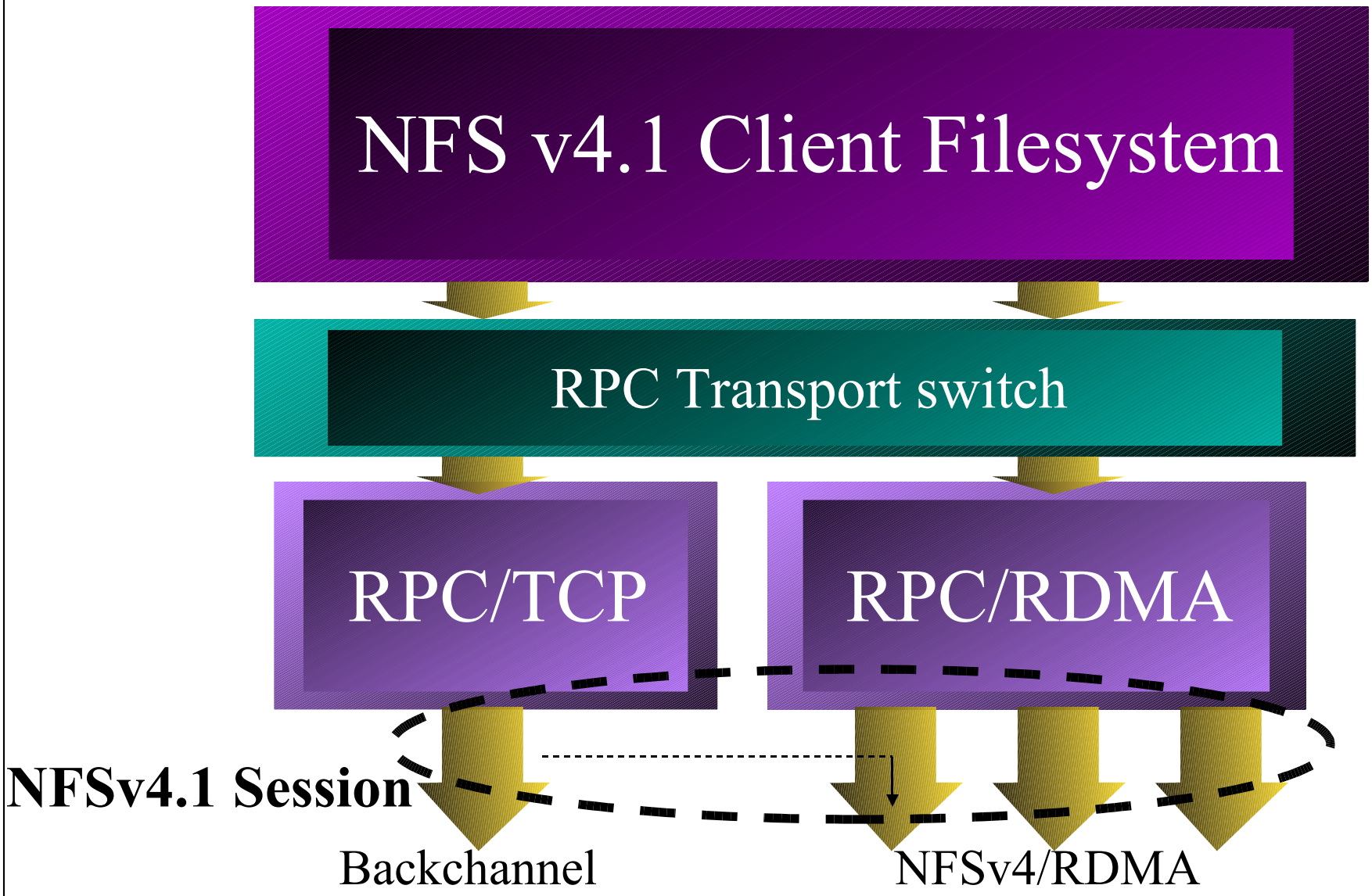
NFSv3/RDMA stack



NFSv4/RDMA stack



NFSv4.1/RDMA stack



OS Componentry

- RPC switch, mount API
  - Selectable RPC transport, and parameters
- NFS VFS largely unchanged
  - Only to support new mount API
- NFS VFS *actually* unchanged in Linux 2.4/NFSv3/RDMA prototype
  - Proof of concept and approach established

Mount operation

- mount –o udp,rsize=8192…

  mount –o tcp,rsize=32768…

  mount –o rdma,rsize=…
- Specify alternative transports and transport-specific options
- Side protocol(s) can help, but are not required
- Negotiation built-in to v4.1/session

RPC switch operation

- RPC transport switch handles transport-specific details
- Sets up transport endpoints, connect/reconnect
  - Understands address families, ports/portmap
- Manages buffer marshalling

RPC switch operation

- The switch arbitrates between NFS and:
  - Diverse transport APIs (and buffering)
    - Sockets, RDMA (kDAPL), TOE
  - Diverse address families and protocols
    - IPv4, IPv6, TCP/UDP, SCTP, iWARP, IB, etc…
- Establishes standard layering to maintain a single NFS upper layer implementation

Performance

- Watch this space for Linux 2.6 RPC/RDMA client performance results
  - Hoping for full report at Cthon05
- Pending Linux 2.6 RDMA fabric work
- Basic 2.4/NFSv3/RDMA performance at Cthon04:
  - www.cthon.org

Protocol documents

- RPC/RDMA
- NFS Direct
- NFSv4/Sessions
- Available at NFSv4 working group:
  - http://www.ietf.org/html.charters/nfsv4-charter.html

Implementations

- Linux RPC/RDMA client (2.4)
  - http://sourceforge.net/projects/nfs-rdma
- Linux RPC transport switch (2.6)
  - http://troy.citi.umich.edu/~cel/linux-2.6/
- Linux NFS/RDMA server in plan

Implementations

- Linux NFSv4.1, RDMA, etc
  - http://www.citi.umich.edu/projects/rdma/
- DAPL (Direct Access Programming Library)
  - http://www.datcollaborative.org
  - http://sourceforge.net/projects/dapl
- OpenIB (Infiniband)
- OpenRDMA (iWARP)

Questions?

IETF NFSv4 working group

Sourceforge NFS-RDMA, DAPL

CITI NFS/RDMA

Linux RPC switch patches

tmt@netapp.com