# Management of Cacheable Streaming Multimedia Content in Networks with Mobile Nodes

Muhammad Mukarram Bin Tariq, Atsushi Takeshita
DoCoMo Communications Laboratories USA, Inc.
181 Metro Drive, Suite 300.
San Jose, CA 95110

***Abstract*- In this paper we discuss the delivery of cacheable streaming content in a network with mobile nodes. Caching and regional replication are important techniques for traffic localization and reduction in load on the network and the content server. However traditional caching schemes alone cannot suffice for localization of streaming multimedia traffic in networks with mobile nodes. This is mainly because of limited geographical scope of caches. Mobile nodes can easily move out of such geographical scopes in the duration of a long session such as that of a typical streaming multimedia. In this paper we present a novel technique called cache handoff that allows a mobile node to keep switching to the most appropriate caching proxy as it moves, thus counteracting the adverse mobility effects and maximizing traffic localization. Simulation results show that this technique improves the average delay by almost 25% for a very low cache hit ratio of 0.1. This technique also gives increasingly better performance for larger session durations.**

## I. Introduction And Background

Mobile networks beyond 3G are expected to be fully IP based and with emergence of such networks mobile nodes will become part of larger IP based Internet community. It is also expected that the multimedia traffic will contribute an increasing share in overall traffic in mobile networks. In fact the multimedia traffic may surpass the traditional voice traffic in mobile networks by the year 2005 [23]. Traffic localization achieved by using caching & regional content replication has played an important role in sustaining the massive size of today's Internet [2]. As mobile nodes become part of the Internet, similar caching and replication techniques would need to be introduced in mobile networks as well.

It is desirable to have the caches as close to the client as possible to maximize traffic localization, but due to usual hierarchical nature of networks themselves, placing caching servers close to the edge of the network results in reducing their scope, and thus the number of requests going through a particular caching server is reduced. If caches rely on local storage of responses from previous requests (unlike proactive replication), the cache hit ratio is usually a function of number of requests going through a particular caching server. Thus having a caching server too close to the edge may adversely affect the cache hit ratio. To overcome this problem a number of caching system designs have been proposed [16][18][19] which organize caching servers in hierarchical or mesh like structures and establish co-operation between the different caching servers to pool the content with some (or all) other caching servers in the caching system so that small scope caching servers still remains viable.

A number of caching techniques have been studied for both web content [12][18] and streaming multi-media content [1][3][8][14][17]. These studies consider a fixed caching server present close to a fixed (non-mobile) client. These caching servers serve user requests locally and thus achieve traffic localization and enhancement of user experience. Caching for streaming content has mostly been limited to pre-fetching of small initial segments (or prefixes) due to storage limitations, but we feel that caching of entire or large segments of multimedia streams will become feasible with availability of high speed and large capacity storage [9].

More recently, there have been studies focusing on use of caching proxies for web content in mobile networks [6]. However these studies do not address the caching and traffic localization for streaming multimedia content in mobile networks. One of the factors that makes the traffic localization for streaming content challenging in mobile environments is the fact that the streaming content sessions are longer in duration than the usual web content; this allows a Mobile Node (MN) to travel far from where it started in the duration of a streaming multimedia session. In such circumstances, if the multimedia content stream continues to proxy or originate through a fixed caching server/proxy (which is typically close to the edge of the network), then as MN moves farther from the caching proxy, the stream has to traverse a larger number of hops before it reaches MN. Thus the entire purpose of using a caching proxy for traffic localization is defeated.

Relying on a fixed caching proxy to serve streaming content to a MN results in degraded quality of service and increased delay for individual MN. Furthermore, the aggregate of inter-subnet traffic generated due to streams trying to *follow* the MNs as they move, may overload the network and cause traffic congestion and thus result in overall poor network performance. These problems may be more serious in heterogeneous network environments, which consist of different access technologies. In such networks logical distance (network hops) between two geographically adjacent parts of network with different access technologies may be much larger than the logical distance between geographically adjacent parts of network which provide same access technology. (See Fig. 2, for an illustration).

Our effort is to maximize the traffic localization of

cacheable streaming content in a mobile network while at the same time having the caching servers as close to the edge as possible. In this paper we present a technique called cache handoff, which can be used to localize traffic for mobile nodes that access streaming multimedia content from caching proxies residing very close to the edge of the network. The rest of the paper is organized as follows. Section II explains the cache handoff process. Section III and IV are dedicated to evaluation of cache handoff process, and discussion on simulation results. We conclude our paper in section V.

