Stork Data Scheduler for Windows Azure

Tevfik Kosar
State University of New York at Buffalo

June 2, 2011
Cloud Futures Workshop, Redmond, WA

Big Data ⇨ New Trends

> "In the future, U.S. international leadership in science and engineering will increasingly depend upon our **ability to leverage this reservoir of scientific data** captured in digital form."
>
> *- NSF Vision for Cyberinfrastructure*

Big Data ⇨ New Trends

> “One of the main objectives of the future research programs should be **enhancing the data management infrastructure**... since the users should be able to focus their attention on the information content of the data, rather than how to discover, access, and use it.”
>
> *-Strategic Plan for US Climate Change Program*

Big Data ⇨ New Trends

> "In the same way that the load register instruction is the most basic operation provided by a CPU, so is the **placement of data** on a storage device... It is therefore essential that at all levels data placement tasks be treated in the same way computing tasks are treated [Kosar2004]."
>
> *- DOE Report on Data Management Challenges*

Our Vision

"Data storage resources and the tasks related to data access should be considered as first class entities just like computational resources and compute tasks, and not simply the side effect of computation."

- Ph.D. Thesis, Tevfik Kosar, 1999-2005

Data-Aware Computing

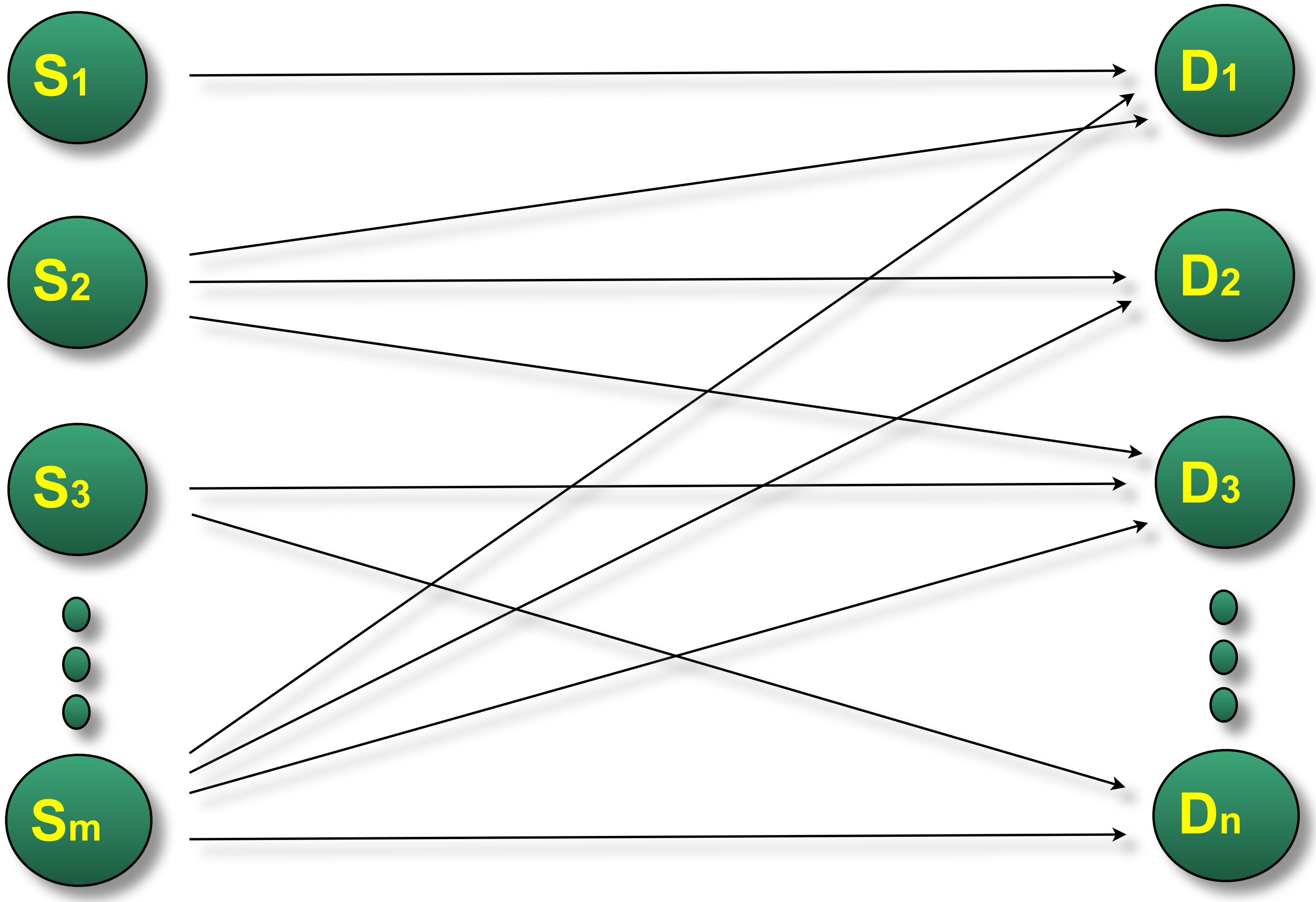
Goal:

- Provide a new computing paradigm where data transfer tasks as well as storage and networking resources are considered first class entities.

Components:

- Data-aware scheduling
- Data-aware storage
- Data-aware workflow management
- Data-aware resource allocation

Data-Aware Scheduling



Transfer *k* files between *m* sources and *n* destinations, optimize by:

- Use physical and semantic metadata in scheduling decisions
- Ordering requests (considering deadlines, file size, storage capacity etc.)
- Choosing the best transfer protocol; translations between protocols
- Tuning protocol transfer parameters (considering current network conditions)
- Throttling - deciding number of concurrent transfers (considering server performance, network capacity, storage space, etc.)

Stork Data Scheduler

- Provides state-of-the art models and algorithms for queuing, scheduling, and optimization of data placement tasks
- Funded by NSF (CAREER, STCI, CiC)
- Futures include:
    - early error detection and classification & recovery
    - data aggregation & connection caching
    - support for multiple transfer protocols
    - dynamic protocol tuning & optimization
    - end-to-end throughput prediction services

