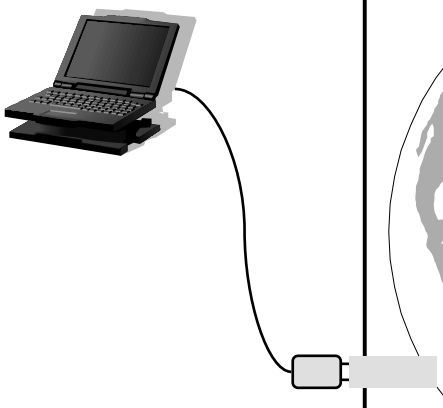
Towards an Application-Aware Multicast Communication Framework for Computational Grids

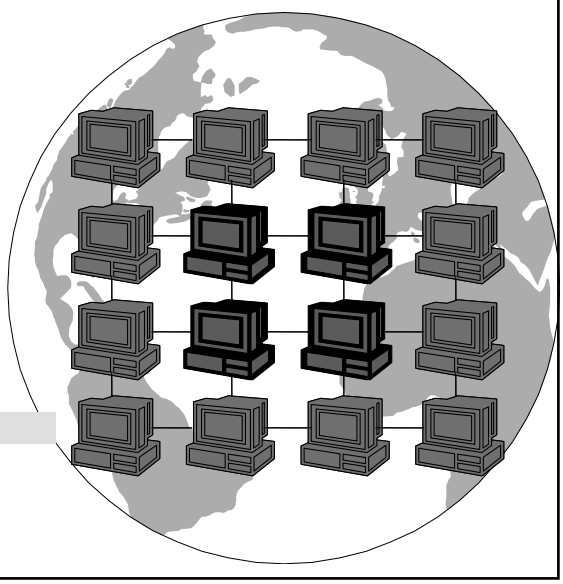
M. MAIMOUR, C. PHAM
RESO/LIP, UCB Lyon

ASIAN'02, Hanoi
Dec 5th, 2002

Computational grids

application user





from Dorian Arnold: Netsolve Happenings

The current usage of grids

- Mostly
  - Database accesses, sharing, replications(DataGrid, Encyclopedia of Life Project…)
  - Distributed Data Mining (seti@home…)
  - Data and code transfert, massively parallel job submissions (task-farm computing)
- Few
  - Distributed applications (MPI…)
  - Interactive applications (DIS, HLA…), remote visualization

WHY?

WHY?

End-to-End performances are not here yet

Not scalable!

Unable to adapt to new technologies and uses

WHY??

People forgot the networking side of grids

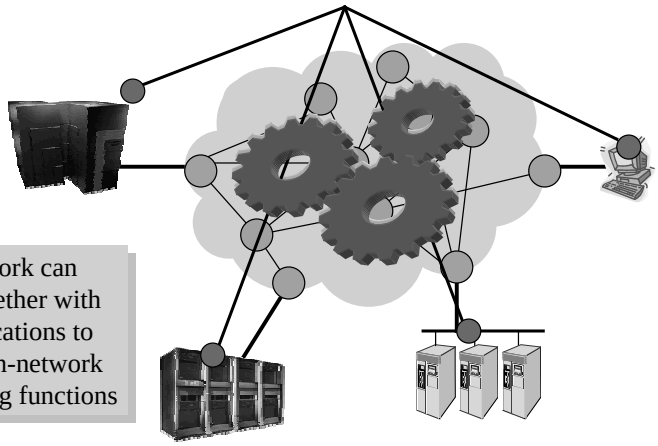
Gbits/s links do not mean E2E performances!