## II. The Cache Handoff Process

Lack of traffic localization may result in degraded QoS (in form of increased delay and jitter) along with inter-subnet traffic to coup with node mobility. Sufficiently large buffering can mitigate jitter at expense of increased delay. However, increased delay may not be desirable for interactive streaming multimedia (e.g. it may result in slow response to VCR like forward and rewind commands). Furthermore, buffering does not address the problem of extra inter-subnet traffic. Problem of lack of traffic localization can only be solved by changing the MN's point of attachment i.e. serving caching proxy, as the MN moves from one region (or subnet) to another. We call this process as cache-handoff.

One way to change MN's point of attachment is that the MN itself discovers any available caching proxies in the new region, which are logically (i.e. in terms of network hops) closer to the mobile node. It can then terminate the session with the previous caching proxy and re-establish the connection with the new caching proxy. The new caching proxy would then be used for both ongoing sessions and new ones. This technique however requires each MN to discover new caching proxy through means such as SLP [6] or other discovery methods.

Another approach for changing mobile node's point of attachment is that once the serving caching proxy learns that a MN has moved, it itself initiates the handover to a caching server which is logically closer to MN's new location. We prefer this second approach because the cache proxies are usually fixed and it is therefore relatively easier for caching proxies to keep track of which neighboring caching servers are available. Relying on MNs to discover the caching service would require the discovery process to be repeated for each movement of each MN and the consequent overhead and delay could be prohibitive for timely handoff execution.

Fig. 1, presents an embodiment of how the cache handoff process can be achieved. MN is initially in subnet 1, and is being served by caching proxy 1. A cache handoff capable caching proxy *subscribes* itself to receive the mobility information of the mobile node (1) by establishing relationship with a Mobility Status Subscription Server (MSSP)[a]. As MN moves to a new subnet (2), the serving

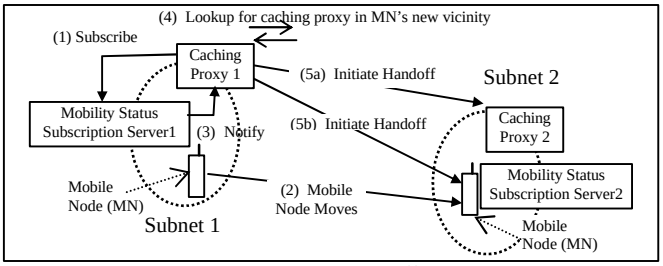


Fig. 1. The Cache Handoff Process

caching proxy is notified[b] by the MSSP about the new location of the mobile node (3). The serving caching proxy then uses the mobility information to learn if there are any caching proxies that are logically closer to MN' current vicinity then the serving caching proxy itself (4). If such a caching proxy is found (caching proxy 2 in this case) then the first caching proxy initiates a handoff to that caching proxy (5), by sending information about the mobile node, current streaming content and other authentication information.

Handoff may fail for various reasons, such as the target caching proxy may be overloaded, or it does not have the desired streaming content cached locally, or some other policy decision. The absence of cached content may not always result in a failed handoff, as the target caching proxy may fetch the content from some other source in the subnet or origin server, such as suggested in co-operative caching [8].

Handoff between the caching proxies alleviates the need of multimedia streams originating at caching proxies to be stretched and forced additional hops to coupe with the mobility of a MN, thus maximizing traffic localization.

## III. Aspects Of The Cache Handoff Process

### A. *Fragmentation of Multimedia Streams*

If the cache handoff is performed when a MN moves from one subnet to another, the session between the old caching server and the mobile node is terminated and a connection with a new caching server is established. If the content was being 'proxied' and simultaneously cached by the older caching proxy, then a cache handoff will result in partial caching of the stream at the old proxy.

If this process continues it would result in caching of (potentially randomly variable sized) fragments of streams across different caching proxies. Due to sequential nature of most streaming multimedia, these partial streams are re-useable and can be used for serving subsequent request by other clients, unlike other web contents, which may be unusable if not cached in entirety (e.g. a partially cached JPEG image may be difficult to re-use). Cache coordination

[a] The Mobility Status Subscription Server is an entity maintaining up-to-date information about the location/mobility of the MN. In a Mobile IP [10][13] network such functionality could be co-located with Home Agent(s) in a subnet, or with the mobile host itself or some other entity which has host mobility information. Caching proxy may also rely on binding update messages that it may receive from MN as a part of Mobile IP route optimization protocol [10][13]. Aki et al. [22] look into methods of making the Mobile IP protocol information available to the applications.

