**Paper title:** Active Networking Support for Mobile Multicast Communication

**Authors:** Flavio Paiva Junqueira E-mail: `flavio@gta.ufrj.br`

Otto Carlos Muniz Bandeira Duarte E-mail: `otto@gta.ufrj.br`

**Affiliation**: Universidade Federal do Rio de Janeiro

**Address**: Grupo de Teleinformática e Automação – GTA

Universidade Federal do Rio de Janeiro

COPPE/EE - Programa de Engenharia Elétrica

Caixa Postal 68504 - CEP: 21945-970 - Rio de Janeiro - RJ - Brasil

**Phone number**: +55 (21) 260-5010 ext. 240

**Fax number**: +55 (21) 290-6626

**Abstract**

This paper proposes a scalable and reliable multicast communication service based on active networking for mobile hosts in cellular environments. This service uses resources offered by active nodes in order to overcome problems in mobile multicast communication such as: high error-bit rates over wireless links, losses due to handoffs, mobile host location, and NAK implosion. Simulations were carried out for evaluating the service performance as both mobile host speed and multicast group size increase. Simulation results show that this service performs well for high handoff frequencies, even to a large number of users.

**Event:** General Conference

# Active Networking Support for Mobile Multicast Communication

Flavio Paiva Junqueira Otto Carlos M. B. Duarte
flavio@gta.ufrj.br otto@gta.ufrj.br
Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro
COPPE/EE - Departamento de Engenharia Elétrica
Caixa Postal 68504 - CEP: 21945-970 - Rio de Janeiro - RJ - Brasil

**Abstract**

This paper proposes a scalable and reliable multicast communication service based on active networking for mobile hosts in cellular environments. This service uses resources offered by active nodes in order to overcome problems in mobile multicast communication such as: high error-bit rates over wireless links, losses due to handoffs, mobile host location, and NAK implosion. Simulations were carried out for evaluating the service performance as both mobile host speed and multicast group size increase. Simulation results show that this service performs well for high handoff frequencies, even to a large number of users.

## 1 Introduction

Due to the cellular technology success and the spread use of mobile computers, an interest in Internet applications adapted to mobile environments has been increasing. Mobile environments have some specific characteristics such as: high error-bit rates over wireless links, losses caused by handoffs, and updates of host position [1, 2]. Position information updating is required for the routing service, and it should be accomplished as soon as possible for supporting high handoff frequencies, required for cellular networks. Internet applications that could be used in mobile environments are reliable multicast applications, as shared whiteboards, and distributed interactive simulation [3, 4]. In these applications, an increase in the number of users should not correspond to an equivalent increase neither in network resources use nor in processing load on sender.

An important reliable multicast problem is acknowledgement implosion [3, 4]. This problem happens when multiple receivers send acknowledgements of the same message, causing both processing overload on sender and network flooding. In addition, for mobile hosts, multicast communications should fulfill some other requirements. Loss recovery must be carried out rapidly, in order to overcome interferences over wireless links and losses due to handoffs. Another requirement is the dynamic updating of a multicast distribution tree, which must provide fast adaptation to mobile host movement. These requirements suggest that loss recovery and topology adaptation should be accomplished near to mobile host location. Therefore, it would be important to perform specific computations in nodes near to mobile receivers to fulfill those requirements.

Recently, researchers have proposed a new networking paradigm, known as active networking [5, 6]. Active networking lets applications insert specific computations in network nodes. Specific computation executions allow not only network service fitting application needs, but also its adaptation to current application and network conditions. Thus, new services can be rapidly developed and deployed, granting performance gains to applications [7]. Computations are inserted through programs executed at active nodes, and a proper programming language must be available to develop these programs. Besides the programming language, an interface to resources should be available to programs in active nodes, as storage capabilities, queries, packet control, etc. This interface is available through a set of primitives used by programs. Program execution occur per packet, i.e., each packet received in an active node fires a specific program execution, and execution results determine packet destination. Thus, some alternatives are suggested to routing service. For instance, packets can be discarded, be replicated, or even wait for other packets for synchronization purposes.

