

# MPI Collective Communication Over OBS Grid

Satoshi Matsuoka<sup>†,††</sup> Shin'ichiro Takizawa<sup>†</sup>

Hidemoto Nakada<sup>†††,†</sup>

Masafumi Koga, Atsushi Takada<sup>††††</sup>

†: Tokyo Institute of Technology

††: National Institute of Information

†††: National Institute of Advanced Industrial Science and Technology

††††: NTT Laboratories

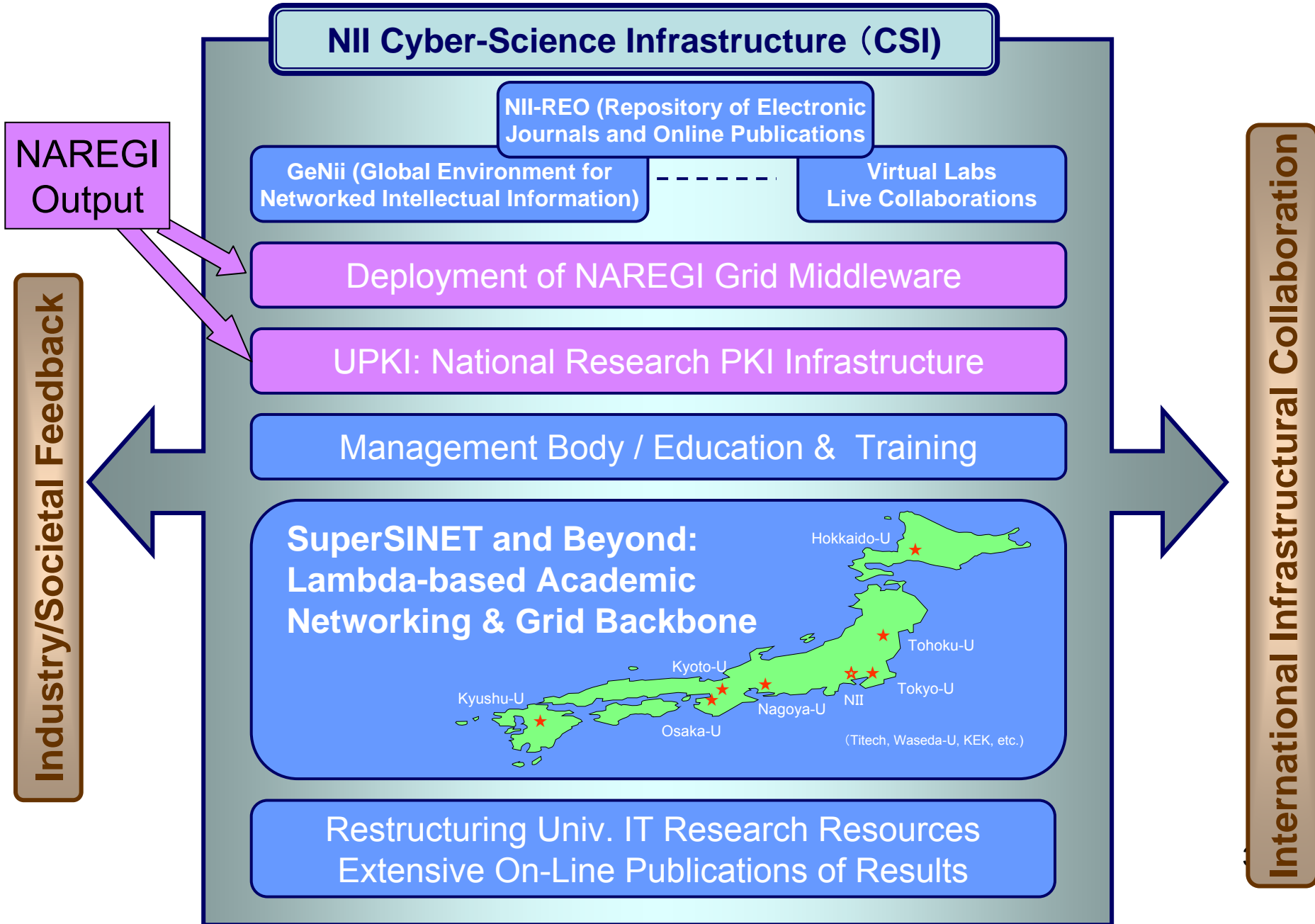
# Yes, I have a HDTV @ Home (and FTTH 100Mbps)

- With build-in HDTV HDD recorder
- Land-based & Satellite (BS/CS110)
- MPEG2 25Mbps
- MPEG4 AVC  
HDTV Internet VoD  
starting  
(NTT On-demand TV  
10Mbps)



- But now for something completely different...

# Towards a Cyber-Science Infrastructure for R & D



# The New 100 TeraFlop “Supercomputing Campus Grid” Core System@ Tokyo Institute of Technology, Spring 2006

Voltaire Infiniband  
10Gbps x 2 x 700 ports



Sun Galaxy 4 (Opteron Dual core 8-Way)  
10480core/655Nodes50Tera  
Flops

OS Linux

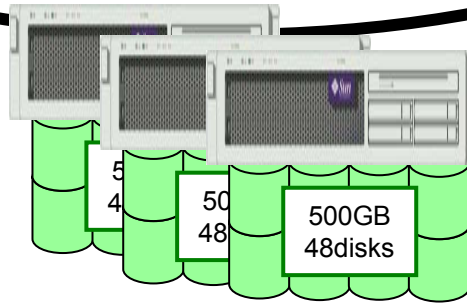
(Future) Solaris, Windows  
NAREGI Grid MW

**UNIFIED CLUSTER/STORAGE**  
**IB NW, 14Tbps Bisection BW,**  
**direct 10 Gbps hookup to**  
**SuperSINET**

10Gbps+External  
Network



NEC SX-8  
Small Vector  
Nodes (under  
plan)

