Implementation of a Fault-Tolerant Real-time Event Channel

Chris Gill & Huang-Ming Huang

Dept. of Computer Science and Engineering

Washington University

One Brookings Drive

St. Louis, MO 63130

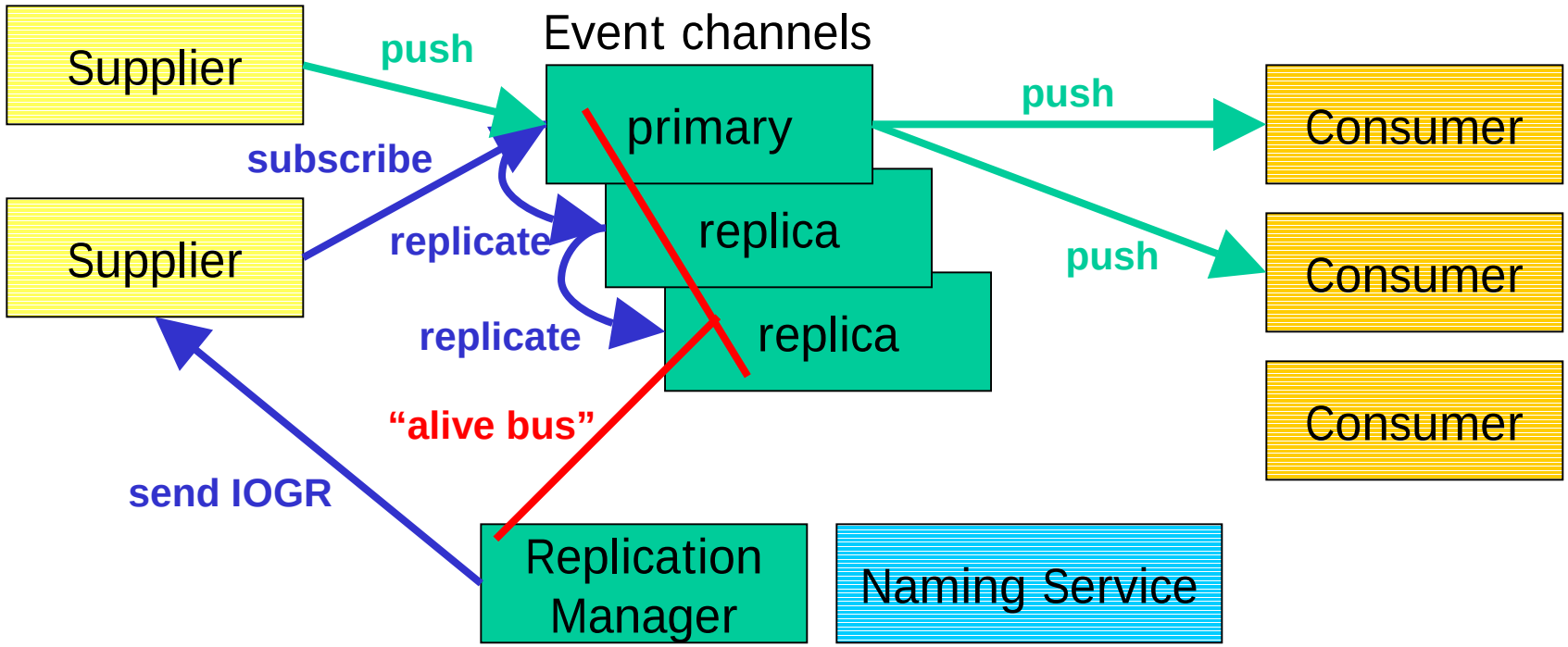
{cdgill,hh1}@cse.wustl.edu

This work was supported in part by the DARPA PCES program, contracts F33651-01-C-1847 and F33615-03-C-4111