[b] We used SIP Events [15] framework for mobility event subscription and notification.

schemes such as those proposed in [8][16][17][18] can be used with minor variations to allow sharing of partially or fully cached multimedia streams among different caching proxies to achieve further localization of traffic. Although if co-operating caches have large overlap in what content they have in their local storage then cache-co-ordination may not yield significant advantage as shown in [20].

Since mobile nodes may move across different subnet boundaries at arbitrary times, the size of the fragments cached at individual proxies can be random, and therefore the co-ordination for sharing of such fragments would be difficult. To overcome this problem, network may only allow discrete sizes of fragments, and disallow random handoffs. We will discuss more of this issue in section IV.

### *B. Using Cache Handoff for Load Balancing*

Although serving content from a near-by source generally results in better quality of service but it may not always be true. Exceptions could be when the closest source of content is overloaded and thus incapable of providing required level of quality of service. Even in such cases cache handoff process can facilitate switching to next most optimal source of content, and achieve load balancing amongst caching proxies.

### *C. Using Cache Handoff for Networks in Motion*

Cache Handoff process can be very useful for a Network in Motion or NEMO. For example mobile nodes with passengers in a train may form an ad-hoc network, and use access point(s) onboard the train to connect to outside world. If the train's network has caching proxy(s) of its own then we have a very interesting situation. We have a moving cache! However such a scenario can be easily accommodated using the cache handoff process. We treat the moving caching proxy as just another mobile node. Since the logical distance between the moving caching proxy and mobile nodes aboard the trains remains fixed, there is no need for mobile nodes to perform cache handoff. The moving caching proxy however needs to keep switching the next level caches so that it fetches content from a caching proxy that is logically closer to the overall NEMO.

## IV. Simulation And Results

### *A. Overview of Simulation Setup*

We performed simulations using MATLAB and OPNET to estimate the improvement in end-to-end delay and jitter experienced by mobile nodes using cache handoff scheme under different network cache hit ratio and session duration conditions. Fig. 2, shows a simplified view of the simulation setup.

We considered a heterogeneous network, such as a primarily cellular network with logically non-adjacent wireless LAN hot spot subnets (as explained previously). For simplicity we assumed that the coverage areas are adjacent but not overlapping. .We added an average delay of 10

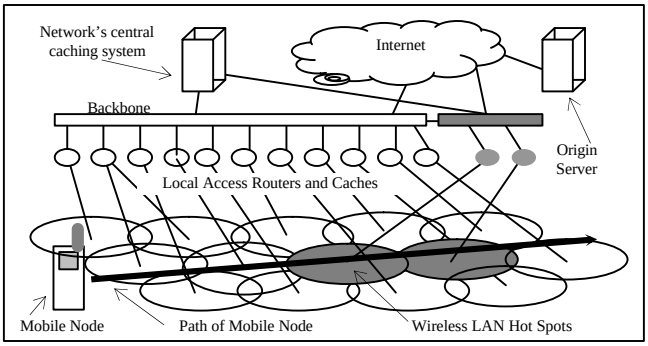


Fig. 2. Simulation Setup

milliseconds, for all the traffic in and out of these hot-spots, so as to mimic the *logical non-adjacency* with rest of the network. Each of the subnets has its own caching server, besides Network's Central Caching System (NCCS) and the origin server. Thus we had overall hierarchical caching system. The local and network cache hit ratios determine the availability of content at subnet's local caching proxy and NCCS respectively.

The simulation consisted of 1500 mobile nodes accessing streaming multimedia content at 64kbps. The mobile nodes comprised of three categories, namely, pedestrians, city traffic, and highway/train traffic, characterized by different average speeds, and moved along arbitrary straight paths.

We performed simulations for three scenarios; in first scenario (No Handoff Scenario), each MN is assigned a subnet caching proxy at start of simulation. MN always attempts to get content from the allocated (home) subnet proxy at a start of session; if the content is not available; its request is sent to NCCS, if there is a cache-miss still, content is served by the origin server. MN remains associated with the selected server for the duration of session, and repeats the process for any new sessions.