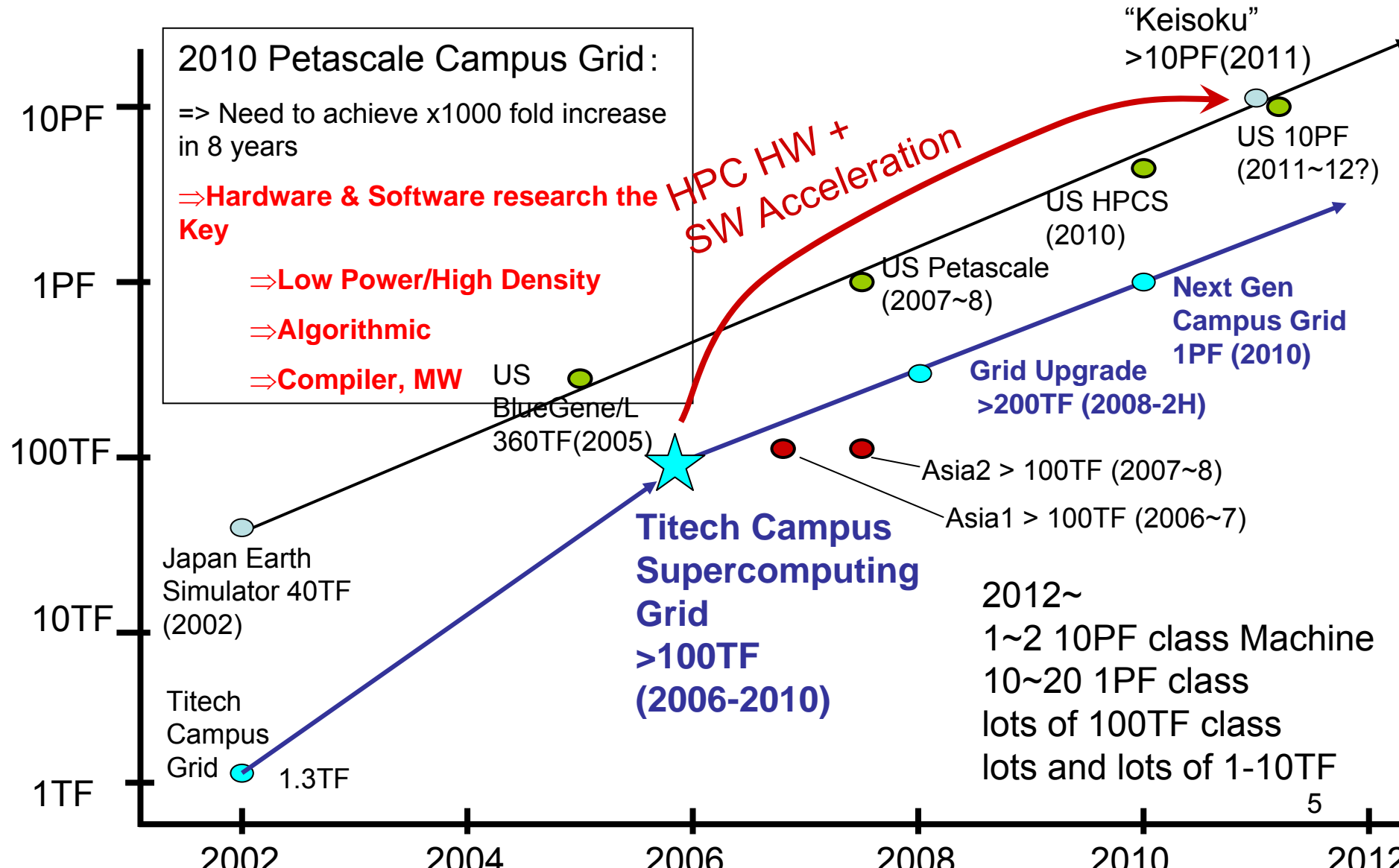


Storage  
1 Petabyte (Sun “Thumper”)  
0.1Petabyte (NEC iStore)  
Lustre FS, NFS (v4?)



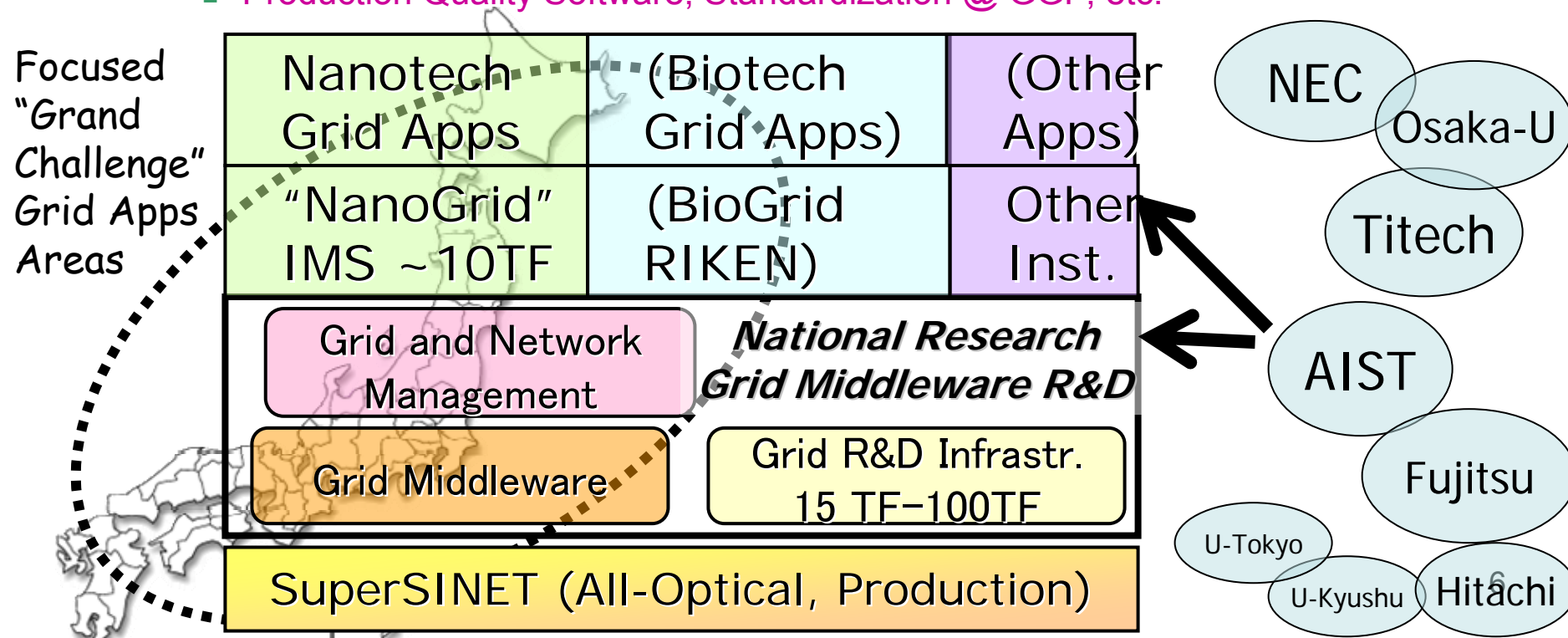
ClearSpeed CSX600  
SIMD accelerator  
35TeraFlops →  
60TeraFlops(1 board  
per node)

# Roadmap to PetaScale Campus Grid

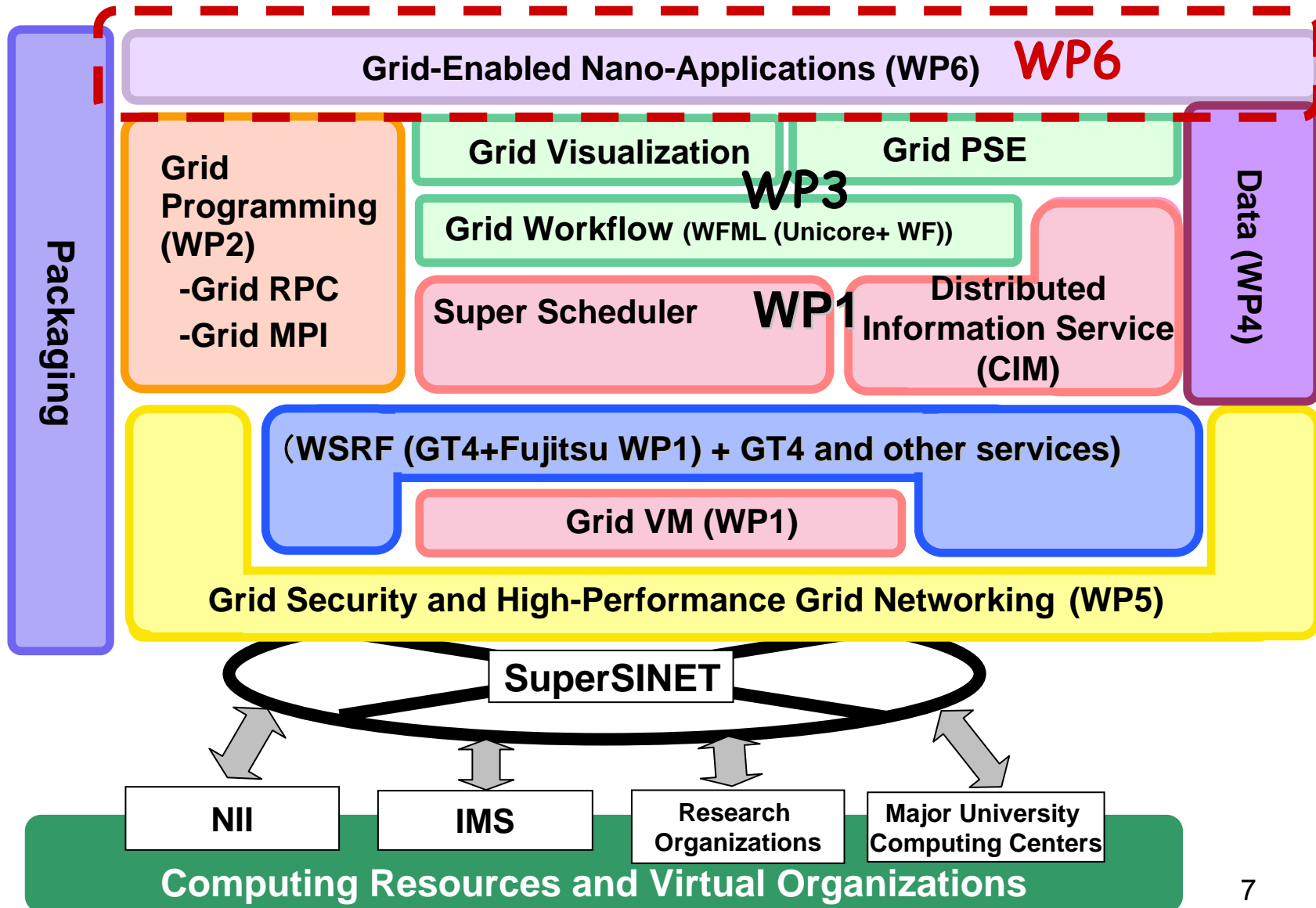


# MEXT National Research Grid Infrastructure (NAREGI) 2003-2007---PetaScale Grids ([www.naregi.org](http://www.naregi.org))