OMG Real-time and Embedded Systems Workshop

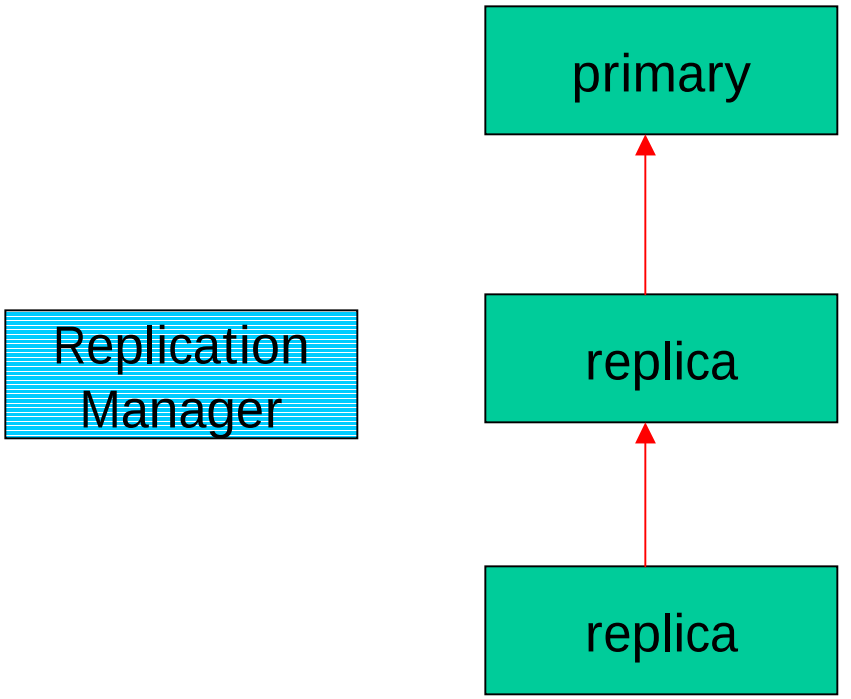
Reston, VA, USA July 14, 2004

FT/RT Event Channel (FTRTEC)



- Provide fault-tolerance (fail-stop) within real-time constraints
- Offer useful configuration knobs, *e.g.*, to Quality Connectors
  - Replicas: where and how many, transactional replication depths
- Service-level implementation
  - Less dependent on ORB-level FT features (mainly need IOGR abstraction)

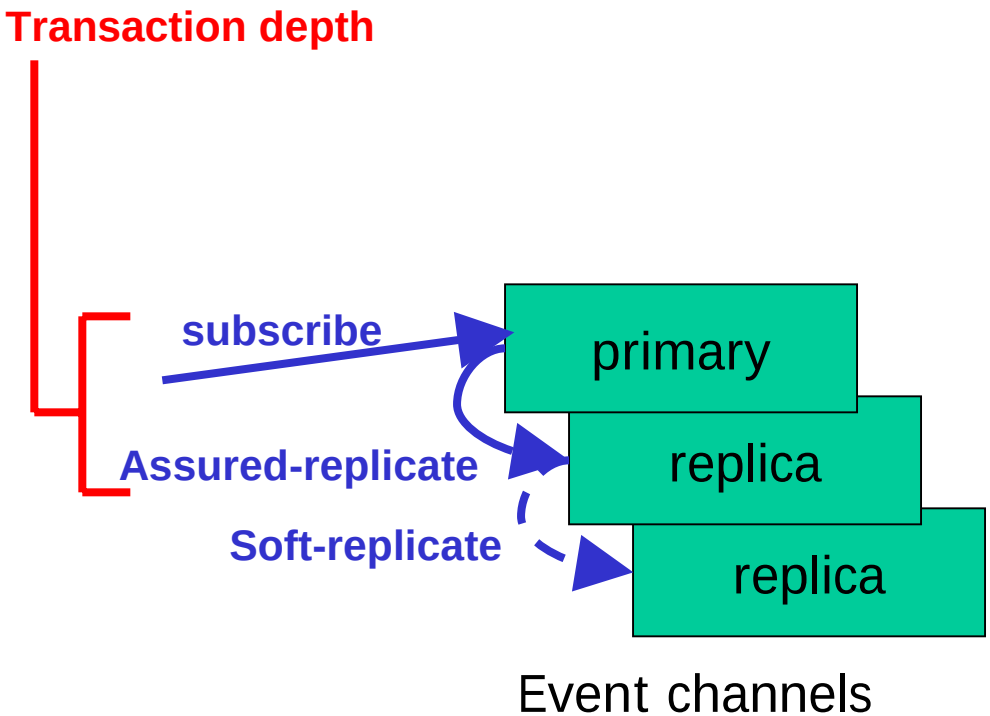
FTRTEC Fault-Detection and Fail-Over



- Maintain connections
  - Connected == alive
  - Among primary/replicas
  - To replication manager service
  - Current implementation uses TCP
- Communication of replica updates
  - Must be transactional
    - To given depth of replication
  - But need not be synchronous
    - CORBA AMI or two-way calls ok
    - But not (unreliable) one-ways