- http://www.storkproject.org

Support for Heterogeneity

FTP

HTTP

SCP

GridFTP

UDT

SRB

iRODS

SRM

Support for Heterogeneity

FTP

HTTP

SCP

GridFTP

UDT

SRB

iRODS

SRM

FTP

HTTP

SCP

GridFTP

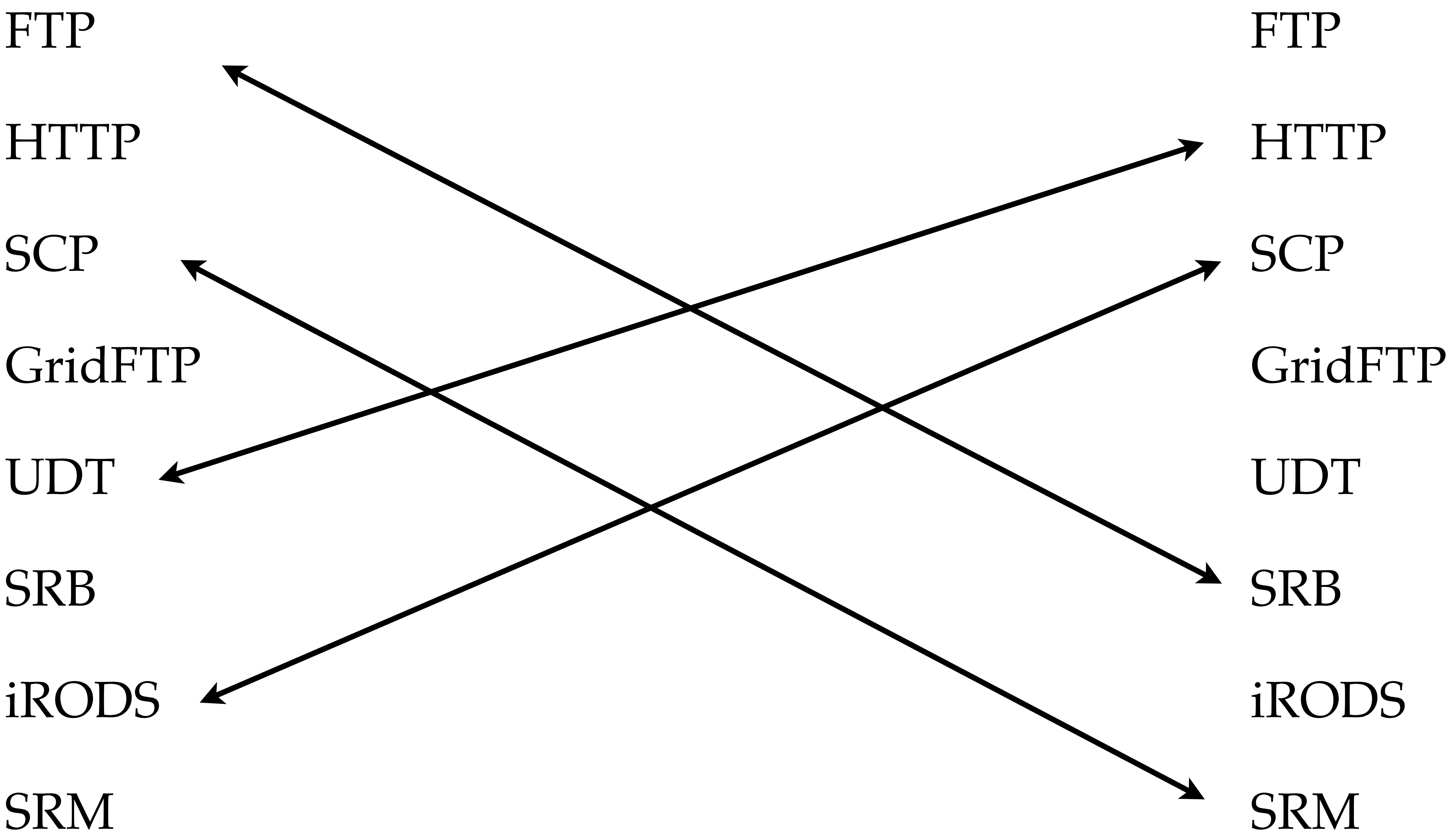
UDT

SRB

iRODS

SRM

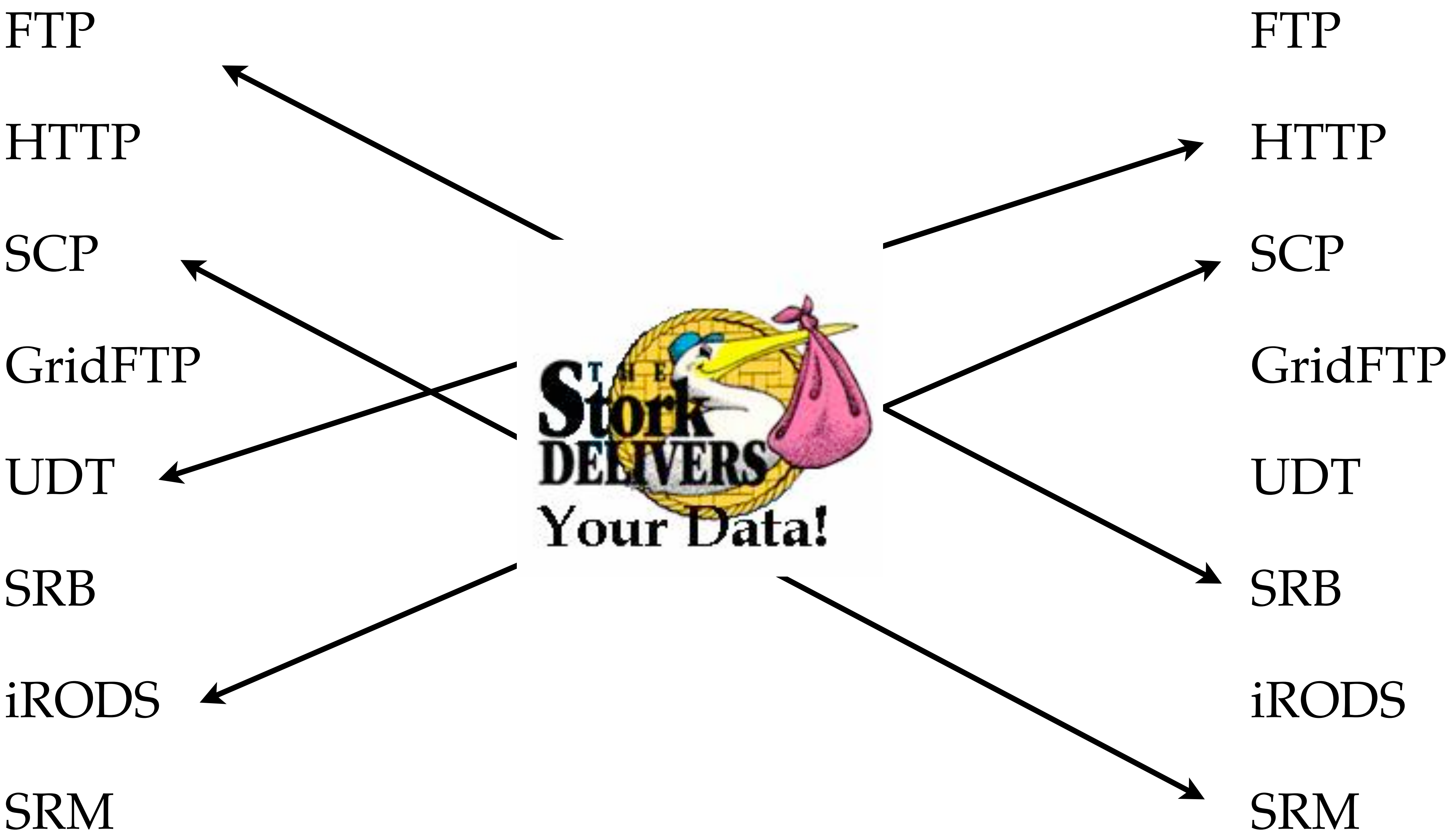
Support for Heterogeneity



FTP
HTTP
SCP
GridFTP
UDT
SRB
iRODS
SRM

FTP
HTTP
SCP
GridFTP
UDT
SRB
iRODS
SRM

Support for Heterogeneity



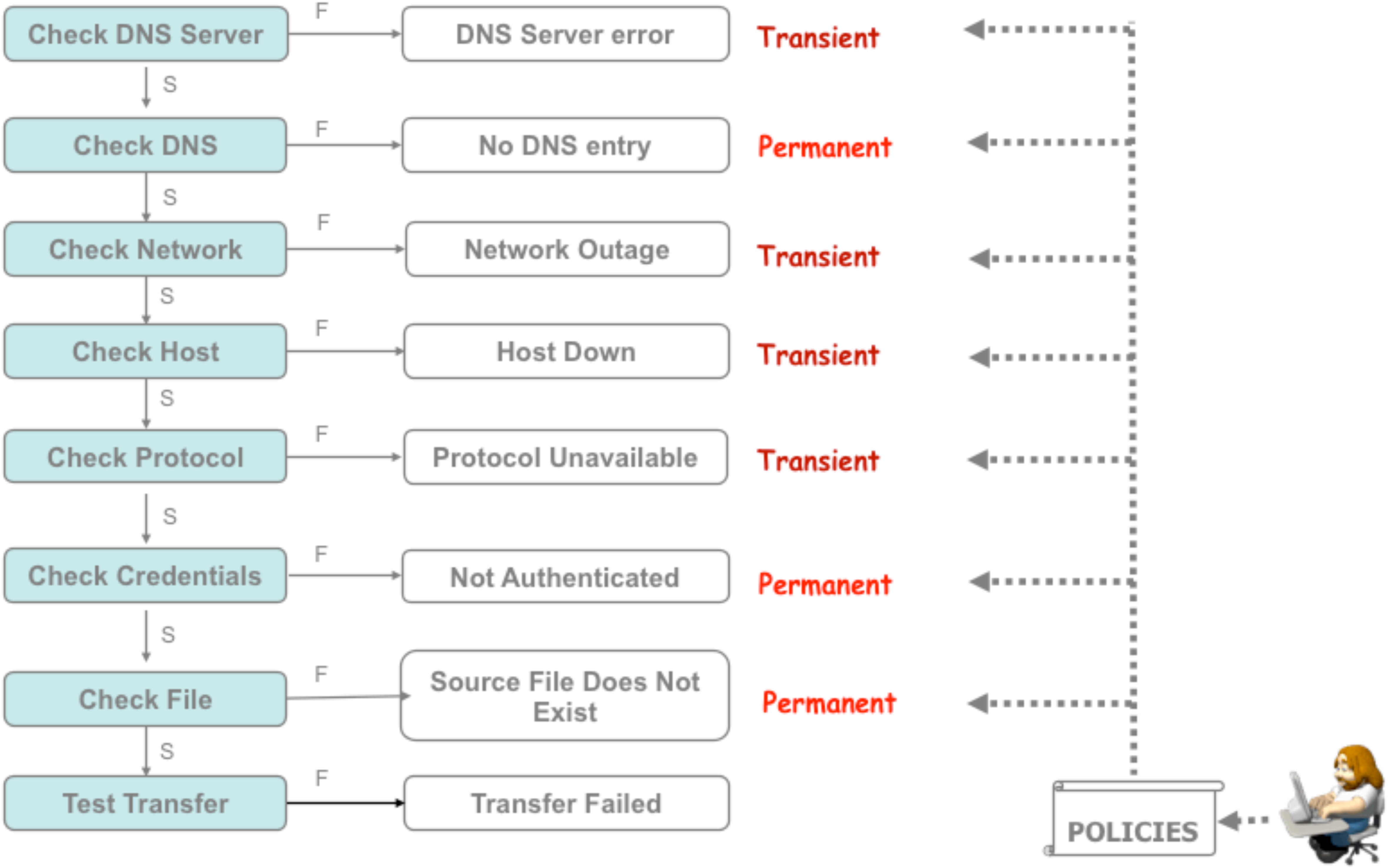
FTP
HTTP
SCP
GridFTP
UDT
SRB
iRODS
SRM

FTP
HTTP
SCP
GridFTP
UDT
SRB
iRODS
SRM

File Transfer Failures

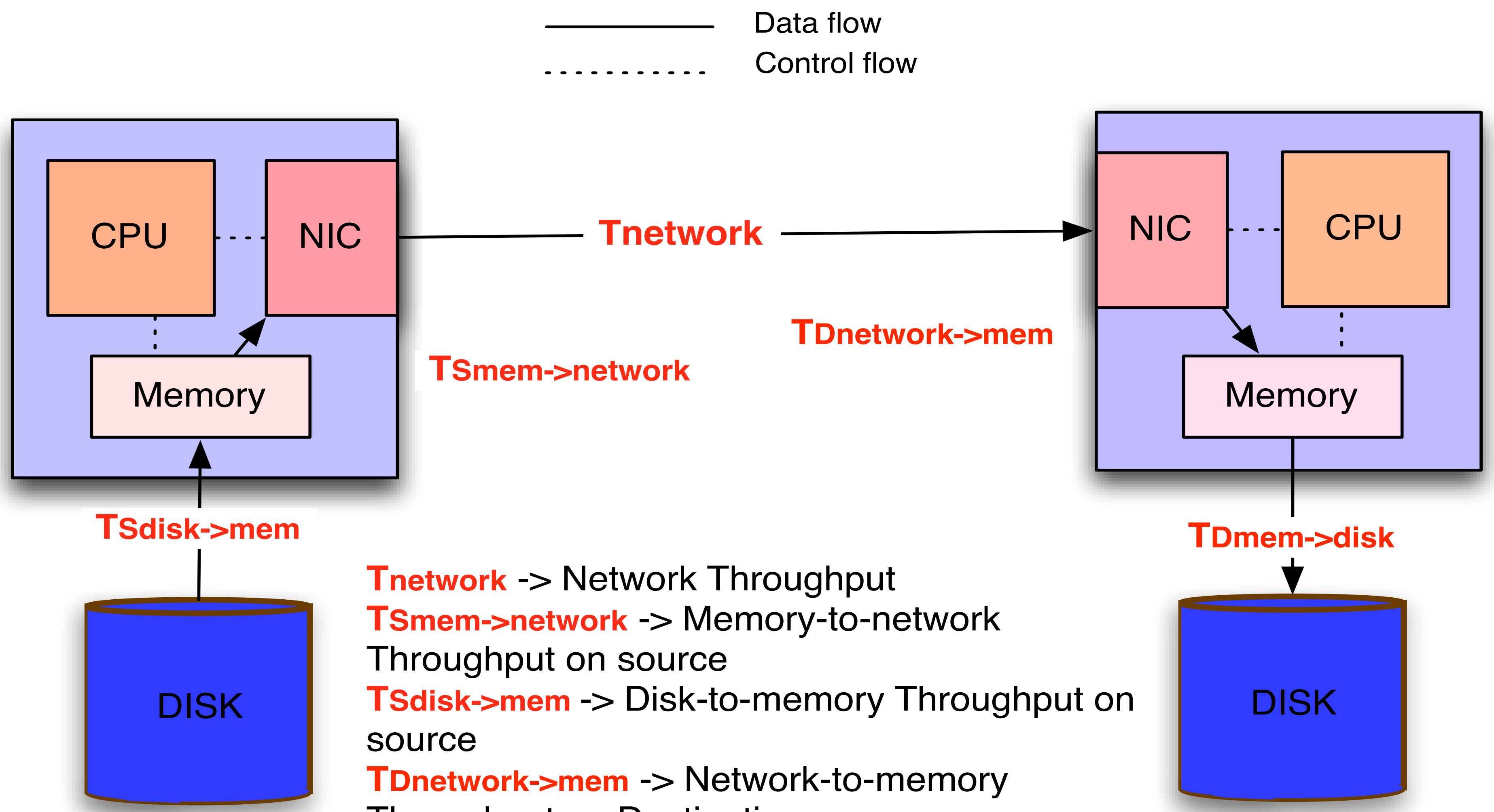
- A data transfer may fail due to different reasons:
  - server down
  - service not running
  - file does not exist
  - authentication failure
  - authorization failure
  - DNS error
  - network error

Error Detection & Clasification



End-to-end Optimization

- In a typical system, the end-to-end throughput depends on the following factors:

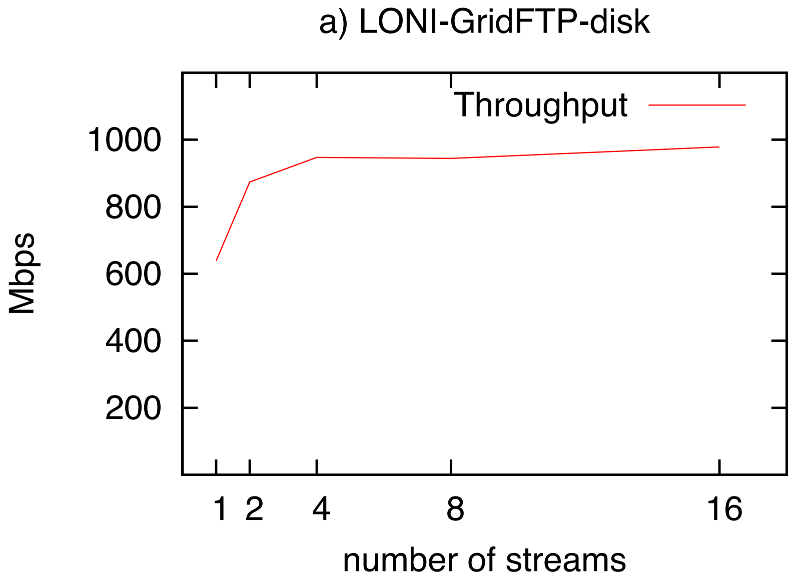


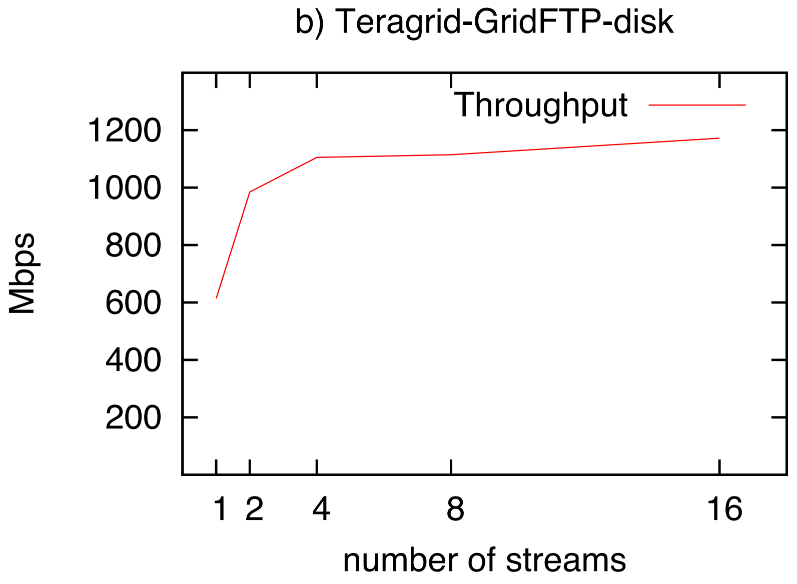
Tnetwork -> Network Throughput
TSmem->network -> Memory-to-network Throughput on source
TSdisk->mem -> Disk-to-memory Throughput on source
TDnetwork->mem -> Network-to-memory Throughput on Destination
TDmem->disk -> Memory-to-disk Throughput on destination

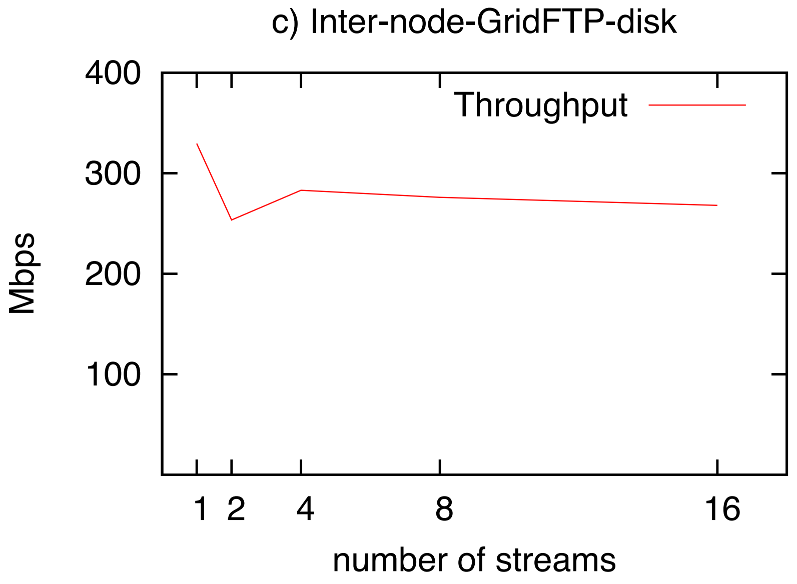
Network Bottleneck

☑ Step1: Effect of Parallel Streams on Disk-to-disk Transfers

- Parallel streams can improve the data throughput but only to a certain extent
- Disk speed presents a major limitation.
- Parallel streams may have an adverse effect if the disk speed upper limit is already reached



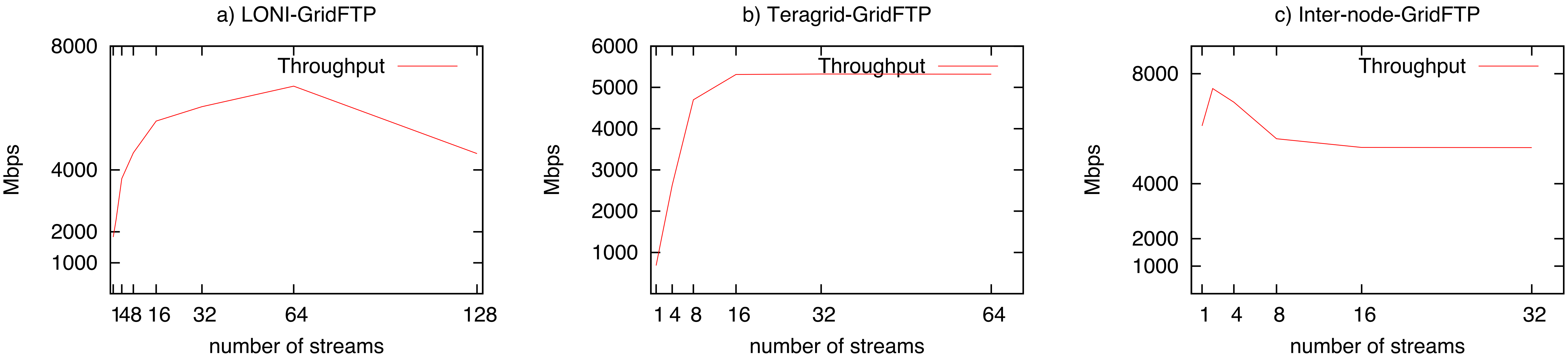




Disk Bottleneck

☑ Step2: Effect of Parallel Streams on Memory-to-memory Transfers and CPU Utilization

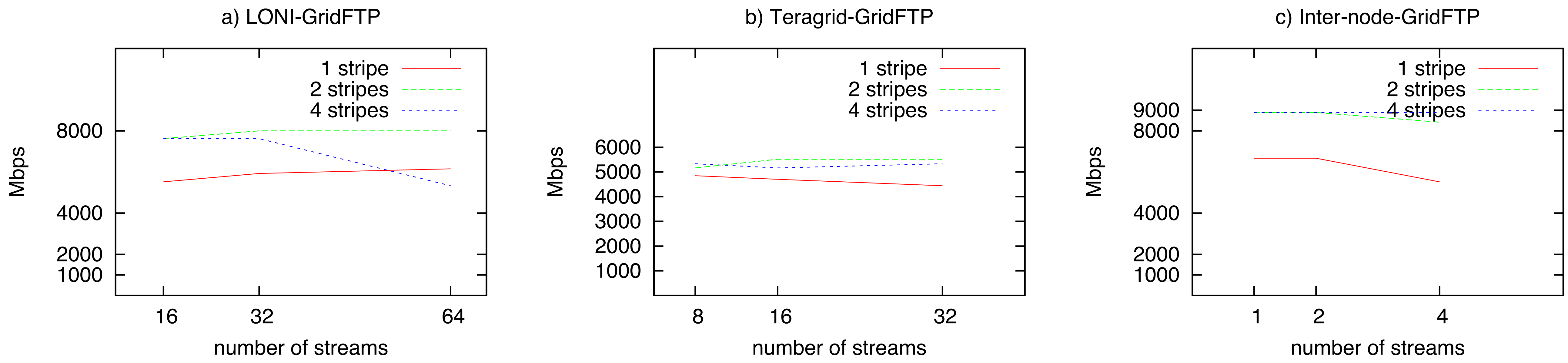
- Once disk bottleneck is eliminated, parallel streams improve the throughput dramatically
- Throughput either becomes stable or falls down after reaching its peak due to network or end-system limitations



CPU Bottleneck

☑ Step3: Effect of Striping and Removal of CPU Bottleneck

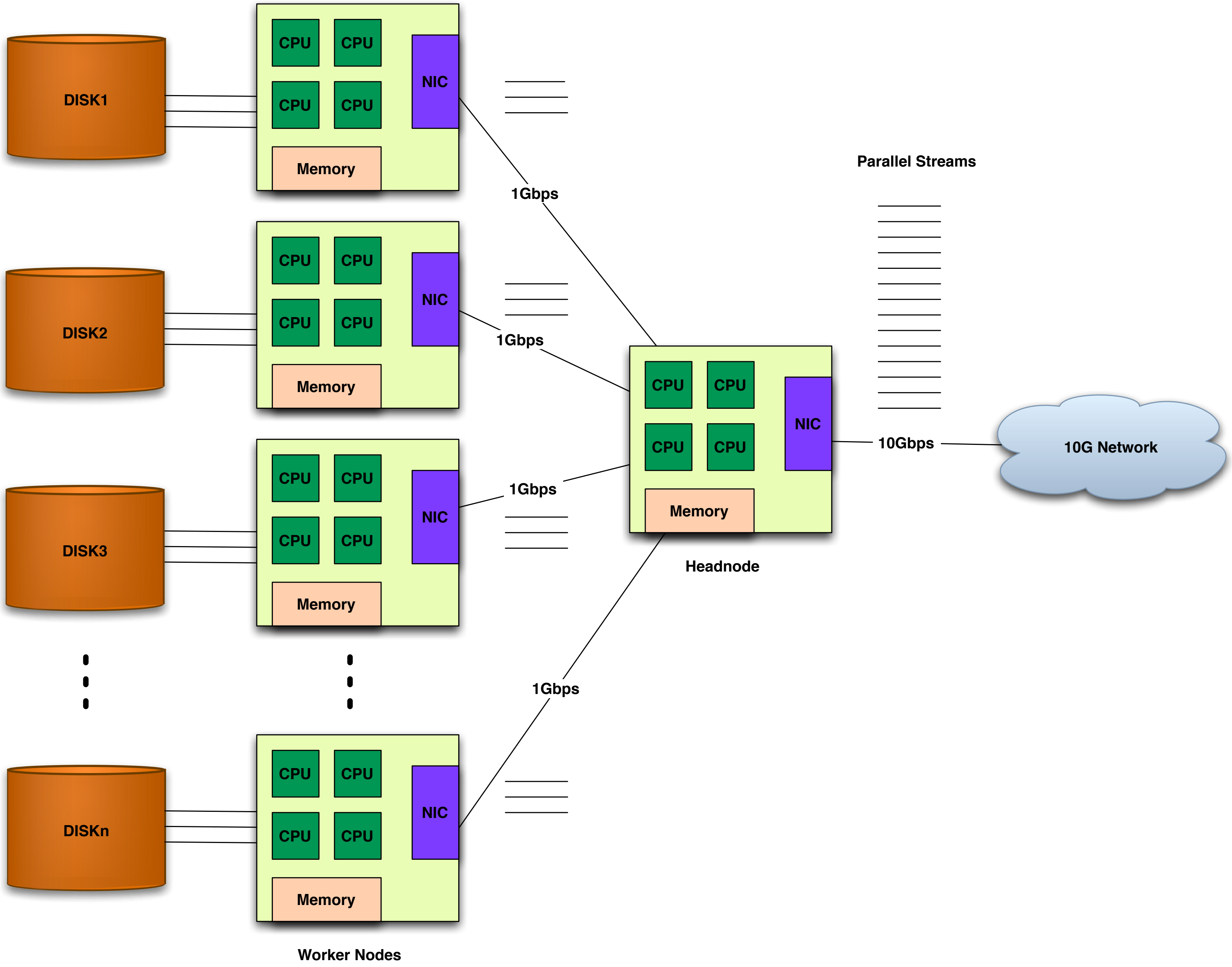
- Striped transfers improves the throughput dramatically
- Network card limit is reached for inter-node transfers (9Gbps)



End-to-end Data Flow Parallelism

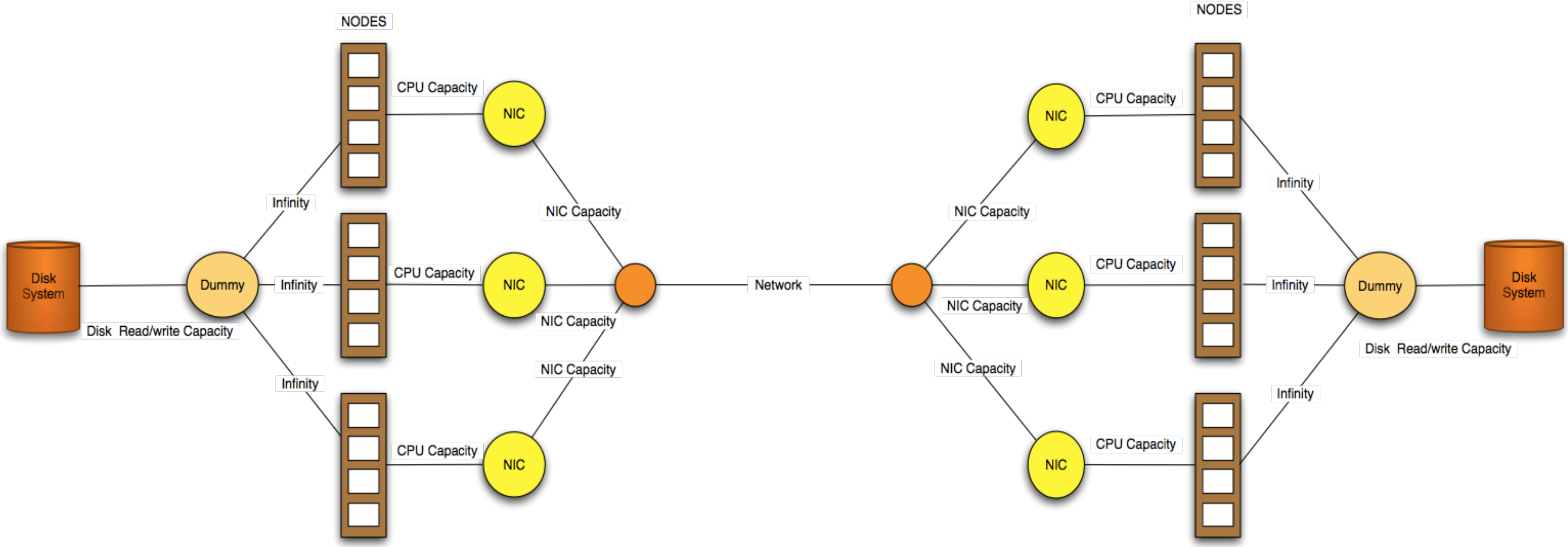
Parameters to be optimized:

- # of streams
- # of disk stripes
- # of CPUs/nodes



Disks and CPUs can be hosted services in a Cloud and dynamically provisioned.