Computing resources and network resources are logically separated

Visions for a grid

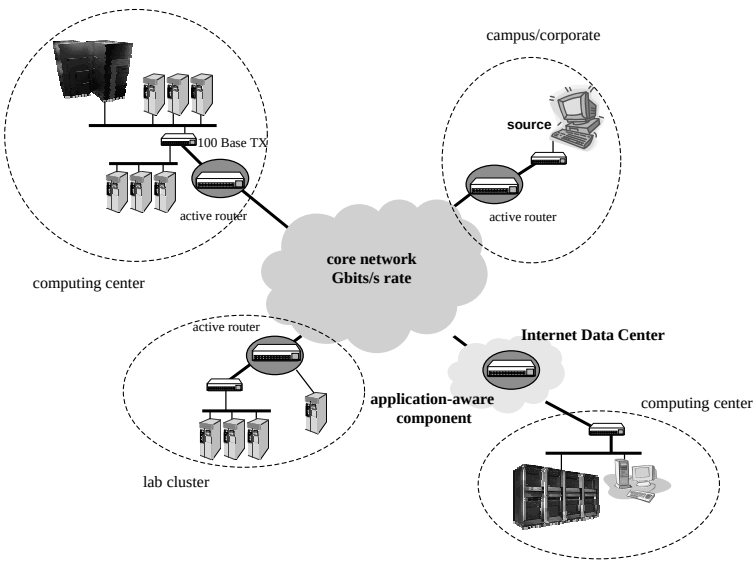
FROM DUMB LINKS CONNECTING COMPUTING RESOURCES

TO COLLABORATIVE RESOURCES



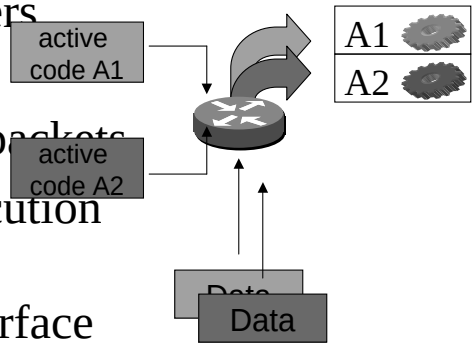
The network can work together with the applications to provide in-network processing functions

Application-Aware Infrastructure on Grids

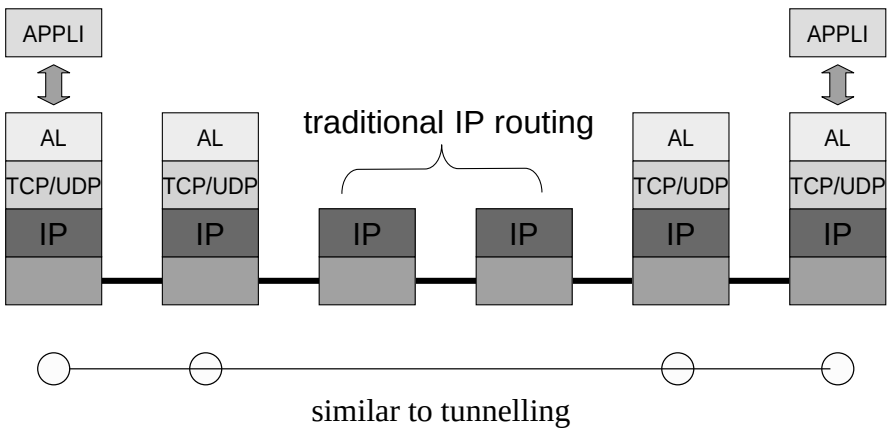


Application-Aware Components AAC

- Based on programmable active nodes/routers
- Customized computations on packets
- Standardized execution environment and programming interface



