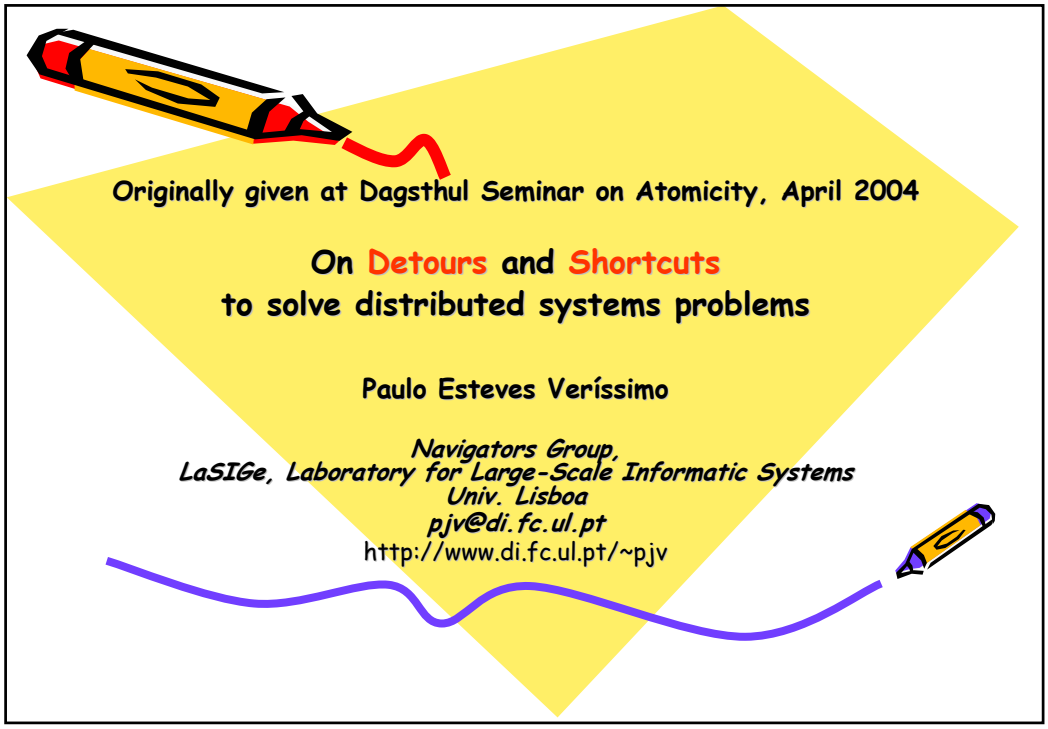
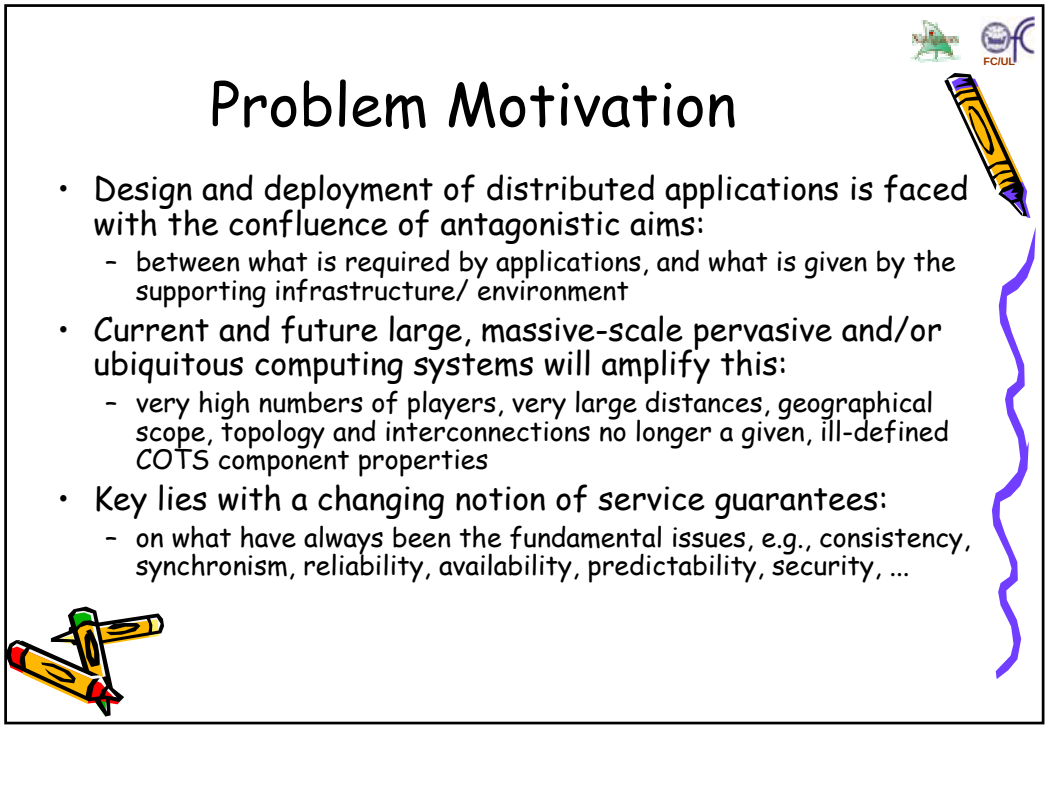