Tennehouse *et al.* [8] have implemented a toolkit called ANTS (Active Network Toolkit System), which replaces conventional packets carrying only data by capsules, that carry data and a reference to a forward routine. Forward routines are implemented using the Java language. Moreover, capsules are classified by type, and each type has an associated routine. In the ANTS, a set of capsule types defines a protocol. The procedures of a protocol reach an active node through an on-demand mechanism, i.e., when a capsule reaches an active node, and its forward routine is not in this node's cache, the node requests this routine to the last node where this capsule came from. Among resources offered by active nodes is the non-transient storage, often known as soft-state, used to store current flow state. In this context, flow is a sequence of capsules which follow the same path and requires the same processing. Soft-state lets capsules store information which remains after routine execution, and can be used by capsules of the same flow. Soft-state is available, however, only for a limited period of time whether no capsule updates it.

A service was proposed by Tennenhouse *et al.* [8] using the concepts introduced by the ANTS to create and maintain one-to-many multicast distribution trees. Receivers interested in receiving multicast data from a group source should periodically send subscribe capsules to this sender. Subscribe capsules store pointers in soft-state to indicate the path to the receivers. Each group source sends multicast data capsules, which read group information at each active node and replicate its contents to each pointer in soft-state.

This information is also used to discard excessive subscribe capsules in order to avoid implosion. This mechanism is robust to topology changes, since a subscribe capsule can easily rearrange multicast tree topology in case either of a node or link failure. Another mechanism, proposed by Lehman et al. and called ARM (Active Reliable Multicast) [9], employs soft-state to store information used to avoid the well-known acknowledgement implosion problem, and to store multicast data in order to perform scoped retransmissions, i.e., to deliver retransmitted packets only to the subset of receivers that did not receive the packets.

In another context, Campbell *et al.* [10] have also been employing soft-state in a proposed scheme for cellular networks on the Internet, using special nodes called cellular IP nodes. A cellular IP network consists of interconnected cellular IP nodes and a gateway connecting this network to the rest of the Internet. Cellular IP nodes route IP datagrams inside cellular networks, and communicate with mobile hosts by air interfaces. Cellular IP nodes with air interfaces, called base stations, send periodically beacon signals, which permit cell change decision to mobile hosts. To route datagrams correctly, cellular nodes have caches to store the temporary path to mobile hosts. This cache permits fast topology adaptation to host movement, since temporary paths are rapidly updated, admitting high handoff frequencies. Cellular IP acts together with Mobile IP [11], which uses intermediate agents to communicate with mobile hosts. An agent in a mobile host home network, called home agent, receives all datagrams sent to the mobile host, and sends them through an IP tunnel to an agent in mobile host current domain, called foreign agent. The foreign agent delivers datagrams to the mobile host. In Cellular IP networks, the gateway represents the foreign agent, and a foreign agent exchange occurs only when a mobile host pass from one cellular network to another.

This article proposes a service to support reliable multicast communication for mobile hosts in cellular networks. The protocol which provides this service uses subscribe capsules to create and maintain one-to-many multicast distribution trees. The proposed service supports high handoff frequencies together with a large number of users, and, in addition, it does not require intermediate agents for mobile hosts. Another feature of this service is the employment of soft-state to provide scoped retransmissions, and to avoid the acknowledgement implosion problem, as proposed by ARM. Simulation results show the service high-performance as both handoff frequency and the number of users per session increase.

The remaining sections are divided as follows. Section 2 presents the general idea of the proposed service. Section 3 describes details of the protocol which provides this service. Simulation model and results are shown in section 4. Finally, section 5 presents the conclusions.

## 2 Service Description

This section describes the proposed multicast communication service for mobile hosts. In this service, we assume that all nodes are active, including base stations in cellular networks. Base stations with air interfaces to communicate with mobile hosts are called active cellular nodes. Moreover, we consider seamless handoff strategy when cell changes occur, i.e., when a mobile host determines that it should change to a new cell, it establishes a connection with the new base station, maintaining the old one, and the old connection is tore down at the same time that the new one is tore up [12]. Cell change decisions are determined through the intensity of beacon signals broadcasted by base stations. Another important assumption is that the source of a multicast group is a fixed node.

This service uses the same mechanism proposed by Tennenhouse *et al.* to create and maintain multicast distribution trees. Multicast distribution tree configuration is illustrated in figure 1. In this mechanism, all multicast group members send subscribe capsules to the group source in order to update the group soft-state in each node on the way to the sender. This mechanism was adopted owing to its fault tolerance and dynamism, since multicast tree reconfiguration can be easily performed. This characteristic is important on the proposed service, because every time a mobile host member changes cells, the path from the sender to this receiver must be reconfigured. Members send subscribe capsules towards the sender, and the subscribe program looks up the forwarding information of its group at each active node. When this information is found, group registration process is complete, and the new member is able to receive multicast data, even if this subscribe capsule needs to continue on the way to sender in order to update the soft-state expiration time from the next nodes.

