Wireless CORBA

Douglas C. Schmidt

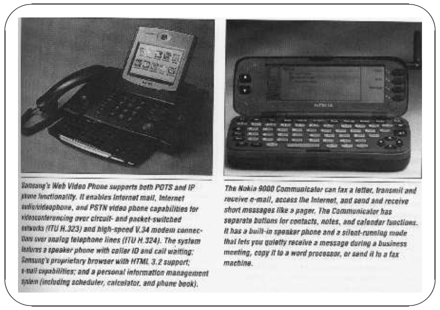
Associate Professor & — Computer Science Dept.
www.cs.wustl.edu/∼schmidt/ — Washington University, St. Louis

Sponsors

NSF, DARPA, Bellcore/Telcordia, BBN, Boeing, CDI/GDIS, Experian, Global MT, Hughes, Kodak, Krones, Lockheed, Lucent, Microsoft, Motorola, Nokia, Nortel, OCI, OTI, Raytheon, SAIC, Siemens SCR, Siemens MED, Siemens ZT, Sprint, USENIX

October 14^{th}, 1999

Motivation: Rapid Growth in Hand-held Devices



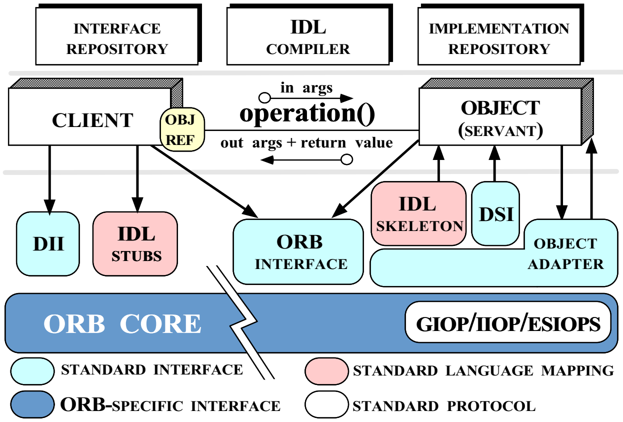
Lewis:98 (IEEE Computer Jan'98)

Comerford:98 (IEEE Spectrum May'98)

- **Devices** → PDAs, WebPhones, WebTVs, handsets
- **Requirements** → Low power consumption, small memory/message footprint, efficient/predictable performance

---

Candidate Solution: CORBA

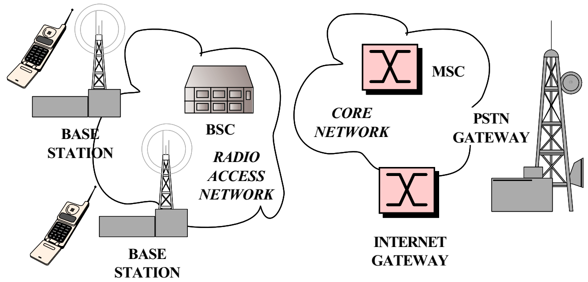


www.cs.wustl.edu/∼schmidt/corba.html

Goals of CORBA

- Simplify distribution by automating
  - Object location & activation
  - Parameter marshaling
  - Demultiplexing
  - Error handling
- Provide foundation for higher-level services

Caveat: Limitations of CORBA for Wireless



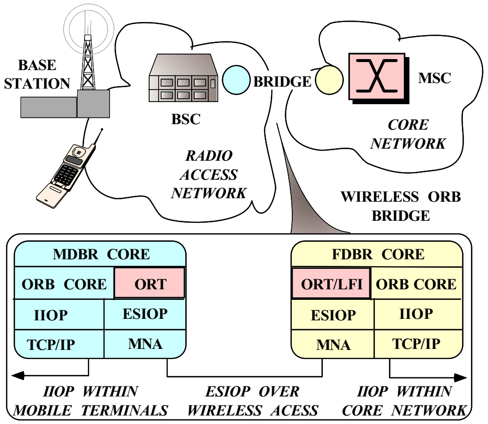
Requirements

- **Location transparency**
- **QoS transparency**
- **Minimal footprints**

Limitations

- **Lack of appropriate protocols**
- **Lack of real-time and async QoS features**
- **Lack of performance optimizations**

