

# Adaptive Multipath Multimedia Streaming Architecture for Mobile Networks with Proactive Buffering Using Mobile Proxies

Sundan Bose and Arputharaj Kannan

Department of Computer Science and Engineering, College of Engineering, Guindy, Anna University, Chennai, India

Real-time multimedia transport has stringent bandwidth, delay and loss requirements. Providing support for such applications in infrastructure-based single hop wireless networks is a great challenge. Since mobile networks are characterized by host mobility, providing continuous streaming service in such an environment is an uphill task. In order to achieve continuous multimedia streaming, we propose an innovative multipath architecture for multimedia streaming. Existing multipath architectures are not efficient for mobile networks, where, in addition to normal streaming requirements, we need to handle the frequently occurring hand-offs. In our architecture, multiple paths, identified using an efficient genetic algorithm, are used to provide robust streaming in case of link failures. Dynamic encoding schemes are used in the server to adapt according to network conditions based on the feedback received from the network. In addition, hand-offs are predicted proactively and mobile agents containing the buffered data are migrated to the predicted base station. Altogether, the architecture provides robust multimedia streaming service under varying network conditions. We have simulated the performance of our architecture using Network Simulator (NS – 2) and the results are promising.

*Keywords:* mobile networks, IEEE 802.11, multimedia streaming, adaptive streaming, multipath transport, video transport, proactive buffering, genetic algorithm

## 1. Introduction

With the advancement in wireless technologies, wireless networks have become significant part of today's access networks. Infrastructure-based single hop wireless networks have gained widespread acceptance in the modern world. With the advent of high performance mobile devices and services such as On-Demand Video

becoming popular, there is a ubiquitous need to provide multimedia streaming service to wireless networks. In multimedia streaming, timely delivery of video/audio data is a stringent requirement. In addition, the path used for video transport should be stable for most of the session period. But in mobile networks, due to host mobility, the path needs to be changed frequently. Additionally, dynamic encoding schemes are required to adapt to changing network conditions like congestion, link failures etc.

Among various mechanisms, multipath transport is highly suitable for multimedia streaming in wireless networks. In multipath transport, multiple paths are used to transport data from one end to the other. Multipath multimedia transport is highly suitable for infrastructure-based wireless networks, where we can exploit the multiple paths existing in the wired network. Multipath transport provides robustness to the multimedia streaming system by having alternate paths in case of path failures. Multipath transport, when combined with dynamic encoding schemes, can produce good results. Dynamic encoding schemes enable the streaming server to adapt to changing network conditions. Our architecture uses feedback-based reference picture selection in which feedback from the network is used to estimate the network path status.

In order to handle host mobility, we use an approach called proactive buffering. Handover is predicted in advance and the buffer size is adaptively increased. Before the actual han-

do over takes place, a mobile proxy containing the buffered data is migrated to the base station concerned. When the mobile host moves to the new access point, the required data will be already available, thus providing continuous streaming.

The remainder of this paper is organized as follows. In Section II, the general multipath architecture is presented. Also, the related issues and prior works are discussed. In Section III, we present our proposed architecture. The various architectural modifications are discussed in detail. Section IV presents the performance analysis of our proposed architecture against existing architectures. Section V presents a discussion and the final conclusion.

## 2. Multipath Video Transport

### 2.1. General Multipath Architecture

Using multiple paths in parallel for a real-time multimedia session, called multi-path transport (MPT), provides a new degree of freedom in designing robust multimedia transport systems. In a typical multipath architecture [3], at the sender side, raw video is first compressed by a video encoder into  $M$  streams. When  $M > 1$ , we call the coder a *multistream coder*. Then the streams are partitioned and assigned to  $K$  paths by a *traffic allocator*. These paths are maintained by a *multipath routing protocol*. When the flows arrive at the receiver, they are first put into a *resequencing buffer* to restore the original order. Finally, the video data is extracted from the resequencing buffer to be decoded and displayed.

