

Applying CORBA Fault Tolerant Mechanisms to Network Management

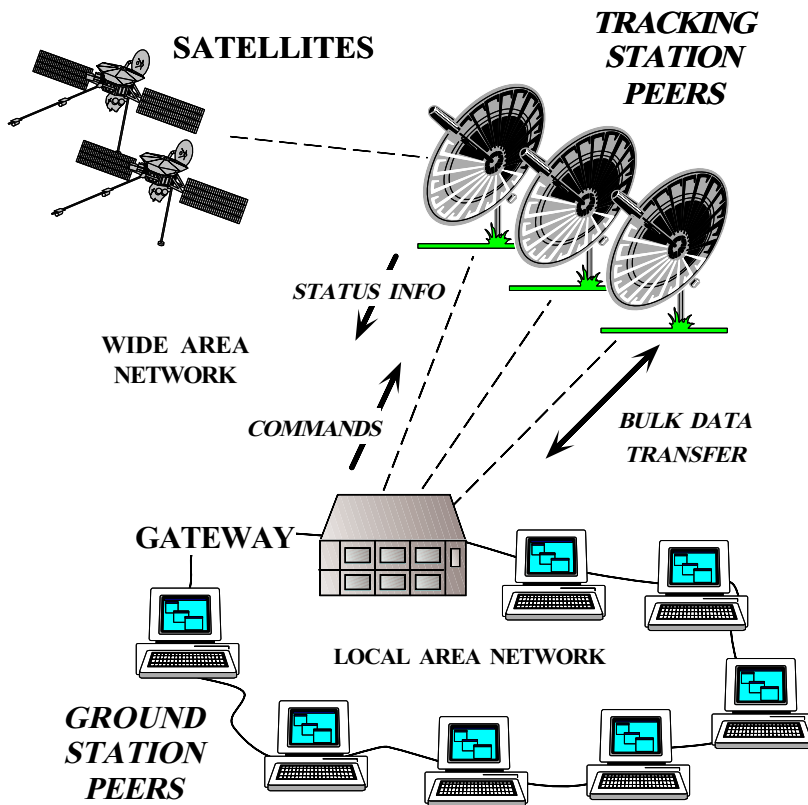
Aniruddha Gokhale
Shalini Yajnik
Bell Laboratories
Lucent Technologies



Douglas Schmidt
B. Natarajan, F. Kuhns, and C. O’Ryan
Distributed Object Computing Group
Washington University at St. Louis