Originally given at Dagsthul Seminar on Atomicity, April 2004

On Detours and Shortcuts to solve distributed systems problems

Paulo Esteves Veríssimo

Navigators Group,
LaSIGe, Laboratory for Large-Scale Informatic Systems
Univ. Lisboa
pjv@di.fc.ul.pt
http://www.di.fc.ul.pt/~pjv



Problem Motivation

- Design and deployment of distributed applications is faced with the confluence of antagonistic aims:
  - between what is required by applications, and what is given by the supporting infrastructure/ environment
- Current and future large, massive-scale pervasive and/or ubiquitous computing systems will amplify this:
  - very high numbers of players, very large distances, geographical scope, topology and interconnections no longer a given, ill-defined COTS component properties
- Key lies with a changing notion of service guarantees:
  - on what have always been the fundamental issues, e.g., consistency, synchronism, reliability, availability, predictability, security, ...

Problem Motivation

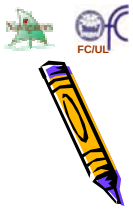
- Take the security dimension
- Many services, beyond mere performance, have to enjoy security properties
- So we should prevent any security breaches
  - But we cannot prevent or detect all attacks/vulnerabilities
  - Even if we could, this would be impractical or too expensive
- Then what if we tolerate them?
  - But it is hard to define a fault model for a hacker…

Grand challenges put by this scenario?

withstanding uncertainty whilst achieving predictability

- Uncertainty:
  - is a common denominator of current systems
  - uncertain synchrony, fault model, and even topology
- Predictability:
  - systems are required to fulfill more and more demanding goals which imply predictability or determinism, e.g, timeliness, security
- Reconciling them means:
  - strong attributes (e.g. on ordering, agreement, timely termination of algorithms) can be secured in settings where usually very little is assumed and very little is expected from
  - current view has been to weaken attributes down to the little that one can expect to get from uncertain environments

The usual path



- If you want efficient/performant solutions to F/T
  - assume controlled failure modes (omissive, fail-silent, etc.)
- If you want to build timely services (even soft RT)
  - assume synchronous models, or at least partially sync
- They only work to the coverage of the assumptions
  - which must be substantiated, else we risk pitfalls such as the "well-behaved hacker" syndrome

Taking detours…

- OBJECTIVE:
  - solve most non-timed problems with highest possible coverage
- tone down determinism
- tone down liveness expectations
- use weaker semantics than ABCAST/Consensus
- tone down allowed fault severity
- OBJECTIVE:
  - solve timed problems with highest possible coverage
- sync, parsync models (coverage ☹)

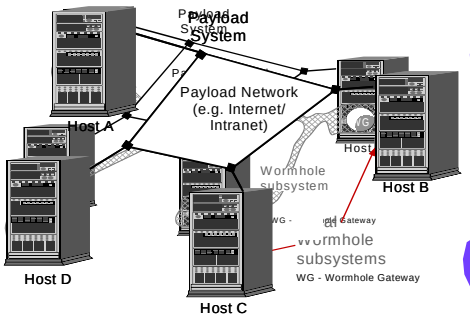
Shortcuts vs. detours

- we propose to render the solution simpler (without changing the problem!)