### 2.2. Issues and Challenges

(1) *Multiple Stream Generation*: One of the challenges when utilizing path diversity for video transmission is how to generate multiple coded substreams to feed the multiple paths. For MPT to be helpful for sending compressed video, one must carefully design the video coder to generate substreams so that the loss in one substream does not adversely affect the decoding of other substreams. However, this relative independence between the substreams should not be obtained at a great expense of coding efficiency. Therefore, the multistream encoder

should strive to achieve a good tradeoff between the coding efficiency and error resilience. In addition, one must consider what is feasible in terms of transport layer error control, when designing the source coder.

(2) *Path Identification*: Similar to substream generation, path identification plays a major role in MPT. Especially in mobile networks, which are characterized by frequent handovers, path identification poses a huge challenge. Also, for real-time applications like multimedia streaming, the path identification should not only generate optimal paths, but also incur less time. In addition, the paths identified should be as disjoint as possible so that when one link fails, only one or a few paths alone fail. In short, the major challenges for the path identification algorithm are as shown below:

- Identification of paths within the timing requirements
- Identification of multiple disjoint paths

(3) *Traffic Allocation*: Once the streams and the paths are in hand, the problem of traffic assignment comes into picture. The available streams must be distributed across the identified paths in a fair and efficient manner. The traffic allocator should maintain the status of the various paths and allocate the traffic accordingly.

(4) *Packet Reordering*: In multipath transport, packets are transmitted to the destination through different paths. These packets experience different delay and reach the destination at different times, thus introducing another problem. At the destination, a resequence buffer is necessary to reorder the packets to the original order. Packet resequencing incurs additional time delay and complexity.

(5) *Handling Handovers*: One additional overhead in providing multimedia streaming to mobile networks is due to host mobility. The mobile hosts roam around resulting in frequent handovers. Handover causes loss of data for certain amount of time. This may be acceptable for data services. But in case of real-time applications like video on demand (VoD) where stringent timing requirements should be met, such discontinuities are intolerable. So handover prediction becomes an integral part of multimedia streaming in mobile networks.

### 2.3. Related Work

In general, the quality of the paths may change with time. Suppose that the system receives feedback about network QoS parameters, such feedback can be used to adapt the coder and transport mechanisms to network conditions (e.g., the encoder could perform rate control based on feedback information, in order to avoid congestion in the network). In [5], an application rate adaptive technique is discussed that uses online encoding for rate adaptation. In [6], the pros and cons of various encoding schemes for MPT are discussed. [7] presents a middleware named ubiQoS for Internet video on demand (VoD). ubiQoS supports QoS tailoring depending on user preferences and terminal properties. ubiQoS exploits the Mobile Agent technology to achieve autonomous deployment and dynamic extensibility. A comprehensive study of various schemes for multipath transport in adhoc networks is presented in [3].

Another essential feature needed for multipath transport is that the encoder should be capable of generating multiple substreams from the single video source. Obviously, one way to generate multiple substreams is to use a standard video codec and split the resulting bit stream into multiple substreams. An intelligent splitting scheme is needed to split the bit stream at the boundary of independently decodable units. Otherwise, a lost substream will make the ones received from other paths useless. A simple way to accomplish this is to send the frames to the paths in a round robin manner, e.g., all odd frames are sent to path 1 and all even frames are sent to path 2. In order to completely avoid the dependency between substreams, the frames sent on one path should be coded with respect to the frames on the same path only. Some efficient coding schemes for MPT, namely feedback-based reference picture selection (RPS), layered coding (LC) with selective ARQ and multiple description motion compensation (MDMC), are discussed in [2]. These schemes differ in terms of their requirements for the transport-layer support.