End-to-end Flow Model



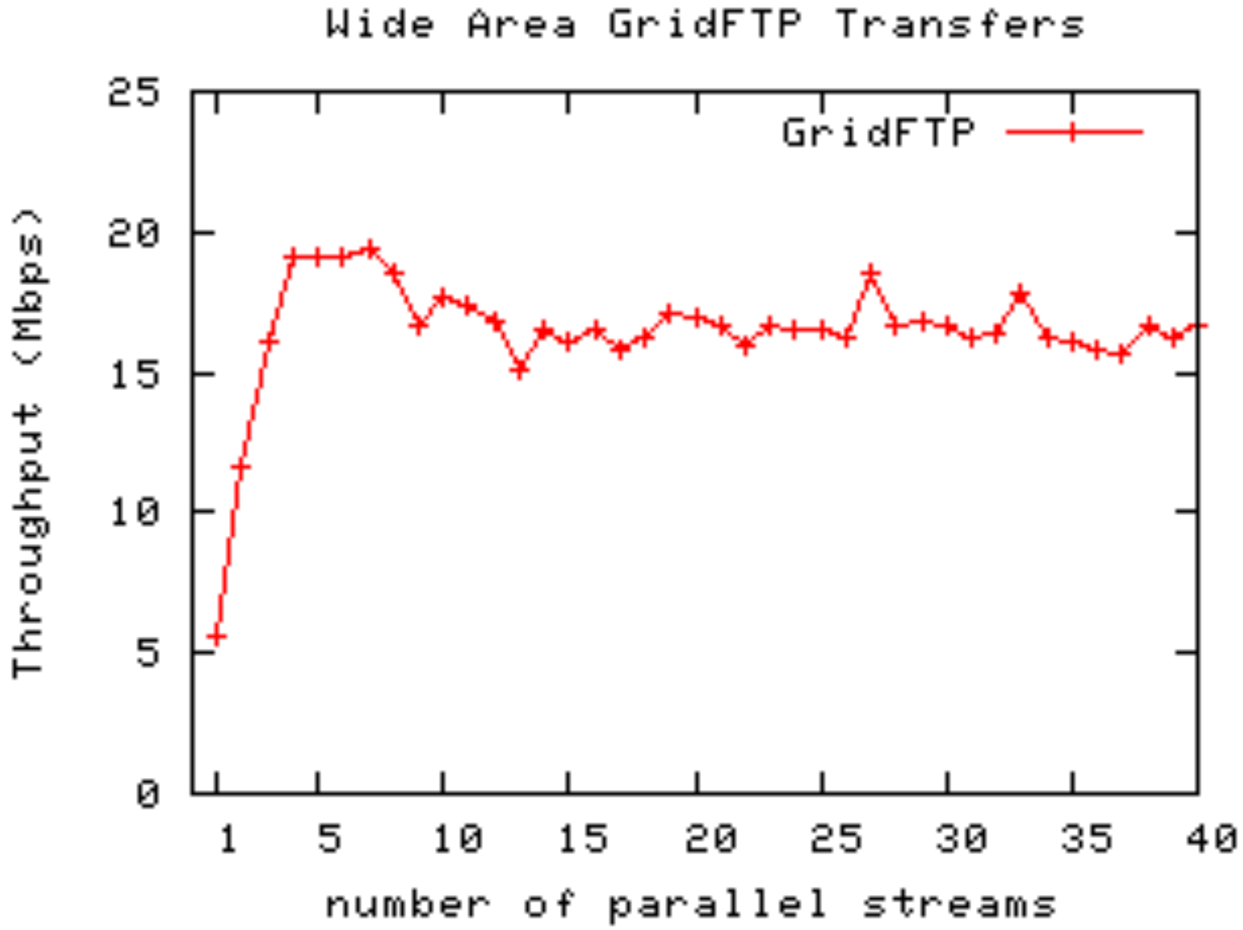
- CPU nodes are considered as nodes of a maximum flow problem
- Memory-to-memory transfers are simulated with dummy source and sink nodes
- The capacities of disk and network is found by applying parallel stream model by taking into consideration of resource capacities (NIC & CPU)

Parallel Stream Optimization

For a single stream, theoretical calculation of throughput (using Mathis Equation):

$$Th <= \frac{MSS}{RTT} \frac{c}{\sqrt{p}}$$

For n streams?

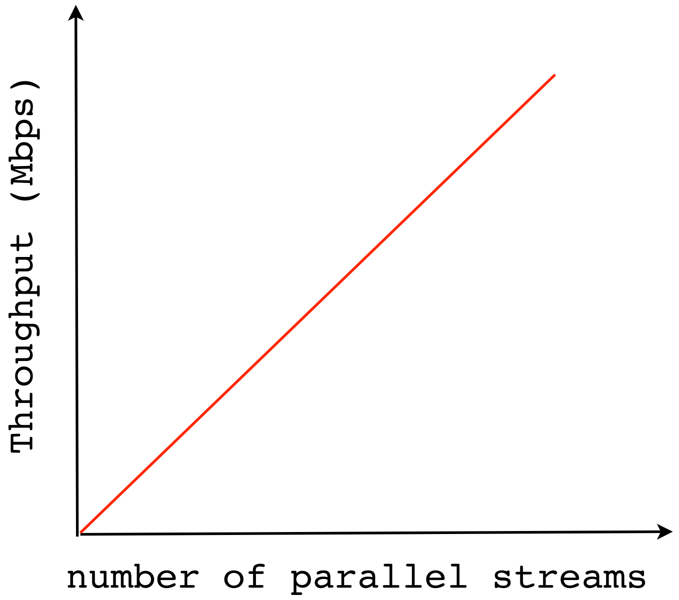


Previous Models

Hacker et al (2002)

An application opening *n* streams gains as much throughput as the total of *n* individual streams can get:

$$Th_n <= \frac{MSS \times c}{RTT}\left(\frac{n}{\sqrt{p}}\right)$$

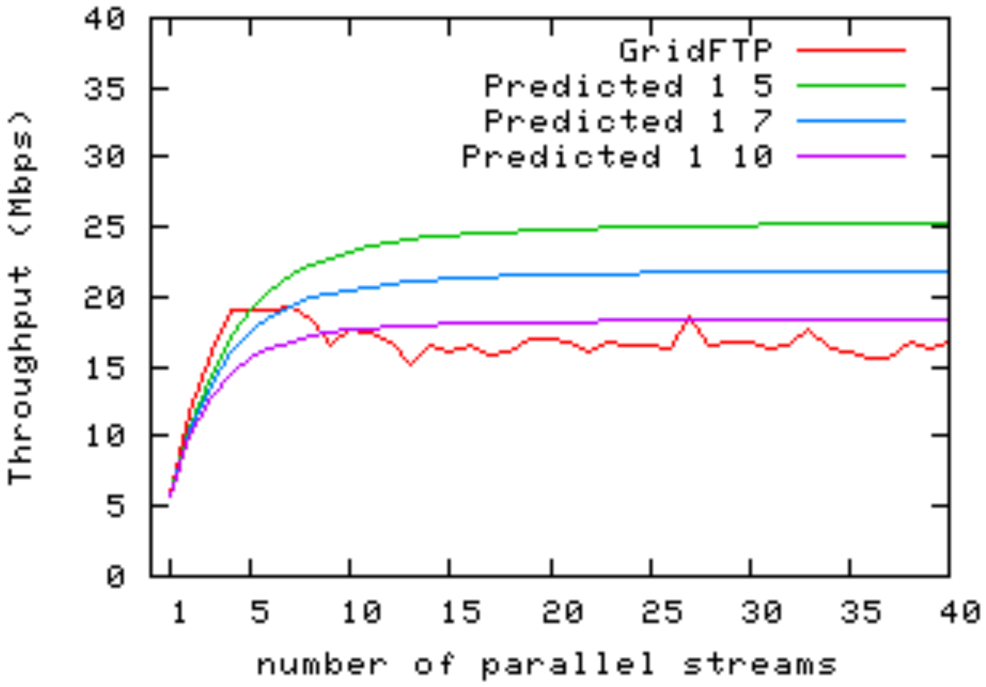


Dinda et al (2005)

A relation is established between *RTT*, *p* and the number of streams *n:*

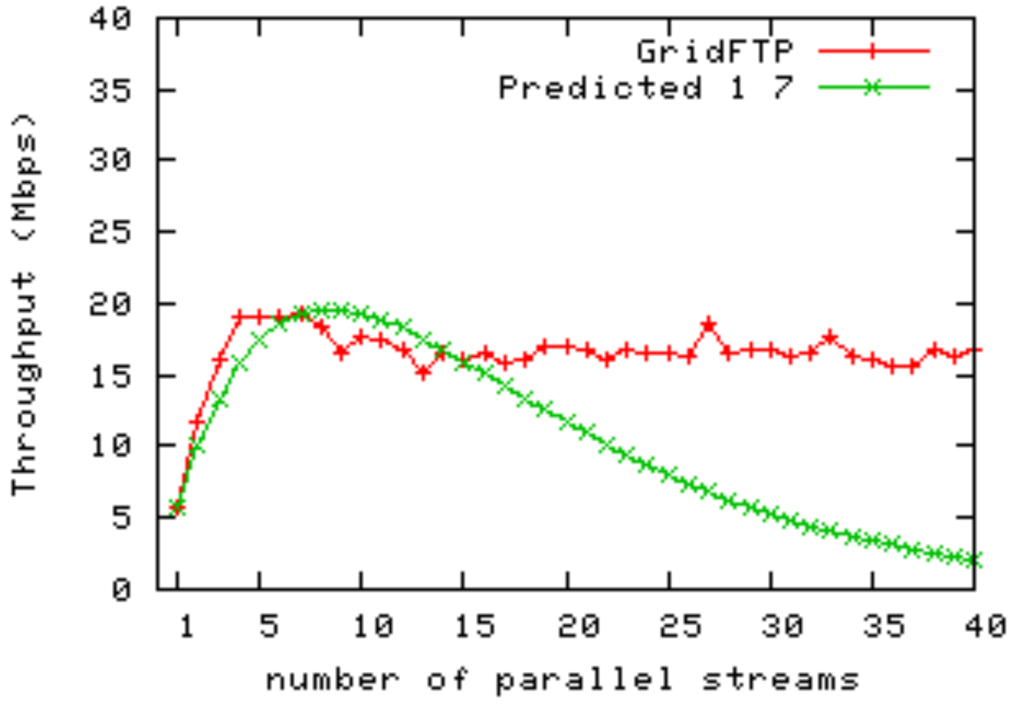
$$p'_n = p_n \frac{RTT_n^2}{c^2 MSS^2} = a'n^2 + b'$$