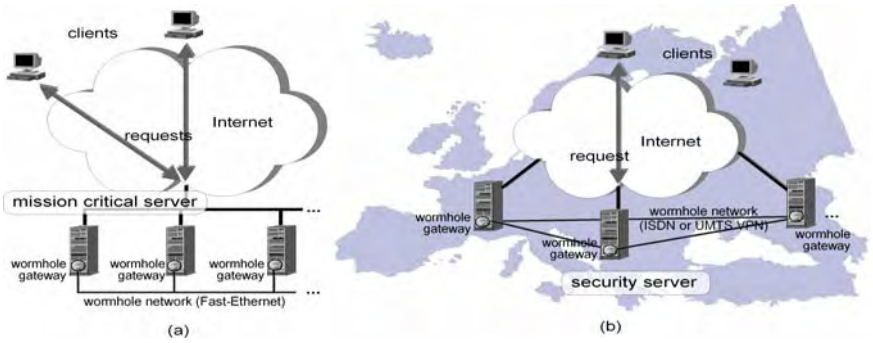
- Architectural hybridization
- Wormholes model

Wormholes

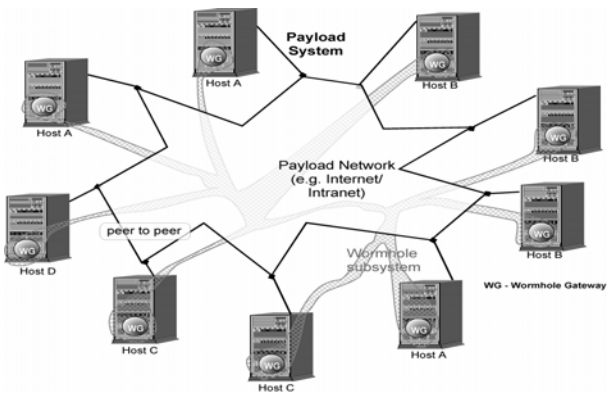
- New design philosophy for architecting and programming distributed systems:
- constructs with privileged properties that endow systems with the capability of evading the uncertainty or weakness of the environment (``taking a shortcut'') for certain crucial steps of their operation, in order to achieve overall strong properties otherwise impossible or complex or expensive



Example of deployment of systems with wormholes



Example of deployment of systems with wormholes



Taking shortcuts i.s.o. detours

- OBJECTIVE:
  - solve most timed or non-timed problems with highest possible coverage
- enforce hybrid behaviour ("strong" and "weak" components) by *architectural hybridization*
- implement strong q.b. components (*trusted-trustworthy*)
- overcome algorithmic hardness (e.g., w.r.t. asynchronism, maliciousness, etc.) through computing models aware of the above (e.g. *Wormholes*)

A (necessarily brief) birds-eye view of some results

Trusted Timely Computing Base (TTCB)

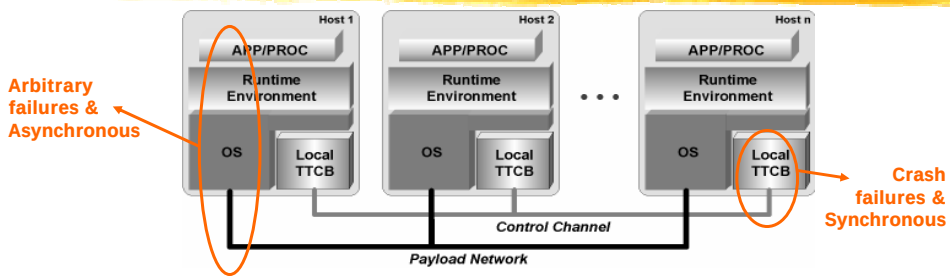
Navigators FC/UL

- **Properties:**
  - trusted and timely execution assistant; trusted timing failure detector
  - secure (can only fail by crashing)
  - real-time (capable of timely behavior)
  - correct processes can interact securely with the TTCB
- **Assists the execution of fault-tolerant algorithms:**
  - provides a trusted environment for crucial steps
- **Can be built (there is a prototype)**

Correia, Veríssimo, and Neves. The Design of a COTS Real-Time Distributed Security Kernel. European Dependable Computing Conf., *EDCC-4*, October 2002