In feedback-based RPS [2], based on the feedback received and predicted path status, the last frame that is believed to have been correctly received is chosen as the reference frame. A feedback message is sent by the decoder for each frame. If any packet in a frame is lost,

the decoder sends a negative feedback (NACK) for that frame. Otherwise, it sends a positive feedback (ACK). The feedback information for a frame may be sent on the same path as the frame, or on a different path. Once a NACK is received for a frame delivered on one path, the path remains “bad” until an ACK is received. Similarly, the path stays in the “good” status until a NACK is received. When encoding a new frame, the encoder deduces the last correctly decoded frame, based on the feedback messages received up to this time, and uses that frame as the reference frame.

LC with selective ARQ [2] is a scheme using layered video coding. With this scheme, a raw video stream is coded into two layers, a base layer (BL) and an enhancement layer (EL). A BL frame is encoded using the standard predictive video coding technique. The BL coding uses only the previous BL picture for prediction. The LC with ARQ codec selects from the three prediction options - upward, forward, bidirection - the one that has the best coding gain for coding EL.

MDMC is a multiple description coding (MDC) [2] scheme which does not depend on the availability of feedback channels. Instead of generating substreams that are unequal in their importance, MDC generates multiple equally important streams, each giving a low, but acceptable quality. A high-quality reconstruction is decodable from all bit streams together, while a lower, but still acceptable quality reconstruction is achievable if only one stream is received.

Once the multiple streams are generated, the paths for transmitting these streams need to be identified. This task is termed as multipath routing. Multiple description coding, combined with multipath routing, has been found to produce good results [2]. However, the difficult problem in implementing multipath routing is the identification of the best paths dynamically. The optimal paths problem can be solved by exhaustive search, which has an exponential complexity. However, a metaheuristic technique like genetic algorithms (GA) is suitable for attacking such complex problems. In addition, genetic algorithms have the intrinsic capability of handling a population of solutions rather than a single solution during each iteration. Such capability gives GA the unique strength in identifying promising regions of search space, rather than getting trapped in local optimum. A GA-

based approach [4] represents a chromosome as a path, which consists of a sequence of genes (nodes). At each iteration a number of genetic operators are applied to the current individuals to generate individuals for the next generation. The driving force in GA is the selection of individuals based on the fitness for the next generation. The survival of the fittest principle ensures that the overall quality of the population improves as the algorithm progresses from one generation to the next. Several algorithms for finding disjoint paths are presented in [9], [10].

The main challenge for multimedia streaming in mobile networks is the discontinuity due to frequent handover caused by host mobility. In order to achieve a robust streaming service in mobile networks, handover prediction is crucial. One primary design idea [1] is to dynamically

deploy mobile proxies on behalf of mobile hosts over the fixed hosts in the network locality that currently offer client connectivity. On the one hand, it permits to migrate the mobile proxies in advance to the wireless cells where mobile clients are going to reconnect, to proactively reorganize user sessions in newly visited network localities. On the other hand, it enables the proactive management of proxy-sided buffers with pre-fetched multimedia streaming content. In particular, the primary guideline is to increase the pre-fetched data in the buffer when the wireless cell handover of the corresponding client is expected to occur soon, while the proxy buffer size is decreased when the predicted probability of a handover in the near future is low. Similar to proactive buffering, [8] proposes a collection of novel middleware-based run-time