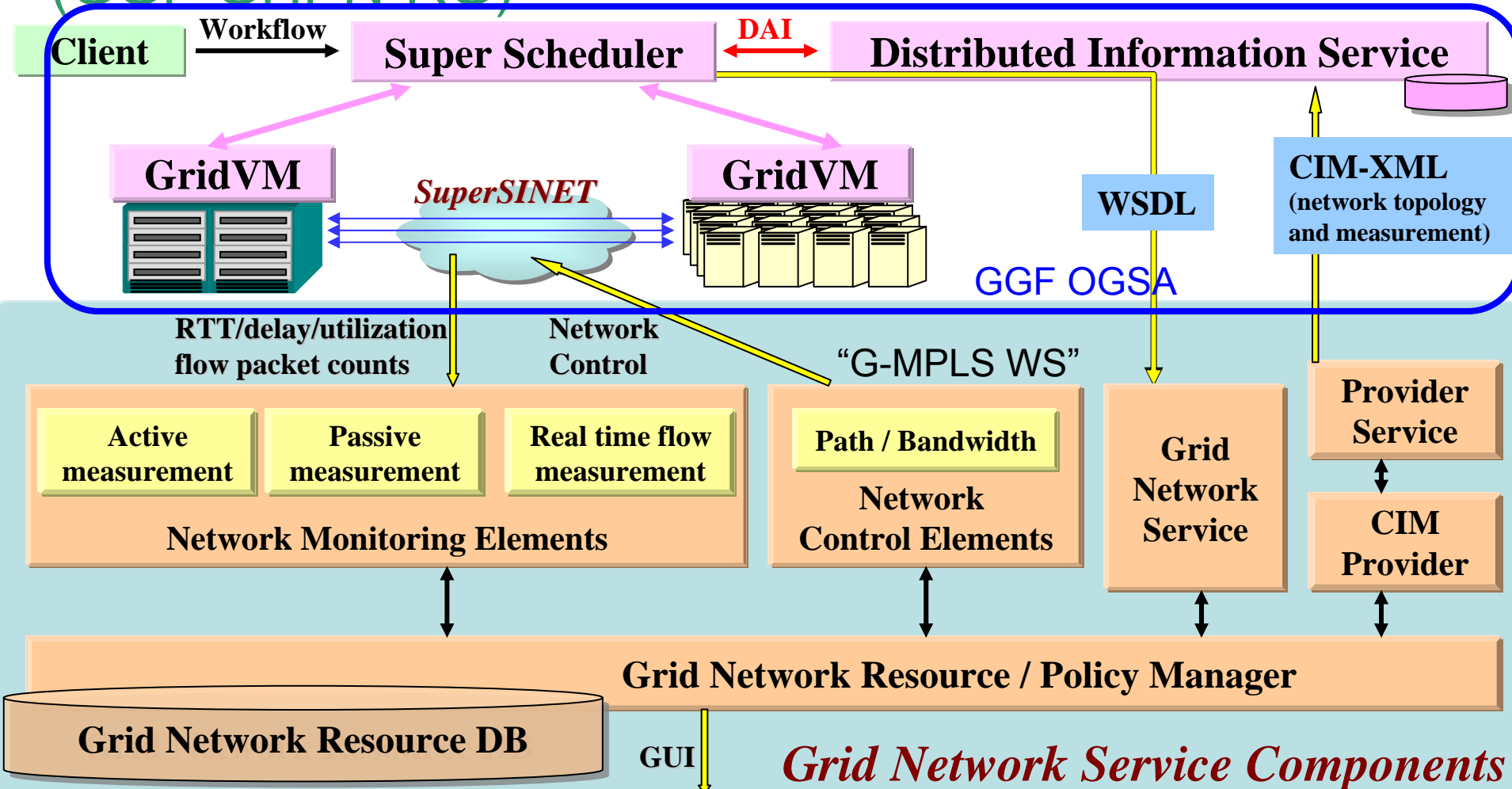
- Petascale Grid Infrastructure R&D for Future Deployment
  - \$45 mil (US) + \$16 mil x 5 (2003-2007) = \$125 mil total
  - Hosted by National Institute of Informatics (NII)
  - PL: Ken Miura (Fujitsu→NII)
    - Sekiguchi(AIST), Matsuoka(Titech), Shimojo(Osaka-U), Aoyagi (Kyushu-U)...
  - **NOT An ACADEMIC RESEARCH PROJECT!!**
    - Participation by Fujitsu, NEC, Hitachi, NTT, etc., >100FTEs
    - Production Quality Software, Standardization @ GGF, etc.



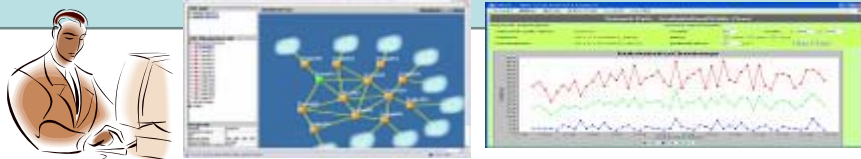
# NAREGI Software Stack (Beta Ver. 2006)



# NAREGI WP5 Network Architecture for Optical Grid (Kyushu Institute of Tech, Osaka-U, NII, Fujitsu, etc.) (GGF GHPN-RG)



Operators



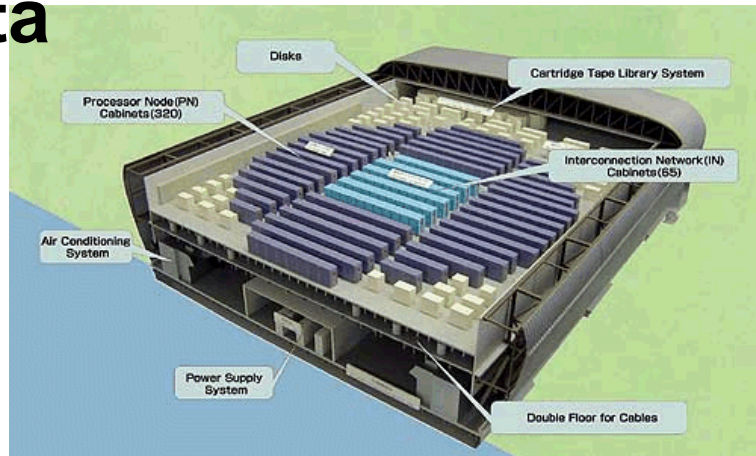
Visualization of Network Topology and Measurement

# Various Types of Grids

- Low  
NW  
BW
- ↑
- Desktop Grid
    - Sharing of Idle PC resources (e.g. [Seti@home](#))
  - Access Grid
    - Large-scale groupware, Video Conferencing
  - Sensor Grid
    - Unifying distributed sensor networks
  - Server Grid
    - Schedule jobs across machines and centers
  - Data Grid
    - Sharing and Managing of Large (Petascale) data
  - Metacomputing
    - Distributed High Performance Computing (e.g., running MPI applications over the grid)
- ↓
- High  
NW  
BW