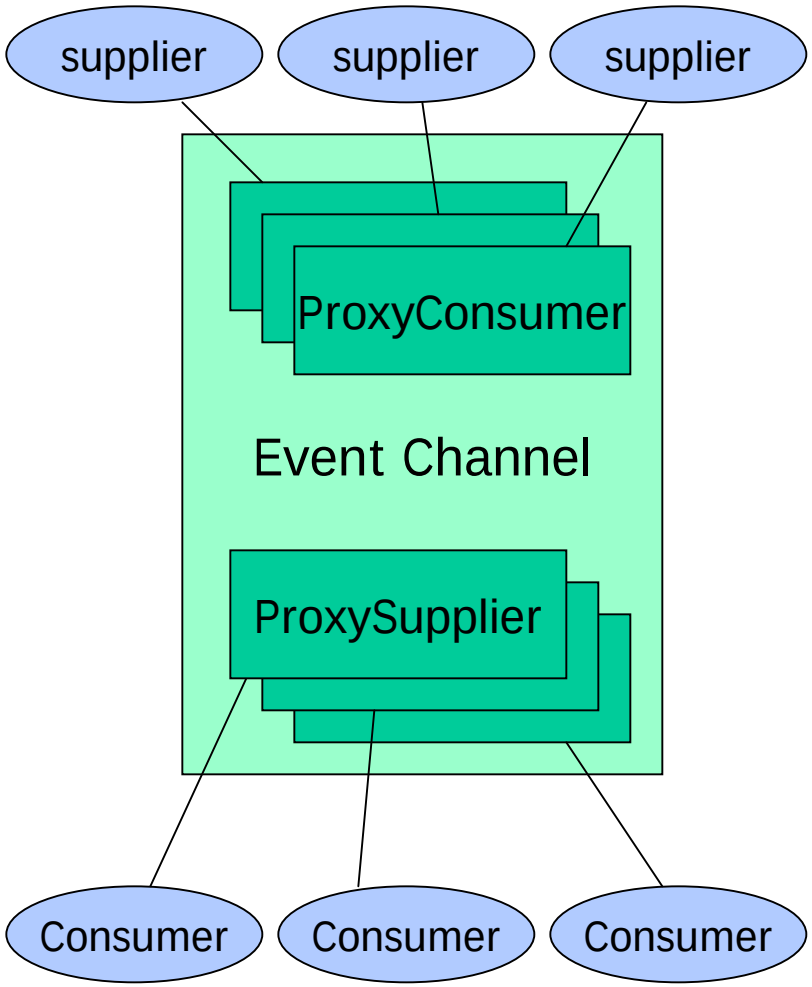
Replicated Event Channels

Subscription Replication Trade-Offs



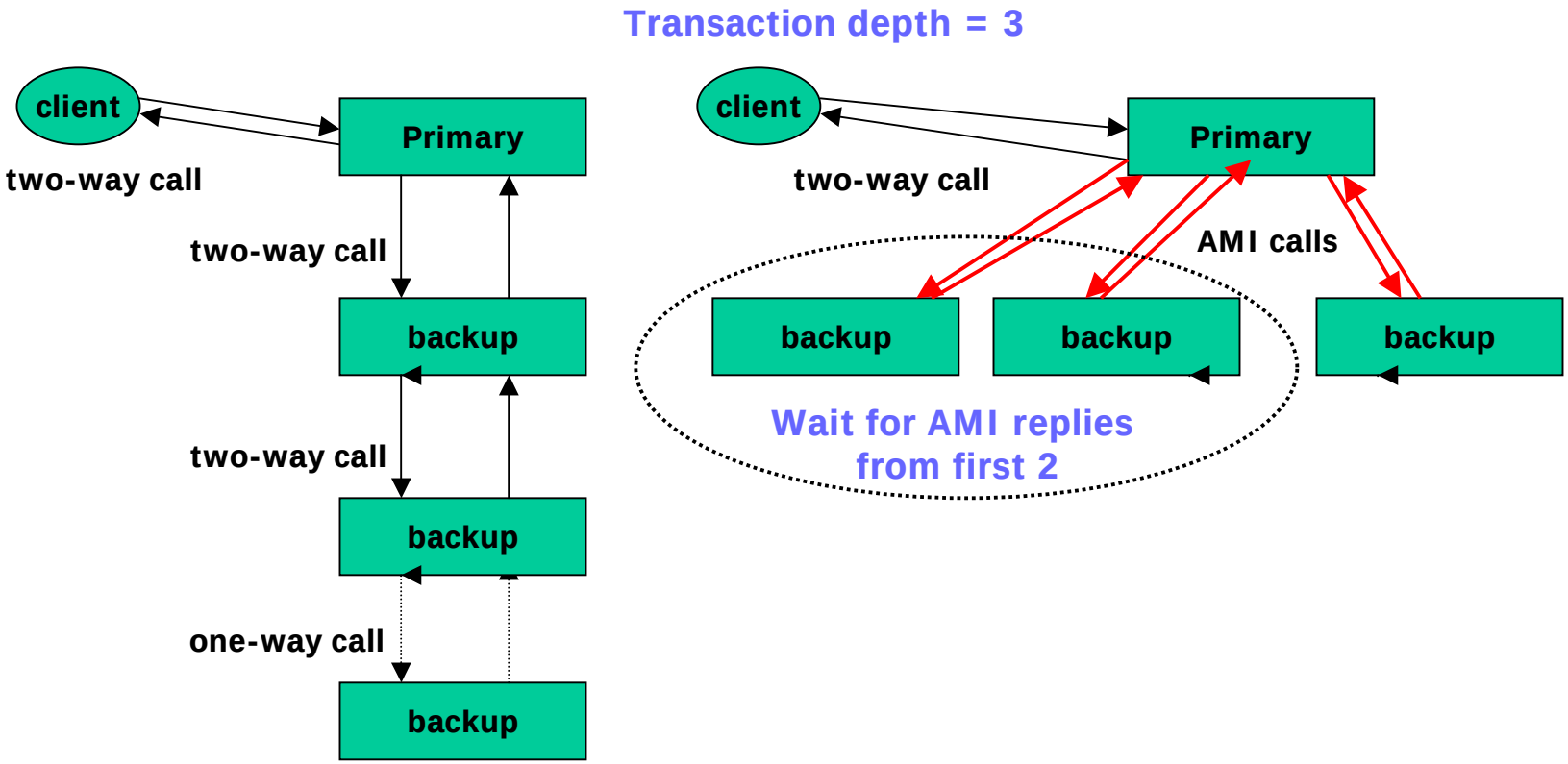
- Risk vs. Blocking times
  - Transaction depth to tradeoff reliability and responsiveness
  - Requires two phase protocol for all replicated objects
  - Use two-way or AMI for assured-replication
    - To specified depth
  - Use oneway operations for soft-replication
    - Beyond specified depth

Message vs. State Replication