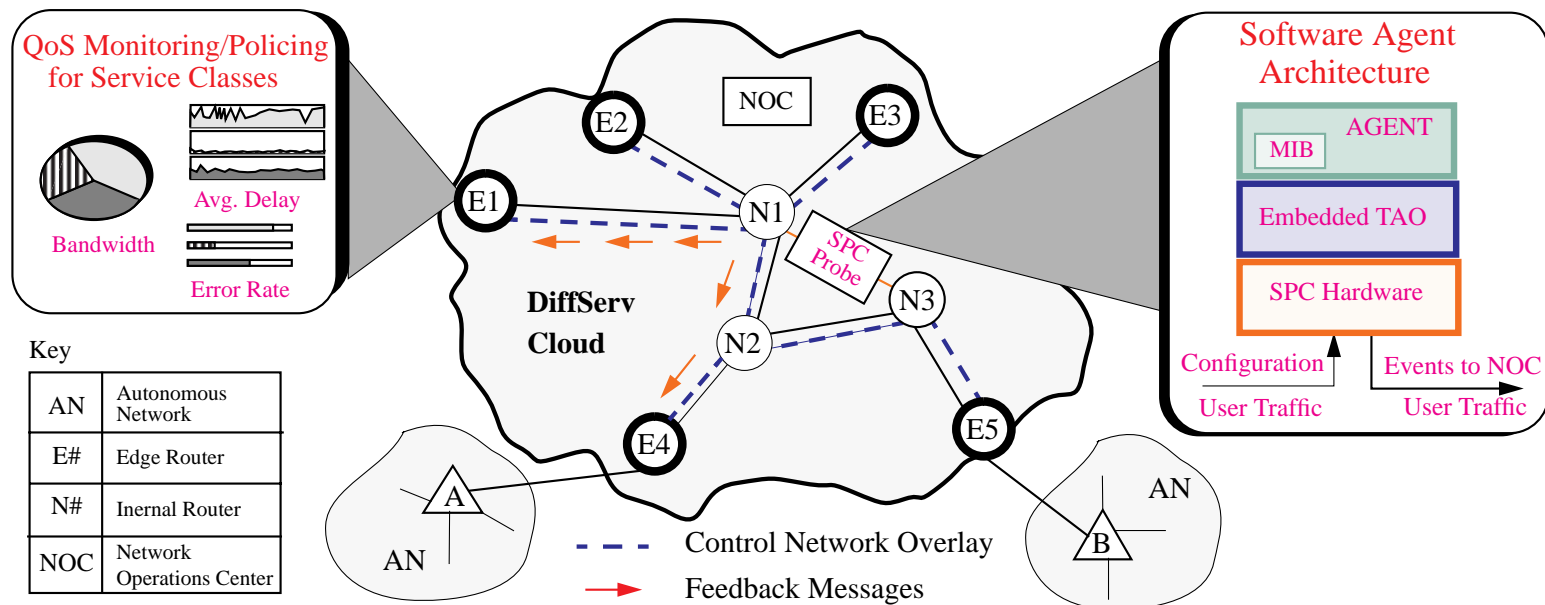
Typical Telecommunications Applications



Common Requirements

- **Network Management**
 - QoS Monitoring
 - Network visualization and control
- **High Availability**
 - 99.999% typical
 - Achieved through monitoring and redundancy