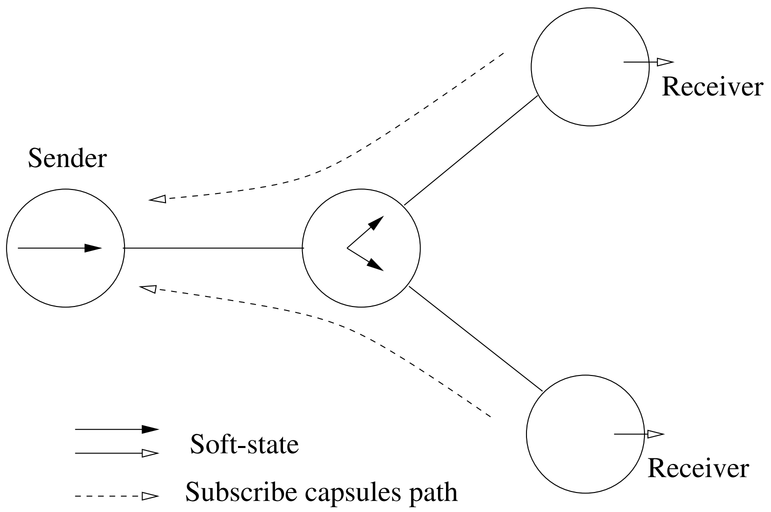



Figure 1: Multicast Distribution Tree.

Multicast-data capsules are delivered to group members following the information left by subscribe capsules. Therefore, next-hop decision is based on group soft-state, instead of destination addresses, that is, an active node forwards multicast data capsules without needing routing service.

To guarantee service reliability, the sender and intermediate nodes retransmit multicast data, as proposed by ARM. There are multicast data caches in the intermediate nodes which take part in the multicast tree, to perform scoped and fast retransmissions. In this context, active nodes near to mobile hosts current location play a fundamental role. If the greater number of losses stands for handoffs, and high error-bit rates over wireless links, these nodes would be responsible for the majority of the retransmissions, otherwise there would be an accumulation of unreceived data capsules, degrading the service performance.

# 3 Protocol Description

Four capsules define the protocol that provides the service: subscribe capsule, multicast-data capsule, repair-request capsule, and end-of-session capsule.

All mobile hosts interested in receiving multicast data of a group should send periodically subscribe capsules to the source of this multicast group. Besides, a mobile host should send a new subscribe whenever a handoff occurs, for updating multicast tree topology, as illustrated in figure 2. With this purpose, the link level indicates all handoffs. The time period used by mobile hosts to send subscribe capsules depends on soft-state expiration time, which is the same for all active nodes. There is a trade-off between communication continuity and network flooding. Subscribe capsules should be sent within a determined period in order to maintain a continuous communication, but this period should not be too short, to avoid network flooding. Thus, subscribe dispatch period ($T_e$) should be less than soft-state expiration period ($T_s$), aiming to compensate queueing delays. Figure 3 shows two situations where $T_e < T_s$. In case 1, that $T_s > T_e > 0.5T_s$, if a subscribe capsule is lost then communication continuity is broken. In case 2, that $0.5T_s > T_e$, if one subscribe capsule is lost, then communication continuity is not broken, since another subscribe capsule reaches the source in time to update the group soft-state, compensating the previous subscribe loss. In the latter case, there would be loss of continuity just in case of congestion. A percentile value of $T_s$ could be calculated to support multiple subscribe losses. Nevertheless, the extra time used to allow one subscribe loss is enough to avoid interruptions in communication, considering that there are other receivers sending subscribe capsules.