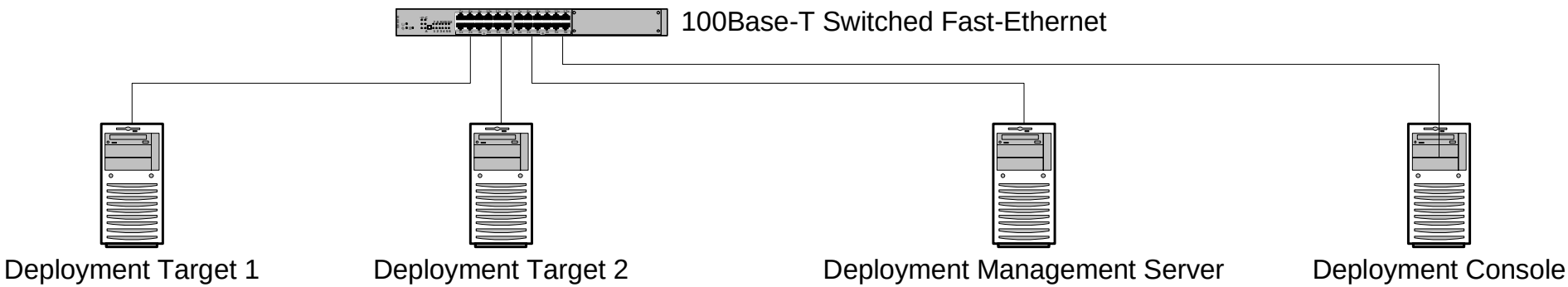
- Multi-faceted Objects:
  - Facade to flatten representation
  - Message-based replication
    - At object, not ORB level
- Transient vs. persistent state
  - Only replicate subscriptions currently
    - Use transactions for assurance
    - Protects the event *stream*
    - *Even during subscription*
  - Protecting events is plausible
    - Using redundant paths
    - But would require split/join semantics
    - Potential area of future work