Network Management - The Big Picture



● Problems

- QoS Monitoring
- Data collection, Control and Visualization

● Solution

- DOVE

Fault Tolerance

- **Problems**

1. Developing highly reliable software
2. Emphasis on components and applications for telecom

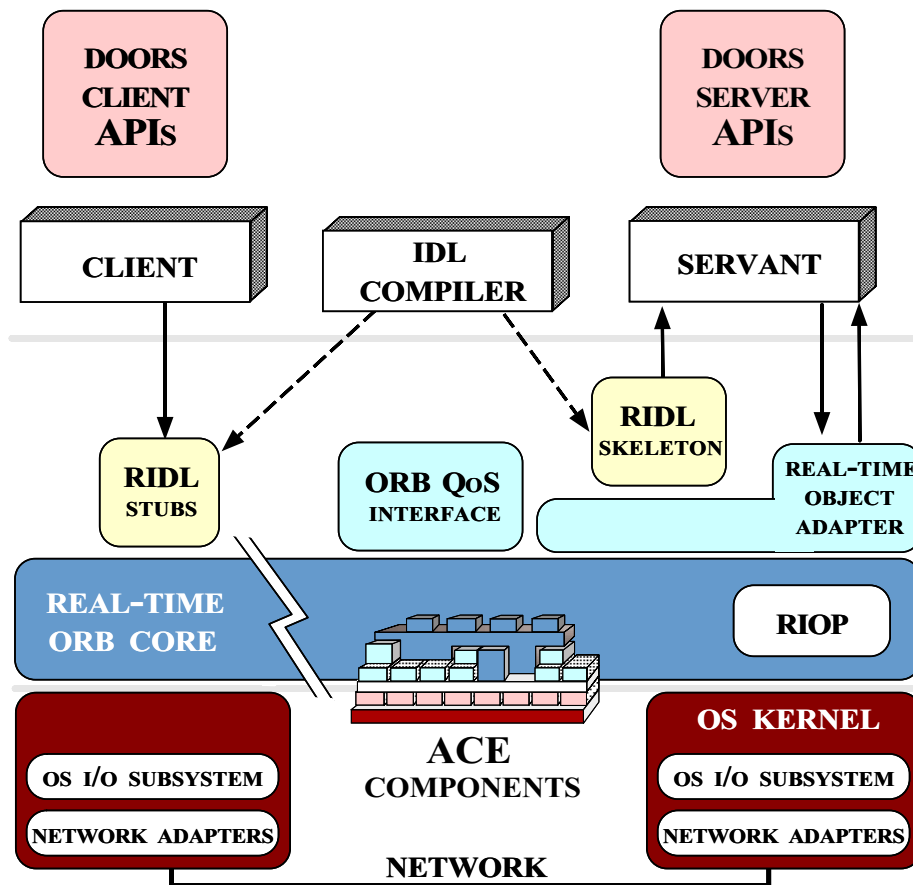
- **Forces**

1. Standard COTS middleware
2. Rapid development cycles
3. Multi-platform support (heterogeneity)

- **Solution**

- **ACE/TAO/DOORS integration**
-

Integrated ACE/TAO/DOORS Platform



DOORS →

- A CORBA service that supports fault-tolerant CORBA Objects