$$Th_n = \frac{n}{\sqrt{p'_n}} = \frac{n}{\sqrt{a'n^2 + b'}}$$

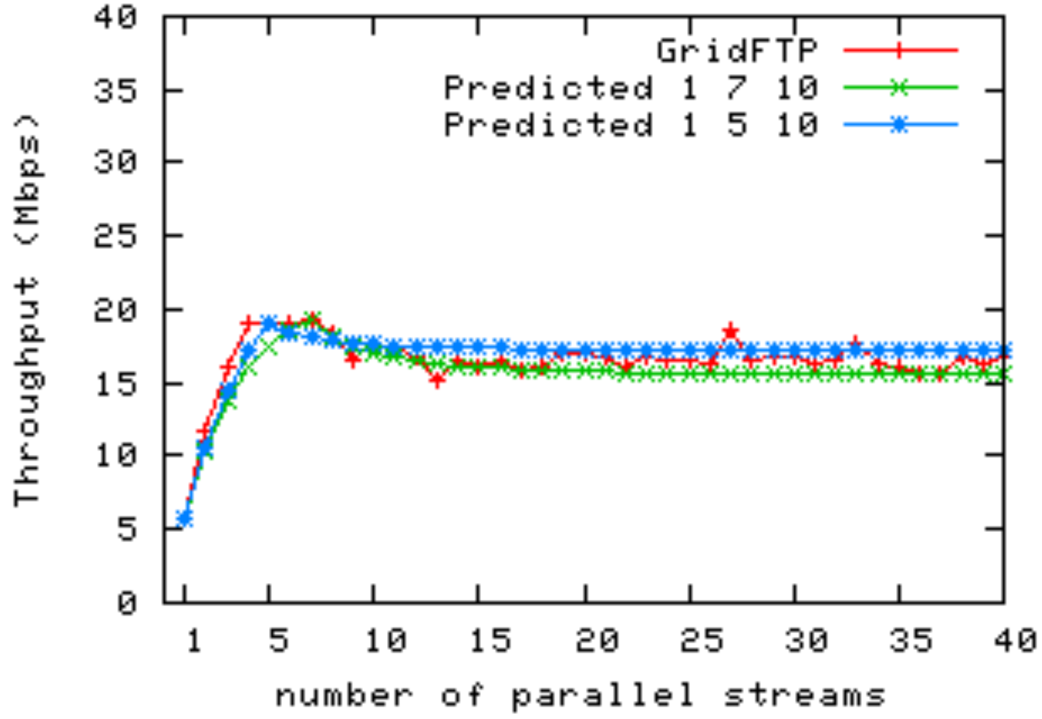


Kosar et al Models

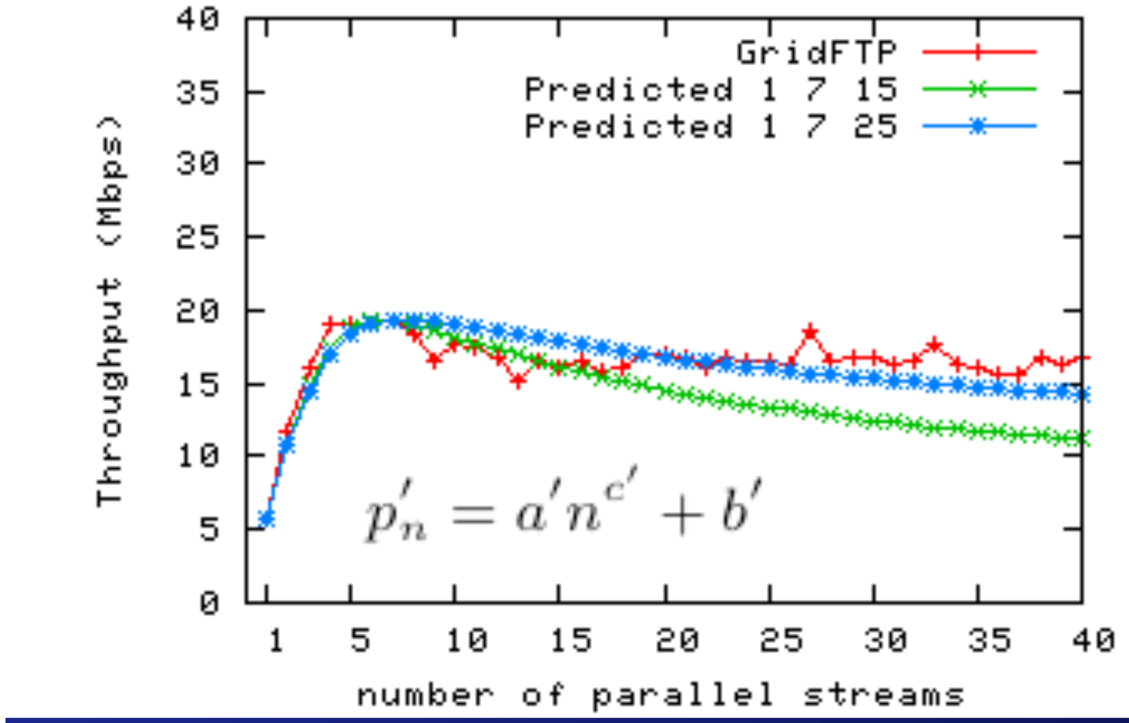
Exponential Packet Loss



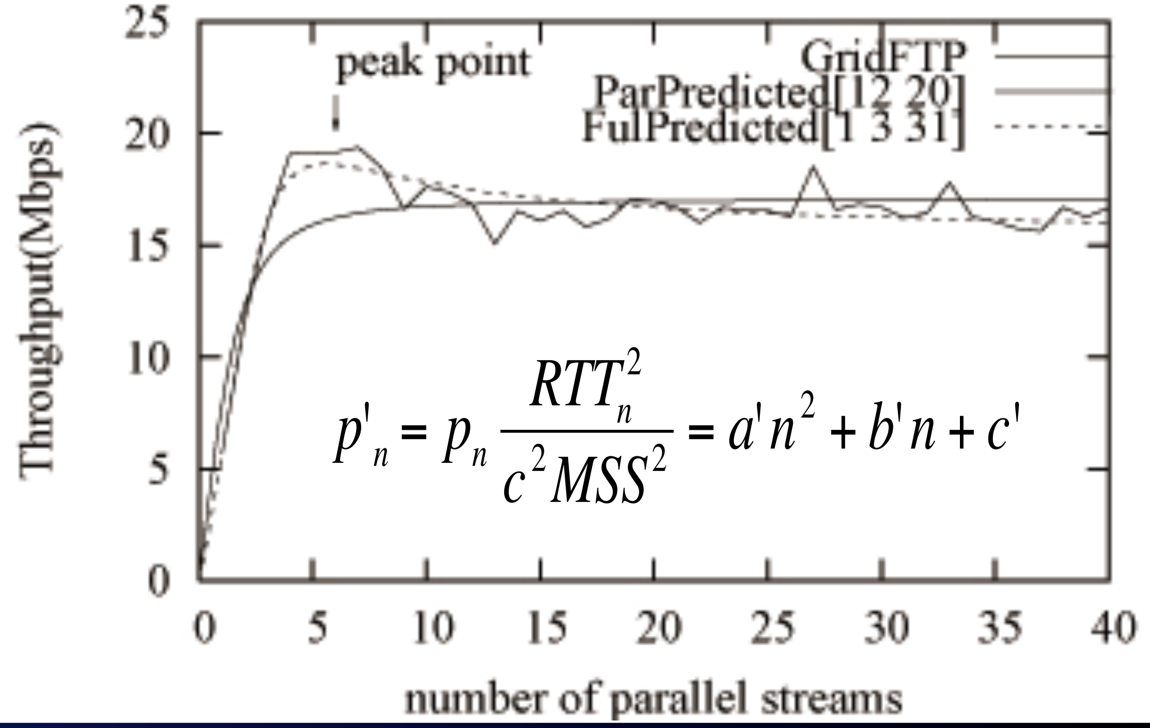
Break Function Modeling



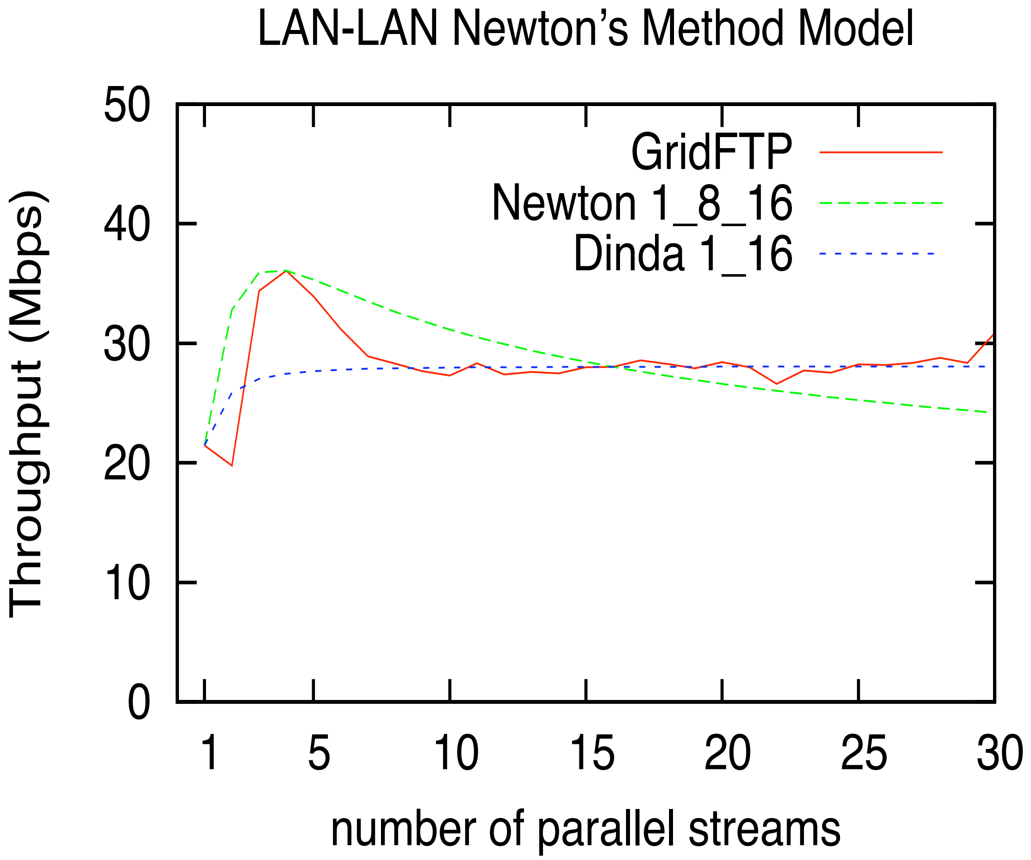
Modeling Based on Newton's Iteration

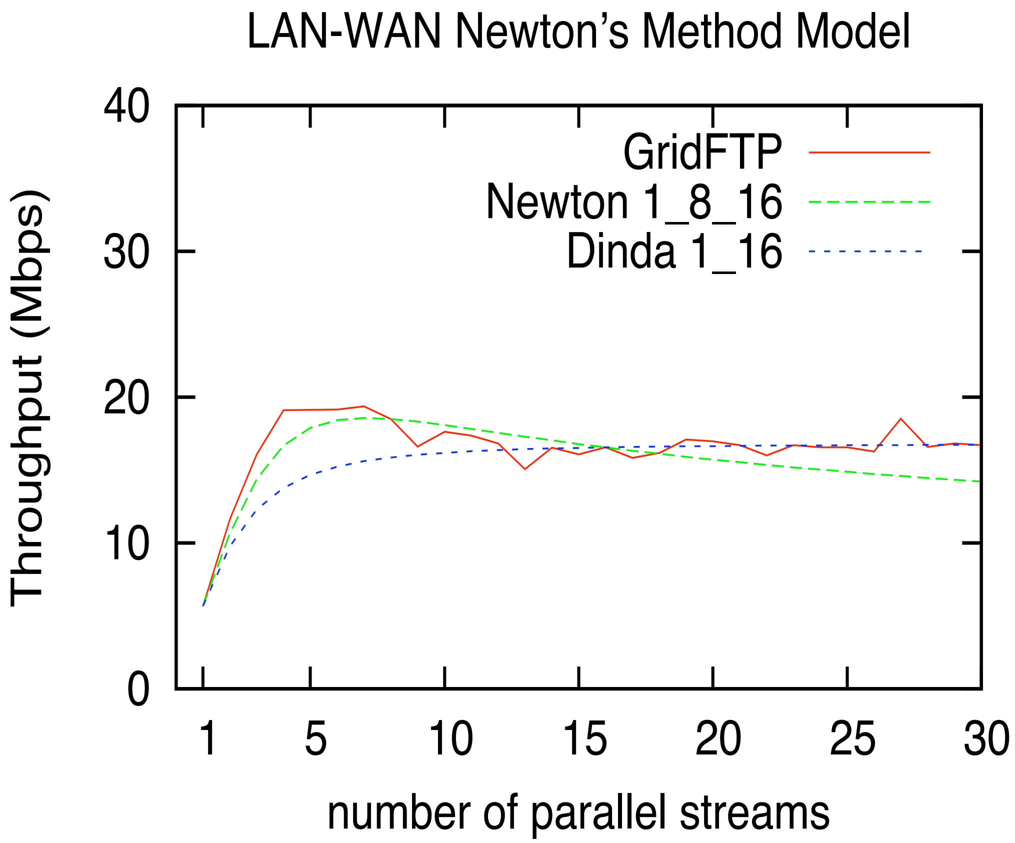


Modeling Based on Full Second Order

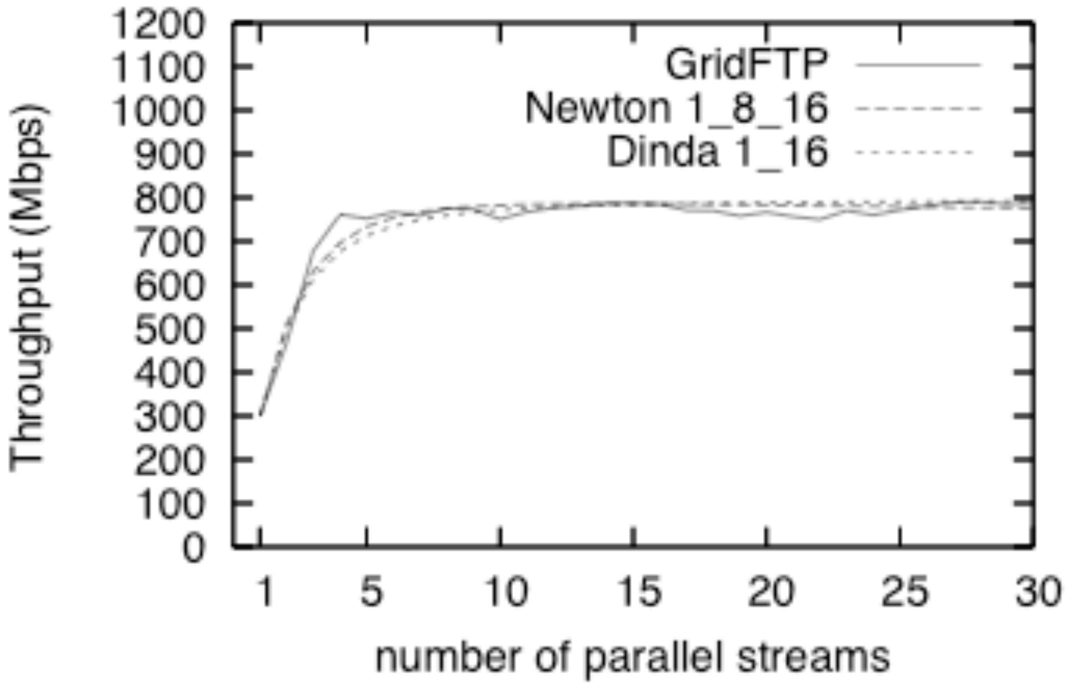


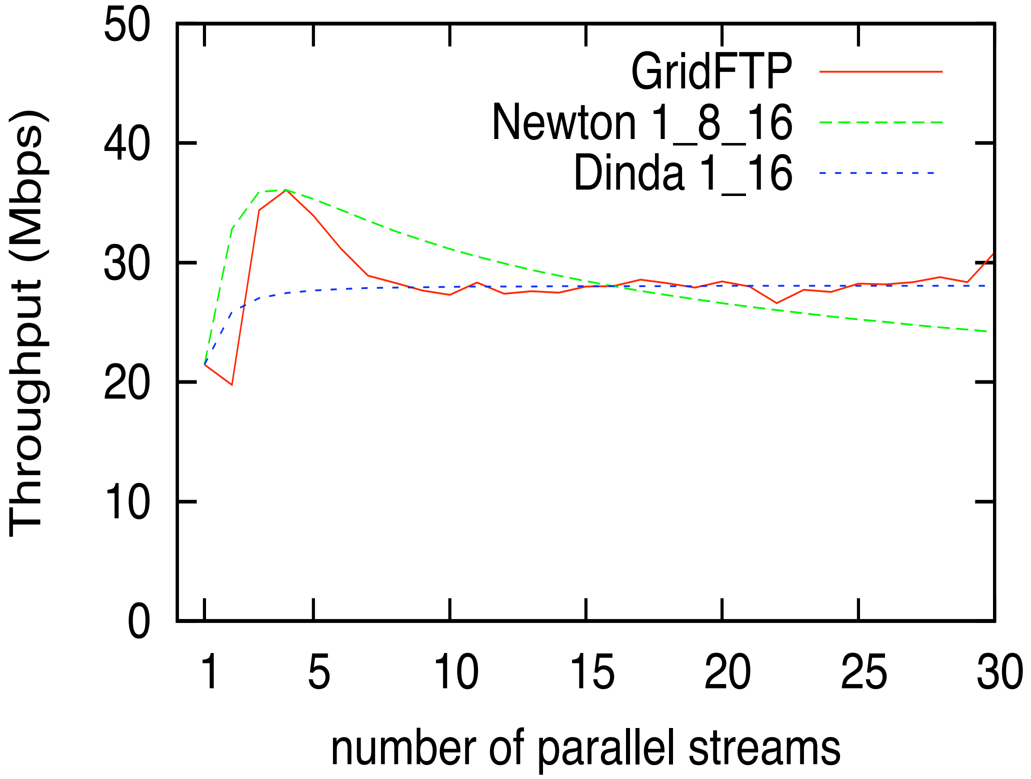
Estimations by the Model

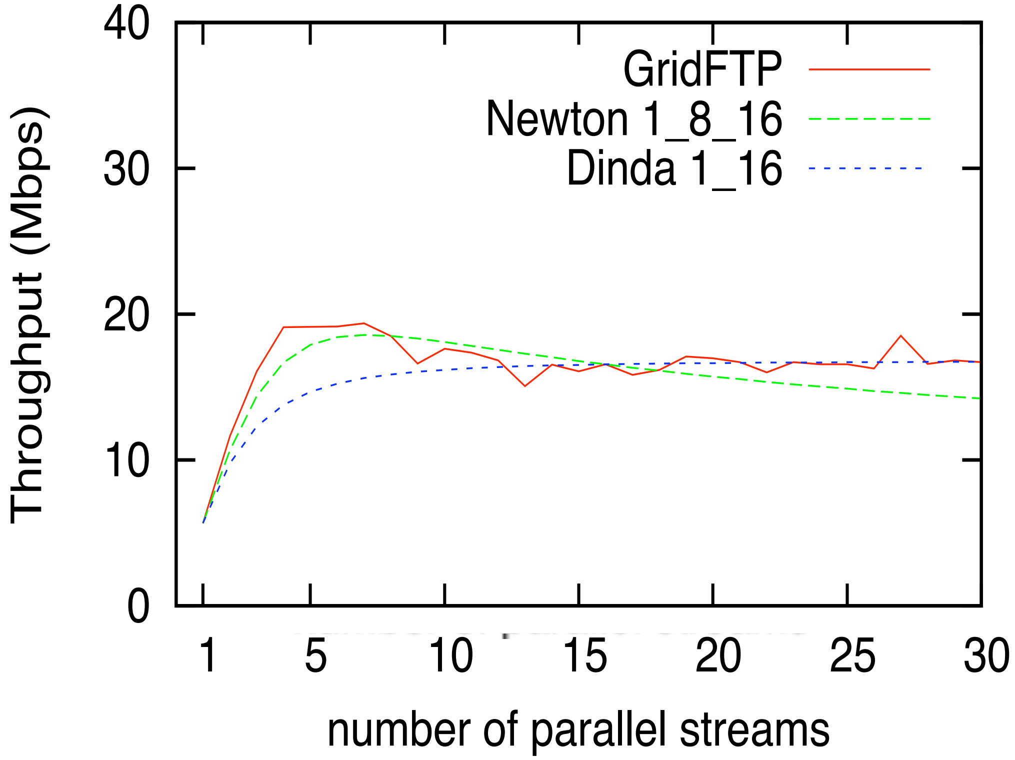




Estimations by the Model

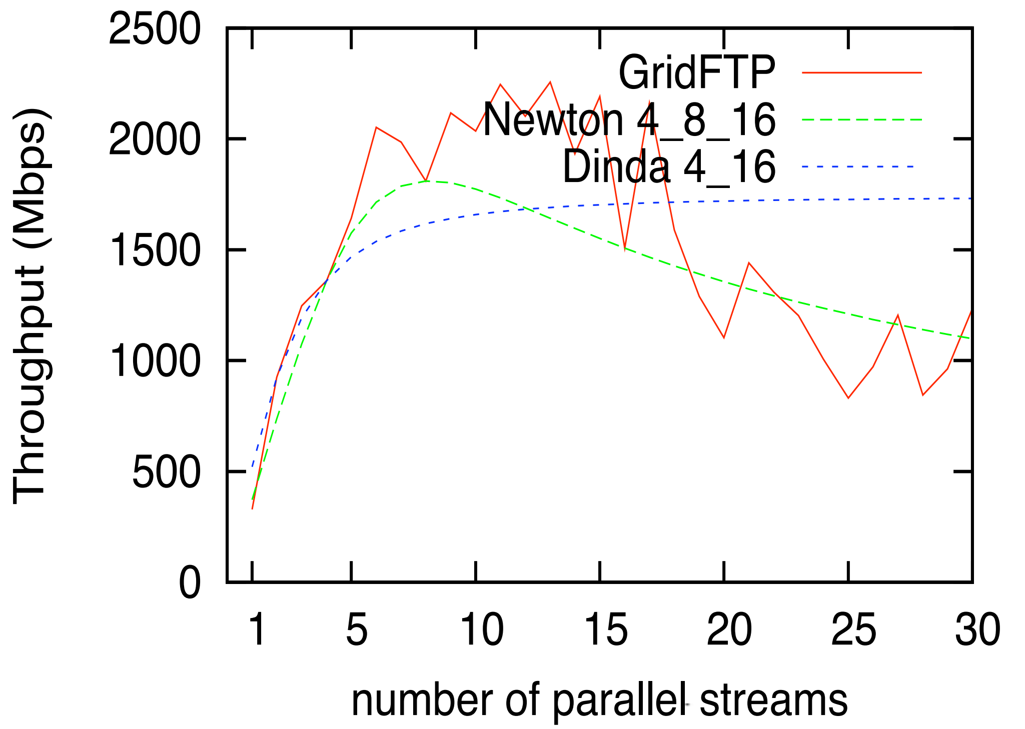

LONI-LONI Newton's Method Model


LAN-LAN Newton's Method Model

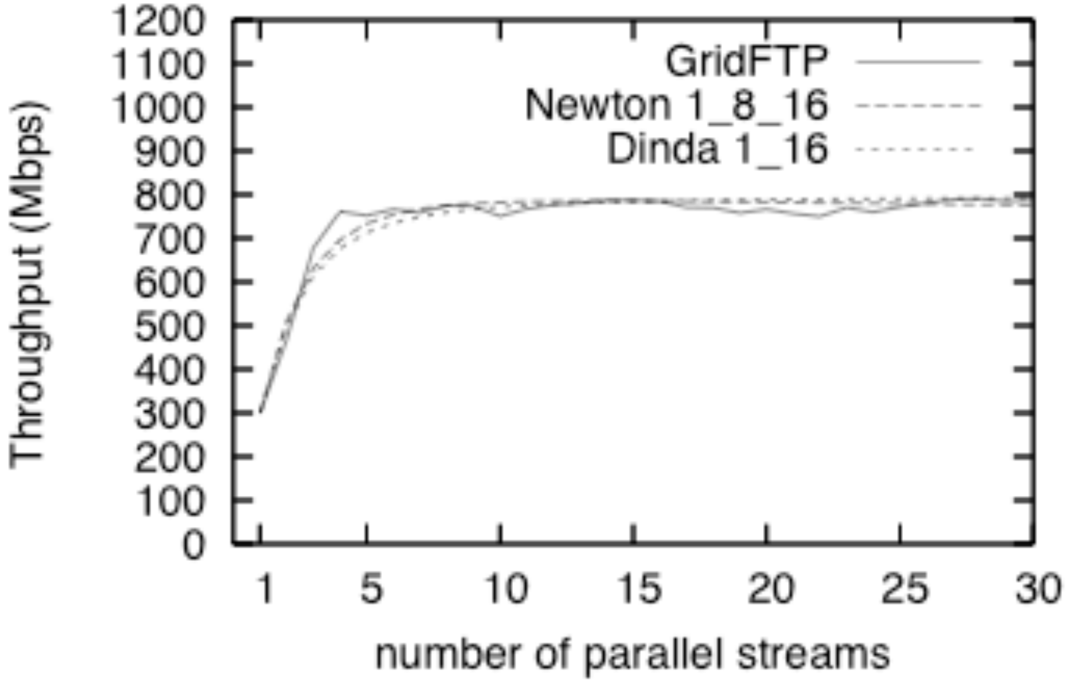

LAN-WAN Newton's Method Model

Estimations by the Model

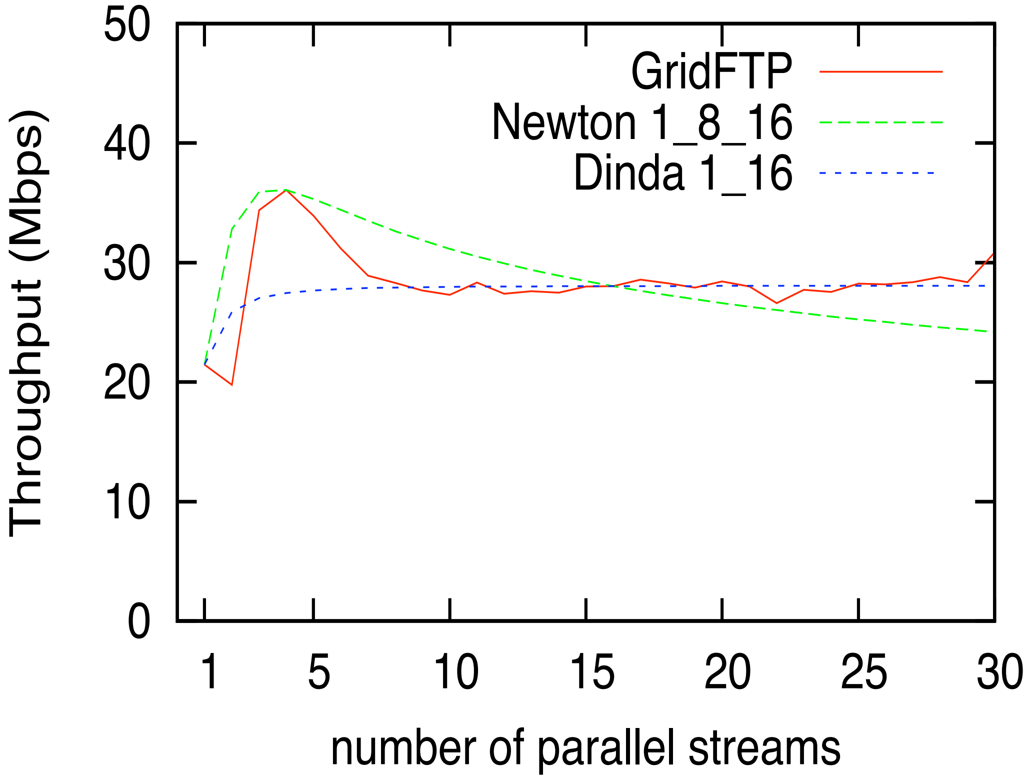
LONI-LONI-10Gbps Newton's Method Model



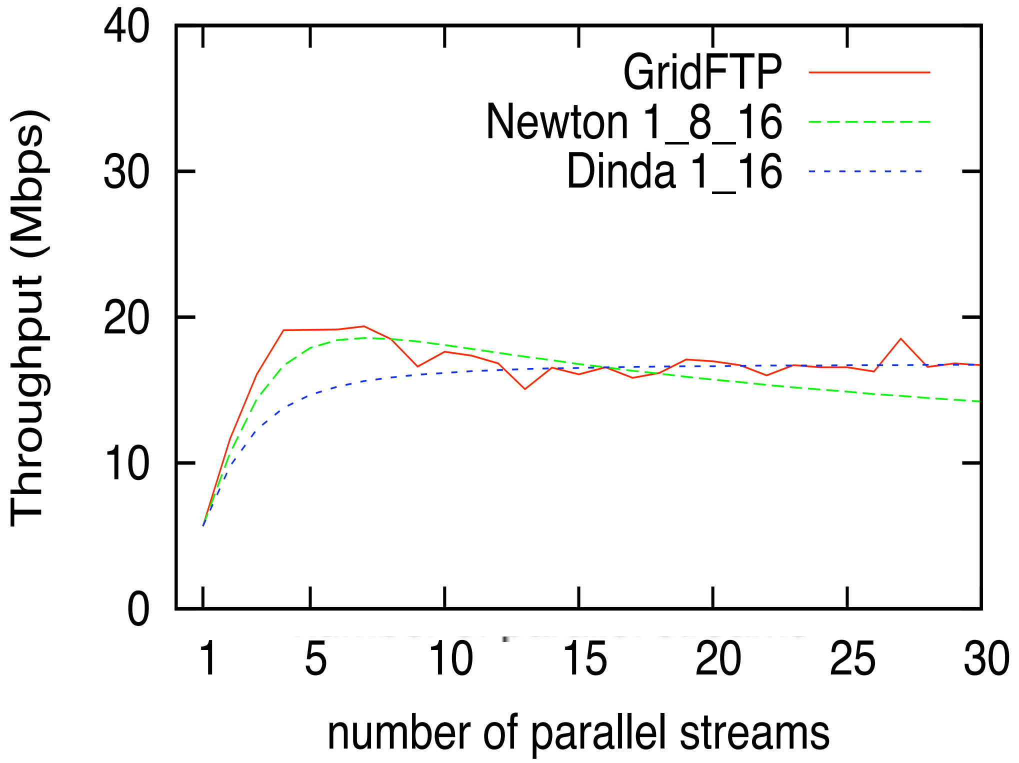
LONI-LONI Newton's Method Model



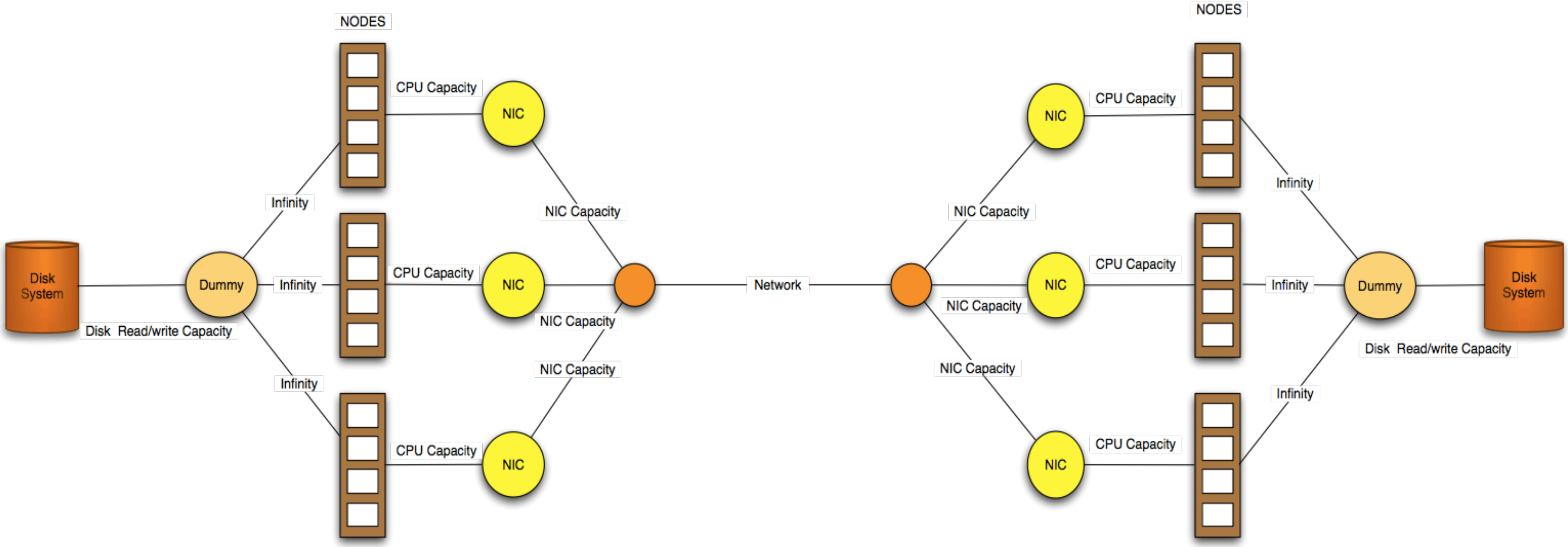
LAN-LAN Newton's Method Model



LAN-WAN Newton's Method Model



End-to-end Flow Model



- Convert the end-system and network capacities into a flow problem
