Reliable Computing over Mobile Networks*

Luís E.T. Rodrigues
IST[†]- INESC[‡]
ler@inesc.pt

Henrique Fonseca
INESC
hjf@inesc.pt

Paulo Veríssimo
IST - INESC
paulov@inesc.pt

Abstract

Mobile networks and hosts have assumed an increasing importance in recent years and will pervade future distributed computing systems. In order to meet user requirements for reliable operation over mobile networks, protocols that mitigate their current limitations, such as slow and unreliable links, need to be found.

In the paper we discuss the difficulties associated with extending existing services to operate on mobile networks. As case studies, the paper discusses two particular protocols: a total order protocol and a fault-tolerant remote invocation scheme. The paper shows that the experience gained with large-scale systems can be successfully applied in this framework.

1 Introduction

Mobile networks and hosts have assumed an increasing importance in recent years and will pervade future distributed computing systems. Unfortunately, with current technologies, mobile communication is still characterized by a number of shortcomings. Wireless cellular technology offers limited bandwidth in a crowded telecommunications space characterized by slow and unreliable links between mobile hosts and "stationary" computers. Also, current methods of packet routing to mobile hosts are still vulnerable to these factors, leading to frequent short-term partitions and high variances in packet latency. In order to meet user requirements for reliable operation over mobile networks, protocols that mitigate these limitations need to be found.

The Navigators team at INESC has been developing, during the last years, a group communication infrastructure designed to operate on large-scale networks. The architecture, called NavTech , is intended to support the development of reliable applications, offering a range of communication and activity-support services including, among others, reliable group communication, group membership, remote-invocation, and support for replication. In the paper we discuss the difficulties associated with extending these services to operate on mobile networks. Similitudes and key differences between the two types of networks are identified. The paper reports how protocols designed for large-scale that can be adapted to accommodate mobility. As case studies, the paper discusses two particular NavTech protocols:

- A total order protocol that is able to adapt itself to the network operating conditions. The protocol uses a new dynamic hybrid approach where processes can switch between two different operational modes depending on traffic load and network delays. The high variation of packet latency exhibited by mobile networks makes this type of approach particularly relevant.
- A fault-tolerant remote invocation scheme, called Generic Remote Invocation Protocol (GRIP), which is independent of the service replication protocol and which places no constraints on the replica consistency model. According to this protocol, clients are not required to maintain fully up-to-date information about the membership of the replicated server. Additionally, they contact the servers through a lightweight remote access protocol based on unreliable point-to-point messages. These weak assumptions are tailored to mobile networks where, due to the presence of partitions and unreliable links, the use of expensive communication and membership protocols is inadequate.

The paper is organized as follows. Section 2 surveys the main properties of mobile networks and Section 3

---

[*] A version of this report was presented in the 5th Workshop on Future Trends in Distributed Computer Systems, Cheju Island, Korea, August, 1995. This work was partially supported by the CEC, through Esprit Project BR 6360 (Broadcast)

[†] Instituto Superior Técnico

[‡] Instituto de Engenharia de Sistemas e Computadores, R. Alves Redol, 9 - 6^o - 1000 Lisboa - Portugal, Tel.+351-1-3100000.

discusses their impact in providing fault-tolerant services. The next two sections present two concrete services: Section 4 presents the total order protocol and Section 5 presents a generic remote invocation protocol. Section 6 gives some concluding remarks.

2 Mobile networks

Today's technological advances allow small notebook and laptop computers to have the computational power and capabilities of what where fully equipped desktop workstations only a few years ago. Still there are problems to be solved like limited power supply and storage capacity, but new breakthroughs in these fields arise almost every day [11, 21]. However, one of the hardest technical challenges is the ability to provide reliable and fast communication facilities to these *mobile computers* [13].