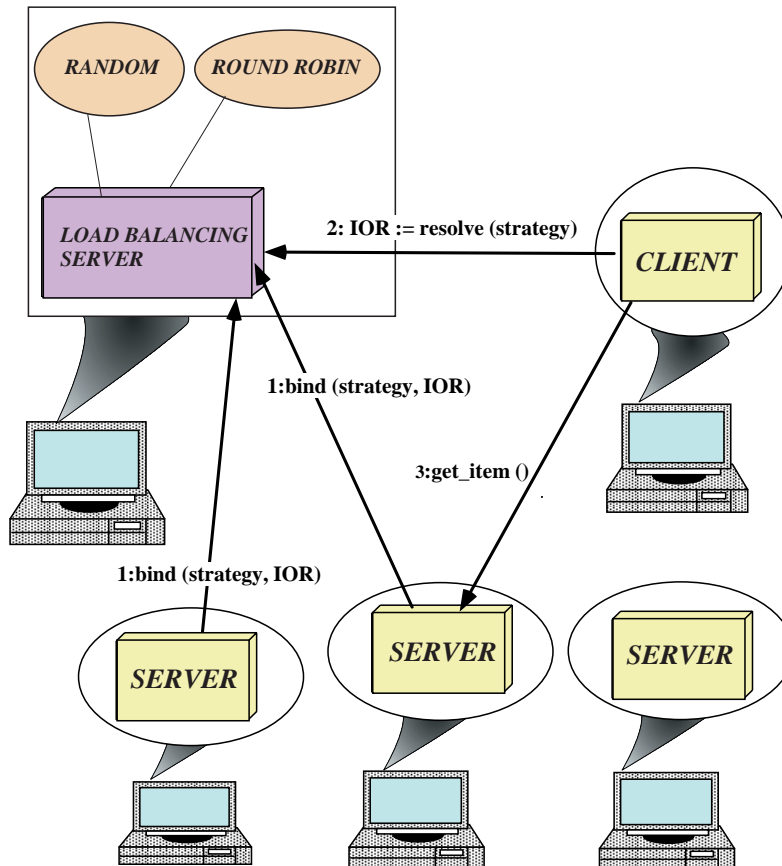
TAO (The ACE ORB) →

- High-performance, real-time CORBA-compliant ORB

ACE →

- Portable OO framework for communications software
- Implements core *patterns*

Example Application: Load Balancing Service



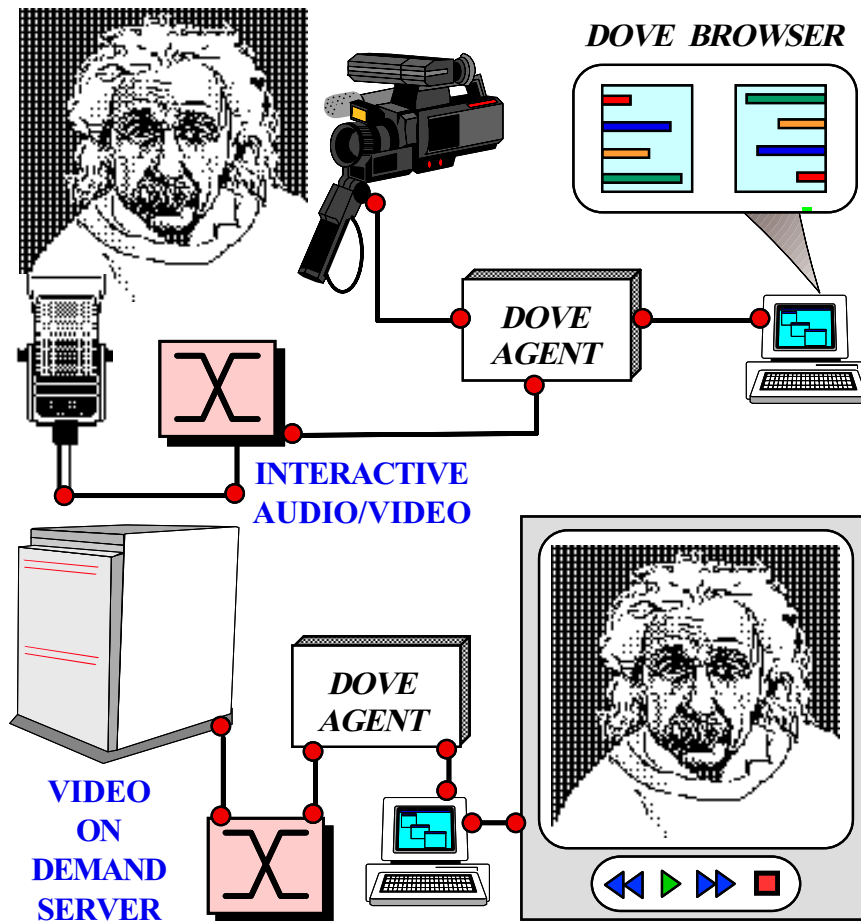
Service Overview

- Improve overall application performance by balancing load
 - *e.g.*, supports both *random* and *round-robin* strategies

Fault Tolerance Issues

- Single point of failure
- Replication for high availability
- Failure detection/recovery