System Model

Navigators FC/UL



- **TTCB is a distributed security kernel that provides a minimal set of trusted and timely services, such as**
  - local authentication
  - agreement on a fixed sized block of data (TBA)
  - globally meaningful timestamps

Navigators FC/UL

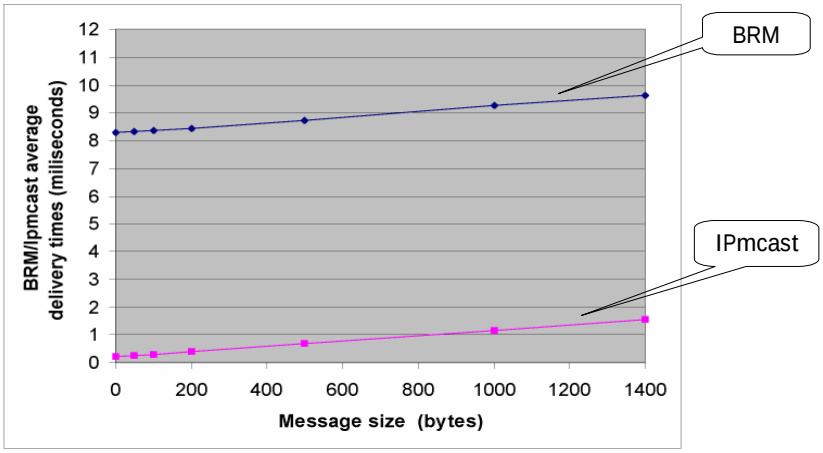
Efficient Byzantine-Resilient Reliable Multicast on a Hybrid Fault Model

www.navigators.di.fc.ul.pt/docs

Correia, M., Lung, L.C., Neves, N.F., Veríssimo, P.: Efficient Byzantine-Resilient reliable multicast on a hybrid failure model. In: Proc. of the 21st Symposium on Reliable Distributed Systems, Suita, Japan (2002)

Navigators FC/UL

Measurements



Typical values in earlier works: ~50ms

Navigators FC/UL

Conclusion

- **Reliable multicast with Byzantine faults requires:**
  - asynchronous system: $n \geq 3f+1$ [Bracha&Toueg]
  - synchronous system: no limit ($n \geq f+2$) [Lamport et al.]
- **We follow a wormhole-aware model:**
  - payload is asynchronous and byzantine-on-failure
  - TTCB is synchronous and crash-on-failure
- **We achieve:**
  - $n \geq f+2$ without asymmetric crypto (signatures)
  - Efficiency: few phases, high performance

Navigators FC/UL

Low Complexity Byzantine-Resilient Consensus

Distributed Computing Journal, 2004/2005

Navigators FC/UL

Termination & FLP result

- **FLP result:**
- ***impossible to deterministically solve consensus in an asynchronous system***
- **Usual solutions:**
- ***randomization, weak synchronous assumptions (e.g., partial synchronous models or unreliable failure detectors)***
- **Our approach:**
- ***avoid violation of safety properties***
- ***ensure termination by finding a way to circumvent the FLP impossibility result***
- **Our assumption**

  eventually there will be a round where at least 2f+1 processes manage to locally call the TTCB on time

Navigators FC/UL

Performance Comparison

- **Use *latency degree* [Schiper 97] criteria extended to include current implementation of TTCB agreement**

Protocol	Latency degree	Requirements
Dwork et al.	**7**	
Dwork et al.	**4**	**signed messages**
Malhki & Reiter	**9 or 6**	**signed messages**
Kihlstrom et al.	**4**	**signed messages**
Block consensus	**1**	**TTCB**
General consensus	**1 or 2**	**TTCB**

Navigators FC/UL

Solving Vector Consensus with a Wormhole

submitted

Navigators FC/UL

Our approach in the FLP scene

- **FLP result:**
- ***impossible to deterministically solve consensus in an asynchronous system***
- **Usual solutions:**
- ***randomization, weak synchronous assumptions (e.g., partial synchronous models or unreliable failure detectors)***
- **Our approach:**
- ***avoid violation of safety properties***
- ***ensure termination by finding a way to circumvent the FLP impossibility result***
- **Our assumption**

  the algorithm running on the payload is fully asynchronous

Performance Comparison

- **Use *latency degree* [Schiper 97] criteria extended to include current implementation of TTCB agreement**