Two-Way vs. AMI Calls for Replication



- Initial two-way call from client to primary
- Two-way calls to the next replica result in sequential processing of updates
- AMI calls to replicas from primary instead allow overlapped processing of updates
- Offers replication speed-up for transaction depth > 2

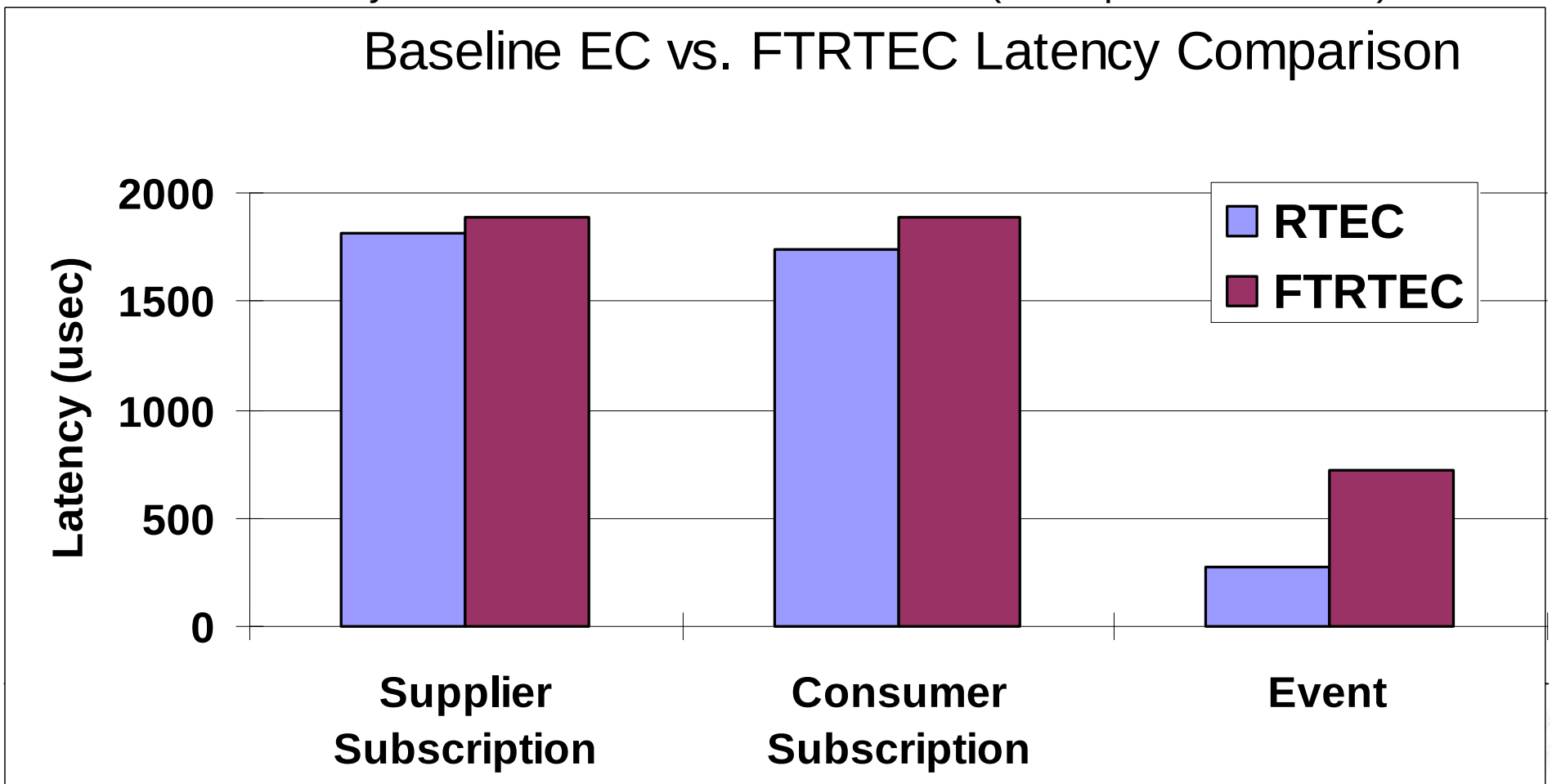
Experimental Testbed



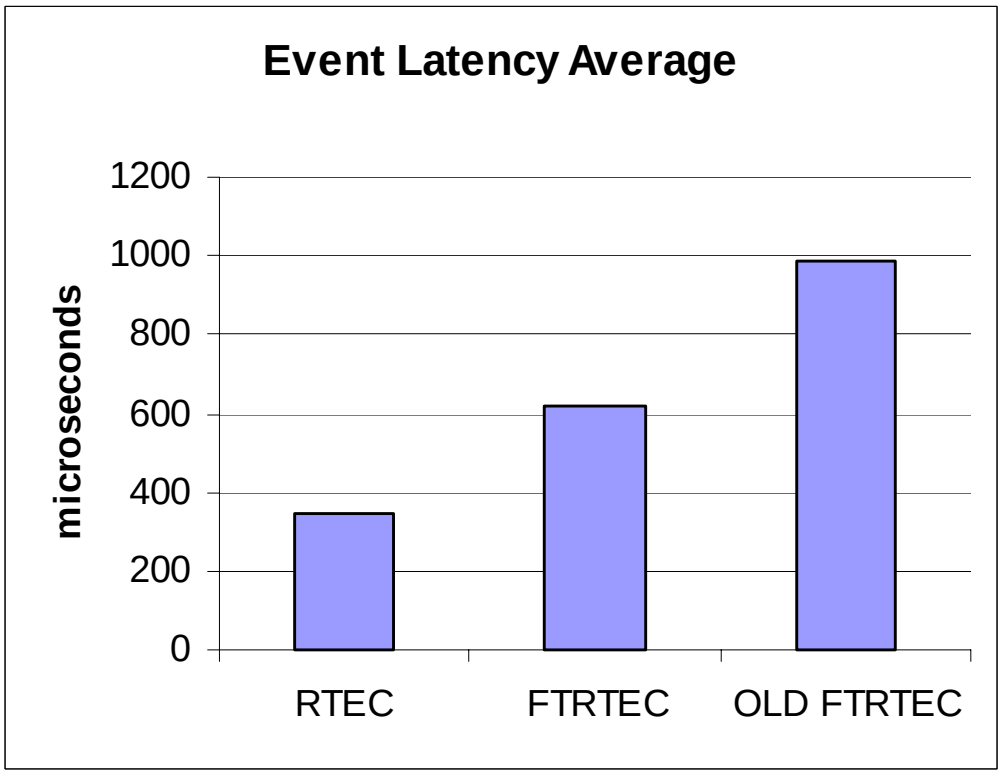
- 2 Pentium-IV 2.5 GHz machines with 500MB RAM, 512KB cache
- 2 Pentium-IV 2.8 GHz machines with 500MB RAM, 512KB cache
- KURT-Linux 2.4.18
- 100 Base-T Ethernet, isolated network for experiment runs
- ACE version 5.3.5 / TAO version 1.3.5 (pre-release version)
- Experiments were run as root in real-time scheduling class

Original FTRTEC vs. RTEC Benchmarks

- Metrics: event throughput, subscription latency (TAO RTEC is baseline)
- TAO FT/RT EC vs. TAO RTEC, no replicas for FTRTEC
- 2 2.8 GHz Pentium boxes, KURT-Linux, Ethernet
- Suppliers/consumers on same machine, EC on a separate machine
- FTRTEC two-way event overhead/fail-over trade-off (AMI optimization WIP)