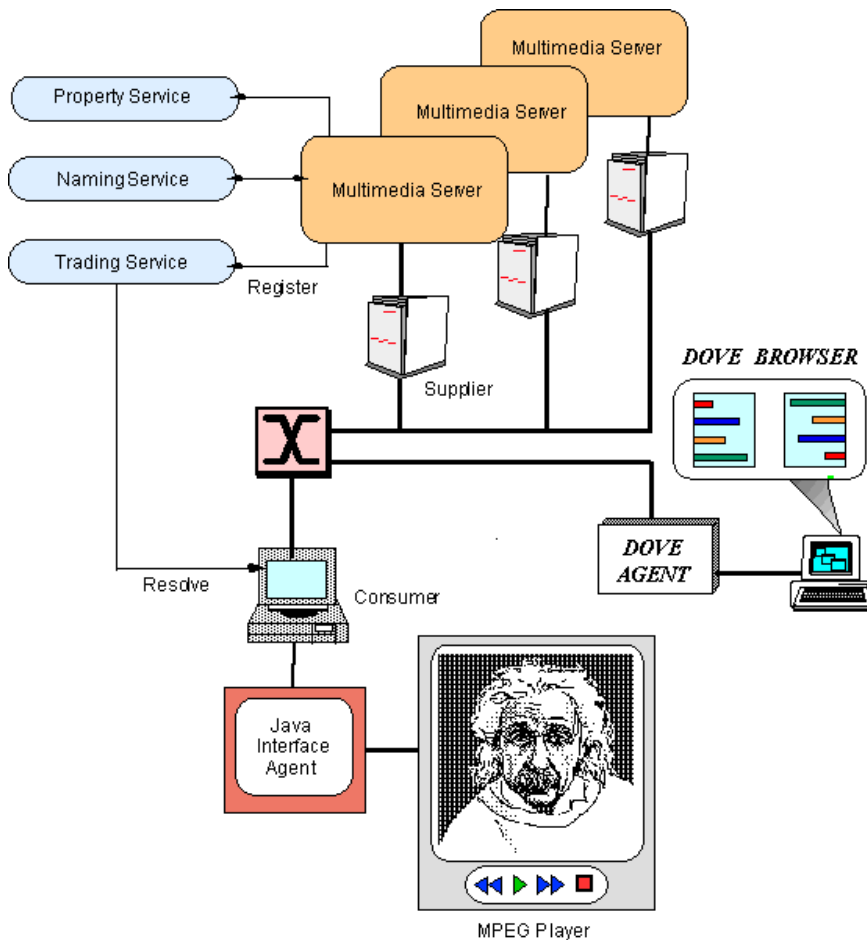
Example Application Architecture: DOVE



Distributed Object Visualization Environment (DOVE)