# Will we achieve “plug&play” of clusters over an optical grid?

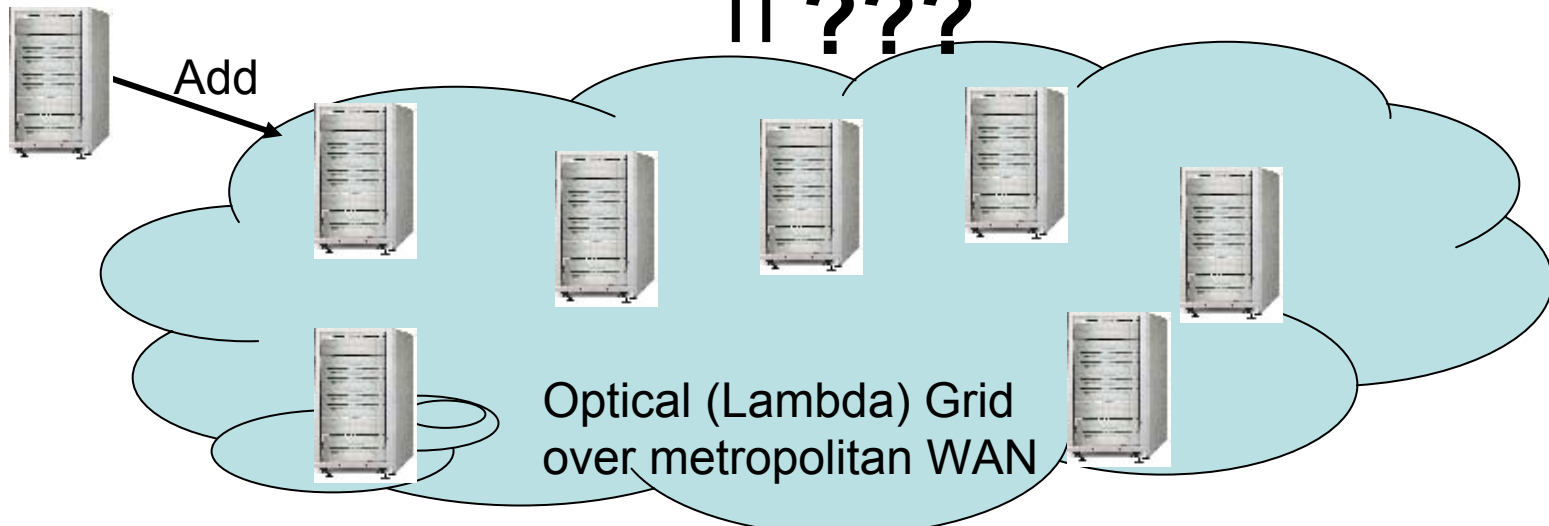
**Peta**



We usually discard this type of vision, but...

**Massive compute grid, SC2000 keynote vision come true?**

|| ???



# Comparison Optical Network vs. Existing Network

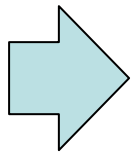
	Existing Network	Optical Network
<div data-bbox="161 378 1732 635" style="background-color: yellow; border: 1px solid black; border-radius: 15px; padding: 10px; text-align: center;">                     Affects applications that communicate frequently                 </div>		
link usage	shared	dedicated
preprocess before communication	none	establish light path
preprocess after communication	none	release light path
#connections	none	#lambda is limited

# Several work in Japan for optical computing grids

- Next-Gen SuperSINET 3 (2007) will have L1, L2, L3 provisioning as a standard service
- AIST GTRC and KDDI on JGN2 (OCS)
  - G-MPLS allocation of BW
  - Co-allocate compute and lambda resources via a prototype co-scheduler
  - Run parallel GridRPC applications
  - Successful experiments over JGN2
- Problems
  - Slow switching, OCS-level usage, lambda depletion

# Our work: Grid-enabling MPI Applications Over OBS

- Message Passing Interface (MPI)
  - Known to scale to Metropolitan (20ms delay) level (Ishikawa et. al.)
  - Various Grid-level MPIs (NAREGI GridMPI, MPICH-G2, PACX-MPI, ...)
- With OBS, it is necessary to dynamically establish /release a light path before/after communication
  - ~10 ms overhead per each communication
  - Collective communications greatly affected



Need to (1) reduce # establishing/releasing light path and (2) hide establish/release latency by overlapping with application communication

# Goal and Achievement of this project

## ■ Goal

- Study MPI collective communication over OBS networks to reduce latency caused by light path establish/release

## ■ Status

- Collaborative work between Titech and NTT Labs
- Propose algorithms that reduces the light path establish/release utilizing multiple ports counts per node
  - Port count = # destination nodes that a node can simultaneously communicate with (CWDM multi-wavelength modulation)
- Confirm effectiveness by modeling, simulation, and actual execution on a testbed

# Analysis of MPI Application

NPB:MG[A, 16] on GbEther

MPI Func	Average Exec Time	call count
Allreduce	358 $\mu$ s	88
Barrier	16.5ms	6
Bcast	16 $\mu$ s	6
Irecv	4 $\mu$ s	664
Reduce	299 $\mu$ s	1
Send	363 $\mu$ s	654
Wait	54 $\mu$ s	664

Exec Time: 0.79 sec

On optical network

- Assume cost of establishing/releasing light path = 10 ms
- If we establish/release the light path at each communication
  - Send : 654 times
    - ▶ + 6.5 sec (654 x 0.01)
  - Irecv,Wait : 664 times
    - ▶ + 6.6 sec (max)
  - collective communication : 101 times
    - ▶ + 101 x 0.01 x (p2p count) sec

Increase of about 10 to 20

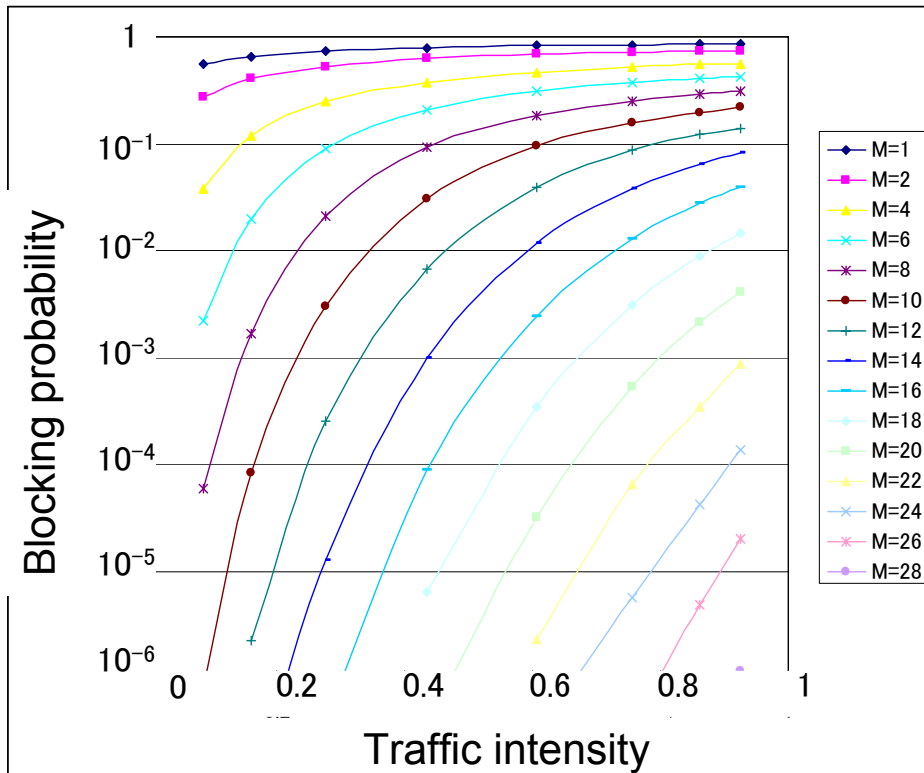
**Significant increase of execution time  
on a long-haul optical network**

# Our Methodology

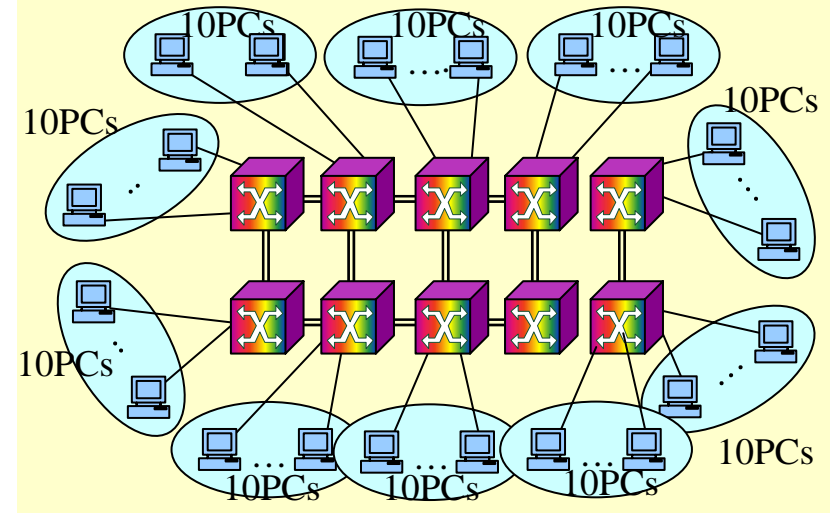
- Controlled establishment of light paths so that MPI Collectives can communicate “just-in-time” without the overhead
  - Simultaneously establish/release paths to multiple destinations (CWDM multi-wavelengths modulation)
  - Do not release a light path if a process frequently sends messages to the same destination
  - Conserve usage of light paths

# Blocking probability in WDM OBS network (NTT Labs)

A burst loss rate under  $10^{-6}$  is achieved even at the traffic intensity of 0.9, using wavelength assignment from 28 wavelengths in 2x5 ladder network accommodating 100 computers.