Scenario 2 (Handoff after Session Scenario) is similar to scenario 1, except that MN sends the request to the nearest caching proxy instead of a fixed (pre-allocated) proxy at the start of each session. In event of a cache-miss the request is forwarded to NCCS. If there is a cache-miss still, content is served by the origin server. The MN does not change the (caching or origin) server in the duration of a session, irrespective of which server is chosen (caching proxy, NCCS, or origin server).

Scenario 3 is similar to scenario 2, except that (caching or origin) server may change to a new local caching proxy each time a MN moves to a new subnet (even within the duration of a session). If there is a cache-miss, content continues to be served via its old point of attachment i.e. previously associated subnet caching proxy, NCCS or origin server.

We measured end-to-end delay (excluding the delay over the air interface), and jitter (as variance of end-to-end delay) experienced by mobile nodes in the network, for varying session durations and varying network and local (subnet) cache hit ratios. We made separate measurements for delay/jitter experienced by mobile nodes in logically adjacent

parts of the network (cellular subnets) and in logically non-adjacent subnets of the network (W-LAN hot spots).

### *B. End-to-End Delay and Jitter Comparison*

Fig. 3, shows the comparison of end-to-end delay for the three scenarios for fixed network cache-hit ratio (0.3) and varying session duration. Not surprisingly, mobile nodes experience least average end-to-end delay for scenario 3 (handoff during session). Also, only the curves for scenario 3 have negative gradient for increasing session duration, i.e. lesser delay for larger sessions, whereas curves for other two scenarios have positive slopes meaning increased delay for longer sessions. This is an important observation because with availability of better bandwidth, the duration of average multimedia session is expected to increase in the future.

Fig. 4, shows end-to-end delay comparison between the three scenarios under varying network and cache hit ratios. If streaming multimedia resources also exhibit Zipf [4] distribution characteristics, then it may be possible to achieve high cache hit ratios. In this comparison too, mobile nodes experience least average end-to-end delay for scenario 3 (handoff during session). It is also clear that mobile nodes in logically adjacent and non-adjacent subnets experience similar average delay for higher cache-hit ratio.

Fig. 5, shows the comparison of average jitter (as variance of end-to-end delay for individual streams) experienced by the mobile nodes in the network. The data is collected for mean session duration of 400 seconds and cache hit ratio of 0.3. Mobile nodes experience less jitter in the scenario when handoff during session is allowed, as compared with other two scenarios. Furthermore the jitter remains almost constant in this scenario for mobile nodes in logically adjacent and logically non-adjacent parts of network. The difference is only 12% or 0.46 milliseconds. Thus cache handoff allows for more homogeneous QoS even in face of heterogeneous network environments.

### *C. Overhead Traffic*

As explained in section II, the cache handoff process requires extra signaling between the caching proxies themselves and between the caching proxy and the mobile nodes the rate of this traffic depends on the size of messages

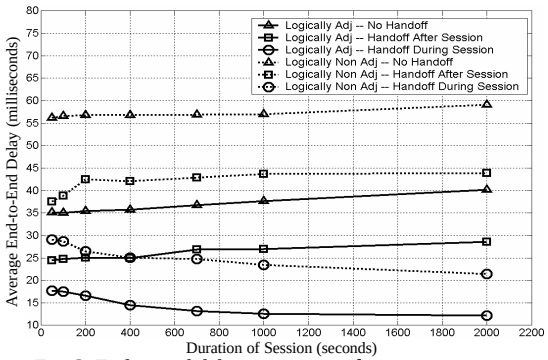


Fig. 3. End-to-end delay comparison for varying average session duration. Network cache hit ratio fixed at 0.3

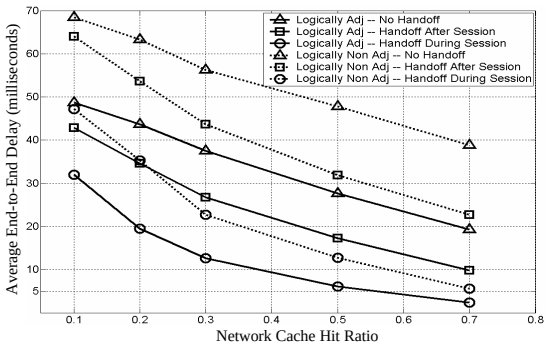


Fig. 4. End-to-End Delay Comparison for Varying Network and Local Cache Hit Ratio, Session duration fixed at 900