1. QoS performance monitoring
2. Fault tolerance
3. Real-time processing
4. Multi-platform

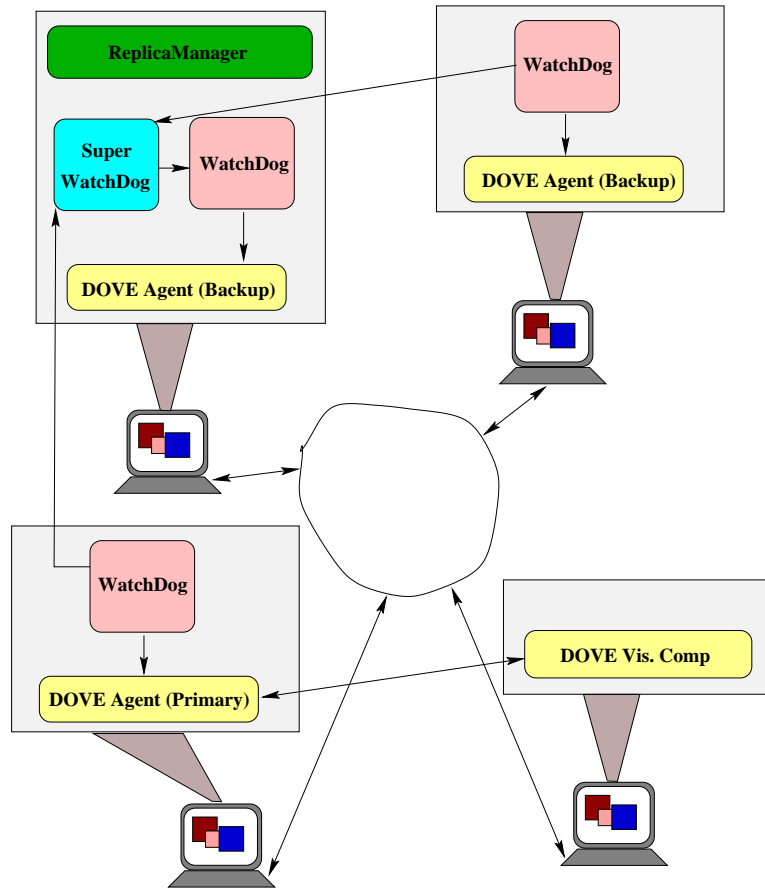
Application Management with DOVE



Multimedia Service Management

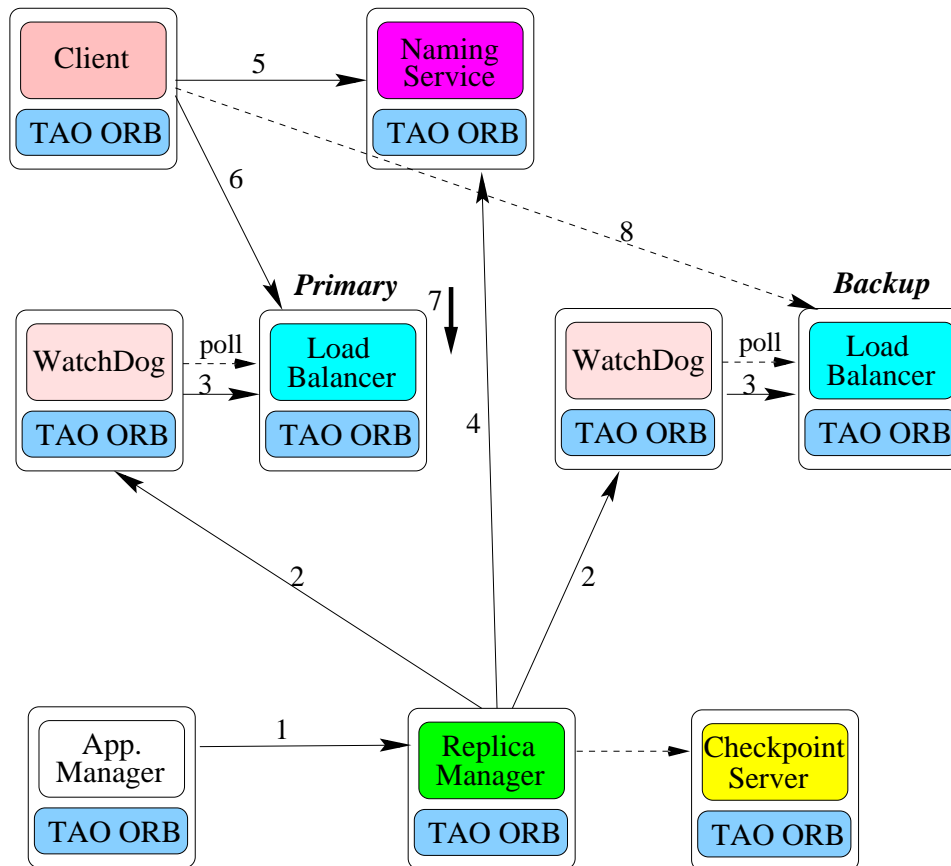
1. Integrated application management
2. DOVE agent
 - Event propagation
3. DOVE Browser
 - Visualize server performance
 - Customizable visualization
4. Fault tolerance through DOORS

Fault Tolerance using DOORS



- **ReplicaManager**
 - Replica configuration and management
- **WatchDog and SuperWatchDog**
 - WatchDog - Local failure detection – *polling, heartbeat*
 - SuperWatchDog - Host-level detection
 - Failure notification
- **Client side failure handling**
 - Transparent switchover from Primary to Backup