## Evaluated NW model

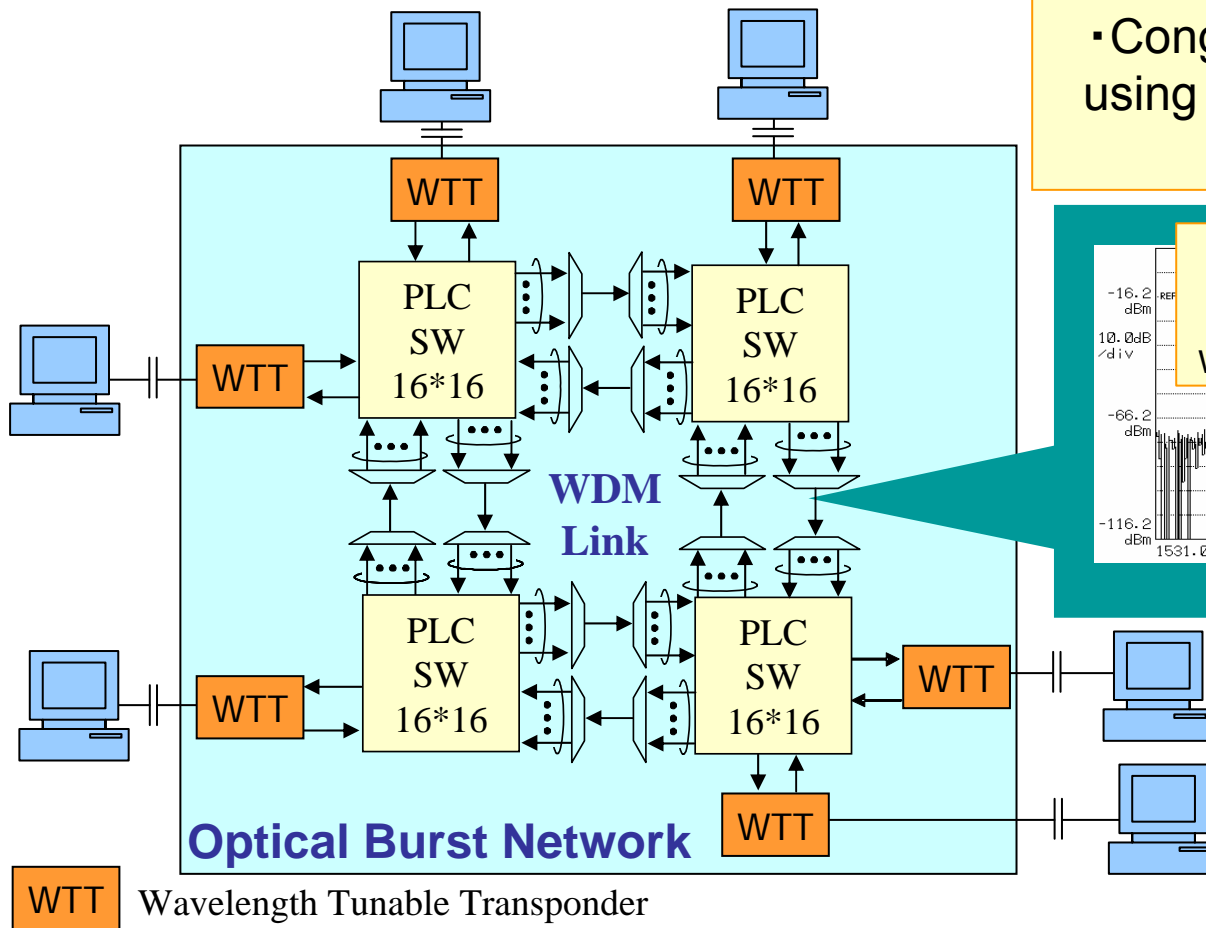


### <TRAFFIC PATTERN>

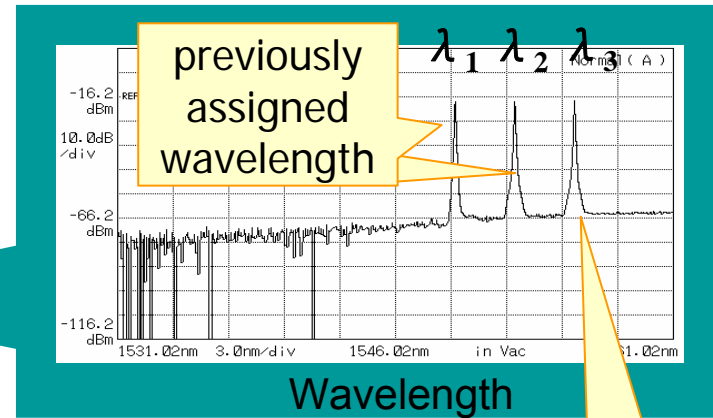
Optical burst data length : 100 ms  
 (about 100 MB in 10GbE)  
 Probability of data generation  
 : Poisson process  
 Destination node : random

# Experimental evaluation of OBS network

- 6 computers are connected by 4-node optical burst NW
- Each node has wavelength tunable transponder
- Each node is connected by WDM links