Mobile computers, like any other modern computing device, require network access; in this case wireless network access. But wireless communication is much more difficult to achieve than wired communication because the surrounding environment interacts with the signal, blocking signal paths and introducing noise and echoes. This results in wireless connections being of much lower quality than wired ones: lower bandwidths, higher error rate and more frequent temporary disconnections and partitions. Moreover, these factors can increase communication latency due to retransmissions, timeout delays and error control protocol processing [15].

Bandwidth is a key issue in current wireless network technologies. Although recent local area wireless solutions achieve 1 to 2 Mbps bandwidth [27], these values are still one order of magnitude away from the 10 Mbps which Ethernet provides, and two orders away from FDDI's 100 Mbps and ATM's 155 Mbps. In the large area environment the scenario is even worse. The only feasible solution up to now is through the use of digital data cellular phones that can achieve a maximum bandwidth of 9.6 to 14.4 Kbps [14].

The cell structure of mobile networks also limits the bandwidth available to the user, because several users have to share the same cell and large concentrations of mobile users in a single cell may overload the network capacity [28]. In consequence, population distribution is an important factor in these networks. Current research leads to the fact that fitting more cells in the same area with reduced transmission ranges can overcome congestion problems. At the same time, it reduces power consumption and signal interference, due to smaller and less powerful transceivers. Bandwidth is also easier to boost in smaller areas of coverage.

Mobile computing networks can exhibit much greater variation in bandwidth than traditional networks. This variation can be perceived when a mobile user shifts from one network provider to another but is even more evident as mobile hosts may sometimes be connected to a wired network, like, for example, when the mobile user is using his portable at work and connects it to the local-area Ethernet present in his office [9]. Additionally, as users move around with a portable device, they may leave the coverage area of network transceivers resulting in wireless connections being lost or degraded as a result of mobility.

Many of today's computer systems and applications are designed to rely heavily on the availability of a network connection, so disconnections are normally not handled in a graceful manner. In mobile systems, wireless network communication is very prone to the loss of connectivity, so the handling of such situations is vital to the correct operation of applications [18].

Also, connection-oriented protocols tend to assume that a permanent and reliable link is available and usually abort when a disconnection or partition occurs. These protocols are usually not prepared to efficiently handle variations from one to four orders of magnitude of the available network bandwidth. Although today's fixed networks exhibit fluctuant traffic loads, especially in the large area field, the variation is not as high as in mobile networks.

3 Fault-tolerant services

We have seen in the previous section that the current state of the art in mobile networks still provides users with a rough environment, very error prone, with low bandwidth and subject to frequent disconnections. In order to provide fault-tolerant services on top of these mobile networks, one has to rely on communication protocols that are able to cope with this environment.

In face of the amount of research work done on developing fault-tolerant services for other communication infrastructures, one should try to evaluate the feasibility of porting existing work to the mobile network environment. Unfortunately, most of previous work was targeted at local-area networks, depending on the high bandwidth, low latency, and reliability offered by these infra-structures. On the other hand, a number of new protocols have been recently designed offering efficient support for distributed fault-tolerance applications over large-scale networks [2, 3]. Such environments, as for instance the Internet in the case of the TCP/IP based internetwork, are not homogeneous, comprising a mesh of long-haul point-to-point links along with many high-bandwidth components such as LANs, MANs and, still to come, ATM fabrics. Besides this large heterogeneity in bandwidth and reliability, large-scale networks are also often prone to physical and virtual network partitions.

The gap between large-scale and mobile networks is being shortened by recent research on network layer routing and addressing protocols for mobile networks, allowing the interconnection of mobile and fixed networks. The Mobile-IP work has addressed the problem of delivering IP packets to mobile hosts regardless of their location [5, 16]. Solutions for delivering multicast messages in mobile networks have also been presented [1] as well as enhancements to TCP protocols for dealing with possible temporary disconnections [4, 29]. These works, provide to application designers similar abstractions as those supported by existing large-scale networks, although they make more acute the need to tolerate unreliable links and network partitions. Thus, given the similarities between both environments, we believe that protocols specially designed for large-scale network systems can be adapted to mobile networks.