Protocol	LatDeg	MSign	GVer	Artifact
DS [15]	5	5	3	Failure detectors
BHRT [1]	3	3	2	Failure detectors
Our protocol	4	1	1	Wormhole

	Crash			Byzantine			
Protocol	LatDeg	MSign	GVer	LatDeg	MSign	GVer	SDeg
DS [15]	5 + 2f	5 + 2f	3 + f	5 + 2f	5 + 2f	3 + f	f
BHRT [1]	3 + f	3 + f	2 + f	3 + f	3 + f	2 + f	f
Our protocol	4	1	1	4 + 2f	1	1 + f	0

Navigators FC/UL

Main Achievements

- **Fully asynchronous payload algorithm**
- **Low complexity**
- **Consensus without FDs:**
- ***Instead failure detectors, uses low level agreement service***
- ***Does not exclude processes, uses all processes that behave correctly at any given time***
- ***Difficult to construct failure detectors in Byzantine systems***
- ***Reliable Byzantine failure detection: an open problem***

Navigators FC/UL

Worm-IT : group communication system for a Byzantine asynchronous environment

submitted

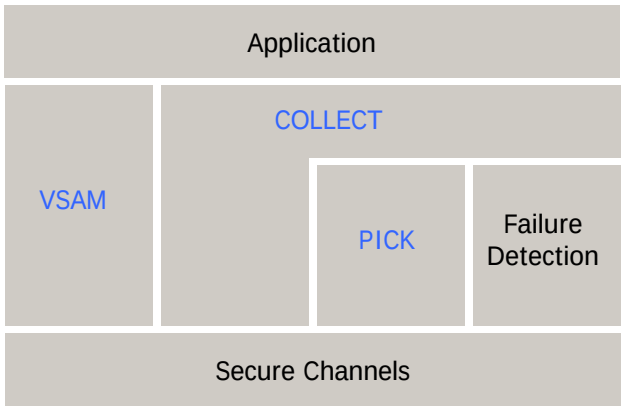
Navigators FC/UL

Worm-IT

- **A group communication system for a Byzantine asynchronous environment**
  - Dynamic Membership Service
  - View-Synchronous Atomic Multicast
- **Intrusion tolerant**
- **The system uses a wormhole that offers a few secure and timely operations**
  - Trusted Timely Computing Base
- **Resilience: f out of 3f+1 (optimal for asynchronous systems)**

Navigators FC/UL

Protocol Stack



Navigators FC/UL

Main Achievements

- **Exemplifies how a reasonably complex system can be built with a wormhole**
- **Make decisions in a distributed way**
- **Good performance since it does not resort to public key cryptography**

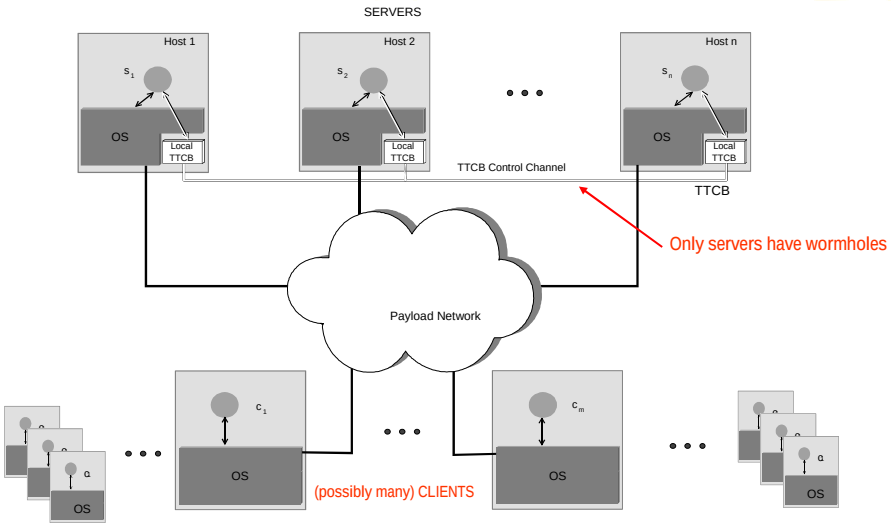
Navigators FC/UL

State machine replication on atomic multicast

IEEE SRDS 04, Florianopolis, Brasil 2004

Navigators FC/UL

System architecture



Navigators FC/UL

Main Achievements

- **First SMA service for practical byzantine distributed systems with resilience f out of 2f+1**
  - Lower number of replicas reduces cost of hardware + cost of designing different replicas (for fault independence)
- **Low time complexity**
- **Probable good performance since it does not resort to public key cryptography**