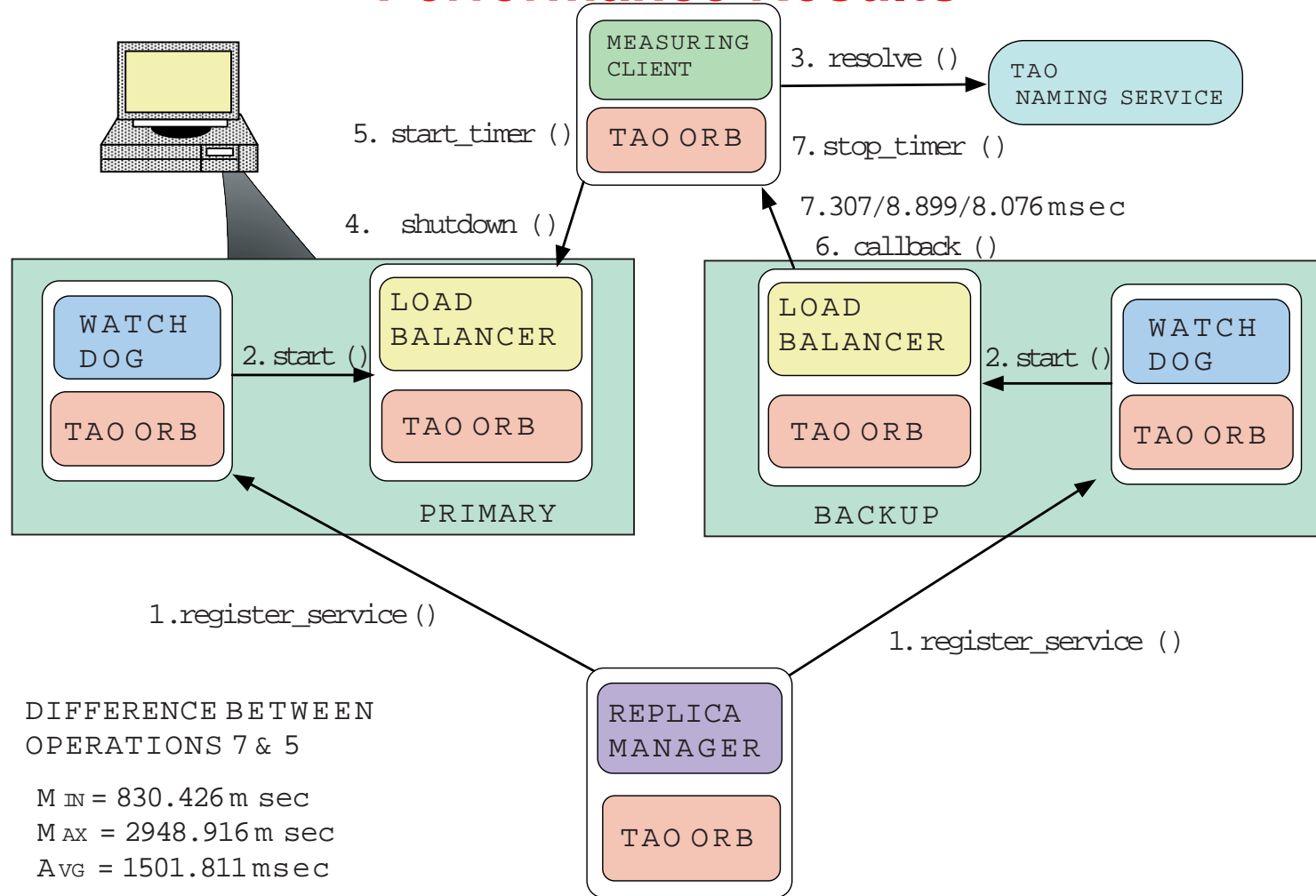
Applying DOORS to Load Balancing Service



• Usage

1. Application registered with RM
2. RM registers appl. with WatchDog
3. WD starts Appl. and polls
4. RM generates IOR and binds to Name Service
5. Client resolves IOR from Name Service
6. Client talks to Primary
7. Primary fails
8. Client transparently talks to Backup

Performance Results

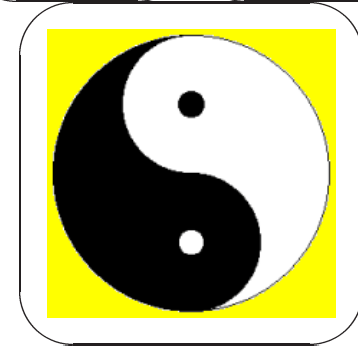
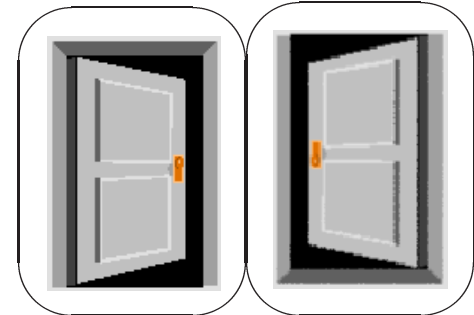