The work presented in the remaining of this paper was developed in the framework of designing and implementing a platform for reliable distributed computing in large-scale systems, called NavTech , Navigators Technology. The NavTech architecture belongs to a new generation of systems [2, 3] attempting at solving the problems posed by large-scale communication. We are currently evaluating the suitability of this architecture for mobile operation. This paper presents two particular NavTech protocols, as case studies: a total order protocol and a fault-tolerant remote invocation scheme. These will be discussed in the next two sections.

4 Total order protocol

Totally ordered multicast protocols have proved to be extremely useful in supporting many fault-tolerant distributed applications. For instance, total delivery order is a requirement for the implementation of a replicated state-machine [26], which is a general paradigm for implement of fault-tolerant distributed applications. Among the several algorithms to implement total ordering, the *token-site* [6, 17] and *symmetric* [22] are the most used approaches. In the token-site approach, one (or more) sites are responsible for ordering the messages on behalf of the other processes in the system. In the symmetric approach, ordering is established by all processes in a decentralized way, using information about message stability. Both methods have advantages and disadvantages.

Token-based protocols provide good performance when a single process is producing messages at a time. In this case, the producer process keeps the token and orders the messages as it sends them. However, when more than one process are transmitting, the latency is limited by the time to circulate the token or to request an order number from the token site. Unfortunately, the message delivery latency for a process that does not hold the token is at least $2D$ (where D is the network delay), i.e., the time to disseminate the message plus the time to obtain either the token or an order number from the token holder. Thus, token-site approaches are inefficient in face of large network delays.

Symmetric protocols have a number of very appealing features. They are fully-decentralized and, since all processes are treated equal, they provide good load distribution. Unfortunately, symmetric protocols require that all processes send messages to enforce message stability. If the appropriate mechanisms are used, symmetric protocols can obtain a latency closer to $D + t$ (where t is the largest inter-message transmission time in the system) [23]. Still, message latency is a function of the transmission rate of the slower processes in the system. Thus, symmetric protocols tend to perform poorly in environments where the majority of processes produce messages at very low rates [20].

In a recent report we propose a new hybrid scheme for implementing total ordering in large-scale systems [23]. In our hybrid scheme all processes multicast the messages directly to all group members. However, only certain processes are allowed to establish message order: these processes are said to operate in *active* mode. Active processes issue order numbers, also called *tickets*, for their own messages and for the messages of the processes that operate in *passive* mode. At a given instant, each passive process is associated with a single active process, which issues tickets for its messages on its behalf (however, this association may change with time). Tickets issued by different active processes are ordered using a symmetric algorithm. Thus, in our scheme, some processes order messages using a symmetric approach and others use a token-site approach in order to minimize overall message latency.

4.1 Ordering messages from mobile hosts

The development of the dynamic hybrid protocol was motivated by the fact that in a large-scale network, the traffic patterns of the processes are usually heterogeneous. The same applies to the properties of the links connecting the processes: some processes will be located within the same local area network, others will be connected through slow links, subject to long delays. In such an environment, none of the previous approaches could provide optimal performance.

The interesting aspect of these assumptions is that they are equally valid in mobile networks:

- Low bandwidth is still a reality in large-scale mobile networks due to current cellular phone technology.
- The high bandwidth variability present in wireless networks sets the base for the development of adaptative protocols that can react to the constant changes of network bandwidth.

In face of these observations, we propose the following approach to offer total order to mobile processes using the hybrid protocol:

1. Mobile hosts can generally assume a passive mode given their low bandwidth and reliability;
2. Some of the fixed hosts will then be in active mode making use of the fixed network infrastructure to achieve more efficient reconfiguration schemes;
3. And finally, given the mobility inherent to mobile hosts, they can easily reconfigure by changing its active sequencer to a closer host on the network.