Navigators FC/UL

Some Recent Publications (urls)

- ***Modeling Wormholes***
- *Uncertainty and Predictability: Can they be reconciled?* Paulo Veríssimo**. Future Directions in Distributed Computing, pages to appear, Springer-Verlag LNCS 2584, month to appear, 2003**
- *The Timely Computing Base Model and Architecture***.** Paulo Veríssimo, António Casimiro. **IEEE Transactions on Computers - Special Issue on Asynchronous Real-Time Systems, vol. 51, n. 8, Aug 2002**
- *The Timely Computing Base: Timely Actions in the Presence of Uncertain Timeliness***.** Paulo Veríssimo, António Casimiro, C. Fetzer. I**n Proceedings of the 1st International Conference on Dependable Systems and Networks, New York, USA, June 2000.**
- The Timely Computing Base**. Paulo Veríssimo and António Casimiro.** Technical Report DI/FCUL TR 99-2, Department of Informatics, University of Lisboa, **May 1999.** ***(original paper, improved in TOCS02)***

- ***Implementing Wormholes***
- *Measuring Distributed Durations with Stable Errors***.** António Casimiro, Pedro Martins, Paulo Veríssimo, Luís Rodrigues. **Proceedings of the 22nd IEEE Real-Time Systs Symposium, London, UK, December 2001**
- *How to Build a Timely Computing Base using Real-Time Linux***.** António Casimiro, Pedro Martins, Paulo Veríssimo. **in Proceedings of the 2000 IEEE International Workshop on Factory Communication Systems, Porto, Portugal, September 2000.**
- *Timing Failure Detection with a Timely Computing Base***.** António Casimiro, Paulo Veríssimo. **3rd Europ. Research Seminar on Advances in Distr. Sys (ERSADS'99), Madeira Island, Portugal, April 23-28, 1999**
- *The Design of a COTS Real-Time Distributed Security Kernel***,** *Miguel Correia, Paulo Veríssimo, Nuno Ferreira Neves***, *Fourth European Dep. Comp. Conf., Toulouse, France, October 2002 © Springer-Verlag.***

Some Recent Publications (urls)

- ***Using Wormholes***
- ***Using the Timely Computing Base for Dependable QoS Adaptation***. António Casimiro, Paulo Veríssimo. **Proceedings of the 20th IEEE Symp. on Reliable Distributed Systems, New Orleans, USA, October 2001**
- ***Generic Timing Fault Tolerance using a Timely Computing Base***. António Casimiro, Paulo Veríssimo. **Procs of the Intern'l Conference on Dependable Systems and Networks, Washington D.C., USA, June 2002**
- ***Efficient Byzantine-Resilient Reliable Multicast on a Hybrid Failure Model***, *Miguel Correia, Lau Cheuk Lung, Nuno Ferreira Neves, Paulo Veríssimo.* ***Proc's of the 21st Symp. on Reliable Distributed Systems (SRDS'2002), Suita, Japan, October 2002***
- ***How to Tolerate Half Less One Byzantine Nodes in Practical Distributed Systems***
  Miguel Correia, Nuno Ferreira Neves, Paulo Veríssimo
  In Proceedings of the 23rd IEEE Symposium on Reliable Distributed Systems. Florianopolis, Brasil, pages 174-183, October 2004
- ***Low Complexity Byzantine-Resilient Consensus***
  Miguel Correia, Nuno Ferreira Neves, Paulo Veríssimo, Lau Cheuk Lung
  Distributed Computing, Accepted for publication, 2004. On-line first:
  http://www.springerlink.com/index/10.1007/s00446-004-0110-7

Navigators FC/UL

- **Navigators group:**
  - http://www.navigators.di.fc.ul.pt/