OMG Approach: Wireless CORBA

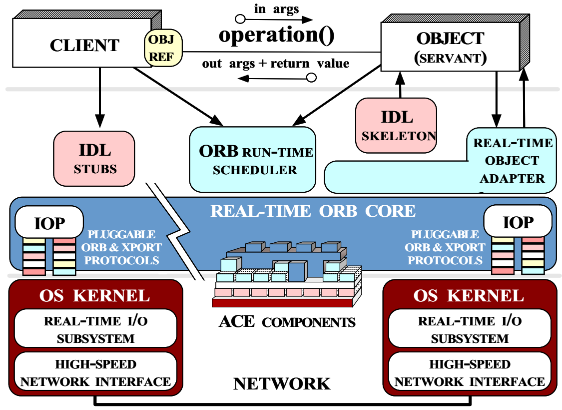


www.omg.org/techprocess/meetings/schedule/Wireless_Access_&_Control_RFI.html

Overview →

- *Mobility domain* and *fixed domain* bridges handle mobility
  - *e.g.*, relocatable object references via location register
- Lightweight ESIOP reduces message footprint

Our Approach: The ACE ORB (TAO)

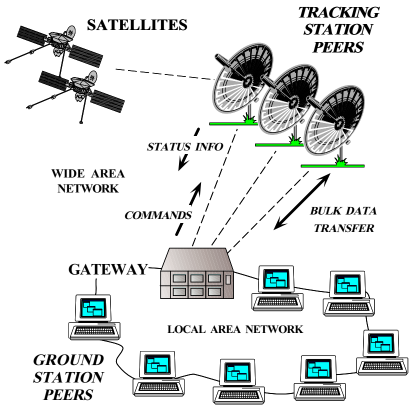


www.cs.wustl.edu/∼schmidt/TAO.html

TAO Overview →

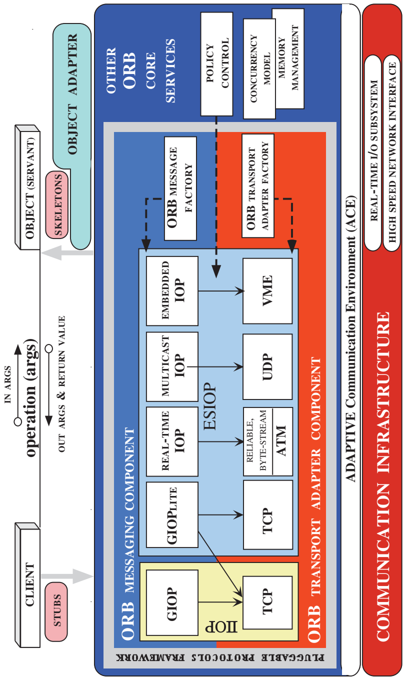
- An open-source, standards-based, real-time, high-performance CORBA ORB
- Runs on POSIX/UNIX, Win32, & RTOS platforms
  - *e.g.*, VxWorks, Chorus, LynxOS
- Leverages ACE

Problem: Hard-coded ORB and Transport Protocols



- GIOP/IIOP are not sufficient for all environments, *e.g.*:
  - GIOP messages may be too large for wireless applications
  - TCP lacks necessary QoS
  - Legacy commitments to certain protocols
- Existing ORBs don't support "pluggable protocols"
  - Both *ORB messaging* and *transport* protocols must be pluggable

Solution: TAO's Pluggable Protocols Framework



Features

- Pluggable **ORB messaging** and **transport** protocols
- Highly efficient and predictable static and/or dynamic configuration
- Multiple profile support

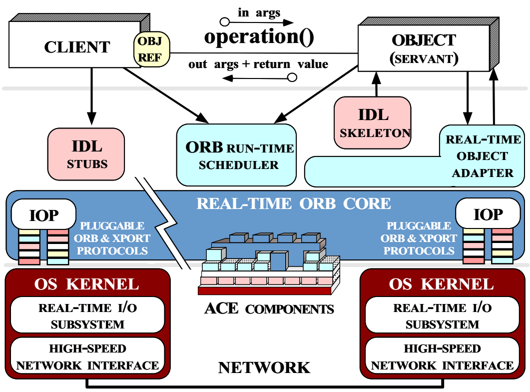
Prior Work: TAO's GIOP-lite Option

- GIOP requests include fields that aren't needed in homogeneous embedded applications
  - *e.g.*, GIOP magic #, GIOP version, byte order, request principal, etc.
- TAO's `gioplite` option save 15 bytes per-request, yielding these calls-per-second:

	Marshaling-enabled			Marshaling-disabled		
	min	max	avg	min	max	avg
GIOP	2,878	2,937	2,906	2,912	2,976	2,949
GIOPlite	2,883	2,978	2,943	2,911	3,003	2,967

- The result is a measurable, but small (2%), improvement in throughput/latency
- Our pluggable protocols framework will all much greater decreases in IOP request sizes, as well as more flexible support for multiple transport protocols
- However, there will be no changes required to the standard CORBA programming model