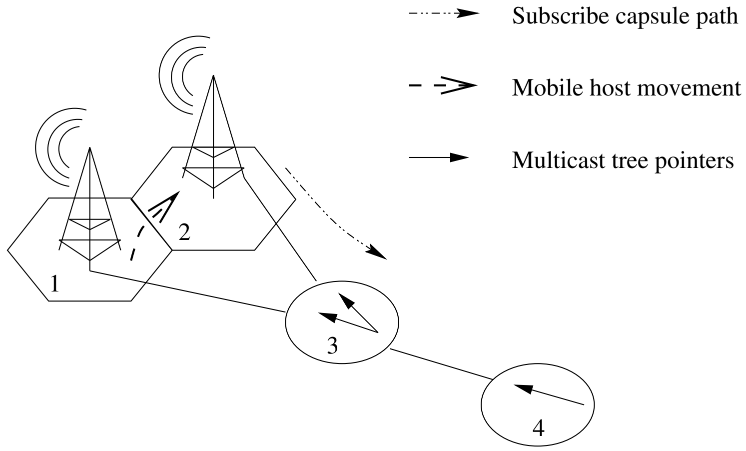
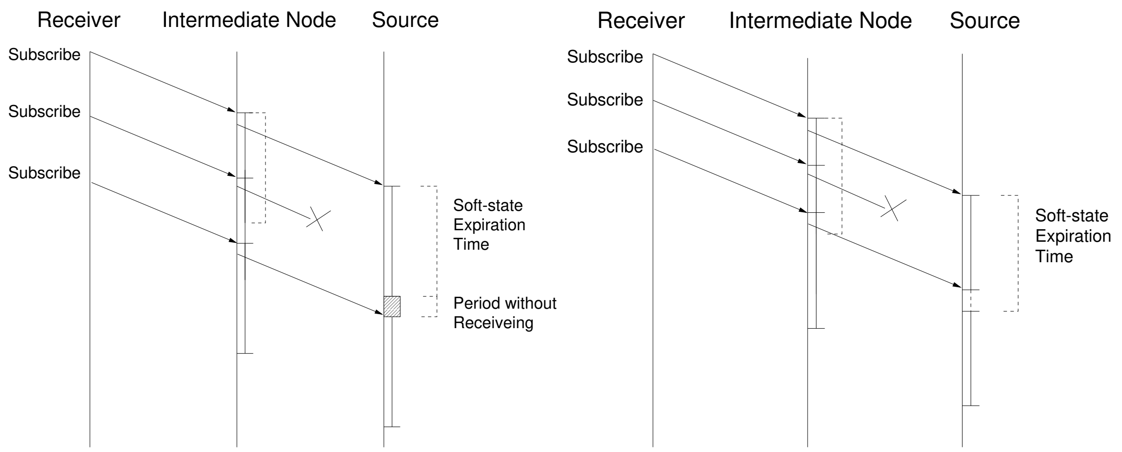



Figure 2: Mobile host changing cells.

The subscribe program uses the value of $T_e$ to filter exceeding subscribe capsules, i.e., only the subscribes capsules essential to update the soft-state of active nodes are forwarded, and the others are discarded. To decide if a subscribe capsule should be discard, the subscribe program checks a time stamp, containing the value of $T_e$, updated by active nodes every time a capsule writes in the soft-state. In addition to this time stamp to the whole group soft-state, each entry must have also a time stamp updated by the subscribe program. The time stamp of each entry is updated by the capsules that come from the interface the entry points to. Moreover, the subscribe program verifies expired entries and eliminate them, so that multicast-data capsules do not follow paths that do not belong




(a) Case 1

(b) Case 2

Figure 3: Subscribe dispatch.

anymore to the multicast tree.

Multicast-data capsules follow indications left by subscribe capsules until they reach the receivers. Thus, the main function of the multicast-data program is reading group indications inside soft-state, and replicating a multicast-data capsule to each entry. Another function of this program is looking up a multicast data cache at each node. If this cache exists, multicast data should be copied to it.

Repair request capsules are sent every time losses are detected. In this service, sequence numbers are used to identify multicast-data capsules. Whenever gaps occur in sequence numbering, receivers send repair-requests to the lost numbers. If the requested sequence number is not received within a determined period of time, based on RTT (Round-time Trip), this member sends a new repair-request. In addition, one repair-request capsule is sent for each sequence number in order to lower processing time per request at each active node. At each node, the repair-request program looks up the sequence number needed in the group cache. If this number exists in the cache, this capsule program requests retransmission, and the repair-request capsule is discarded. Otherwise, the program should mark in the group soft-state the sequence number requested and continue its way to the sender. The next capsules requesting this sequence number are discarded. This indication should persist, however, for a short period of time since a sequence number can be lost more than once.