- Goal: Provide the user with parallelism parameters
- Number of streams per stripe (N_{si})
- Number of stripes per node (S_x)
- Number of nodes (N_n)

Assumptions:

- Parameters not given and found by the model:
  - Available network capacity
  - Available disk system capacity
- Parameters given
  - CPU capacity (100% assuming they are idle at the beginning of the transfer)
  - NIC capacity
  - Number of available nodes

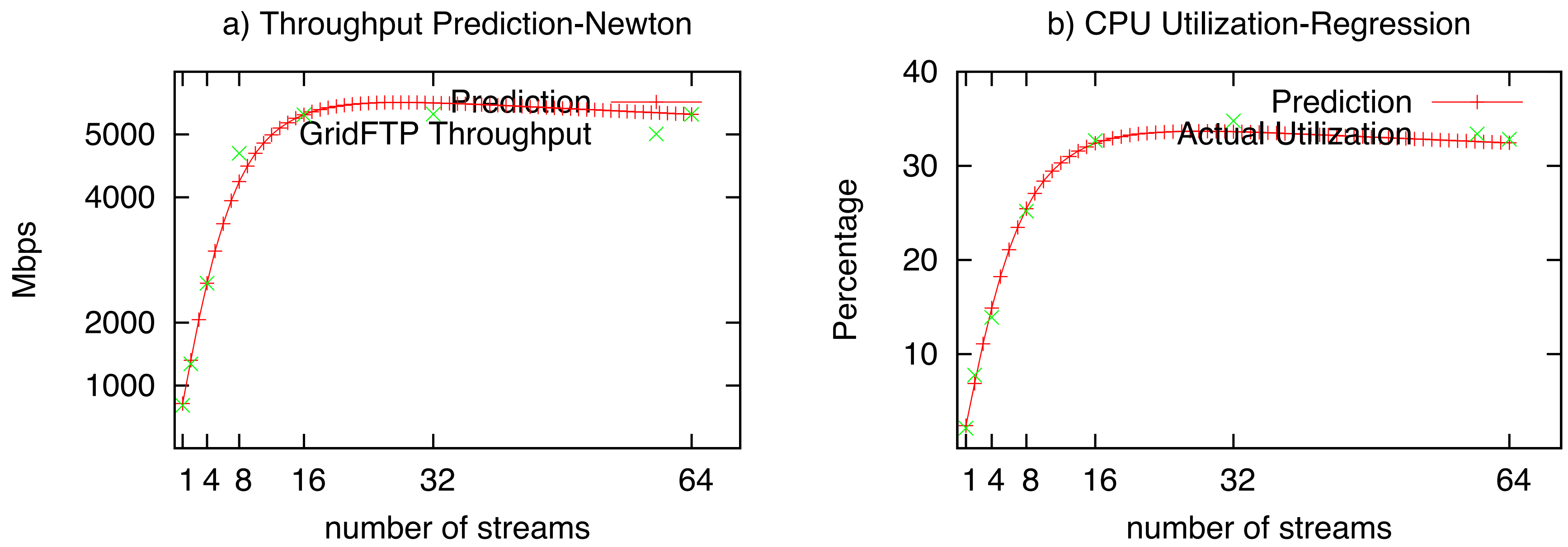
Modeling CPU utilization

- There is a high positive correlation between the throughput of parallel streams and CPU utilization
- The linear relation between CPU utilization and Throughput is presented as -->
- The variables could be calculated by using method of least squares and the sampling throughput & utilization values from the parallel streams optimization model -->

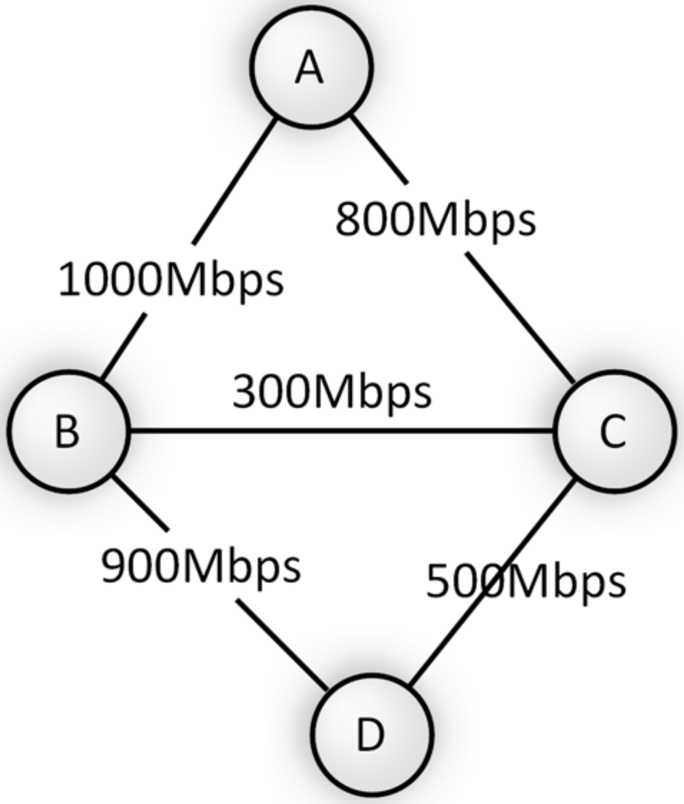
$$Ucpu = a + b \times Th$$

$$a = Mean(U) - b \times Mean(Th)$$

$$b = \frac{\sum ThU - (\sum U \sum Th / size)}{\sum Th^2 - (\sum Th)^2 / size}$$

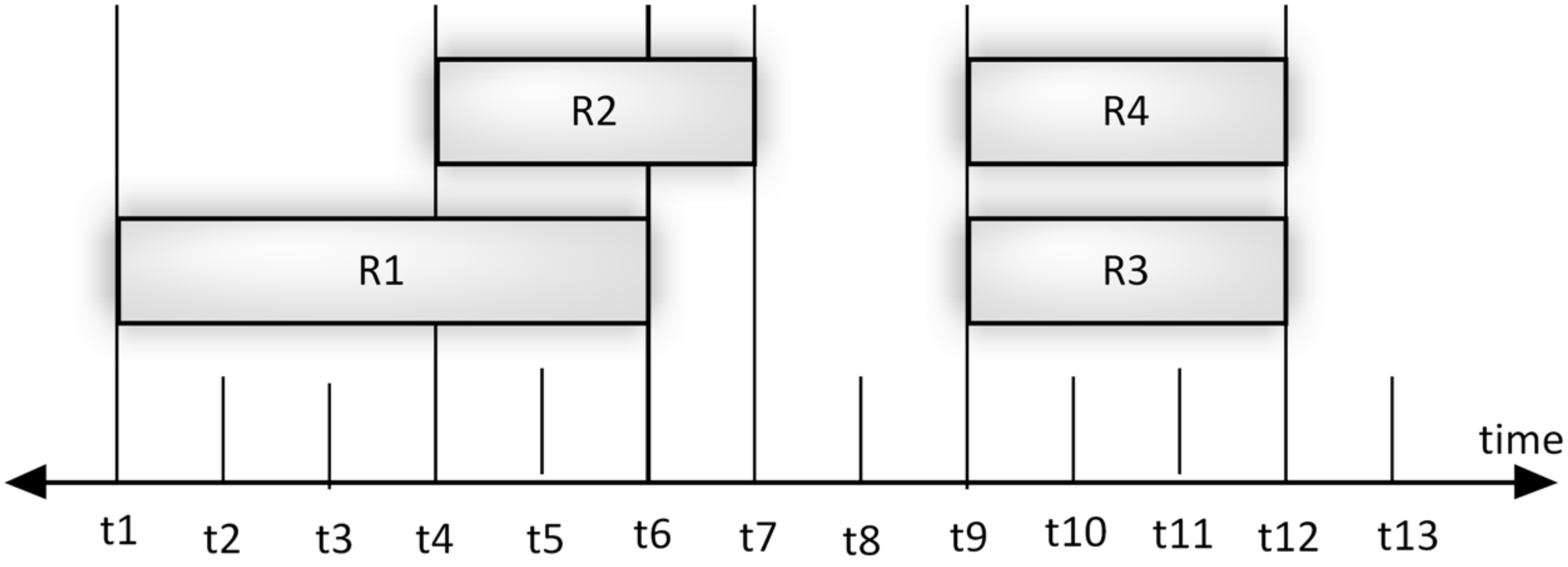


Advanced Reservations



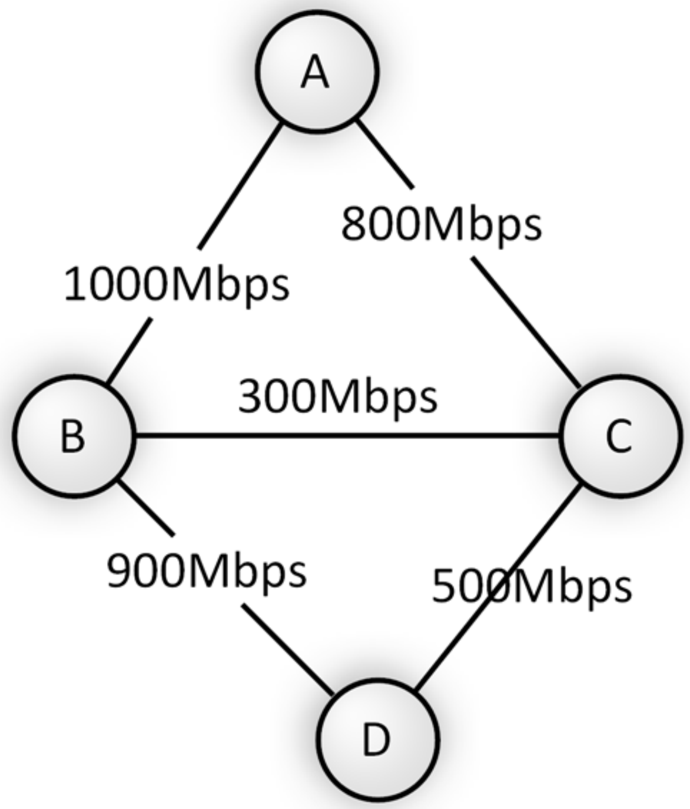
Network Graph with edges
showing maximum bandwidth