DIFFERENCE BETWEEN OPERATIONS 7 & 5

M_{IN} = 830.426 m sec
 M_{AX} = 2948.916 m sec
 AVG = 1501.811 msec

Ongoing and Future Work

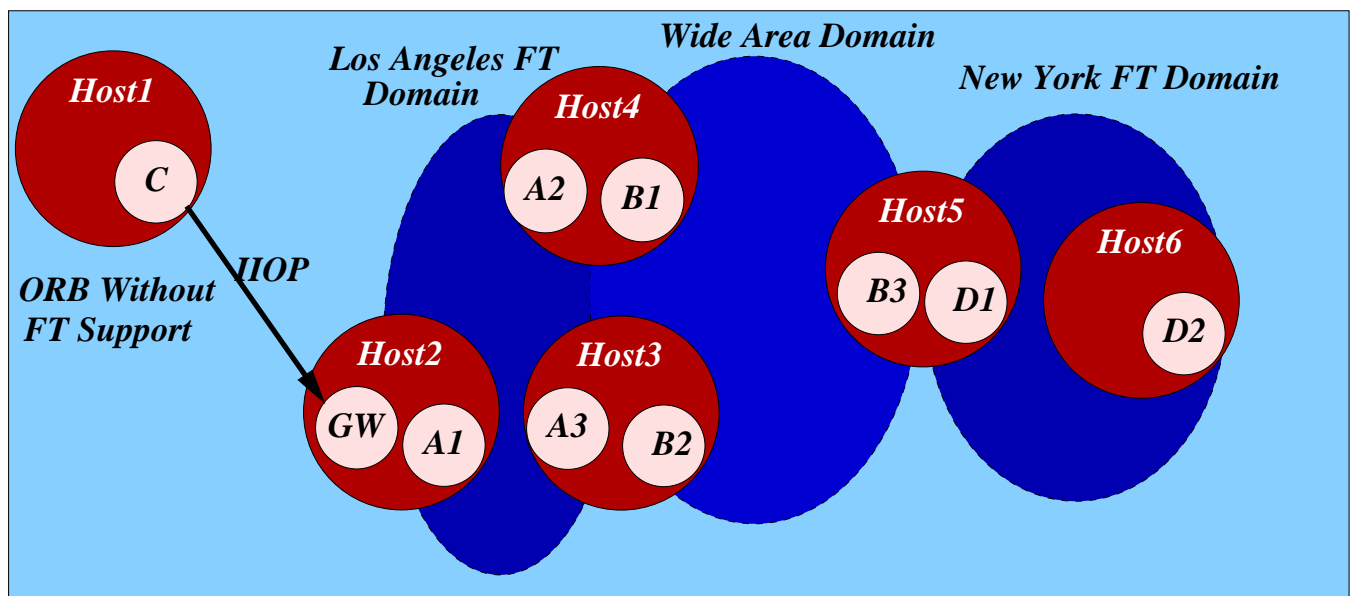
- Systematic empirical benchmarks to measure impact of fault tolerance on efficiency and predictability
- Integrate DOORS/TAO with a variety of telecom applications
- Enhance DOORS and TAO to conform to forthcoming OMG CORBA fault tolerance spec (November 1999)
- Solicit feedback from Lucent Business Units



Overview Of CORBA Fault Tolerance

Objectives

- Wide range of fault tolerance
 - Simple low cost clients
 - Highly reliable servers
- Passive and Active replication
- Automatic and application-controlled FT mechanisms
- Strong replica consistency



[www.omg.org/techprocess/meetings/schedule/
Fault_Tolerance_RFP.html](http://www.omg.org/techprocess/meetings/schedule/Fault_Tolerance_RFP.html)

Concluding Remarks

- **Summary**

- Cost-effective reliability is important to next-generation telecom applications
- Benefits of DOORS/TAO integrated framework
 - * Provides the desired fault-tolerance
 - * Enables rapid application development
 - * Uses standards-based COTS CORBA middleware

- **URLs**

- `www.cs.wustl.edu/~schmidt/TAO.html`
 - `www.cs.wustl.edu/~schmidt/nmvc.html`
 - `www.cs.wustl.edu/~schmidt/dove.html`
 - `www.bell-labs.com/~shalini/papers/doors97.doc`
-