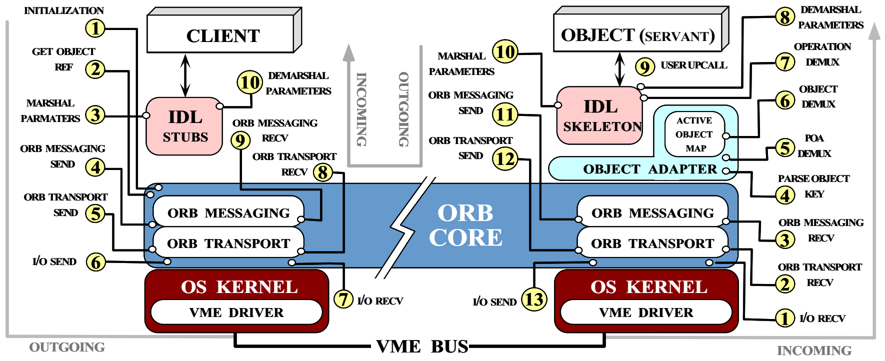
Integrating Pluggable Protocols into TAO



www.cs.wustl.edu/~schmidt/
pluggable_protocols.ps.gz

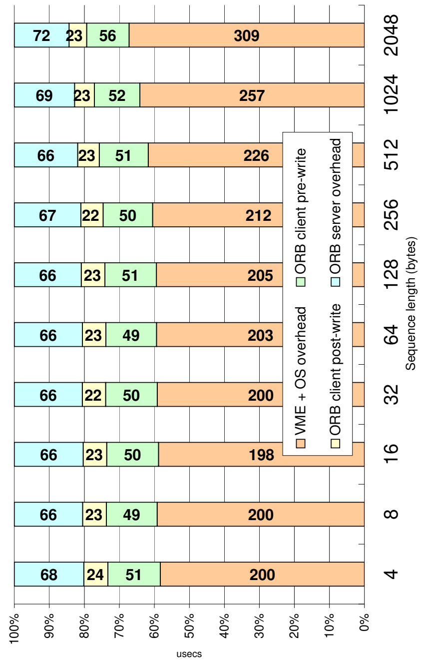
- **ORB Responsibilities**
  - Connection/memory management
  - Data transfer
  - Endpoint/request demuxing
  - Concurrency control
- TAO ORB is highly *pluggable*
  - Demuxing strategies
  - Concurrency strategies
  - Protocol strategies

Pinpointing ORB Overhead with VMEtro Timeprobes



- Timeprobes use VMEtro monitor, which measures end-to-end time
- Timeprobe overhead is minimal, *i.e.*, 1 μsec

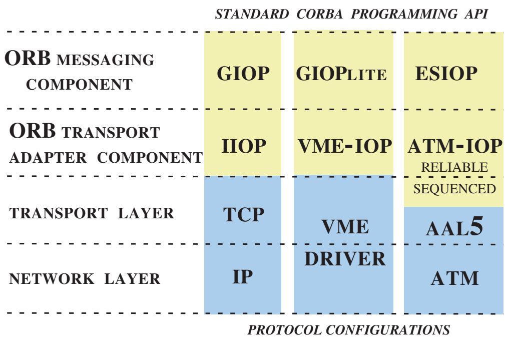
ORB & VME One-way Overhead Results



Synopsis of Results

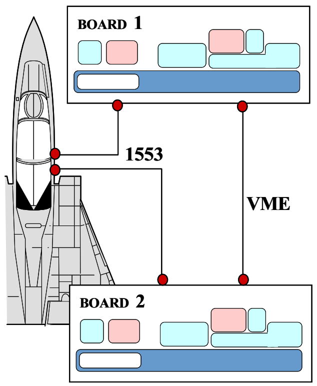
- ORB overhead is relatively constant and low
  - *e.g.*, ~110 μsecs per end-to-end operation
- Bottleneck is VME driver and OS, not ORB

Open Issues for Wireless CORBA



- Where should terminal mobility be supported?
  - *i.e.*, in *Request Bridge* vs. *Mobile IP*
- What types of ESIOPs are necessary?
  - *e.g.*, IIOP variants or entirely new ORB messaging protocols
- How will CORBA services be affected?

Concluding Remarks



- Pluggable protocols are essential to meet the needs of wireless CORBA applications
- The advent of *standard* pluggable protocol APIs will support configuration of multiple **ORB messaging** and **transport** protocols beneath the standard CORBA programming model
- Hopefully, next-generation ORBs will provide much better support for standard pluggable protocols...

Web URLs for Additional Information

- Middleware 2000:
  `//www.research.ibm.com/Middleware2000/`
- These slides:
  `~schmidt/TAO/pluggable_protocols4.ps.gz`
- Information on pluggable protocols:
  `~schmidt/pluggable_protocols.ps.gz`
- Information on CORBA:
  `~schmidt/corba.html`
- Information on ACE:
  `~schmidt/ACE.html`
- Information on TAO:
  `~schmidt/TAO.html`