The end-of-session capsule exists to indicate when a session of a multicast group is over. Since a sender does not have neither information about the receivers of its group nor of what was received by them, this message forwards the last sequence number sent. So, receivers which did not receive the last multicast-data capsules can request their retransmission. To avoid that some group member do not receive this capsule, each receiver has a silence timer. After a determined period of time without receiving any multicast data, the receiver sends a repair-request capsule asking for the last expected sequence number. When the sender receives this request, if the requested number is higher than the last sequence number sent, it sends an end-of-session capsule. It is important to remark that end-of-session capsules are not cached, because repair-requests do not know if they are asking for the end-of-session capsule or a lost sequence number.

# 4 Simulation Results

Simulations were accomplished in order to assess the service performance. The main purpose is to demonstrate that the subscribe mechanism rearranges multicast tree fast enough to enable high handoff frequencies, and also that the use of capsule caches in intermediate nodes provides recovery delays that avoid the accumulation of lost data capsules, as mobile hosts move through cells.

The *ns* (Network Simulator) [13] was used to simulate this service. Researchers from the VINT Project (Virtual Internetwork Testbed) have been developing this simulator for assessing new protocols and services for the Internet. All support to active networks was implemented as extensions to *ns*. Furthermore, active nodes were implemented to support customized program execution, offering a set of primitives. For instance, primitives to manipulate non-transient storage were implemented.

The simulations used transit-stub topologies to achieve a better representation of the current Internet infrastructure. This kind of topology reflects the locality and hierarchy present in the Internet. Each routing domain in the Internet can be classified as either a stub domain or a transit domain. A stub domain carries only traffic that originates or terminates in the domain. Transit domains do not have this restriction [14]. An example of transit-stub topology is illustrated in figure 4. The GT-ITM (Georgia Tech Internetwork Topology Models) was used to generate random graphs that characterize this kind of topology. The model used by the GT-ITM is based on some averages and probabilities given by the user to generate nodes and links in each domain. Values used in the simulations are shown in table 1.

Table 1: Transit-Stub Topology Parameters.

| | |
|---|---|
| Average number of nodes per transit domain | 9 |
| Average number of nodes per stub domain | 4 |
| Link probability between two transit nodes | 60% |
| Link probability between two stub nodes | 60% |
| Number of stub domains per transit node | 1 |
| Number of transit domains | 1 |
| Number of links between two transit domains | 0 |

Each stub domain in the generated topologies represents a cellular network, and each stub node an active cellular node responsible for a cell. In each cellular network, cells are

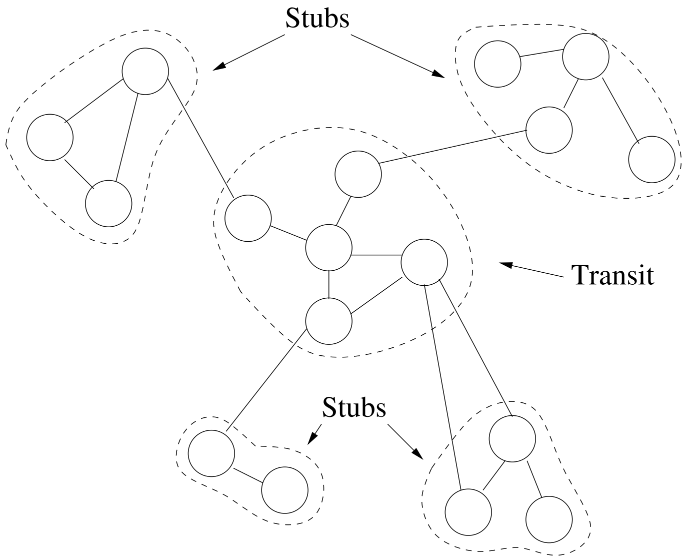



Figure 4: Transit-stub topology example.

disposed so that mobile hosts can circulate among them. Figure 5 shows cell arrangements and the simulated movement for a mobile host in cellular networks with three, four, and five cells. Links between active cellular nodes, and between a cellular node and the transit core were generated randomly by GT-ITM, with the latencies ranging from 0.0005 to 0.01 s. Moreover, data rate values of 2 Mb/s and 1 Mb/s were assumed for wireless and wired links, respectively. In order to simulate interferences over wireless links, a 10% packet loss rate was considered. Another important issue concerns simulated handoff. Handoffs of a mobile host is not based on distance, but on an average frequency, i.e., a handoff occurs after a random period of time in a specified interval, and the next cell is determined by the cell arrangement of the cellular network.

