UDT BASED APPLICATION PERFORMANCE FOR HIGH SPEED DATA TRANSFER

Joshua Miller (a) uchicago. edu], Allison Heath

OVERVIEW

Problem

Over high performance wide area networks, TCP has been shown to reach a data transfer bottleneck before UDP. UDT is a proven and reliable UDP based application level protocol, but is not user friendly.

Purpose:

- Improve performance of UDT based rsync wrapper (UDR)
- Create a new application to eliminate possible overhead (UDTCAT)
- Create an ecnryption package to match the transfer rate

UDT

UDT is proven for transporting large datasets over high performance networks and has powered applications that have won the Supercomputing Bandwidth Challenge in 2006, 2008 and 2009.

Benefits:

- Disk-disk transfer rate is 9 Gbps over 10 Gbps WAN
- Forms the underlying technology for products already on the market
- High performance with a single flow

UDR

UDR (developed by the Laboratory for Advanced Computing):

- One dependency, openSSL encryption
- Simple prefix wrapper to rsync
- Compatible with all rsync options

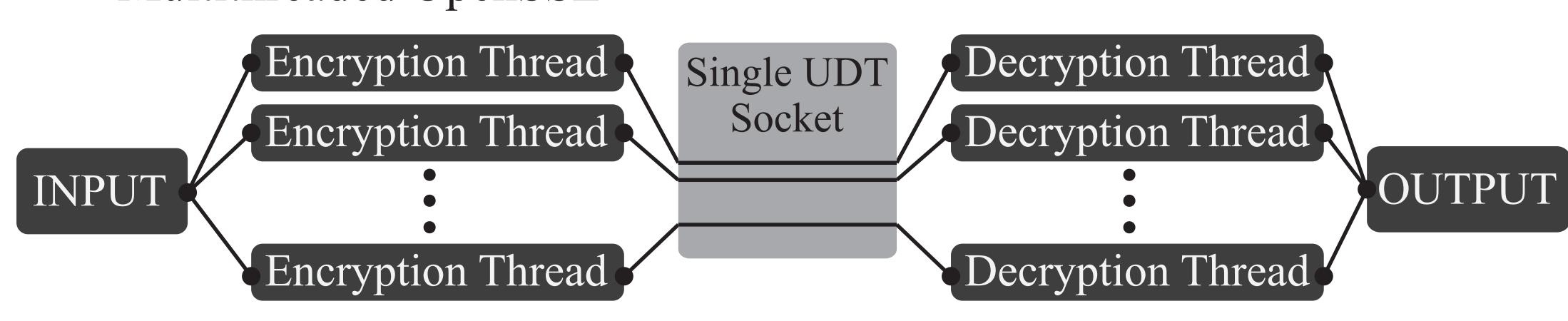
UDR uses rsync and scp as they are, there is no need to patch or install a new version.

UDT APPLICATIONS

Threaded Encryption:

An encryption package was needed to match the throuput of the UDT tools.

Multithreaded OpenSSL



UDR Contributions:

- Re-implemented upr for use with scp to reduce overhead
- Threaded encryption
- Attempts to buffer data to optimal block size did not increase throughput

local\$ udr [rsync/scp command]

UDTCAT:

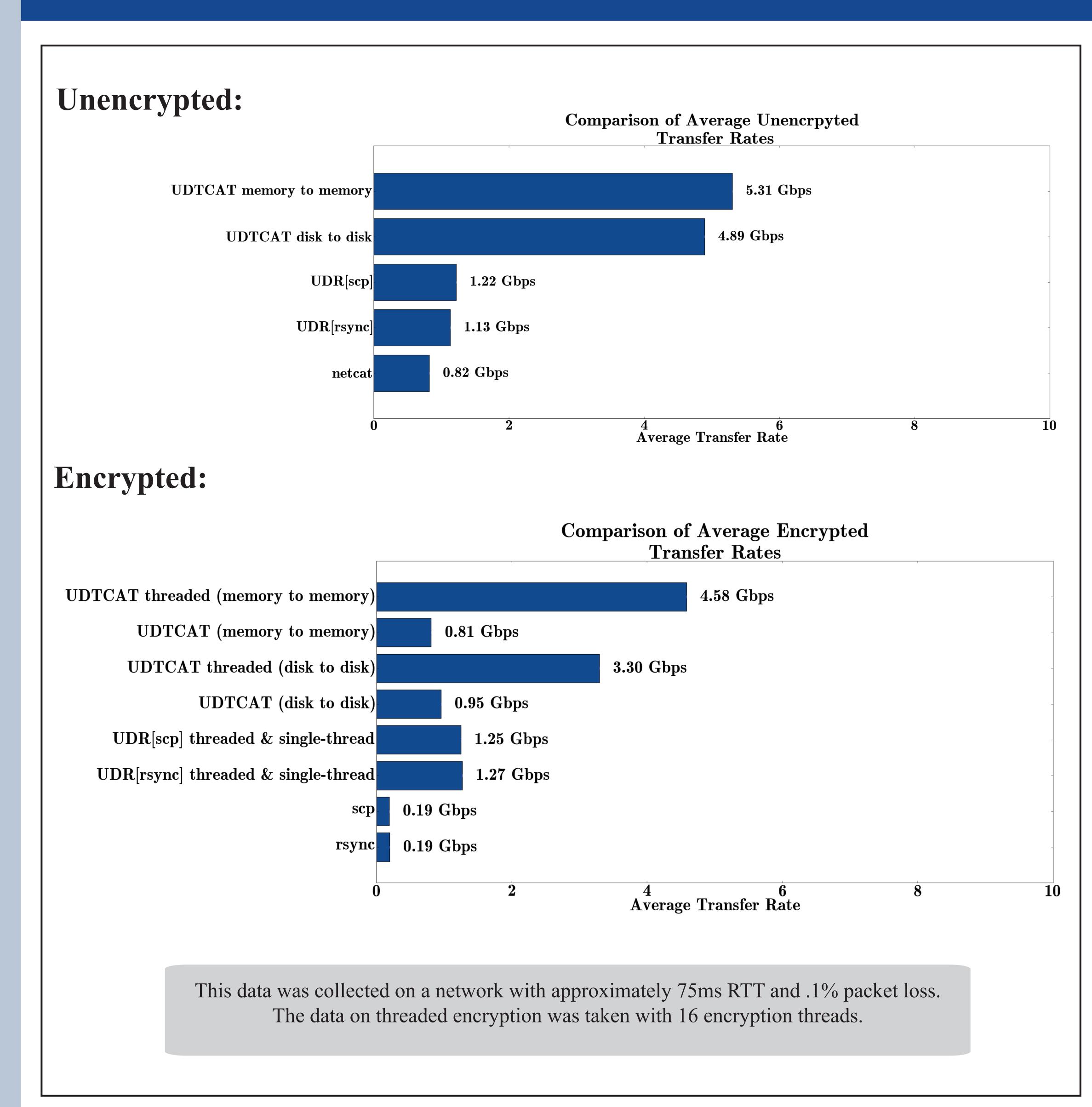
- Originally based on netcat functionality
- Includes multithreaded encryption and authentication
- Useful for piping large data streams securely over UDT
- When used with netcat and a two way pipe, UDTCAT can bind any TCP port on localhost to be piped over UDT

client\$ uc host port [< infile]
server\$ uc -l port [> outfile]

UDCAT works just like a secure netcat

Example Transfers:

PERFORMANCE



LOOKING FORWARD

Future plans in this project include:

- Adding option to patch rsync to remove buffer size bottleneck
- Continued development of UDTCAT for easier file transfer and more robust piping options