4.2 Comparative performance

In the limit situations, the dynamic hybrid protocol will resemble either a pure symmetric or a pure token-site protocol. In an intermediate scenario, operational modes are selected according to the properties of a process and of its links. Processes are allowed to dynamically switch between the passive and active modes. We have shown that the dynamic hybrid scheme can be successfully applied in systems where the topology/traffic patterns are not known *a priori*, exhibiting a lower latency than the static hybrid approach [23]. This is the case in mobile networks where users often change location, thus changing the network topology, as well as in large-scale systems where messages pass through unknown routers and links before reaching their destinations.

Using this rationale as a guideline, we have made a preliminary evaluation of our hybrid protocol in a mobile environment. The hybrid approach was compared with a symmetric algorithm and a token-site algorithm. The pure symmetric algorithm we have used is a non-optimized variant of the protocol in [12]. The pure token-site algorithm used for comparison terms was the non fault-tolerant version of [17], where a single site issues tickets on behalf of all other processes in the group.

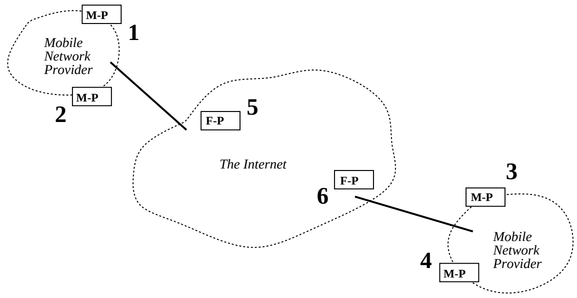

Figure 1: Simulation configuration

Initial tests show promising results. We have simulated a system composed of two groups of two mobile networks interconnect through the Internet, as illustrated in Figure 1. There are two mobile processes in each wireless network. The following simulation parameters were used: $200ms$ latency in the wireless networks; $50ms$ latency in the Internet; processes 1 and 3 have a inter-message transmission time of $1m/s$; processes 2 and 4 a inter-message transmission time of $5m/s$; a quasi-periodic message distribution was used for all processes. In the pure token-site algorithm node 2 was the sequencer while in the hybrid approach the two fixed-host processes were assigned with the active mode and all mobile processes with the passive mode. Under this configuration, the average latency of the total order multicast is the following: $996ms$ for the pure symmetric algorithm, $772ms$ for the pure token-site algorithm, and $588ms$ for the hybrid approach. Further evaluation tests have to be performed using more complex scenarios with a higher number of fixed and mobile hosts. However we are confident that results from these tests will show reduced latency when compared to previous approaches.

5 Remote invocation protocol

The choice of a replication algorithm for a particular replicated service can be based on many constraints, most of which depend on some combination of the service semantics and environment. The inevitable evolution of system components or service requirements is thus likely to affect the choice of replication algorithm, and makes the availability of a wide range of protocol options advantageous. This is particularly relevant in large-scale systems given the large spectrum of application requirements and the heterogeneity of interconnecting links, characteristics which are also shared by mobile systems. Unfortunately, in most systems a replicated service can be accessed remotely only via protocols tailored to its specific replication protocol [8, 19], causing an undesirable violation of both transparency and modularity.

In order to maintain the transparency of an object replication scheme, we have designed a fault-tolerant Generic Remote Invocation Protocol (GRIP) [24] which is independent of the service replication protocol and which places no constraints on the replica consistency model. Clients use a lightweight remote access protocol which allows them to remain independent of the details of the replication protocol and of the resulting inter-replica synchronization. GRIP design provides explicit support for the implementation of weakly consistent replication schemes.