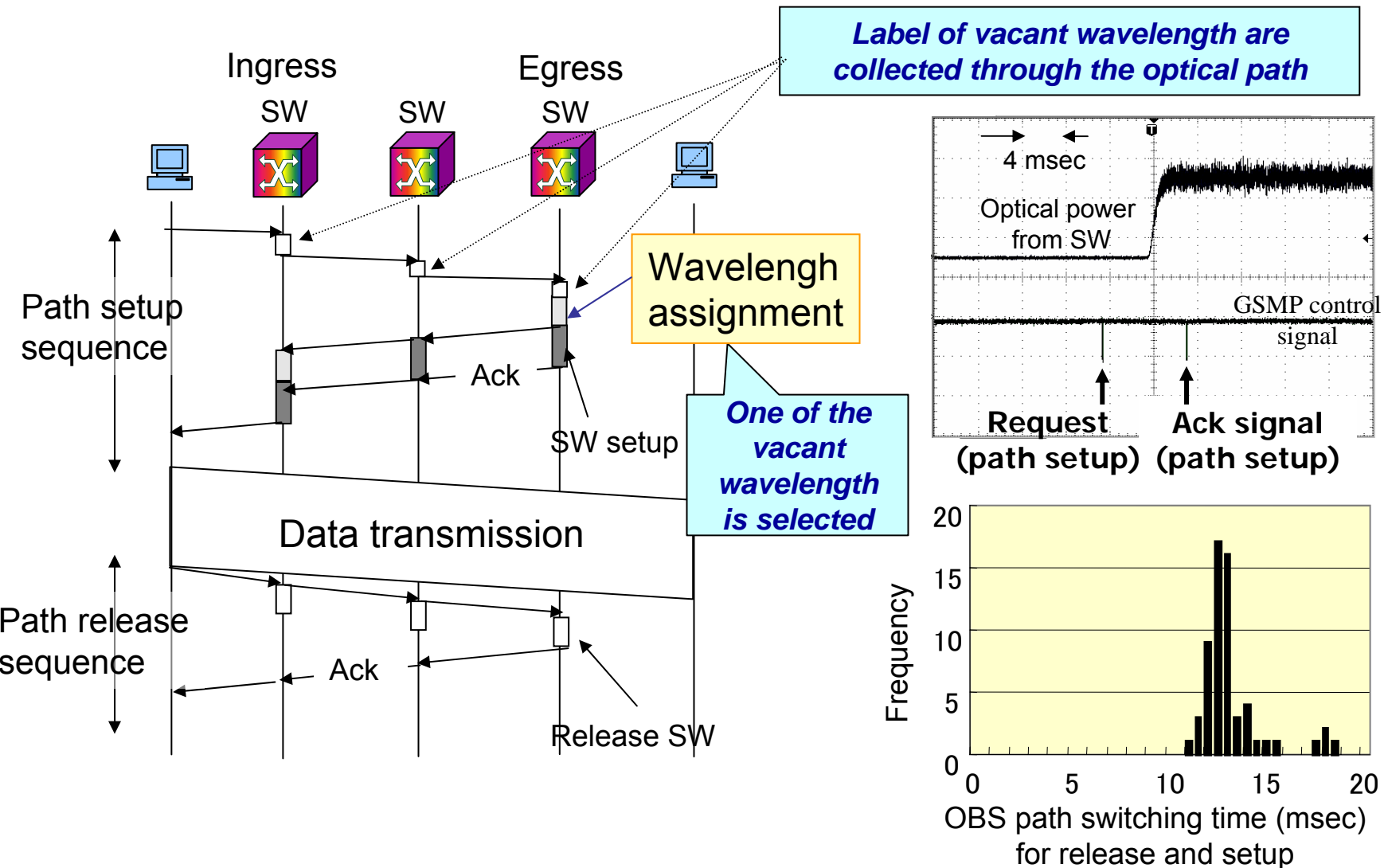


• Congestion control function using wavelength assignment is installed



congestion controlled wavelength

# Congestion control by wavelength assignment and Path switching time

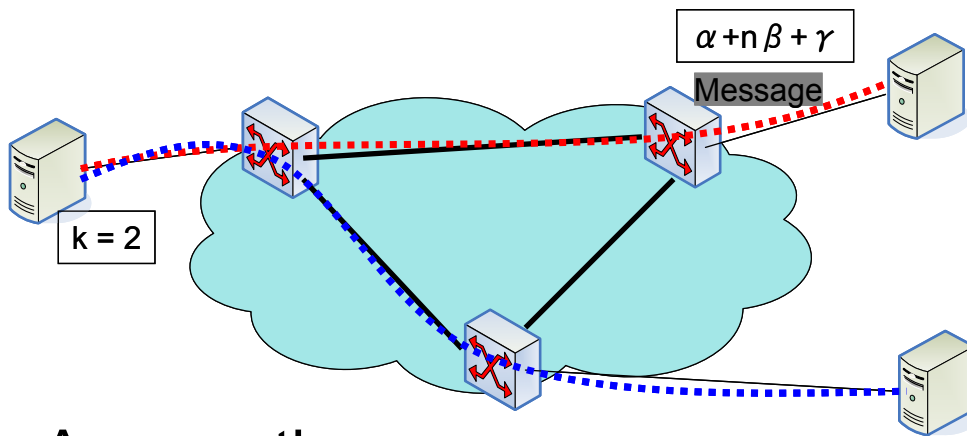


# Adapting MPI Collectives Algorithms to Optical Grids

- Existing MPI collectives algorithms adapted according to our methodology
  - Linear, Ring, Recursive Doubling, Binomial Tree
- Used in standard MPI implementation
  - E.g., MPI\_Allgather (MPICH 1.2.6 or later)
    - Ring (< 512KB) + Recursive Doubling (>= 512KB)
- Other algorithms are their variants

Model the execution cost of each algorithm adapted to the Optical Grid

# Underlying Optical Networks



## Assumption

- 1 MPI process per node
- Port count is equal in all processes
- full duplex transmission over a path
- The cost of establishing paths is constant, independent of how many paths are being established from a single node

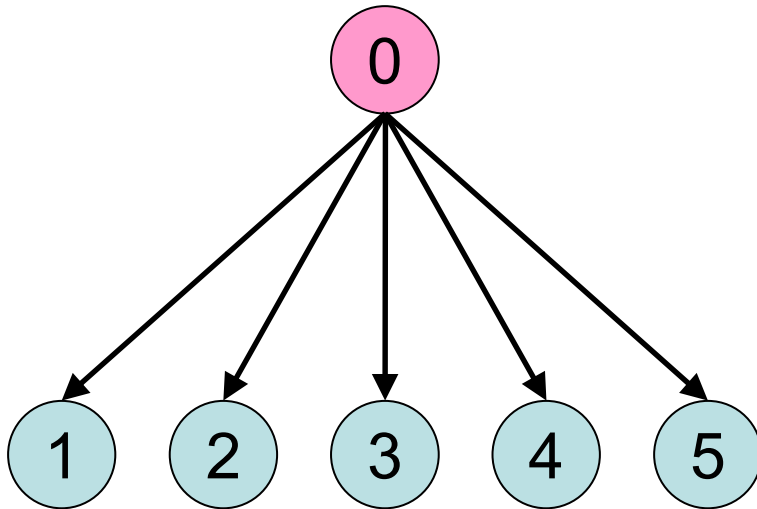
## Network Property

$p$	# of processes
$k$	port count (# of destinations that a node can simultaneously communicate with)
$n$	message size
$\alpha$	communication latency
$\beta$	per byte communication time
$\gamma$	path establishing/releasing time

## Communication Cost

- w. path management
  - $\alpha + n\beta + \gamma$
- w/o path management
  - $\alpha + n\beta$

# Linear: Behavior, Cost on Existing Network



- Root sends its  $n$  bytes message to other processes sequentially
- Step count  
 $p - 1$
- Communication cost per step  
 $\alpha + n\beta$

$$\text{Total Cost: } (p - 1)(\alpha + n\beta)$$

# Linear: Cost on Optical Network

- Establish/release paths at each communication (Naive method)

$$\underline{(p-1)(\alpha + n\beta + \gamma)}$$

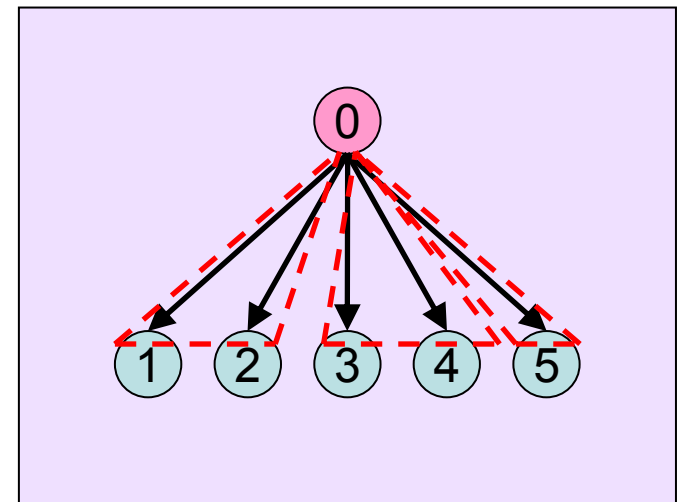
- Strategy

- Establish/release paths per port count independent of communication

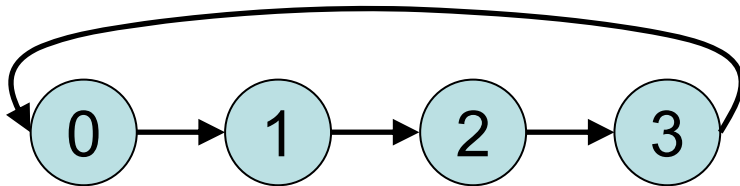
$$(p-1)(\alpha + n\beta) + \underline{\left[ \frac{p-1}{k} \right] \gamma}$$

- Amount of delay reduction

$$\left( (p-1) - \left[ \frac{p-1}{k} \right] \right) \gamma$$



# Ring: Behavior, Cost on Existing Network



- Each process sends  $n/p$  bytes message using ring topology network

- Step Count

$$p - 1$$

- Communication cost per step

$$\alpha + \frac{n}{p} \beta$$

$$\text{Total Cost: } (p - 1) \left( \alpha + \frac{n}{p} \beta \right)$$

# Ring: Cost on Optical Network

- Naive method

$$\underline{2(p-1)\left(\alpha + \frac{n}{p}\beta + \underline{\gamma}\right)}$$