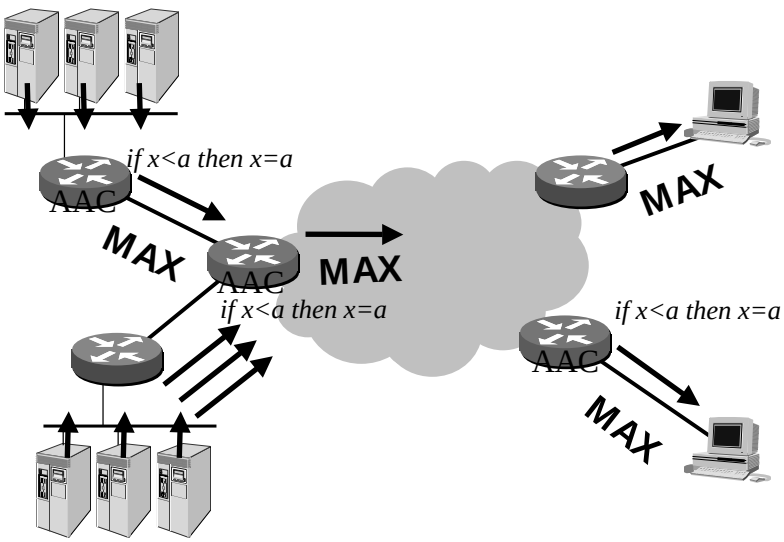
Interoperability with legacy routers



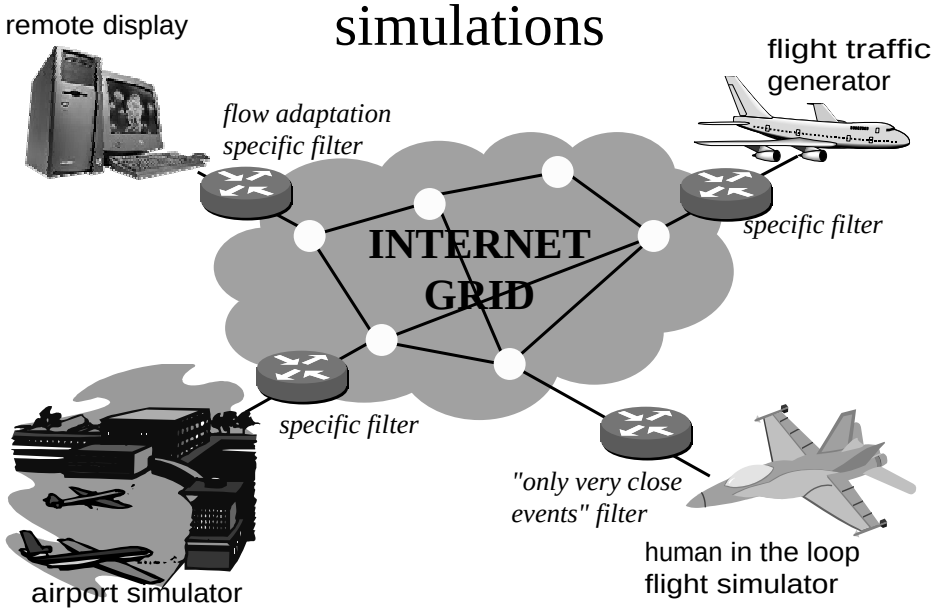
Deploying new services

- Collective/gather operations
- Interest management, filtering (DIS, HLA)
- On-the-fly flow adaptation (compression, layering…) for remote displays
- Intelligent directory services
- Distributed, hierarchical security system
- Distributed Logistical Storage
- Custom QoS policy

Ex: Collective operations *max computation*



Ex: Wide-area interactive simulations

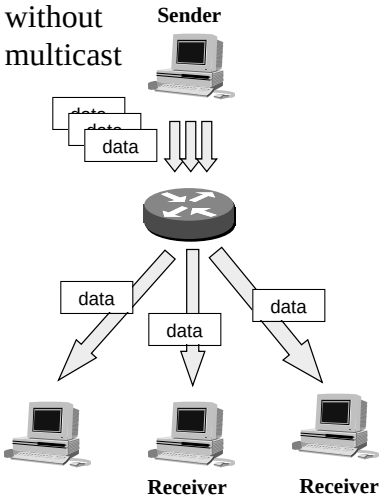


Deploying reliable multipoint data distribution services

- For
  - Database accesses, sharing, replications
  - Data and code transfert, massively parallel job submissions (task-farm computing)
  - Distributed applications (MPI…)
  - Interactive applications (DIS, HLA…)
- Desired features
  - scalable
  - low latencies

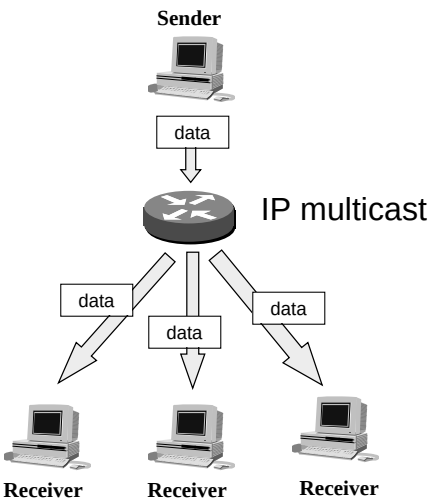
Deploying reliable multipoint data distribution services

- For
  - Database accesses, sharing, replications
  - Data and code transfert, massively parallel job submissions (task-farm computing)
  - Distributed applications (MPI…)
  - Interactive applications (DIS, HLA…)
- Desired features
  - scalable
  - low latencies



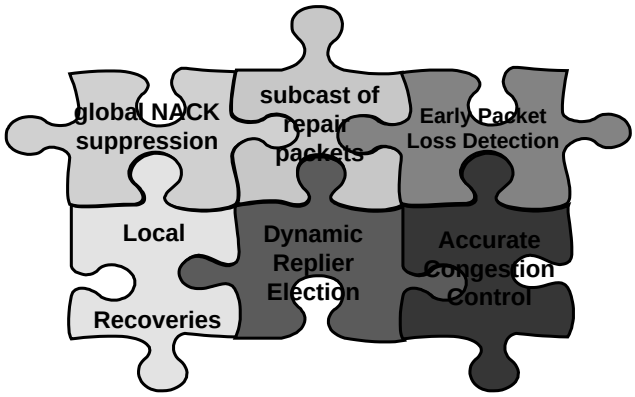
Deploying reliable multipoint data distribution services