In order to evaluate the service behavior for a large number of users, an experiment was accomplished varying the number of users in a session, and measuring the average throughput considering all group members. The sender is randomly chosen among transit nodes, and it transmits a CBR (Constant Bit Rate) traffic below 1Mb/s (data rate of wired

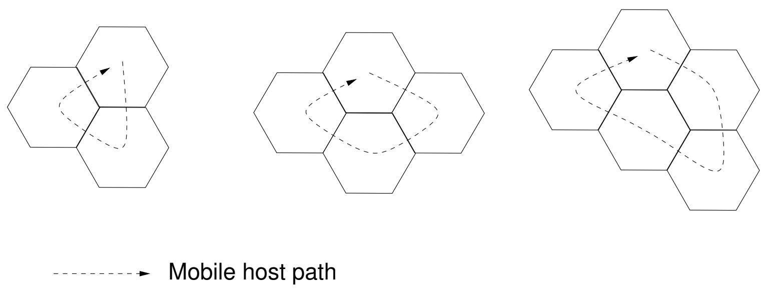



Figure 5: Cell arrangements.

link) of 400 kb/s rate (50 capsules/s, 1000 bytes/capsule). The average period between consecutive handoffs during simulation time used for each mobile host was in the interval 0.4-0.8 s, since it represents high handoff frequencies ranging from 1,25-2,50 handoffs/s. Mobile hosts were equally distributed by all cellular networks in the generated topology. Figure 6 shows the results for this experiment. The average throughput for group members is approximately constant for a group size ranging from 10 to 140 users. As the group size increases, the number of subscribes and repair-request capsules increases as well. The average throughput do not decrease, however, because of the filtering mechanisms, and scoped retransmissions. Filtering mechanisms together with scoped retransmissions avoid network flooding, since they discard excessive capsules, and do not send retransmitted data to the whole group, respectively.

Another experiment was performed to determine cached retransmissions influence. The average throughput of a multicast group was measured in face of handoff frequencies (varying the average period between consecutive handoffs of a mobile host) and considering intermediate nodes with and without cache. A CBR traffic source was used with two different data rates below 1 Mb/s: 400 kb/s (50 capsules/s, 1000 bytes/capsule), and 600 kb/s (66.67 capsules/s, 1000 bytes/capsule). Figure 7 shows the results for 400 kb/s traffic. In figure 7.a, the throughput obtained with cache in intermediate nodes are slightly higher

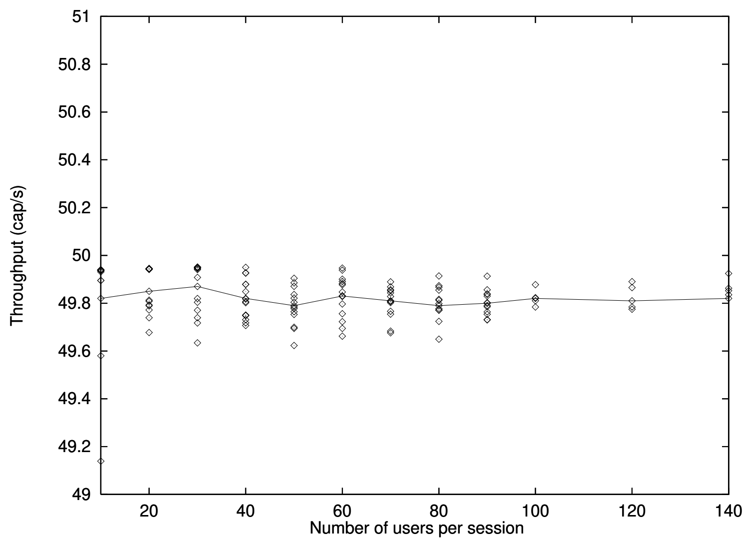
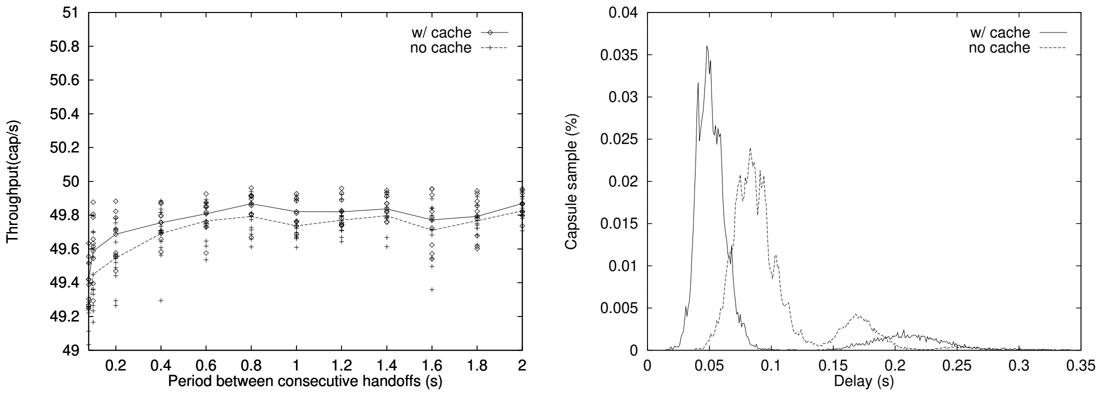