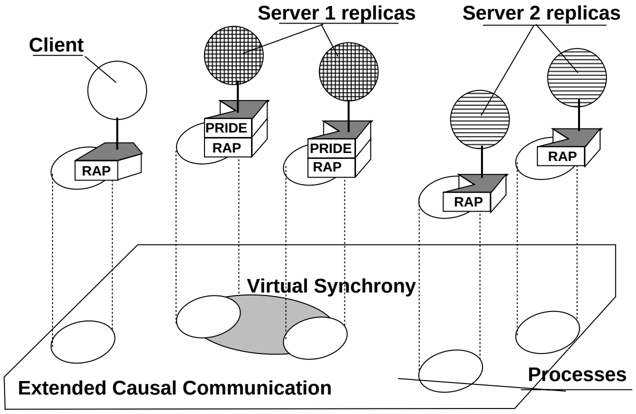

Figure 2: Interface Layer

GRIP is designed as a collection of modular services which can be configured according to the needs of the application (see Figure 2). The lowest layer is the communication service. It is based on a group communication model with some optional services designed for large-scale operation (in this paper, we just refer to the *transparent messages* service). Clients contact replicated services by using a lightweight Remote Access Protocol (RAP) to contact an individual replica. The corresponding RAP server entity relays the request to its local replica, where it is handled according to the semantics of the chosen replication protocol; replies are relayed back to the client in a similar manner. Because it is designed to be light-weight, distributed at-most-once semantics are not explicitly guaranteed by RAP. If desired, they can be implemented for a particular service with the help of the optional Protocol for Repeated Invocation DEtection (PRIDE) layer.

Although modularity and flexibility were the major goals of GRIP design, special attention was devoted to issues such as light-weight client-server interaction. As a result, GRIP has three characteristics which are particularly useful for mobile operation, namely:

1. the ability to preserve causal relation in presence of unreliable links (transparent causal messages);
2. dynamic re-binding;
3. and a powerful retry detection mechanism (PRIDE) providing at-most-once failure semantics.

These mechanisms will be discussed in next paragraphs.

5.1 Communication services

GRIP protocols assume the availability of (logical) reliable multicast and causal communication among server replicas but does not require their use by service clients, allowing servers and clients to operate with low synchronization costs. This is an important goal in large-scale systems where unreliable links can coexist with local area networks. In mobile networks which are even more subject to packet loss and long delays this requirement gains additional relevance.

Still, to offer consistency guarantees to the clients, GRIP uses a mechanism that preserves causal relation in presence of unreliable links. This mechanism was called transparent causal messages [25]. The purpose of this scheme is to circumvent one of the criticisms made of causal multicast communication systems [7] which concerns the mandatory reliability requirement. Once a message introduces a causal dependency, that message must be reliably delivered; otherwise, successive messages will be prevented from being delivered. In some cases the delivery of a causal message is delayed until there is a guarantee that the message will be successfully delivered at all destinations; the sender may even be prevented from sending new messages until this guarantee is obtained.

relation	delivery order
$M_1 \rightarrow M_2$	M_2 after M_1
$M_1 \rightarrow m_2$	m_2 after M_1
$m_1 \rightarrow M_2$	undefined
$m_1 \rightarrow m_2$	undefined

Table 1: Transparent messages

To avoid this problem, the communication service distinguishes two types of messages: (normal) *opaque* causal messages and *transparent* causal messages. The delivery order for transparent messages with regard to opaque messages is summarized in the Table 1 (where opaque messages are represented in upper-case, transparent messages in lower-case, and right-arrows, →, represent the logical precedence).

Transparent causal messages are messages that are delivered in causal order with respect to (normal) opaque messages but that do not themselves introduce causal dependencies. Thus, they have the following characteristics relevant to mobile operation:

- no message is ever delayed by a transparent message;
- no reliability constraints are imposed on the transmission of transparent messages.

5.2 Dynamic re-binding

Also of special relevance in the context of mobile communication is the GRIP's dynamic re-binding feature, which is designed not only to optimize access to the replica set in a large-scale setting, i.e. using the most accessible replica, but also in redirecting to ensure availability in case of replica failure. This feature is relevant for mobility since it automatically compensates the effect of migration (of client or server).