- For
  - Database accesses, sharing, replications
  - Data and code transfert, massively parallel job submissions (task-farm computing)
  - Distributed applications (MPI…)
  - Interactive applications (DIS, HLA…)
- Desired features
  - scalable
  - low latencies

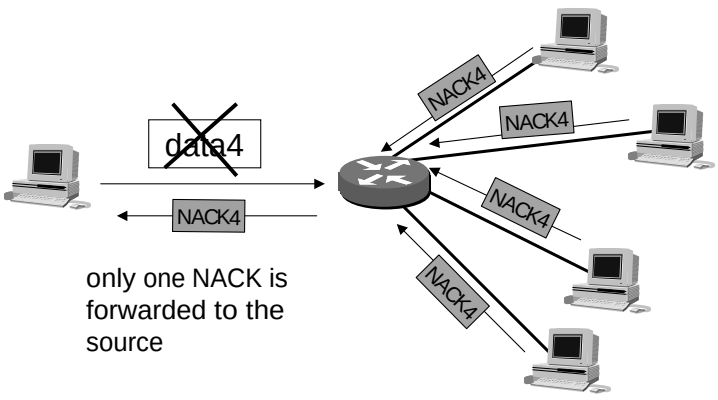


DyRAM

Protocol with modular services for achieving reliability, scalability and low latencies



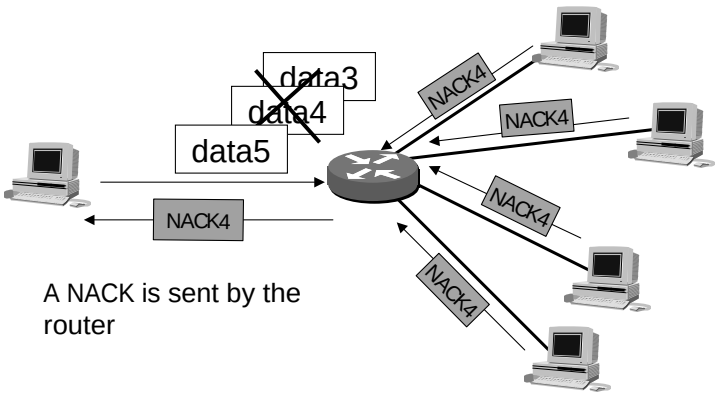
Ex: Global NACKs suppression



only one NACK is forwarded to the source

Ex: Early lost packet detection

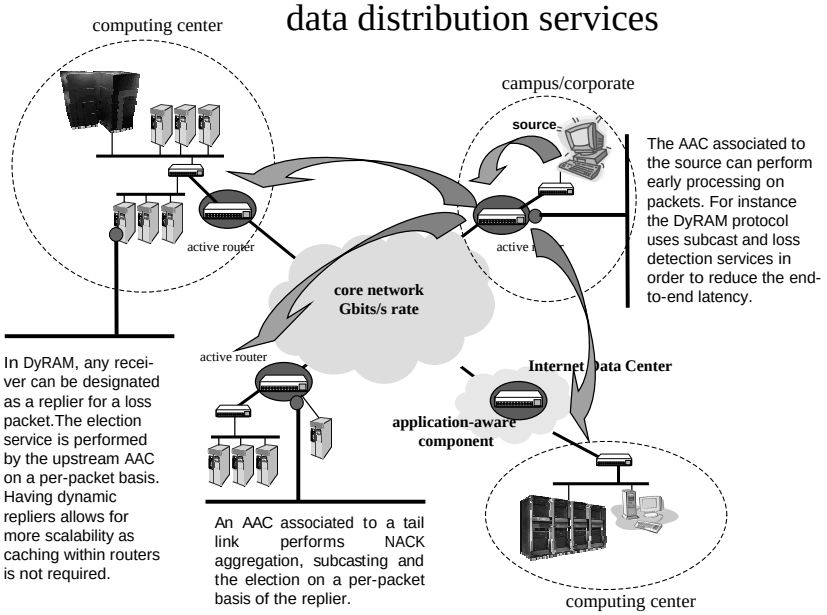
The repair latency can be reduced if the lost packet could be requested as soon as possible



A NACK is sent by the router

These NACKs are ignored!