exchanged and the number of attempted and successful handoffs. In our simulation a typical handoff attempt requires exchanging 8 messages (starting from subscribing to MSSP to attempting handoff to target cache), totaling roughly 10 Kbytes. A successful handoff requires exchange of 4 additional messages, totaling 5 KB. In our simulations, we measured 0.26 attempted handoffs per second per subnet on the average, resulting in 20.8 kbps per subnet, and only 31.2 kbps per subnet of traffic for successful handoffs when cache hit ratio is 0.5. The peak overhead signaling traffic can be many times the average rate, such as when users traveling together (e.g. in a train) leave a subnet and enter another subnet (avalanche effect). The exact amount of signaling overhead is highly dependent on implementation (such as handoff protocol) and network configuration & conditions, such as size of subnets, user density and movement patterns.

### *D. Size of Stream Fragments*

As explained in section III.A the cache handoff process may result in caching of fragments of streams at individual subnet caches. The sizes of these fragments can be random, due to random user movement patterns. Fig. 6, shows the probability mass (using 50 second containers, e.g. fragments lying in 50~100 second range counted together) and cumulative distribution for the size of these fragments for different local and network cache hit ratio. It is evident that most of the fragments are concentrated below 300 seconds duration. (This value is network configuration dependent).

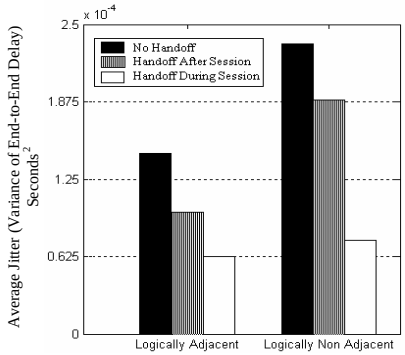


Fig. 5. Comparison of Jitter (variance of end-to-end delay) for the three simulation scenarios. (Session duration = 400s, cache-hit-ratio = 0.3)

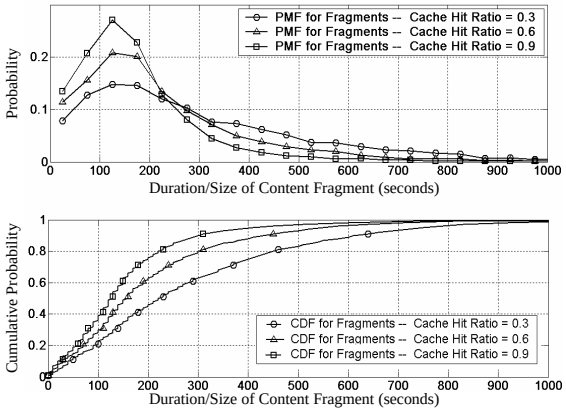


Fig. 6. Probability Mass and Cumulative Distribution for Fragments Size

This is an important trend that can be used in implementation of larger scale streaming multimedia distribution networks.

As explained in section III.A, using randomly sized content fragments for co-ordination and sharing between different caching proxies can be difficult, and it is, therefore, desirable to enforce some minimal and quantized content fragment size for handoffs. Enforcing such fragment size is likely to result in QoS degradation, because of the possibility of delayed handoffs. However, if a quantized fragment size is chosen based on careful analysis of handoff patterns such as those depicted in Fig. 6, and then there would be minimal impact on QoS.

We simulated the cache handoff process with enforced quantized fragment sizes; cache handoff is delayed until the session duration reaches a quantized level. We found that with quantized fragment durations set at 100 seconds interval and disallowing handoffs at random, the number of handoffs (and thus overhead traffic) can be reduced by 12.2% with only 6% increase in delay on the average, for a mean cache hit ratio of 0.3 and mean session duration of 700 seconds. We also observed that for sessions that are smaller or equal to the forced quantized fragment size, there is no or little advantage over the scenario wherein the handoff is only allowed after the session (i.e. simulation scenario 2). Enforcing such quantized fragment sizes mitigates the avalanche effect (introduced in previous subsection) by reducing likeliness of synchronized cache-handoffs attempts for multiple MN traveling together.