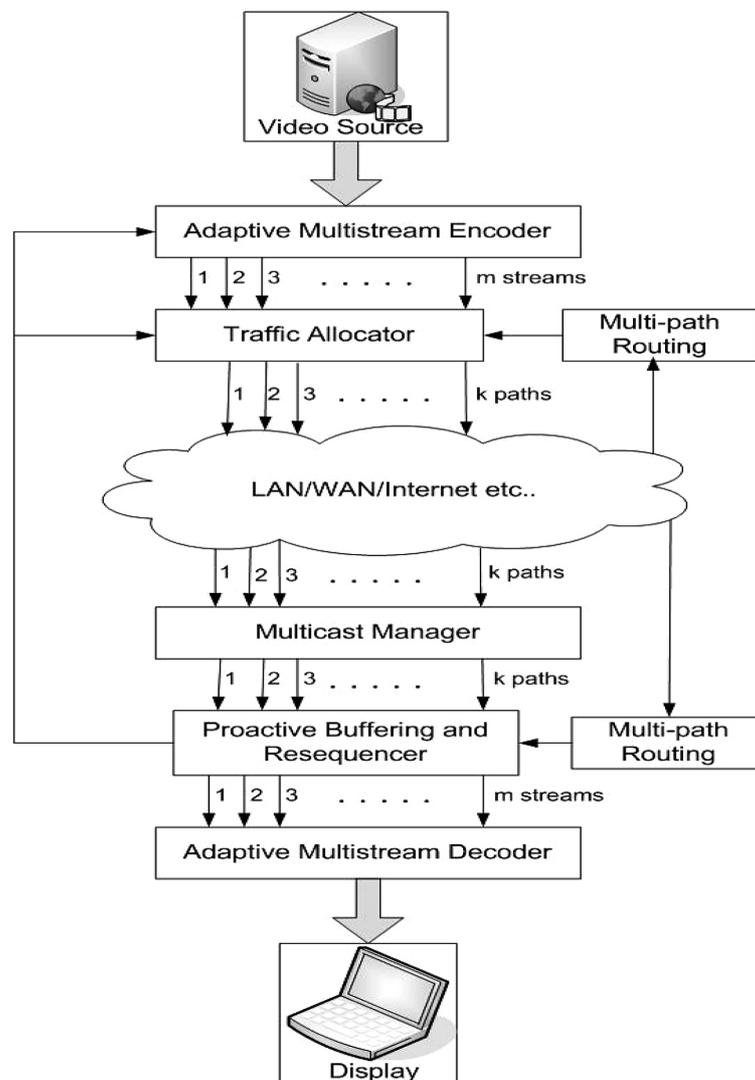


Figure 1. Architecture overview.

algorithms that ensure continuous availability of such multimedia streaming services, while minimizing the overhead involved. In this technique, the network-wide continuous streaming coverage is achieved by partition prediction and service replication on the streaming sources and assisted by distributed selection of streaming sources on regular mobile nodes and users.

### 3. Proposed Architecture

Figure 1 gives an overview of the proposed multimedia streaming architecture. The various logical modules are depicted clearly in the diagram.

The entire streaming architecture can be logically classified into three major modules as shown below:

- Adaptive Multistream Encoder / Decoder
- Multipath Routing
- Proactive Buffering

The following sections provide an insight into each of these modules.

#### 3.1. Adaptive Multistream Encoder / Decoder

In our proposed architecture, we use a modified form of feedback-based RPS as the dynamic encoding scheme.

*Feedback-based RPS:* Feedback-based reference picture selection (RPS) uses the time domain partitioning method, which selects the reference picture based on feedback and estimated path status. The coded frames are sent on separate paths. The mapping of frames to paths depends on the available bandwidth on each path.

For example, in the two-path case, if both paths have the same bandwidth, then even frames are sent on path 1, and odd frames on path 2. We assume that a feedback message is sent by the decoder for each frame. If any packet in a frame is lost, the decoder sends a negative feedback (NACK) for that frame. Otherwise, it sends a positive feedback (ACK). The feedback information for a frame may be sent on the same path as the frame, or on a different path. An encoder receives the feedback message for frame  $n - \text{RTT}$  when it is coding frame  $n$ , where round-trip time (RTT) is measured in frame intervals. Furthermore, once a NACK is received for a frame delivered on one path, we assume that the path remains “bad” until an ACK is received. Similarly, we assume the path stays in the “good” status until a NACK is received. When encoding a new frame, the encoder deduces the last correctly decoded frame, based on the feedback messages received up to this time, and uses that frame as the reference frame. A simple demonstration of feedback-based RPS is shown in Figure 2.