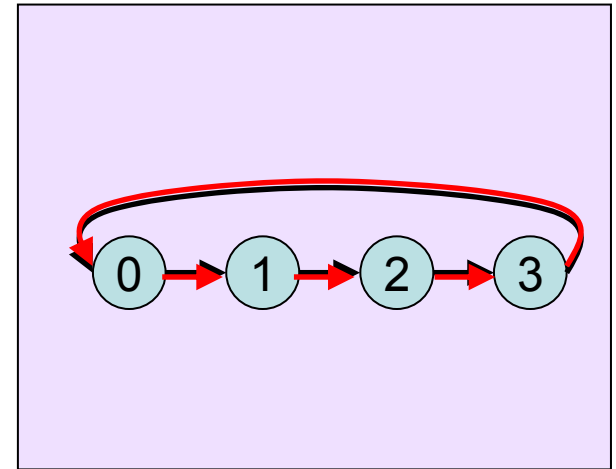
- Methodology

- If there are multiple ports, establish path to form a ring, communicate and release path

$$(p-1)\left(\alpha + \frac{n}{p}\beta\right) + \underline{\gamma}$$

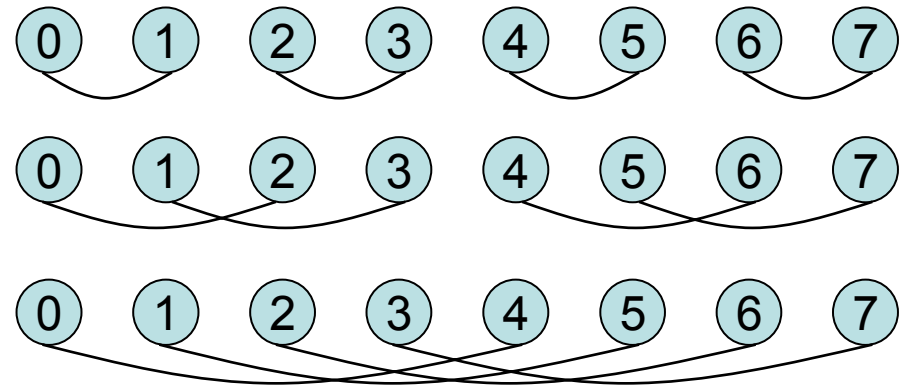
- Amount of delay reduction

$$(2p-3)\gamma$$



# Recursive Doubling: Behavior, Cost on Existing Network

- Each process exchange its  $2^{i-1}n/p$  bytes message with a process away by  $2^{i-1}$



- Step count  
 $\log p$

- Communication cost per step

$$\alpha + 2^{i-1} \left( \frac{n}{p} \right) \beta$$

$$\text{Total Cost: } (\log p)\alpha + (p-1)\frac{n}{p}\beta$$

# Recursive Doubling: Cost on Optical Network

- Naive method

$$\underline{(\log p)(\alpha + \gamma)} + (p-1)\frac{n}{p}\beta$$

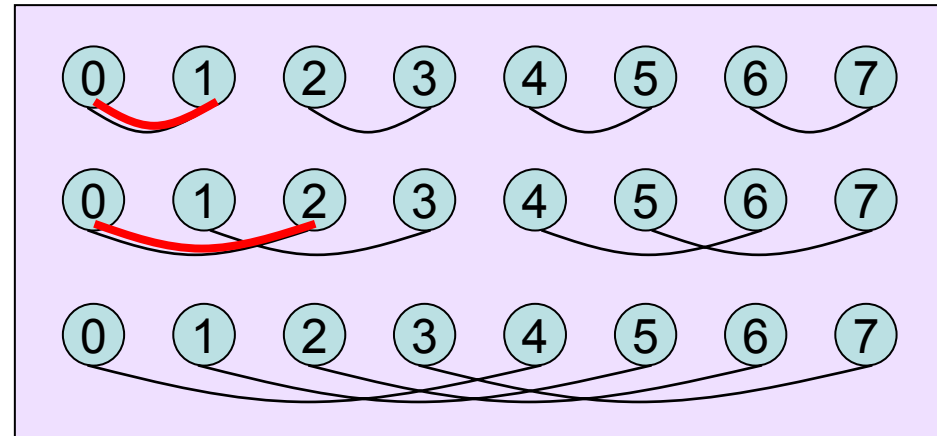
- Methodology

- Establish/release paths per port count independent of communication

$$(\log p)(\alpha) + (p-1)\frac{n}{p}\beta + \underline{\left\lceil \frac{\log p}{k} \right\rceil \gamma}$$

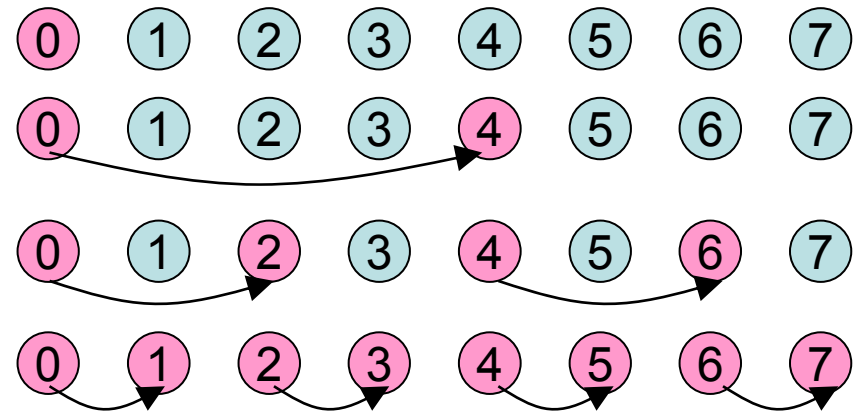
- Amount of delay reduction

$$\left( \log p - \left\lceil \frac{\log p}{k} \right\rceil \right) \gamma$$



# Binomial Tree: Behavior, Cost on Existing Network

- Root sends its  $n$  bytes message to other processes using binomial tree



- Step count

$$\log p$$

- Communication cost per step

$$\alpha + n\beta$$

$$\text{Total Cost: } (\log p)(\alpha + n\beta)$$

# Binomial Tree: Cost on Optical Network

- Naive method

$$(\log p)(\alpha + n\beta + \gamma)$$

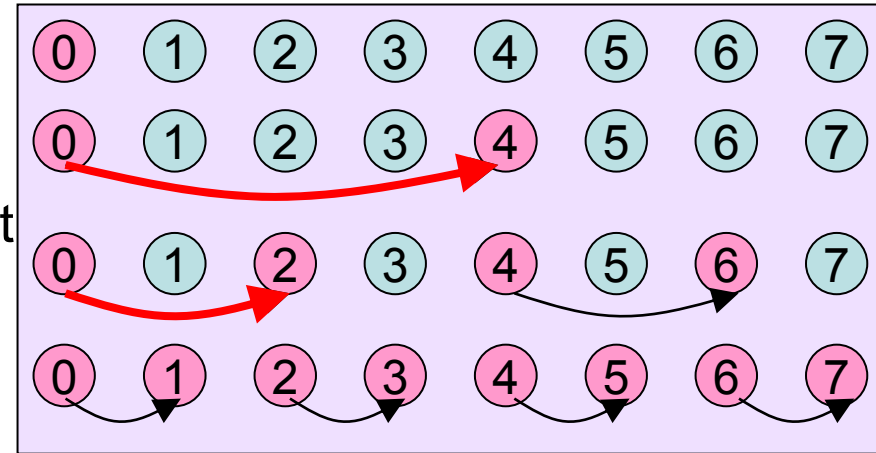
- Methogology

- Establish/release paths per port count independent of communication

$$(\log p)(\alpha + n\beta) + \left\lfloor \frac{\log p}{k} \right\rfloor \gamma$$

- Amount of delay reduction

$$\left( \log p - \left\lfloor \frac{\log p}{k} \right\rfloor \right) \gamma$$



# Collective Communication in Real Optical Network Environment

- # Light paths bound by available lambdas
  - Cannot establish paths / communicate (**congestion**)
- If congestion occurs, collective communication fails
  - Problems for algorithms that use multiple paths at a time, such as “Ring”
    - ▶ A Retry mechanism is necessary
- Can also result in non-optimal path allocation
  - Problems in algorithms that imposes hierarchical order between processes, such as “Binomial Tree”
    - ▶ A smart path allocation algorithm is necessary