## V. Conclusion

In this paper we addressed the problem of traffic localization for cacheable streaming multimedia in a network with mobile nodes. Using fixed caching proxies to serve mobile nodes may defeat the entire purpose of using caches for traffic localization; therefore it is important that the mobile node can switch to appropriate caching proxy as it moves. Our simulation results show that delay experienced by mobile nodes can be significantly reduced if MN keeps switching to nearest caching proxy. This also maximizes the traffic localization. We also show that random user movement may result in handoffs after random periods of. time, making co-ordination and synchronization difficult. But this problem can be alleviated by enforcing quantized times in session, at which a handoff can occur, with out sacrificing much QoS.

## Acknowledgement

We would like to thank Mr. Gerald Powell, Dr. Ravi Jain, Dr. James Kempf and Mr. Shahid Shoaib of DoCoMo USA Labs, for their valuable feedback and guidance at various stages of this work.

## References

[1] S. Acharya, B. Smith. “MiddleMan: A Video Caching Proxy Server” in the proceedings of NOSSDAV 2000. June 2000.

[2] G. Barish, K. Obraczka. “World Wide Web Caching: Trends and Techniques”, IEEE Communications Magazine, Vol.38, No. 5, pp.178-185, May 2000.

[3] E. Bommaiah, K. Guo, M. Hofmann, S. Paul. “Design and Implementation of a Caching System for Streaming Media over the Internet” in the proceedings of IEEE RTAS 2000.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker. “Web Caching and Zipf-like Distributions: Evidence and Implications”. In the proceedings of IEEE Infocom, 1999.

[5] D. Funato, X. He, C. Williams, A. Takeshita. “Geographically Adjacent Access Router Discovery Protocol”. Internet Draft – Work in progress. IETF, November 2001.

[6] E. Guttman, C. Perkins, J. Veizades, M. Day. “Service Location Protocol, Version 2”. IETF RFC 2608, June 1999.

[7] S. Hadjiefthymiadas, L. Merakos. “Using Proxy Cache Relocation to Accelerate Web Browsing in Wireless/Mobile Communications”. In the proceedings of WWW10, May 2001.

[8] M. Hofmann, T. Eugene Ng, K. Guo, S. Paul, H. Zhang. “Caching Techniques for Streaming Multimedia over the Internet”. Technical Report, Bell Laboratories, April 1999.

[9] IBM Storage, http://www.storage.ibm.com/

[10] D. Johnson, C. Perkins. “Mobility Support in IPv6”. Internet draft – Work in progress. IETF, July 2001.

[11] S. Lee, W. Ma, B. Shen. “An Interactive Video Delivery and Caching System Using Video Summarization”. In the proceedings of WCW01, June 2001.

[12] A. Luotonen, H. Nielsen, T. Berners-Lee. “CERN httpd”, http://www.w3.org/Daemon/. July 1996.

[13] C. Perkins. “IP Mobility Support” IETF RFC 2002. October 1996

[14] R. Rejaie, J. Kangasharju. “Mocha: A Quality Adaptive Multimedia Proxy Cache for Internet Streaming”, In the proceedings of NOSSDAV 2001.

[15] A. Roach. “SIP Specific Event Notification”. Internet draft – Work in progress. IETF, February 2002.

[16] The Adaptive Web Caching Project. http://irl.cs.ucla.edu/AWC/

[17] S. Sen, J. Rexford, D. Towsley. “Proxy Prefix Caching for Multimedia Streams”, in proceedings of IEEE INFOCOM, April 1999.

[18] The Squid Web Proxy Cache. http://www.squid-cache.org/

[19] D. Wessel, K. Claffy. “Internet Cache Protocol (ICP) version 2”. IETF RFC- 2186~2187. September 1997.

[20] A. Wolman, g. Voelker, N. Sharma, N. Cardwell, A. Karlin, H. Levy. “On the scale and performance of cooperative Web proxy caching”. In the proceedings of 17th ACM SOSP. December 1999.

[21] K. Wu, P. Yu, J. Wolf. “Segment-Based Proxy Caching of Multimedia Streams”. In the proceedings of WWW10, May 2001.

[22] A. Yokote, A. Yegin, M. Tariq, C. Williams, A. Takeshita. “Mobile IP API”. Internet Draft – Work in Progress. IETF, February 2002.

[23] H. Yumiba, K. Imai, and M. Yabusaki, "IP-Based IMT Network Platform”. IEEE Personal Communications, October 2001, pp 18-23.