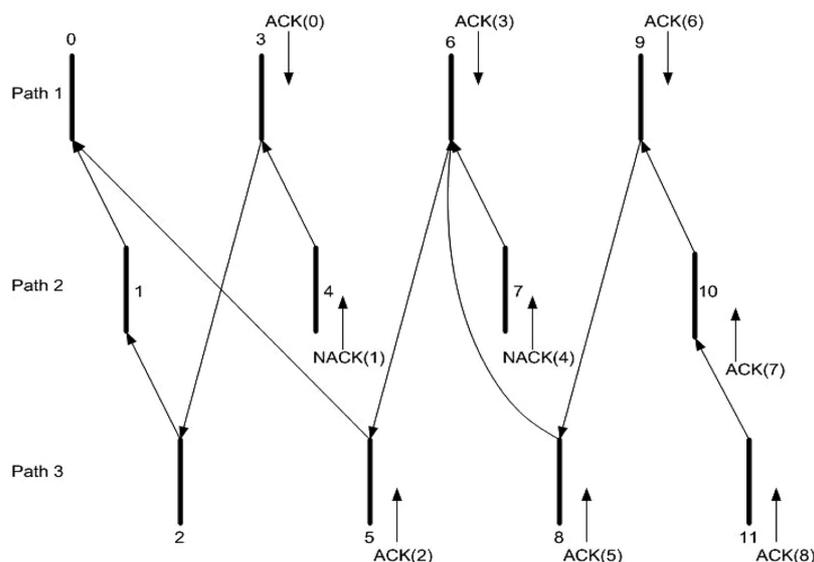


Figure 2. Feedback-based RPS – demonstration.

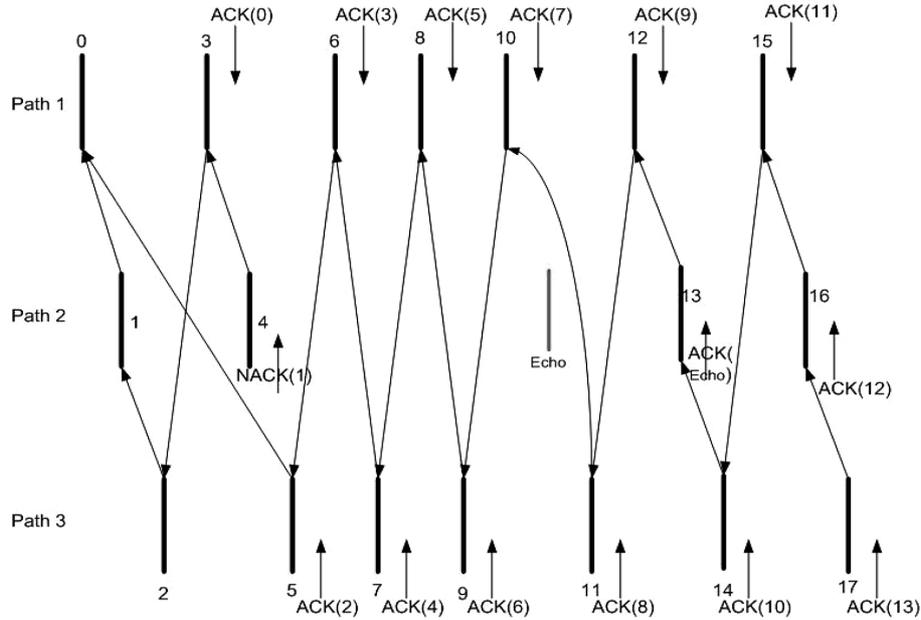


Figure 3. Modified feedback-based RPS – demonstration.

This scheme works well when the loss of a path is bursty, which is typical in ad hoc networks. A more sophisticated scheme may adopt a threshold for the NACKs and ACKs for a given window of time and switch the reference frame only when the threshold is exceeded.