Figure 6: Number of users per session. CBR source = 50 cap/s.

than without cache, because of faster recoveries. The average throughput is approximately constant in the interval 0.2-2 s in both cases, and it decreases for values below 0.2 s due to loss recovery delay; as the time a mobile host spends to recover from a capsule loss closes the average handoff frequency, throughput decreases. This fact can be observed in figure 7.b which shows retransmission delay distribution considering all simulated average periods between handoffs. All retransmission delays for this sample are below 0.35 s. Therefore, for time periods above 0.35 s, a loss due to a handoff is recovered before a new handoff occur. In addition, errors originated by interferences over wireless links do not cause unreceived data accumulation in this case.

Using the 400 kb/s traffic source, the service without cache is able to recover from lost capsules in time to avoid unreceived data accumulation, despite the lower throughput. Nevertheless, for a 600 kb/s CBR traffic source, as presented in figure 8.a, mobile hosts are not able to recover from lost capsules in time, therefore accumulating unreceived data, and resulting in a lower average throughput. In figure 8.b, the retransmission delay

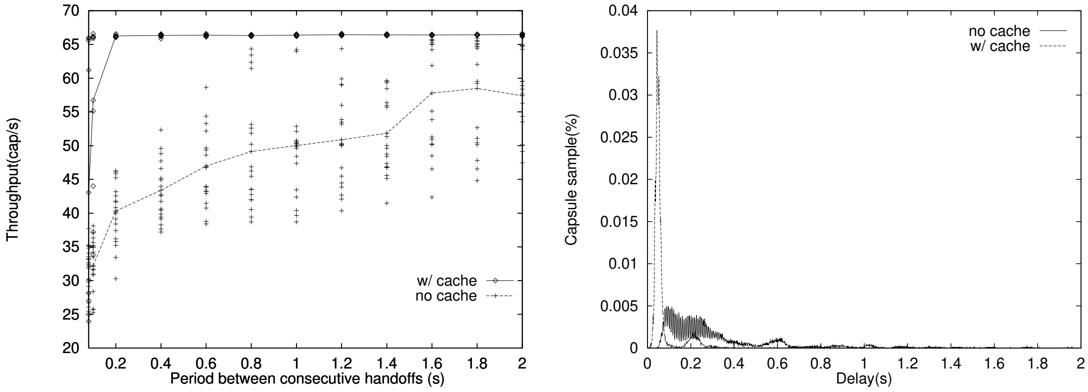



(a) Throughput (b) Retransmission delay

Figure 7: 400 kb/s CBR Traffic Source.

distribution for the service without cache shows that there are retransmission delays up to 2 s and higher, because of the network congestion caused by this accumulation. When the service is used with cache, however, retransmission delays are still limited by 0.35 s, allowing the maximum throughput for time periods between consecutive handoffs higher than 0.2 s.

Table 2 provides more details about retransmission delays. This table shows the percentile of retransmission delays lower than time periods between handoffs. For the service without using cache, and with the 600 kb/s source, the percentile of retransmission delays achieved for the time period between handoffs of 2 s is not 100%, what indicates that loss accumulation occurred. It can be noticed in this case that as the percentile increases, the throughput increases as well. A difference of 2.5% from 1.4 s to 1.6 s, and the approximation to 100% causes a higher increase in the throughput. The other three cases achieve a percentile of 100% for time periods above 400 ms.




(a) Throughput (b) Retransmission delay

Figure 8: 600 kb/s CBR Traffic Source.