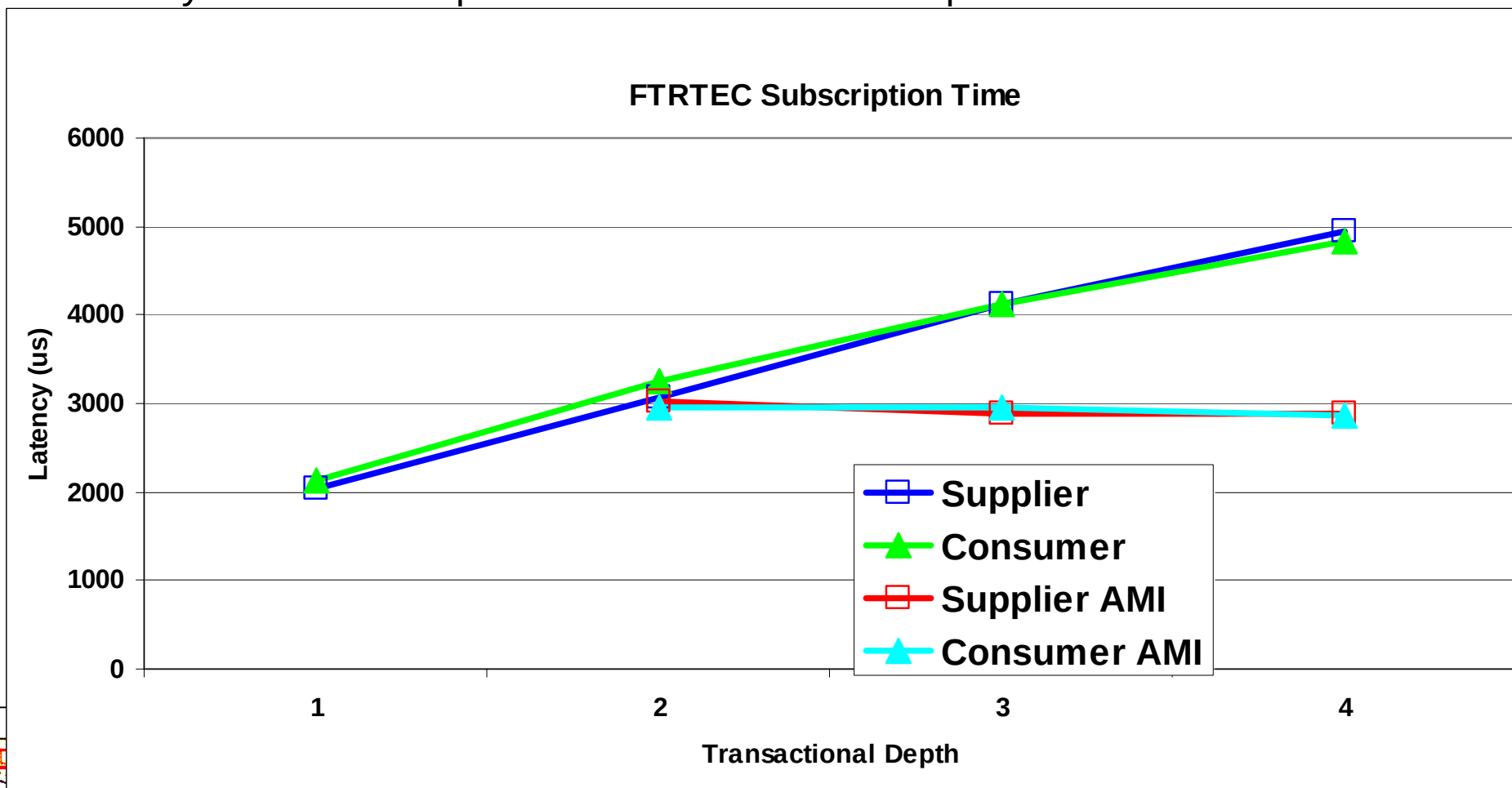
Current FTRTEC vs. RTEC Latency Benchmark



- We made several key FTRTEC improvements
  - Client EC gateway uses a separate ORB (event pushes bypass IOGR interceptors)
  - Moved IOGR service context processing to not impact events
  - Removed a redundant service context id check from FTCORBA core
- And re-ran experiments with pre-release TAO 1.4