R1: (time t1, t6) A -> B ->D (900Mbps)
R2: (time t4, t7) A -> C -> D (400Mbps)
R3: (time t9, t12) A -> B -> D (700Mpbs)
R4: (time t9, t12) A -> C -> D (500Mpbs)



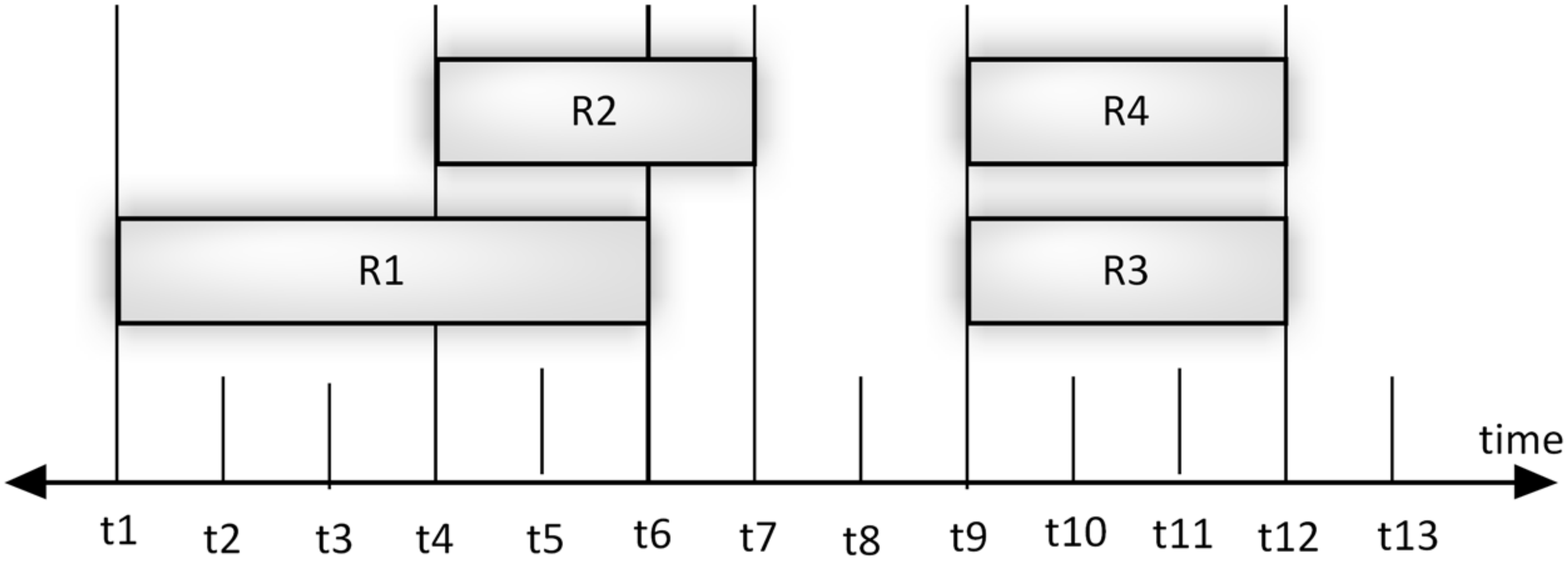
Reservations

Advanced Reservations

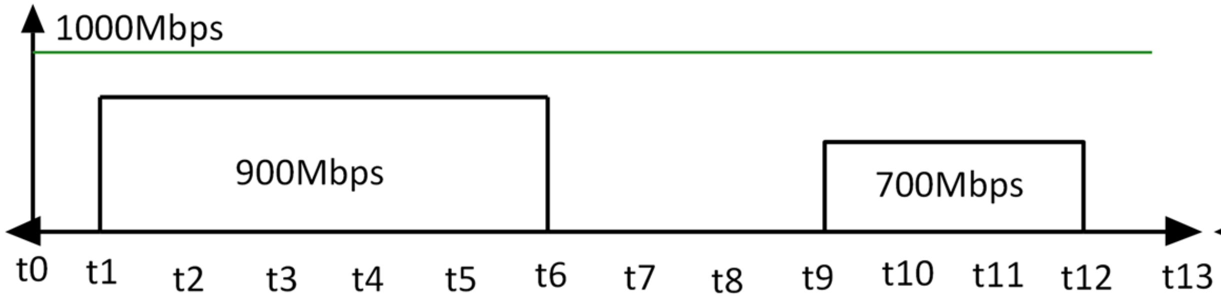


Network Graph with edges showing maximum bandwidth

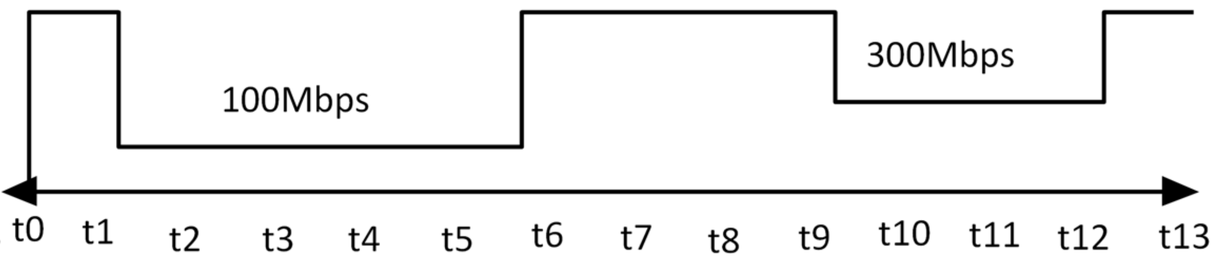
R1: (time t1, t6) A -> B ->D (900Mbps)
R2: (time t4, t7) A -> C -> D (400Mbps)
R3: (time t9, t12) A -> B -> D (700Mpbs)
R4: (time t9, t12) A -> C -> D (500Mpbs)



Reservations

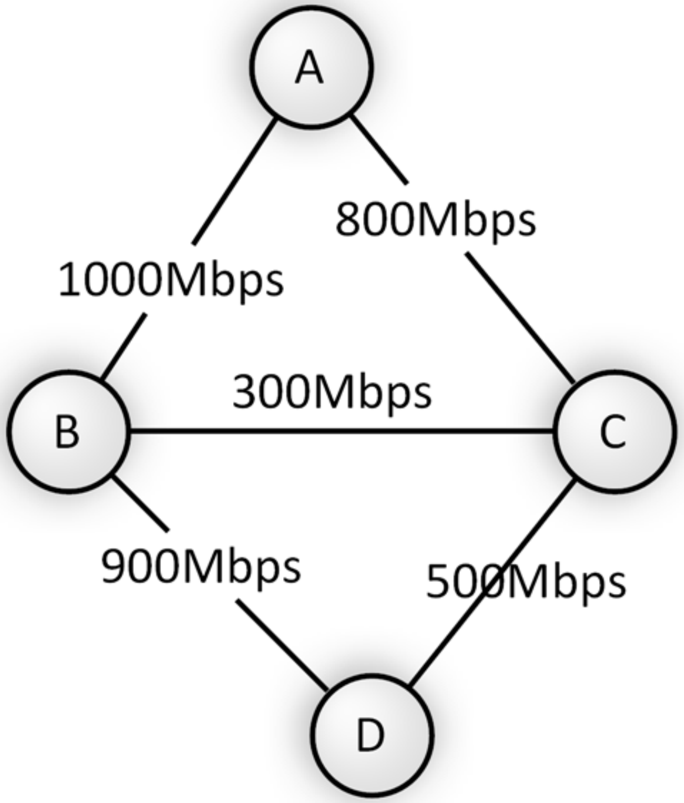


allocated



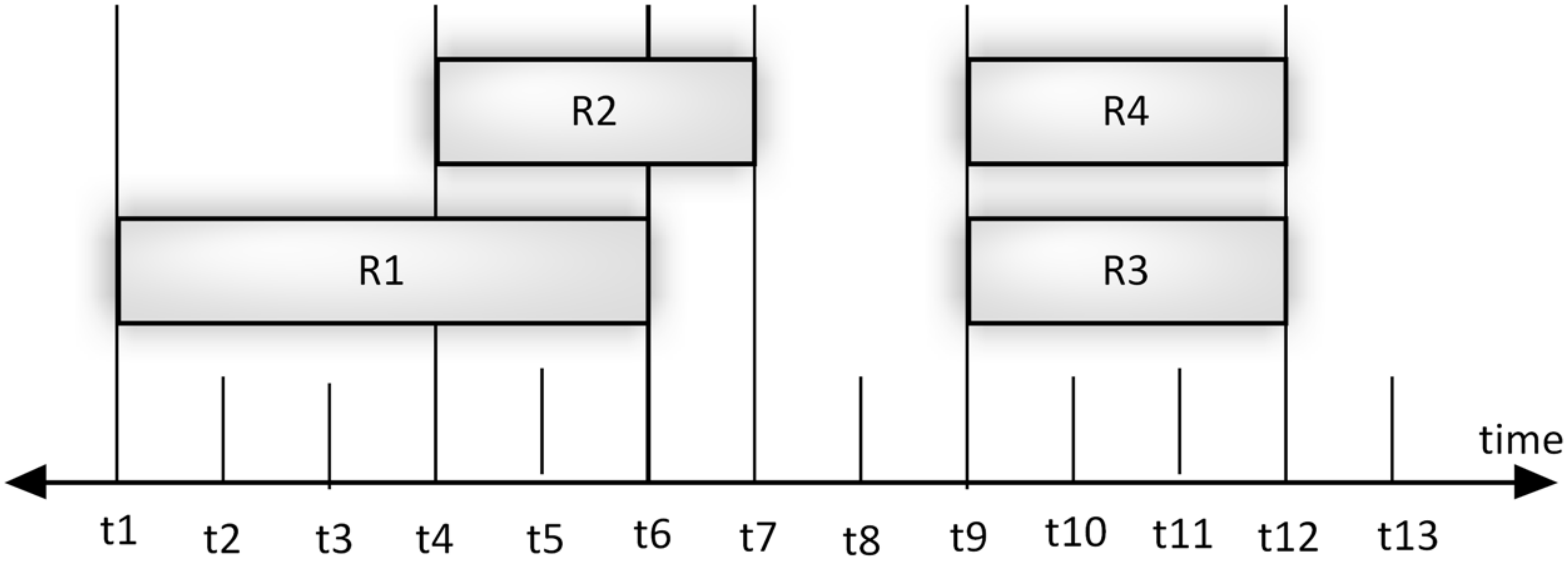
available

Advanced Reservations

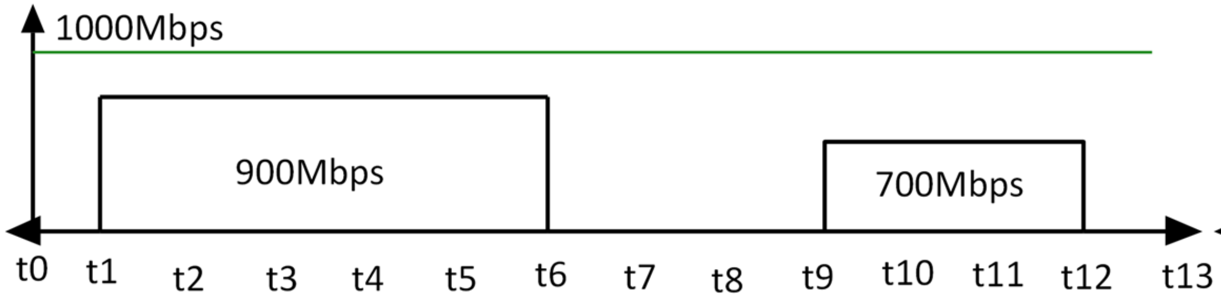


Network Graph with edges showing maximum bandwidth

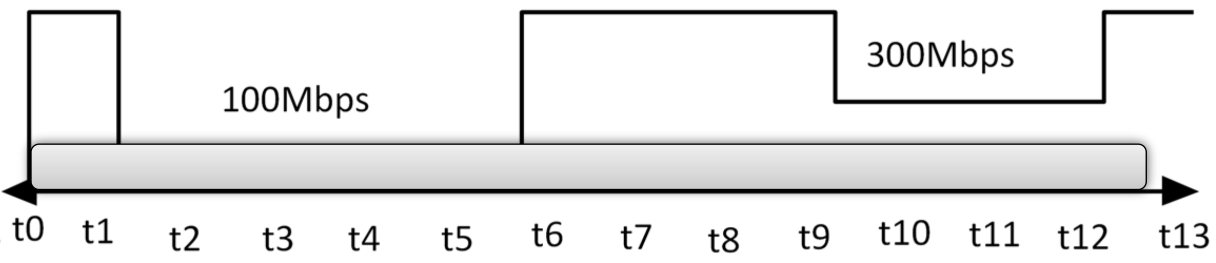
R1: (time t1, t6) A -> B ->D (900Mbps)
R2: (time t4, t7) A -> C -> D (400Mbps)
R3: (time t9, t12) A -> B -> D (700Mpbs)
R4: (time t9, t12) A -> C -> D (500Mpbs)



Reservations

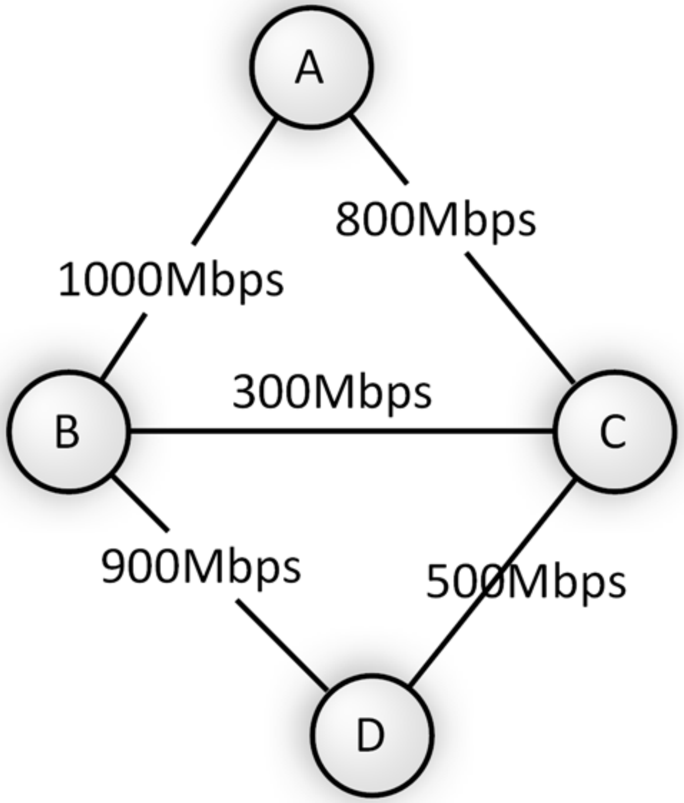


allocated



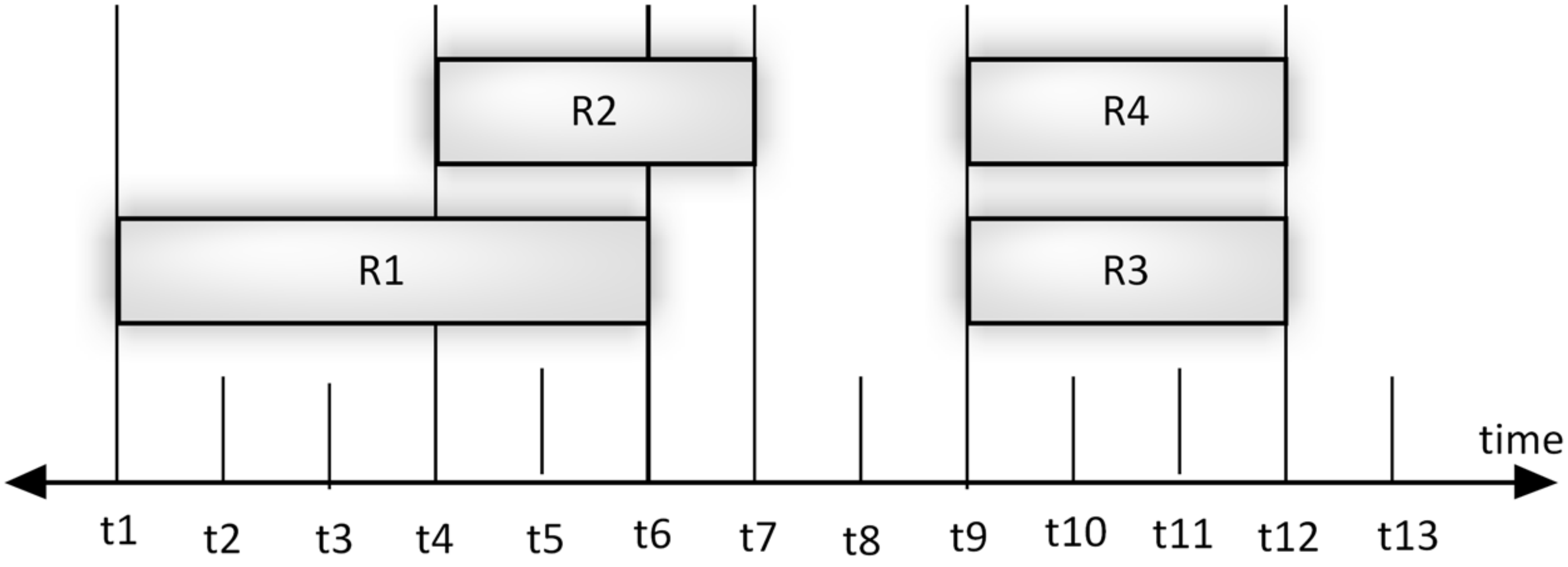
available

Advanced Reservations

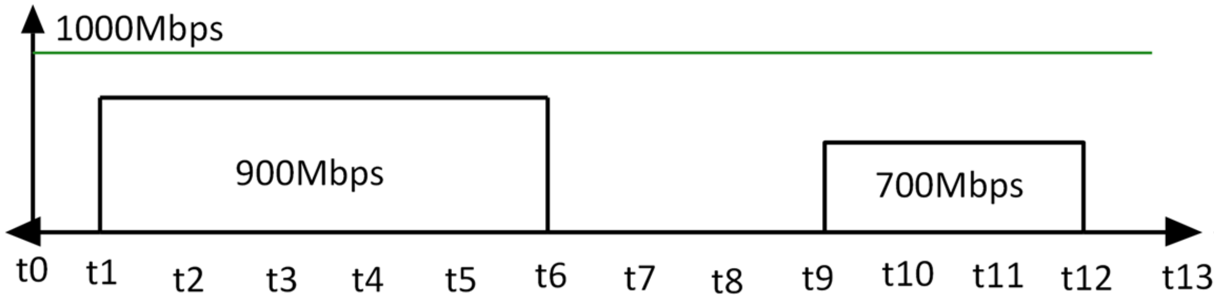


Network Graph with edges showing maximum bandwidth

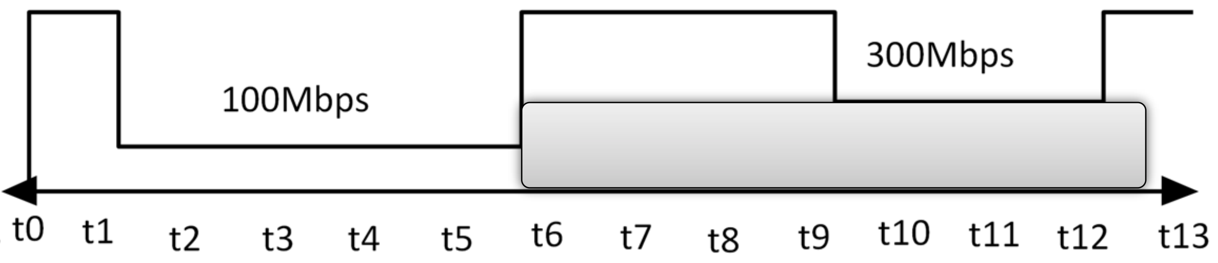
R1: (time t1, t6) A -> B ->D (900Mbps)
R2: (time t4, t7) A -> C -> D (400Mbps)
R3: (time t9, t12) A -> B -> D (700Mpbs)
R4: (time t9, t12) A -> C -> D (500Mpbs)