Table 2: Proportion of retransmission delays below average period between consecutive handoffs.

| | CBR - 400 kb/s | | CBR - 600 kb/s | |
|---|---|---|---|---|
| Handoff (s) | No cache (%) | Cache (%) | No cache (%) | Cache (%) |
| 0.08 | 37.43 | 78.52 | 4.47 | 63.49 |
| 0.1 | 70.74 | 81.58 | 15.03 | 85.76 |
| 0.2 | 98.09 | 91.75 | 33.89 | 90.96 |
| 0.4 | 100.0 | 100.0 | 67.03 | 100.0 |
| 0.6 | 100.0 | 100.0 | 80.25 | 100.0 |
| 0.8 | 100.0 | 100.0 | 92.39 | 100.0 |
| 1.0 | 100.0 | 100.0 | 95.62 | 100.0 |
| 1.2 | 100.0 | 100.0 | 96.61 | 100.0 |
| 1.4 | 100.0 | 100.0 | 96.83 | 100.0 |
| 1.6 | 100.0 | 100.0 | 99.33 | 100.0 |
| 1.8 | 100.0 | 100.0 | 99.36 | 100.0 |
| 2.0 | 100.0 | 100.0 | 99.59 | 100.0 |

# 5 Conclusions

This paper proposed a reliable multicast service for mobile hosts in cellular networks. The use of active nodes allowed fast retransmission of lost data, and dynamic adaptation of the multicast tree when mobile hosts accomplished handoffs.

Simulation results showed that the service performs well even for high handoff frequencies, and a large number of users. Using topologies similar to Internet structure, the service supported handoff frequencies up to 5 handoffs/s. Comparisons between cached and non-cached retransmissions shows that fast capsule recovery, which is allowed by cached retransmissions, provides an expressive better throughput. In another experiment, group size was evaluated, demonstrating that this service is able to support up to 140 users in a multicast group, maintaining average data capsule throughput.

Therefore, the results presented in this paper demonstrates that this service fits well for a mobile multicast communication service. The flexibility and inside network computations provided by active networking allowed a good performance, even to a large number of mobile hosts moving fast in a cellular network with small cells.

# References

[1] H. Balakrishnan, S. Seshan and R. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", *ACM Wireless Networks*, vol. 1, no. 4, Dec. 1995.

[2] P. Manzoni, D. Ghosal and G. Serazzi, "Impact of Mobility on TCP/IP: An Integrated Performance Study", *Journal on Selected Areas in Communications*, vol. 13, no. 5, pp. 858–867, June 1995.

[3] D. Towsley, J. Kurose and S. Pingali, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", *Journal on Selected Areas in Communications*, vol. 15, no. 3, Apr. 1997.

[4] S. Floyd, V. Jacobson, C. Liu, S. McCanne and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784–805, Dec. 1997.

[5] K. Calvert, E. W. Zegura, S. Bhattacharjee and J. Sterbenz, "Directions in Active Networks", *IEEE Communications Magazine*, vol. 36, no. 10, pp. 72–78, Oct. 1998.

[6] F. Junqueira, L. H. K. Costa and O. C. M. B. Duarte, "Uma Arquitetura de Roteamento com Qualidade de Serviço baseada em Redes Ativas", in *II Seminário Franco-Brasileiro de Sistemas Informáticos Distribuídos*, Fortaleza, Ceará, pp. 257–268, Nov. 1997.

[7] M. Calderon, M. Sedano, A. Azcorra and C. Alonso, "Active Network Support for Multicast Applications", *IEEE Network*, pp. 46–52, June 1998.

[8] D. J. Wetherall, J. V. Guttag and D. L. Tennenhouse, "ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols", in *IEEE OPENARCH'98*, San Francisco CA, EUA, Apr. 1998.

[9] L. H. Lehman, S. J. Garland and D. L. Tennenhouse, "Active Reliable Multicast", in *IEEE INFOCOM'98*, San Francisco CA, EUA, Apr. 1998.

[10] A. Valko, A. Campbell and J. Gomez, "Cellular IP". draft-valko-cellularip-00.txt Internet Draft, Nov. 1998.

[11] C. Perkins, "IP Mobility Support". Request For Comments 2002, Oct. 1996.

[12] B. Jabbari, G. Colombo, A. Nakajima and J. Kulkarni, "Network Issues for Wireless Communications", *IEEE Communications Magazine*, pp. 88–98, Jan. 1995.

[13] K. Fall and K. Varadhan, *NS Notes and Documentation.* The VINT Project, Nov. 1997.

[14] E. W. Zegura, K. L. Calvert and M. J. Donahoo, "A Quantitative Comparison of Graph-Based Models for Internet Topologies", *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 770–783, Dec. 1997.