Deploying reliable multipoint data distribution services



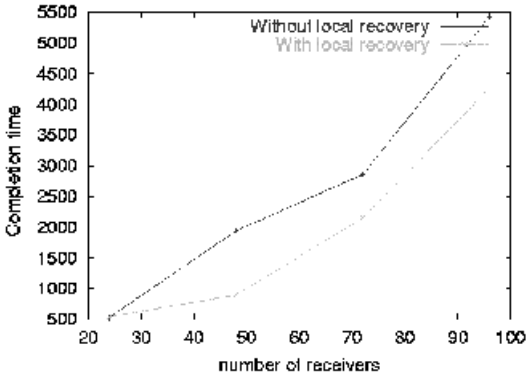
The AAC associated to the source can perform early processing on packets. For instance the DyRAM protocol uses subcast and loss detection services in order to reduce the end-to-end latency.

In DyRAM, any receiver can be designated as a replier for a loss packet.The election service is performed by the upstream AAC on a per-packet basis. Having dynamic repliers allows for more scalability as caching within routers is not required.

An AAC associated to a tail link performs NACK aggregation, subcasting and the election on a per-packet basis of the replier.

Local recovery & replier election

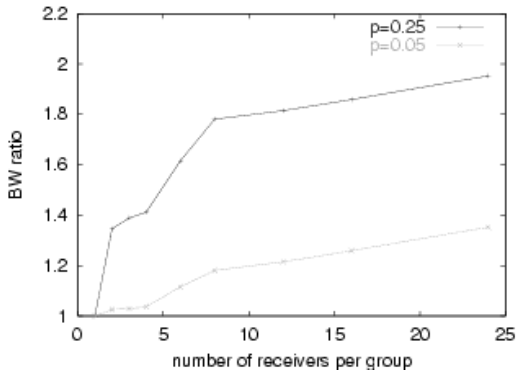
4 receivers/group



grp: 6…24 p=0.25

Local recoveries reduces the end-to-end delay (especially for high loss rates and a large number of receivers).

Local recovery & replier election

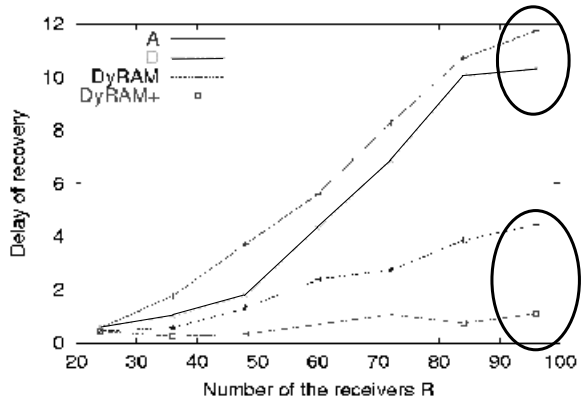


48 receivers distributed in g groups ➡ # grp: 2…24

As the group size increases, doing the recoveries from the receivers greatly reduces the bandwidth consumption