Reservations

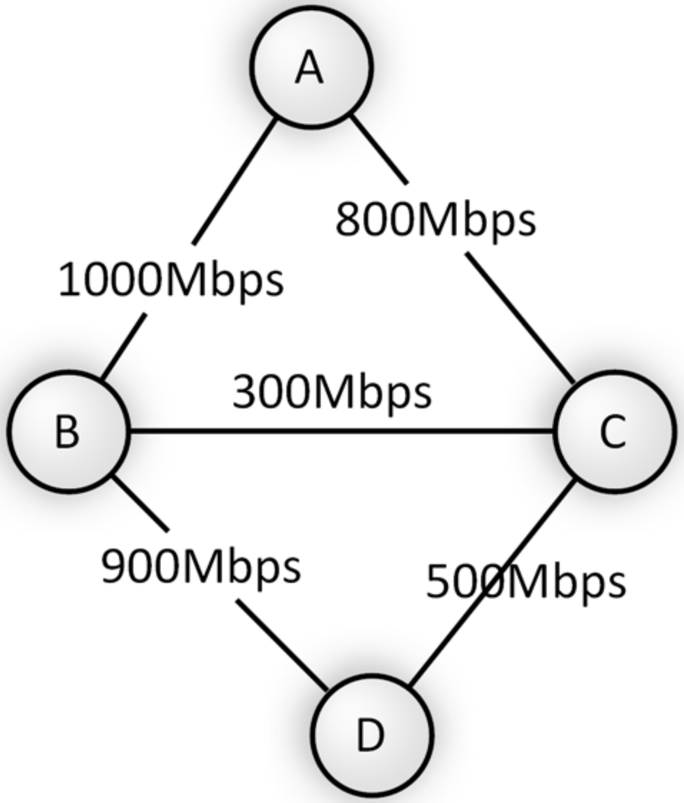


allocated



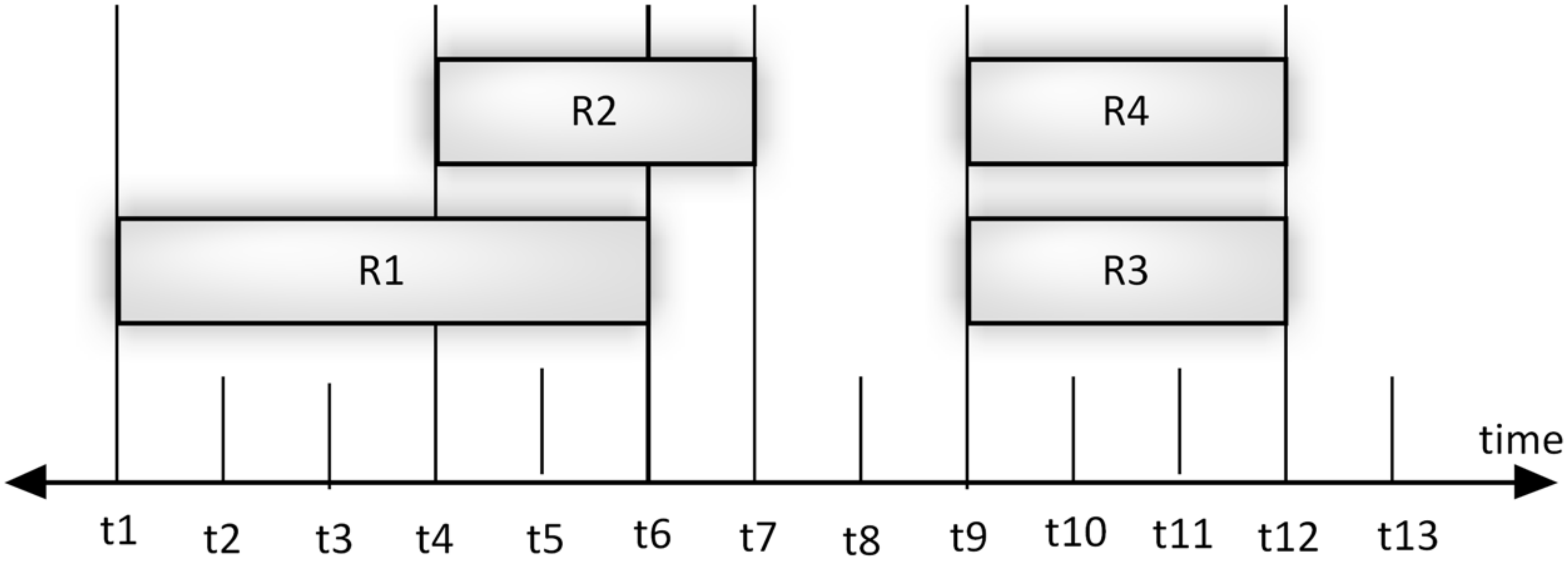
available

Advanced Reservations

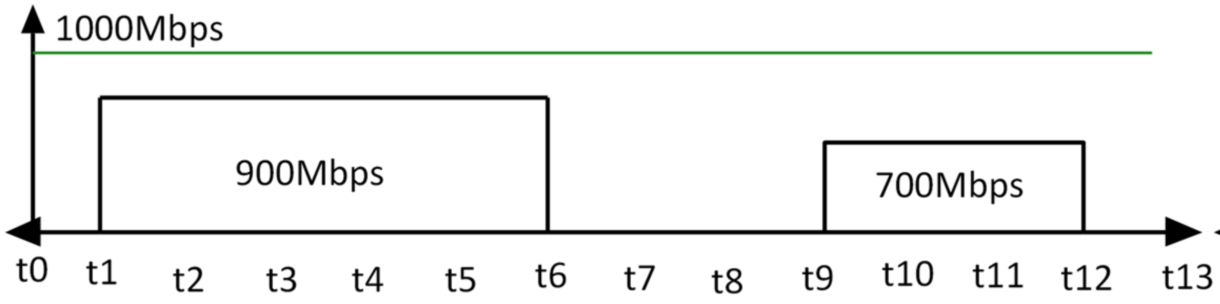


Network Graph with edges
showing maximum bandwidth

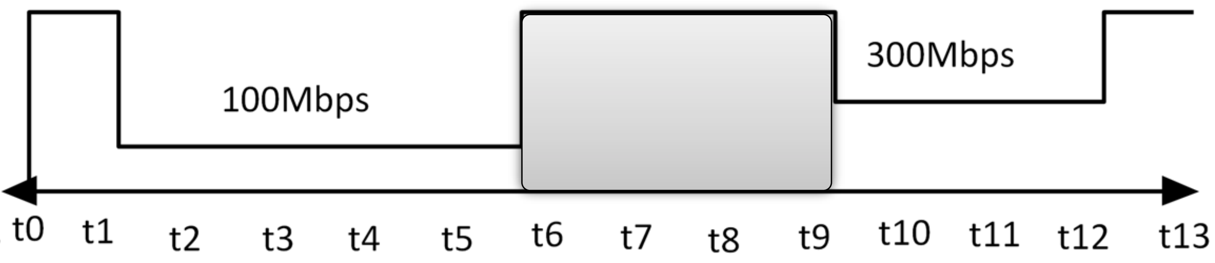
R1: (time t1, t6) A -> B ->D (900Mbps)
R2: (time t4, t7) A -> C -> D (400Mbps)
R3: (time t9, t12) A -> B -> D (700Mpbs)
R4: (time t9, t12) A -> C -> D (500Mpbs)



Reservations

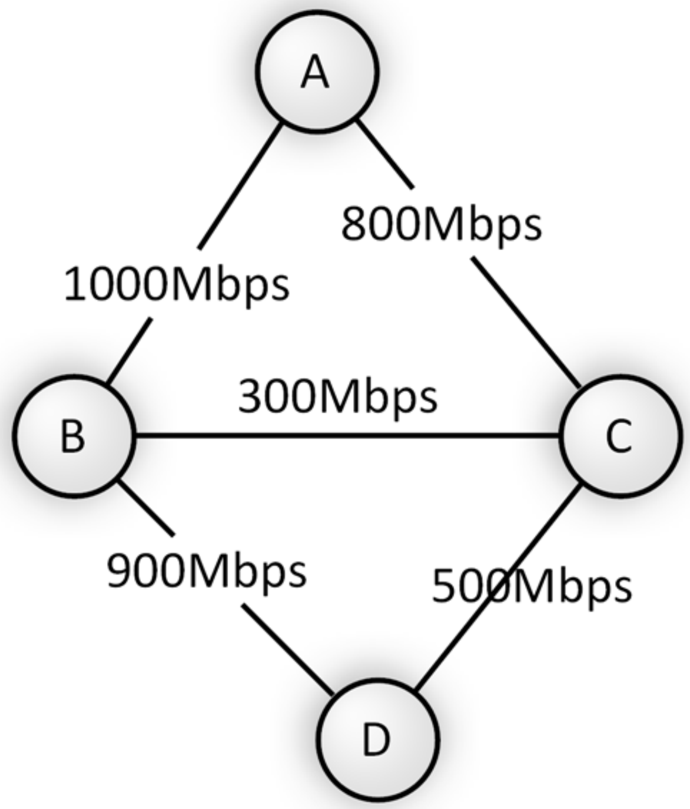


allocated



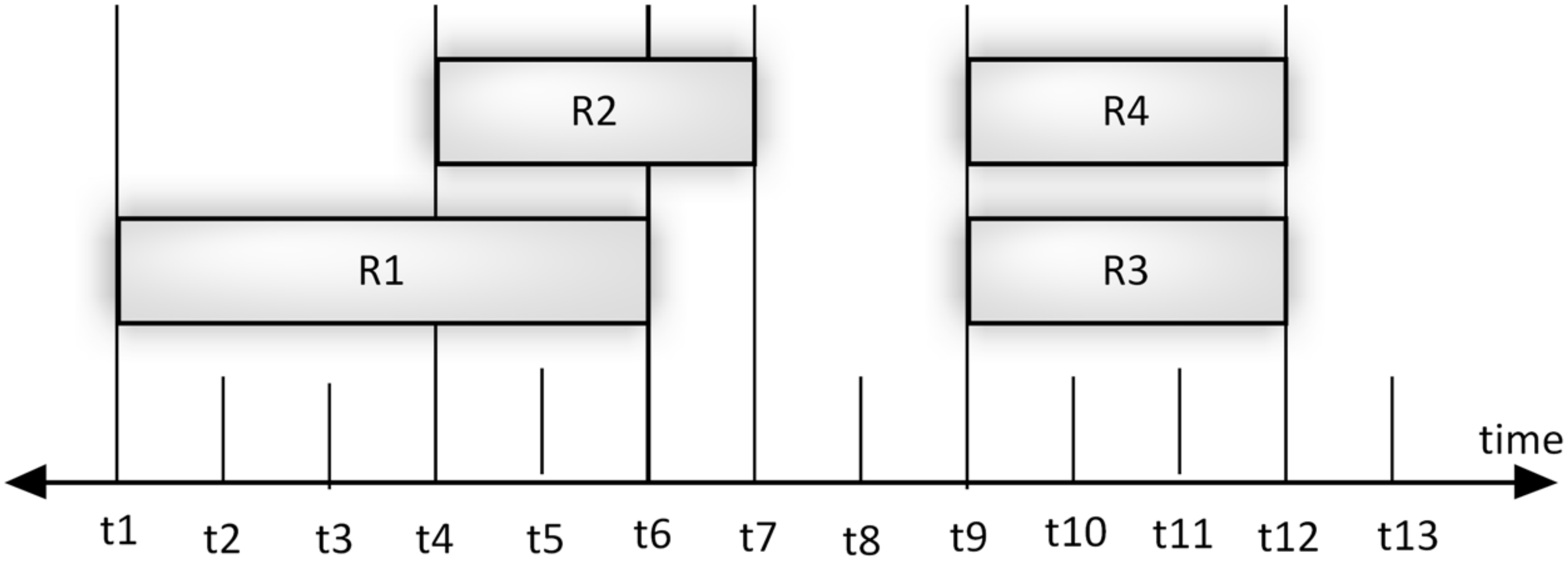
available

Advanced Reservations

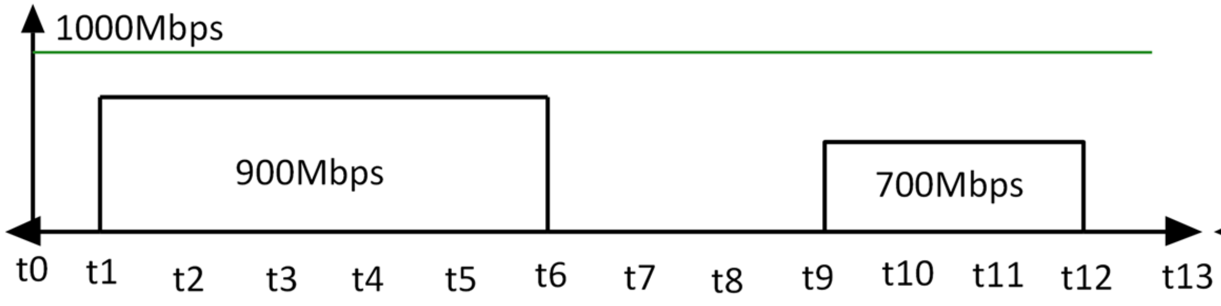


Network Graph with edges showing maximum bandwidth

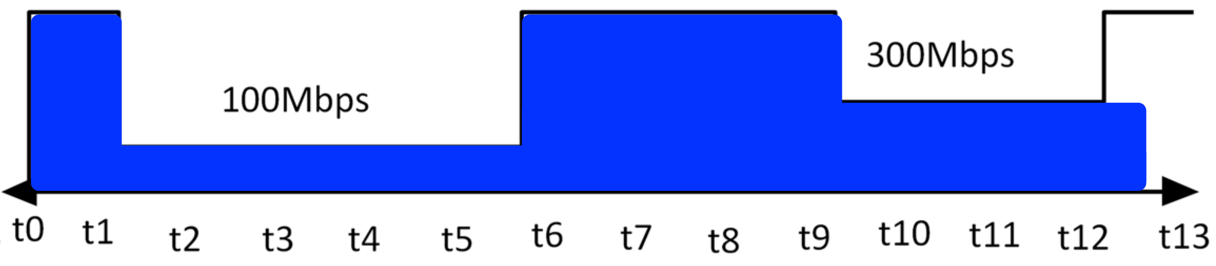
R1: (time t1, t6) A -> B ->D (900Mbps)
R2: (time t4, t7) A -> C -> D (400Mbps)
R3: (time t9, t12) A -> B -> D (700Mpbs)
R4: (time t9, t12) A -> C -> D (500Mpbs)