*Modified Feedback-based RPS:* In the existing feedback-based RPS scheme, when a path status becomes “BAD”, the subsequent frames do not use the frames sent on the “BAD” path as the reference picture until an ACK is received; but the frames are sent as usual on the “BAD” path. This leads to a major problem. Suppose the NACK is received due to congestion on the path. Then sending subsequent frames on the same path aggravates the congestion. In our modified scheme, a path marked “BAD” is not used for sending frames subsequently. After a specified period of time, an “ECHO” packet is sent to check the status of the path. If an ACK is received for the “ECHO” packet, i.e., the “ECHO” packet is returned within a particular threshold time, the path status is restored as “GOOD” and continued as normal; otherwise no change is made to the path status. Thus no extra traffic is added to the congested route, which enables the route to be restored quickly.

### 3.2. Multipath Routing

Multi-path routing for video traffic can be formulated as a cross-layer combinatorial optimization problem, where the objective is minimizing video distortion and the constraints include connectivity, loop-free paths, and stable links. The solution space, which is exponential, consists of combinations of all feasible paths that provide a connection from the source to the destination.

Multi-path routing consists of two main sub-modules, namely:

- Path Identifier
- Traffic Allocator

*Path Identifier:* The path identifier in our architecture uses a genetic algorithm [4] to identify the multiple paths from the source to the destination. GA is a population-based meta-heuristic that is inspired by the survival of the fittest principle. Starting with a set of solutions (i.e., a population), in each iteration a number of genetic operators are applied to the individuals of the current population in order to generate offspring. Individuals with a higher degree of fitness (in the form of an objective function value) are more likely to be chosen for the next generation. The survival of the fittest principle ensures that the overall quality of the population

increases as the algorithm progresses from one generation to the next. Figure 4 shows the flow chart for the GA-based approach to the multipath routing problem. The termination condition in the diagram could be based on iteration count (generations), maximum computing time or a threshold of desired video distortion.

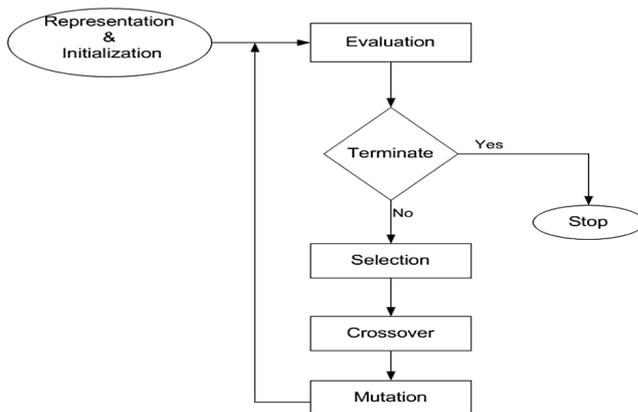


Figure 4. Path identifier.

**Solution Representation & Initialization:** In order to encode a feasible solution in the genetic format, we must define a gene first and then map a solution to a sequence of genes (chromosome). For a routing problem, a natural encoding would be to define a node as a gene. Then, an end-to-end path consisting of an ordered sequence of nodes can be represented as a chromosome. Before entering the main loop, we need to generate an initial population – initialization.

**Evaluation:** Each individual is evaluated based on the fitness function. The fitness function is closely related to the objective function (i.e. Distortion).

**Selection:** During this operation, individuals that have a greater chance or potential to produce good offspring in terms of fitness function are selected. A selection scheme known as Tournament selection randomly chooses  $m$  individuals and then selects the best of these individuals in terms of their fitness values.

**Crossover:** Crossover mimics the genetic mechanism of reproduction in which genes from parents are recombined and passed to offspring. Suppose there are two parent individuals. Then a crossover would be simply picking randomly

one path from each parent. If one or more common nodes exist, then a new offspring may be obtained by taking the head (till the common node) of one parent and the tail of the other.