Early Packet Loss Service

4 receivers/group # grp: 6…24



EPLD is very beneficial to DyRAM

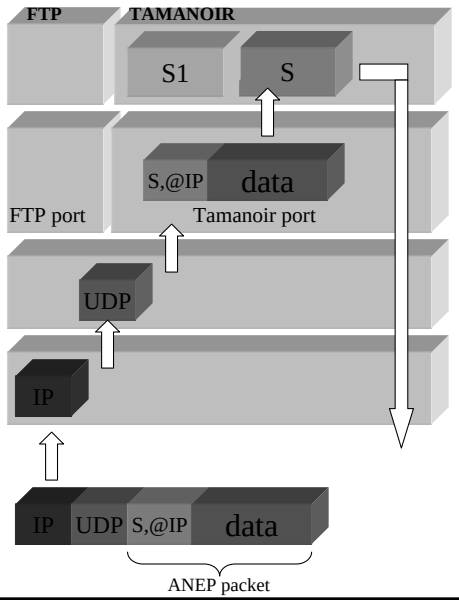
p=0.25

DyRAM implementation

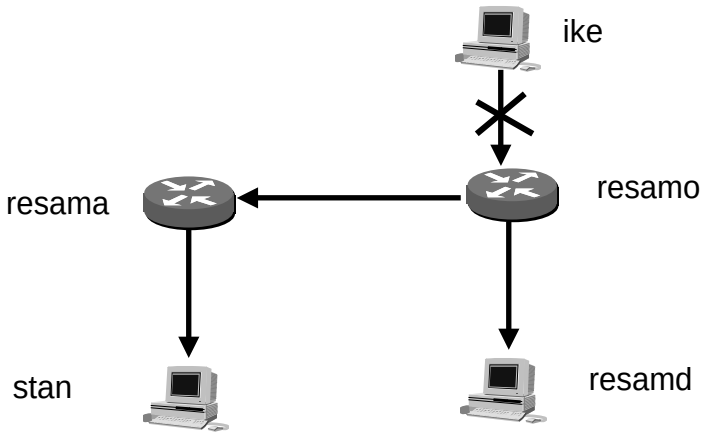
testbed configuration

- TAMANOIR active execution environment
- Java 1.3.1 and a linux kernel 2.4
- A set of PCs receivers and 2 PC-based routers (Pentium II 400 MHz 512 KB cache 128MB RAM)
- Data packets are 4 KBytes

The data path

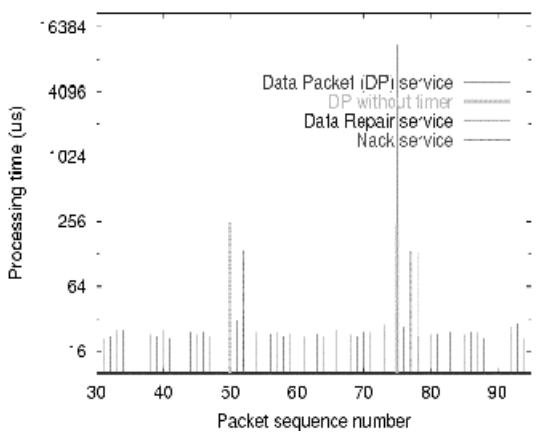


Cost of Data Packet Services

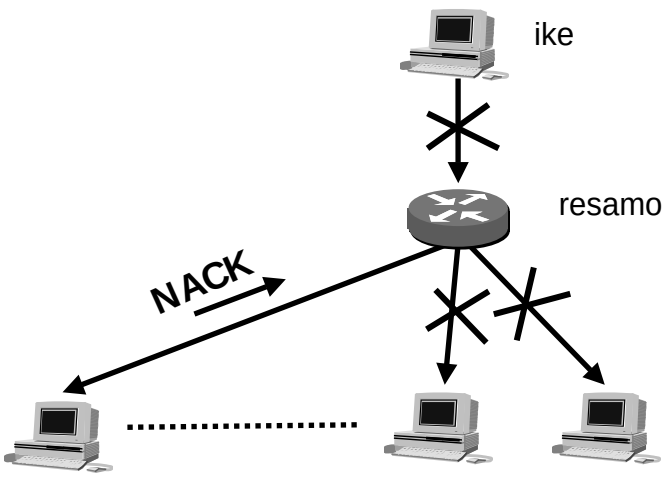


Cost of Data Packet Services

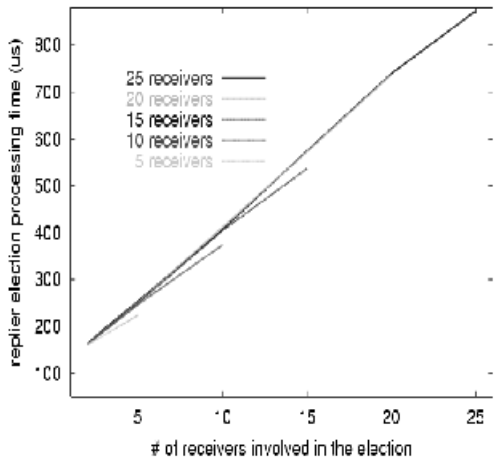
- NACK: 135µs
- DP : 20µs if no seq gap, 12ms-17ms otherwise. Only 256µs without timer setting
- Repair: 123µs



Cost of Replier Election



Cost of Replier Election



The election is performed on-the-fly.

It depends on the number of downstream links.

Costs range from 0.1 to 1ms for 5 to 25 links per router.

Conclusions (1)

- Grids can be more than end-host computing resources interconnected with network links
- High-bandwidth links is not enough to provide E2E performances for distributed, interactive applications
- Application-aware components can be deployed to host high-value services
- In-network processing functions can make grids more responsive to applications' needs

Conclusions (2)

- The paper shows how an efficient multipoint service can be deployed on an application-aware infrastructure
- Simulations and experimentations shows that low latencies can be obtained with the combination and collaboration of light and simple services