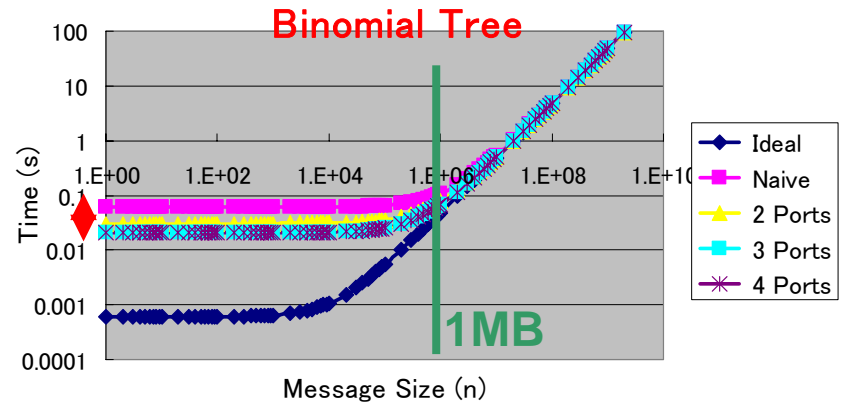
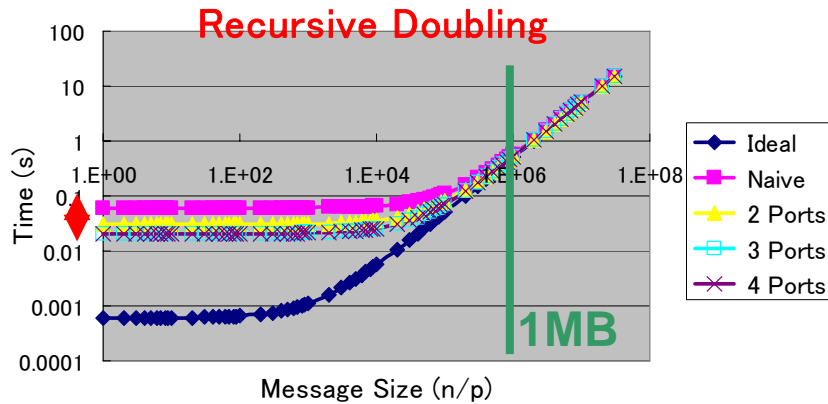
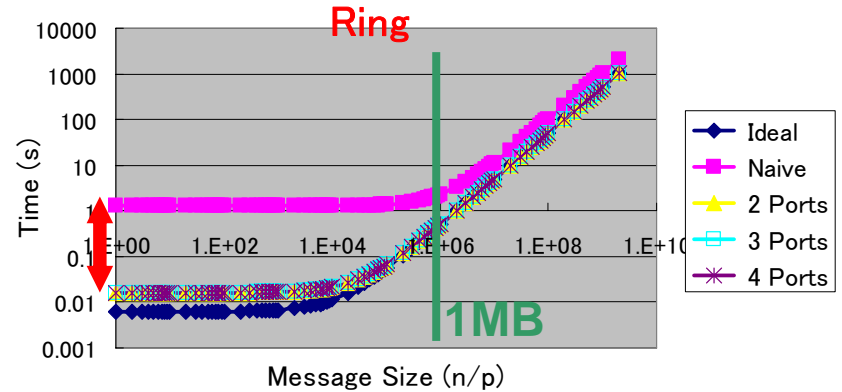
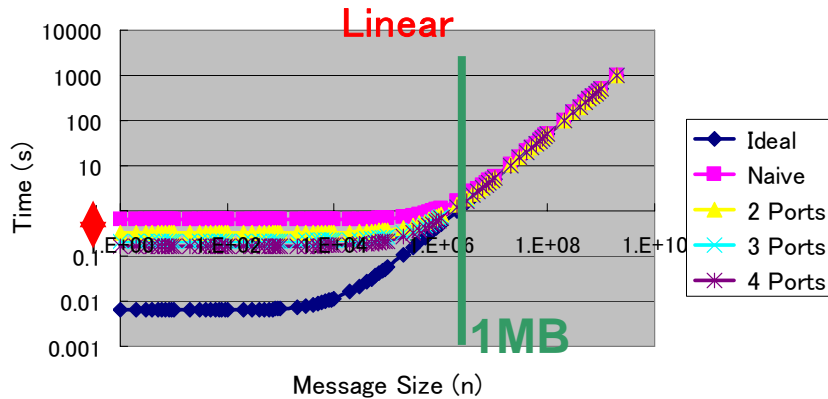
# Simple Evaluation

- Evaluate cost formulas of each algorithm by changing the message sizes for:
  - Ideal case
    - cost of path establishment/release is 0
  - Naive method
    - establish/release paths for every communication with cost  $\gamma$
  - Proposed method, with varying available # ports

## Available Parameters

Var	Value	Note
p	64	# of processes
k	1,2,3,4	# of ports (1 equals to Naive method)
$\alpha$	0.0001	latency (100 $\mu$ s)
$\beta$	$10^9$	bandwidth (1Gbps)
$\gamma$	0.01	establishing/releasing a path (10ms)

# Results of various MPI collectives



**Our method is effective when message size is less than 1MB**

# Evaluation on Optical Grid Simulator

- NAS parallel Benchmarks EP and FT on a optical grid MPI simulator
- Execution Options : Class A, 16 processes
  - Compare with Ideal case, Naïve, and proposed method

## Network Setting

Variable	Value
Bandwidth	1Gbps
Latency	100 $\mu$ s
Latency for path management	10ms

## Collective Comm. Implementation

MPI Function	Employed Algorithm
MPI_Allreduce	Recursive Doubling
MPI_Alltoall	Recursive Doubling
MPI_Barrier	Recursive Doubling
MPI_Bcast	Binomial Tree
MPI_Reduce	Binomial Tree

# The Optical Grid MPI Simulator

- Log MPI application events and calculate execution time
  - Event data is a repetition of “CPU part” and “MPI part”
    - Generated from MPI application execution log
  - Cost of CPU part directly obtained by parsing log data
  - Cost of MPI part calculated by simulating communication
- Simulating optical networks
  - Establish a light path before communication
    - +0.005 sec
  - Release a light path after communication
    - +0.005 sec
  - Communication time

$$latency + \frac{message\_size}{bandwidth}$$

# Result of EP and FT

EP			FT		
Method	Exec Time (s)	Increase Ratio (%)	Method	Exec Time (s)	Increase Ratio (%)
Ideal	2.837	0.00	Ideal	2.001	0.00
Naive	3.034	6.94	Naive	2.666	33.23
2 Ports	2.934	3.41	2 Ports	2.328	16.53
3 Ports	2.934	3.41	3 Ports	2.328	16.35
4 Ports	2.887	1.76	4 Ports	2.158	7.83

- execution time is affected directly by # of collective communications
  - EP : 5 times , FT : 17 times
- More communication ports result in less overhead

# Related Work (1)

- Collective communication over optical network [Afsahi et al '02]
  - Proposes algorithm dependent on port count
  - Effective in simulation, but difficult to implement
  - no handling of path congestion
- Prediction algorithms for communication occurrence in MPI application [Afsahi et al '99]
  - Predict communication and establish path in advance
  - Path establishment/release time ( $\sim$ msec) is much bigger than CPU time ( $\sim \mu$  sec), as such establish/release latency cannot be hidden

# Related Work (2)

- Nodes that combine optical and conventional electrical NICs [Barker'05]
  - In point-to-point communications, initially the electrical network is used, and when message size exceeds a certain threshold, then communication is delegated to optical networks
  - Collective communications are always executed on the electrical network

# Summary and Future Work

- Proposed and evaluated implementations of MPI collectives on optical grids that by simultaneously establish/release light paths according to the port count
  - Confirmed reduction of execution time at both model and simulation levels
  - Testing on real optical switched networks in the works in collaboration with NTT Labs.
- Future work
  - Consider collective communication algorithms under heavy congestion, in the presence of external processes