Clients of a replicated service use the lightweight Remote Access Protocol to contact an individual replica. RAP does not requires up-to-date information about the membership of the replicated service. Thus, clients do not need to be permanently connected with the server group. Instead, RAP simply caches a list of addresses where servers might be running and, using unreliable transparent messages, tries to contact a reachable replica by parsing this list. Usually, one of the cached addresses will be reachable but, in worst case, the client will need to contact a name server to obtained a fresh copy of the server address list.

5.3 Distributed retry detection

GRIP provides a module, called PRIDE, that guarantees at-most-once semantics for client requests. PRIDE's semantic guarantees are optional on a per-service and a per-invocation basis. Retry-detection is achieved by communication among the servers, and at-most-once guarantees are provided despite the fact that clients are free to submit different, or even the same, requests to different replicas.

In this paper we are not concerned with the actual protocols used to actually detect and eliminate duplicate request. Instead, we emphasize the importance of such mechanism in a mobile environment. Since in this type networks virtual partitions are frequent, it is possible that a mobile host suddenly looses contact with is associated server. As soon as communication is re-established, *eventually through another server*, the mobile client can freely re-submit all pending request to that server without risking violating consistency. Since PRIDE involves exclusively inter-server communication (where links are expected to be faster and more reliable), this strongly reduces the complexity of mobile client software.

6 Conclusion

Users demand reliable and efficient services over mobile networks. To support such distributed applications powerful communication protocols need to be developed. Using two concrete examples, the paper has shown that the experience gained with large-scale systems can be successfully applied to mobile networks.

At INESC, we are currently evaluating the feasibility of applying the NavTech architecture to mobile networks. Although the described features of this architecture are suitable for such a task, there are other aspects which deserve further study. Among these, one of the most important aspects is the development of efficient membership protocols that operate correctly in presence of partitions [10, 3]. At this moment, the total order protocol assumes a linear membership service and GRIP avoids the membership problem by not requiring clients to have up to date information about server membership.

References

[1] A. Acharya and B. Badrinath. Delivering multicast messages in networks with mobile hosts. In *Proceedings of the 13th International Conference on Distributed Computing Systems*, pp 292–299, Pittsburgh, Pennsylvania, U.S.A., May 1993.

[2] Y. Amir, L. Moser, P. Melliar-Smith, D. Agarwal, and P. Ciarfella. Fast message ordering and membership using a logical token-passing ring. In *Proceedings of the 13th International Conference on Distributed Computing Systems*, pp 551–560, Pittsburgh, Pennsylvania, U.S.A., May 1993.

[3] O. Babaoglu and A. Schiper. On group communication in large-scale distributed systems. In *Proceedings of the 6th ACM-SIGOPS Europe Workshop*, Dagstuhl, Germany, Sep. 1994.

[4] A. Bakre and B. Badrinath. I-TCP: Indirect TCP for mobile hosts. Technical Report DCS-TR-314, Rutgers University, Oct. 1994.

[5] P. Bhagwat and C. Perkins. A mobile networking system based on internet protocol (IP). In *USENIX Symposium on Mobile and Location Independent Computing*, pp 69–82, Cambridge, Massachusetts, U.S., Aug. 1993.

[6] J. Chang and N. Maxemchuck. Reliable broadcast protocols. *ACM, Transactions on Computer Systems*, 2(3), Aug. 1984.

[7] D. Cheriton and D. Skeen. Understanding the limitations of causally and totally ordered communication. In *Proceedings of the 14th Symposium on Operating Systems Principles*, Asheville, NC, U.S.A., Dec. 1993.

[8] E. Cooper. Replicated procedure call. In *Proceedings of the 3rd ACM symposyum on Principles of Distributed Computing*, Berkeley, U.S.A., Aug. 1984.

[9] N. Davies, G. Coulson, and G. Blair. Supporting quality of service in heterogeneous networks: from ATM to GSM. Technical Report LUCD Int. Rep. MPG-93-26, Lancaster University, 1993.

[10] D. Dolev, D. Malki, and R. Strong. An asynchronous membership protocol that tolerates partitions. Research Report, The Hebrew University of Jerusalem, 1993.