Reservations

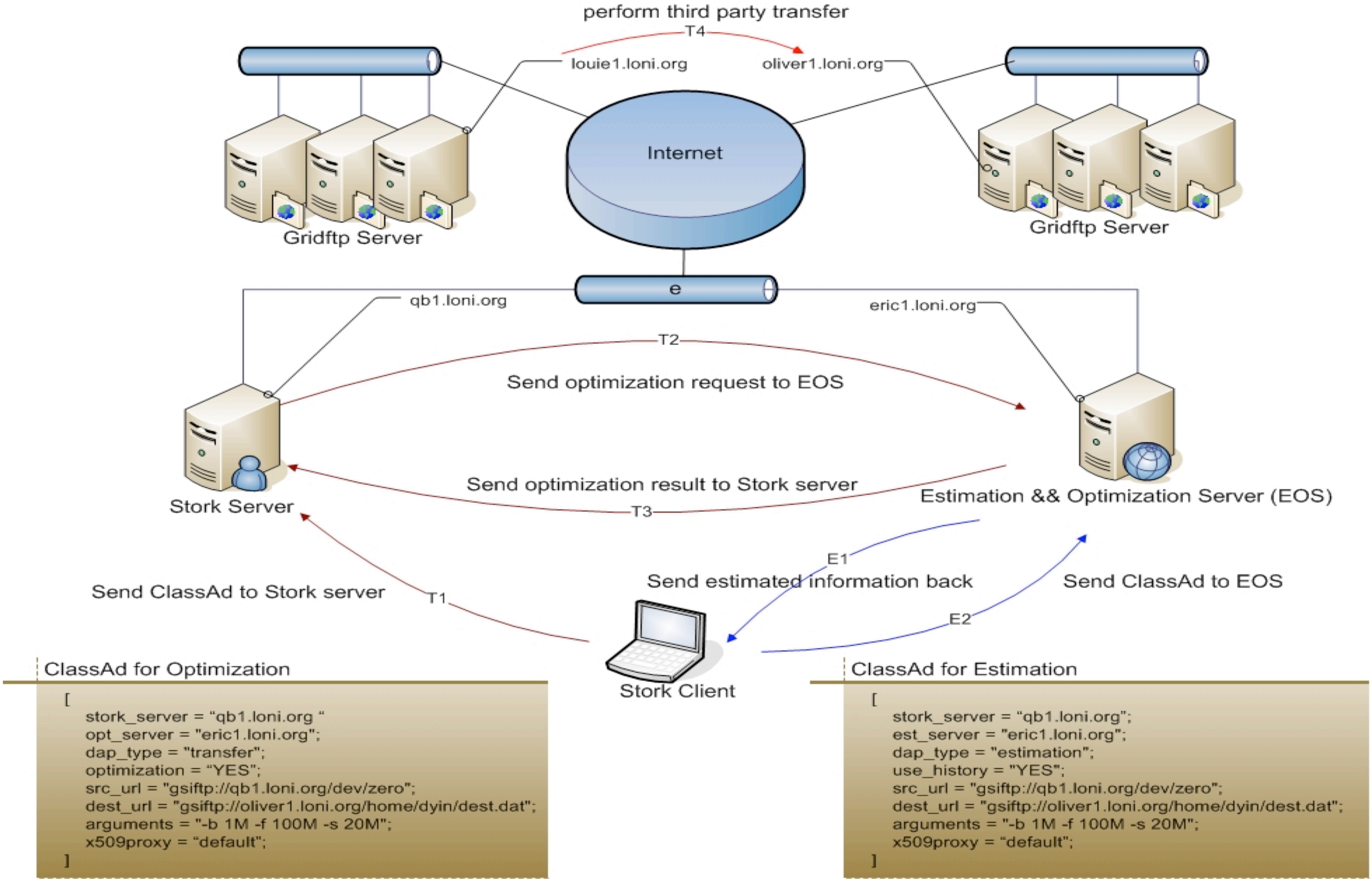


allocated



available

Implementation in Stork


perform third party transfer
T4
louie1.loni.org
oliver1.loni.org
Internet
Gridftp Server
Gridftp Server
e
qb1.loni.org
eric1.loni.org
T2
Send optimization request to EOS
Send optimization result to Stork server
T3
Stork Server
Estimation && Optimization Server (EOS)
E1
Send estimated information back
Send ClassAd to EOS
Send ClassAd to Stork server
T1
E2
Stork Client

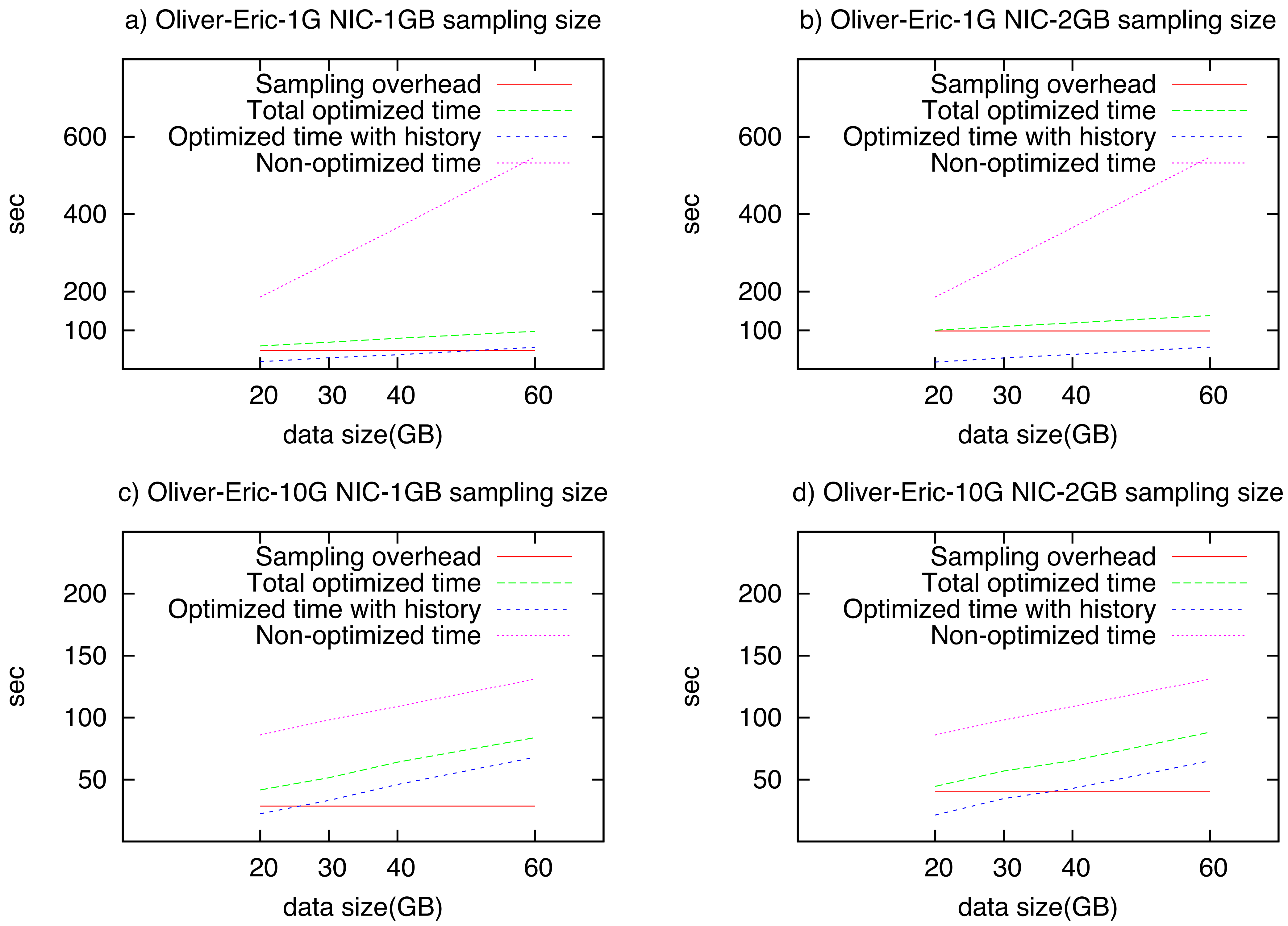
ClassAd for Optimization

```
[
    stork_server = “qb1.loni.org “
    opt_server = "eric1.loni.org";
    dap_type = "transfer";
    optimization = “YES";
    src_url = "gsiftp://qb1.loni.org/dev/zero";
    dest_url = "gsiftp://oliver1.loni.org/home/dyin/dest.dat";
    arguments = "-b 1M -f 100M -s 20M";
    x509proxy = “default";
]
```

ClassAd for Estimation

```
[
    stork_server = “qb1.loni.org";
    est_server = "eric1.loni.org";
    dap_type = "estimation";
    use_history = "YES";
    src_url = "gsiftp://qb1.loni.org/dev/zero";
    dest_url = "gsiftp://oliver1.loni.org/home/dyin/dest.dat";
    arguments = "-b 1M -f 100M -s 20M";
    x509proxy = “default";
]
```

Optimization Overhead



a) Oliver-Eric-1G NIC-1GB sampling size

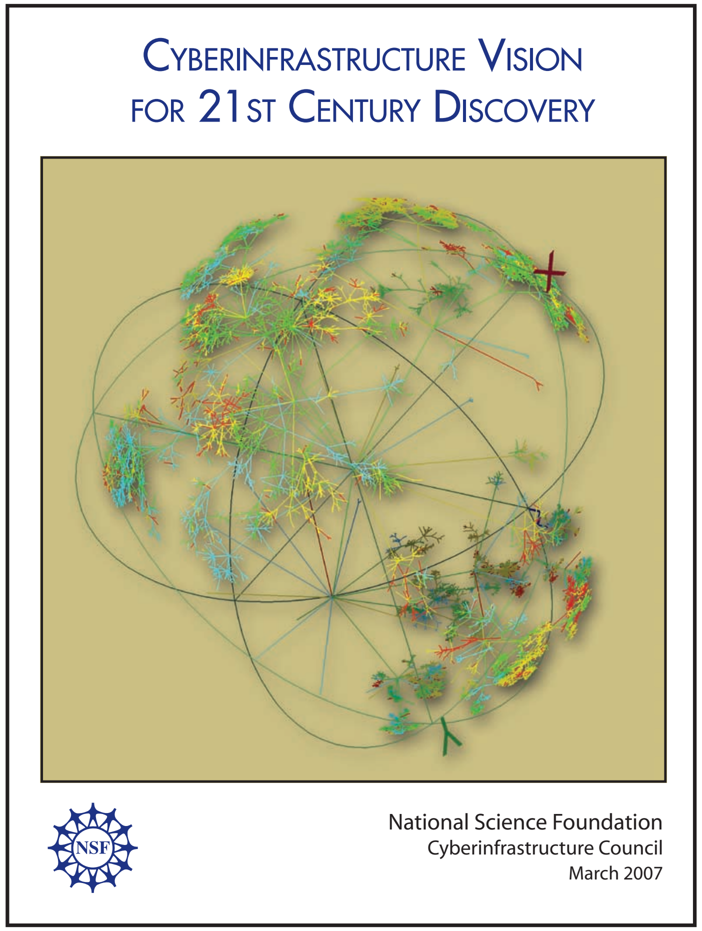
b) Oliver-Eric-1G NIC-2GB sampling size

c) Oliver-Eric-10G NIC-1GB sampling size

d) Oliver-Eric-10G NIC-2GB sampling size

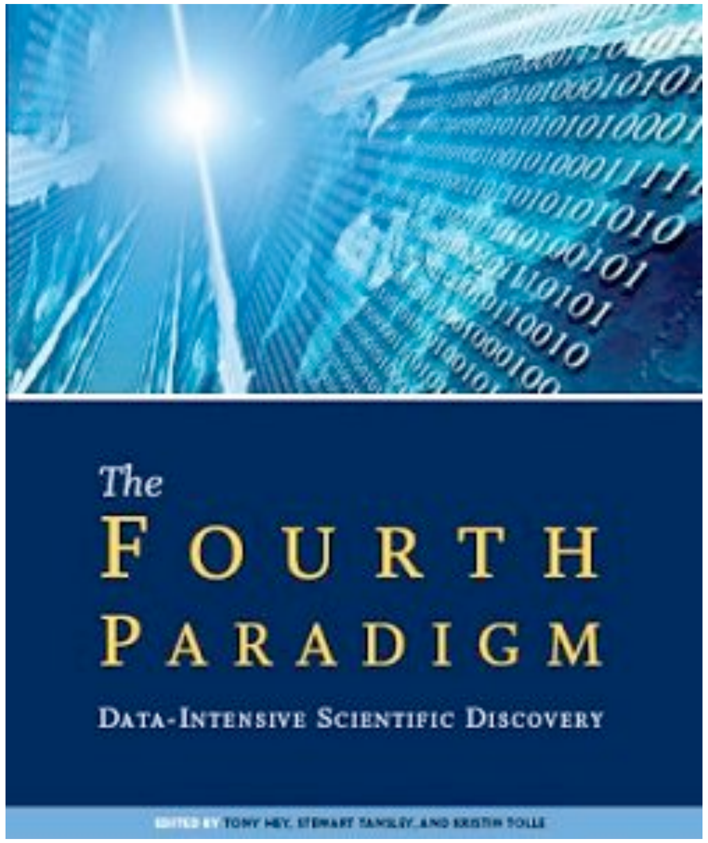
Summary

- Scientific applications are getting more and more data intensive
- Data sharing and I/O is the major bottleneck in front of multi-institutional and inter-disciplinary collaborative science
- Stork project focuses on developing models, algorithms, and real systems to mitigate the data bottleneck
- Recent NSF CiC Grant allows us to bring scheduled data placement to Windows Azure platform



CYBERINFRASTRUCTURE VISION
FOR 21ST CENTURY DISCOVERY

National Science Foundation
Cyberinfrastructure Council
March 2007



The
FOURTH
PARADIGM

DATA-INTENSIVE SCIENTIFIC DISCOVERY

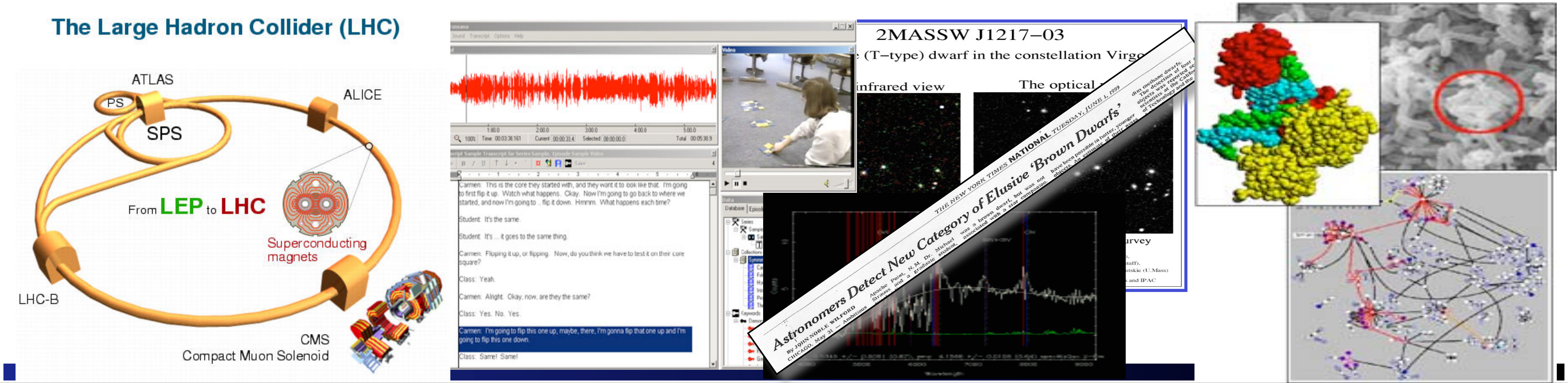
EDITED BY TONY HEY, STEWART TANSLEY, AND KRISTIN TOLLE

This work has been sponsored by NSF Awards:

CNS-1131889 (CAREER) – Research & Theory
OCI-0926701 (STCI) – SW Design & Implementation
CCF-1115805 (CiC) – Stork for Windows Azure

For more information:

Stork web page: http://www.storkproject.org



Carmen: This is the core they started with, and they want it to look like that. I'm going to first flip it up. Watch what happens. Okay. Now I'm going to go back to where we started, and now I'm going to .. flip it down. Hmmm. What happens each time?

Student: It's the same.

Student: It's ... it goes to the same thing.

Carmen: Fliping it up, or flipping. Now, do you think we have to test it on their core square?

Class: Yeah.

Carmen: Alright. Okay, now, are they the same?

Class: Yes. No. Yes.

Carmen: I'm going to flip this one up, maybe, there, I'm gonna flip that one up and I'm going to flip this one down.

Class: Same! Same!

THE NEW YORK TIMES NATIONAL TUESDAY, JUNE 1, 1999

Astronomers Detect New Category of Elusive 'Brown Dwarfs'

By JOHN NOBLE WILFORD

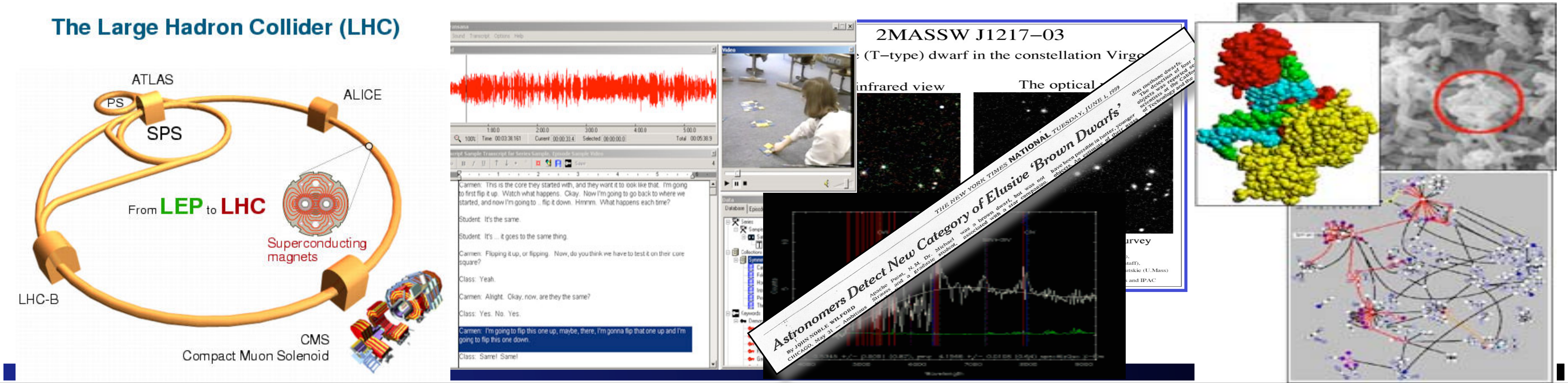
CHICAGO, May 31 — Ambitious Apache Point, N.M. Dr. Michael Strauss and a graduate student, was a brown dwarf, but was not associated with a star companion. have been possible in hotter, younger objects. An estimate of their mass dim methane dwarfs. The detection of four objects was reported scientists at the Califo of Technology and the

This work has been sponsored by NSF Awards:

CNS-1131889 (CAREER) – Research & Theory
OCI-0926701 (STCI) – SW Design & Implementation
CCF-1115805 (CiC) – Stork for Windows Azure

For more information:

Stork web page: http://www.storkproject.org


The Large Hadron Collider (LHC)
ATLAS
PS
SPS
ALICE
From LEP to LHC
Superconducting magnets
LHC-B
CMS
Compact Muon Solenoid
2MASSW J1217−03
(T−type) dwarf in the constellation Virgo
infrared view
The optical
THE NEW YORK TIMES NATIONAL TUESDAY, JUNE 1, 1999
Astronomers Detect New Category of Elusive 'Brown Dwarfs'
By JOHN NOBLE WILFORD