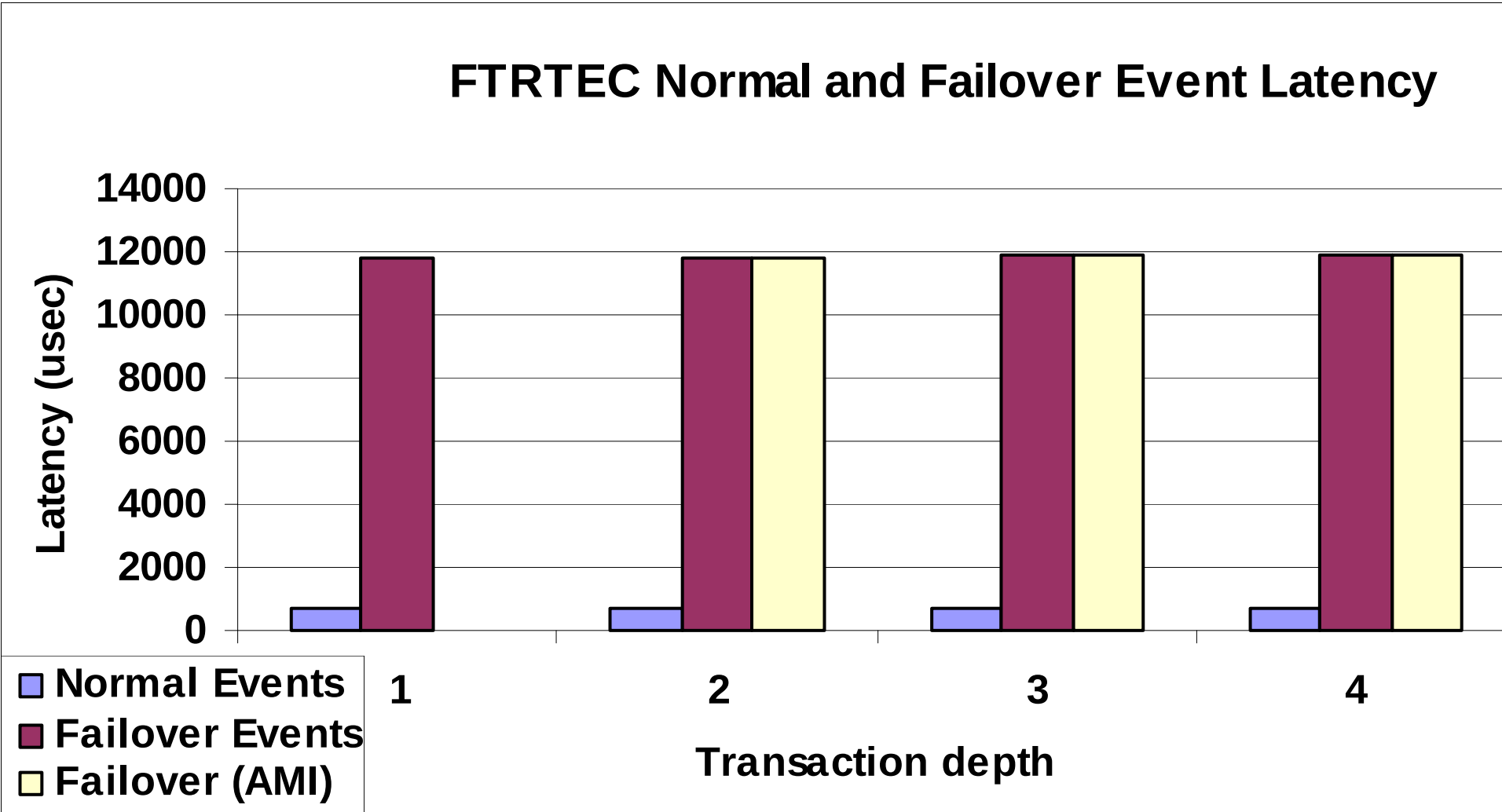
Subscription Time Scalability

- Metrics: subscription latency/depth with two-way vs. AMI replication
- 2 2.8 GHz + 2 2.6 GHz Pentium boxes, KURT-Linux, Ethernet
- AMI/two-way active replication from primary to replicas up to transaction depth
- One-way semi-active replication after transaction depth



FTRTEC Normal, Fail-Over Event Latency

- Metrics: fail-over latency versus normal event latency
- 2 2.8 GHz & 2 2.6 GHz Pentium boxes, KURT-Linux, Ethernet
- Approximately a factor of 15 greater latency for fail-over (still, > 80 Hz)



Concluding Remarks

- Using CORBA AMI optimizes primary to replica updates
  - Exploits overlapped processing of updates
  - While still allowing transactional semantics
- Able to trade-off real-time performance with fault-tolerance
- Removing replication mechanisms from event paths is key
- FT/RT Event Channel is available open-source
  - ACE version 5.3.5 / TAO version 1.3.5 and later
- FT/RT EC integrated into LMCO MINERS demo in December
- Thanks to Joe Cross (DARPA), and Sylvester Fernandez and David Mattioli (LMCO)
- Thanks also to Andy Gokhale, Bala Natarajan, and Doug Schmidt at Vanderbilt University