[11] D. Duchamp. Issues in wireless mobile computing. In *Third Workshop on Workstation Operating Systems*, pp 2–10, Key Biscayne, Florida, U.S.A., 1992.

[12] P. Ezhilchelvan, R. Macedo, and S. Shrivastava. Newtop: A fault-tolerant group communication protocol. In *Proceedings of the 15th International Conference on Distributed Computing Systems*, Vancouver (BC), Canada, May 1995.

[13] G. Forman and J. Zahorjan. The challenges of mobile computing. Technical Report UW CSE TR 93-11-03, University of Washington, Mar. 1994.

[14] S. Hild. A brief history of mobile telephony. Technical report, Cambridge University Computer Laboratory, U.K., Apr. 1995.

[15] T. Imielinski, A. Acharya, and B. Badrinath. Impact of mobility on distributed computations. *ACM Operating Systems Review*, 27:15–20, 1993.

[16] J. Ioannidis, D. Duchamp, and G. Q. Maguire. IP-based protocols for mobile internetworking. In *SIGCOMM'91*, pp 235–245, Zurich, Switzerland, 1991.

[17] M. Kaashoek and A. Tanenbaum. Group communication in the Amoeba distributed operating system. In *Proceedings of the 11th International Conference on Distributed Computing Systems*, pp 222–230, Arlington, Texas, U.S.A., May 1991. I

[18] K. Keeton, B. Mah, S. Seshan, R. Katz, and D. Ferrari. Providing connection-oriented network services to mobile hosts. In *USENIX Symposium on Mobile and Location Independent Computing*, pp 83–102, Cambridge, Massachusetts, U.S.A., 1993.

[19] R. Ladin, B. Liskov, and L. Shrira. Lazy replication: Exploiting the semantics of distributed services. In *Proceedings of the Ninth Annual ACM Symposium of Principles of Distributed Computing*, pp 43–57, Quebec City – Canada, Aug. 1990.

[20] L. Mahlis, W. Sanders, and R. Schlichting. Analytic performability evaluation of a group-oriented multicast protocol. Technical report, University of Arizona, 1992.

[21] B. Marsh, F. Douglis, and R. Caceres. System issues in mobile computing. Technical Report MITL-TR-50-93, Matsushita Information Technology Laboratory, U.S.A.., Feb. 1993.

[22] L. Peterson, N. Buchholz, and R. Schlichting. Preserving and using context information in interprocess communication. *ACM Transactions on Computer Systems*, 7(3), Aug. 1989.

[23] L. Rodrigues, H. Fonseca, and P. Veríssimo. A dynamic hybrid protocol for total order in large-scale systems. Technical report, INESC, Lisboa, Portugal, 1994.

[24] L. Rodrigues, E. Siegel, and P. Veríssimo. A replication-transparent remote invocation protocol. In *Proceedings of the 13th Symposium on Reliable Distributed Systems*, Dana Point (CA), U.S.A., Oct. 1994.

[25] L. Rodrigues and P. Veríssimo. How to avoid the cost of causal communication in large-scale system. In *Proceedings of the 6th ACM-SIGOPS Europe Workshop*, pp 106–111, Dagstuhl, Germany, Sep. 1994.

[26] F. Schneider. Implementing fault-tolerant services using the state machine approach: a tutorial. *ACM Computing Surveys*, 22(4):290–319, Dec. 1990.

[27] B. Tuch. An ISM band spread spectrum local area network: WaveLAN. In *Workshop on Wireless Local Area Networks*, Worcester Polytechnic Institute, 1991.

[28] J. Walker and B. Gardner. Cellular radio. In J. Walker, editor, *Mobile Information Systems*, pp 59–104. Artech House, 1990.

[29] R. Yavatkar and N. Bhagwat. Improving end-to-end performance of TCP over mobile internetworks. In *Workshop on Mobile Computing Systems and Applications*, Santa Cruz (CA), U.S.A., 1994.