**Mutation:** The objective of mutation operation is to diversify the genes of the current population, which helps the solution being trapped in a local optimum. Mutation is achieved by randomly picking a node and use random constructive approach to find a partial path from the random node to the target node. If no such path exists, then the path is kept intact; otherwise a new path is constructed by concatenating the two partial paths.

**Traffic Allocator:** The traffic allocator takes its input from the multistream encoder. It partitions the streams and assigns them to the available path. It makes use of the path identifier for identifying the various paths. The traffic allocator works on the basis of the modified feedback-based RPS technique.

### 3.3. Proactive Buffering

Proactive buffering is the feature that enables continuous streaming possible, even during handover. The main functional components of proactive buffering are resequencer and proxy migratory.

**Resequencer:** The resequencer takes as input the frames from all the ‘ $k$ ’ paths. Frames arriving through different paths experience different delays depending upon the transmission link characteristics. The main task of the resequencer is to restore the frames to their original order. The resequencer uses the sequence number to resequence the arriving packets. The resequencer places the sequenced packets in the adaptive buffer.

**Proxy Migration:** Continuous streaming in a mobile environment suffers a major problem due to Client Mobility i.e., when a mobile host moves from one BTS to another BTS, or from one MSC to another MSC, profile information of the mobile host need to be transferred to the new AP. During this interval, all the frames that arrive for the mobile host are lost. This is critical in case of applications like real time multimedia streaming.

To handle the host mobility, we use an agent-based approach in which a mobile agent is assigned to each mobile client. The mobile agent tracks the motion of the host and updates the profile information. Based on this, the agent predicts the movement of the host. A Soft Proactive (SP) strategy triggers handover only if

- The current AP RSSI is lower than a Fixed Handover Threshold (FHT).
- There is an Access Point (AP) with Received Signal Strength Indication (RSSI) greater than current AP RSSI plus Hysteresis Handover Threshold (HHT); HHT is introduced mainly to prevent heavy bouncing effects.

When the predictor notifies a proxy that its associated client is going to change its wireless cell, the proxy sets buffer size to maximum, waits for buffer fulfillment, and then commands the migration of its clone, with the fulfilled buffer, to the predicted location. If the client disassoci-

ates from the origin AP before buffer is full, the proxy immediately sends its clone to the predicted location with the already buffered data.

After clone migration, the proxy in the origin locality sets buffer size again to minimum and continues serving its client until it leaves the cell. If client entrance in the predicted cell occurs too late with regard to clone migration, part of the migrated buffer becomes obsolete. For this reason, in the case of client not arrived yet, an updated buffer is automatically re-sent. When the host actually disconnects from the original AP to the new AP, the clone agent provides the buffered data without any discontinuity. This helps achieving continuous streaming, allowing host mobility.

*Mobile Proxy – Architecture:* Figure 5 shows the architectural components of a mobile proxy. The architecture of the mobile proxy can be functionally classified into the following modules, namely:

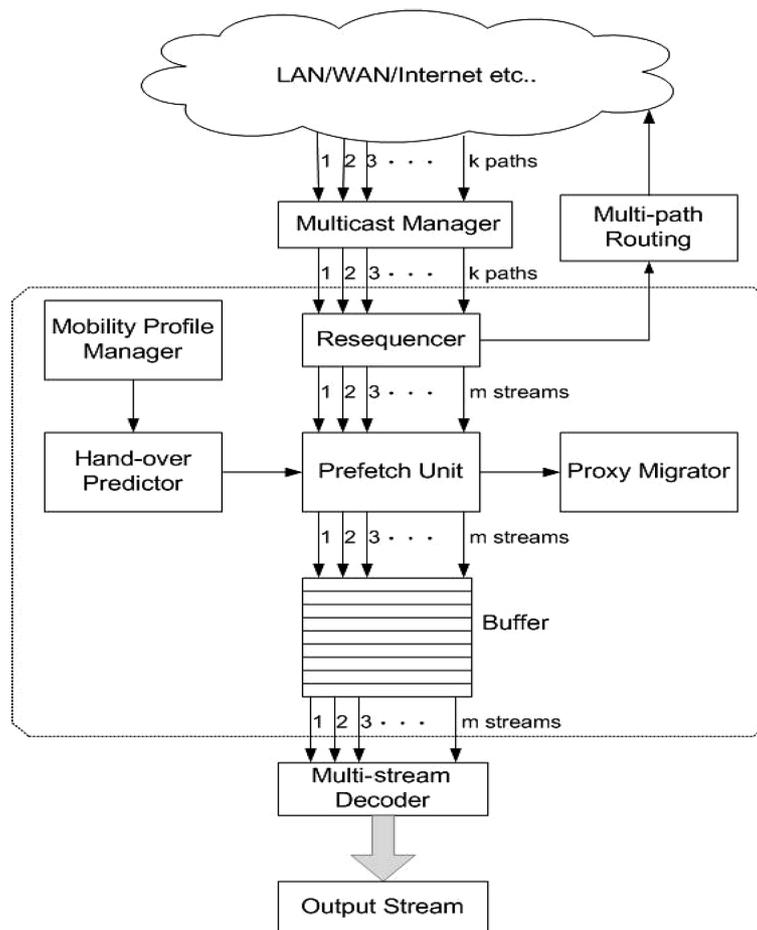


Figure 5. Mobile proxy – architecture.

- Mobility Profile Manager
- Handover Predictor
- Prefetch Unit
- Proxy Migrator

*Mobility Profile Manager:* The mobility profile manager plays a major role in the handover prediction. The mobility profile manager continuously monitors the mobility pattern of the mobile host concerned. It periodically calculates the position of the mobile host and stores it in the database. The mobility history is useful in predicting the access point to which the mobile host is expected to get connected.

*Handover Predictor:* The handover predictor is responsible for the actual predictions. For making the predictions, it uses the RSSI values as well as the mobility history collected by the mobility profile manager. When the threshold conditions are exceeded, the handover predictor triggers an alarm. Once the handover predictor triggers a handover, proactive buffering starts followed by the migration of the cloned agent called shadow proxy.

- The predictor triggers a handover when both following conditions are met:
  - $\text{RSSI of a visible AP} > \text{current RSSI} + \text{HPT}$
  - $\text{Current RSSI} < \text{FPT}$
- Handover occurs when the following conditions are met:
  - $\text{RSSI of a visible AP} > \text{Current RSSI} + \text{HHT}$
  - $\text{Current RSSI} < \text{FHT}$

*Prefetch Unit:* The prefetch unit is responsible for the management of adaptive buffers. When the handover predictor triggers a handover prediction alarm, the prefetch unit adaptively increases the buffer size and prefetches data. When the buffer gets filled, the prefetch unit alerts the proxy migrator.

*Proxy Migrator:* The proxy migrator is responsible for the migration of the proxy to the predicted AP. When the prefetch unit alerts the proxy migrator, it creates a clone of the mobile proxy called the shadow proxy. The shadow proxy is then migrated to the predicted AP along with the buffered data. Once the proxy migration is over, the prefetch unit restores the buffer to its original size.

### 3.4. Multicasting

Multicasting is another common service required for video applications like video-on-demand, video conferencing etc. Multicasting a video to a set of mobile devices is quite different from multicasting in wired network. This is due to the fact that the mobile nodes keep on moving and therefore the video needs to be buffered at different locations redundantly. Also when handoffs occur, there will be discontinuities. To avoid this, we propose a novel approach using multicast manager. The multicast manager is a static agent present in each of the MSCs and is depicted in Figure 6.

In our proposed architecture, each mobile node has a corresponding mobility profile manager (part of the mobile proxy). It collects the

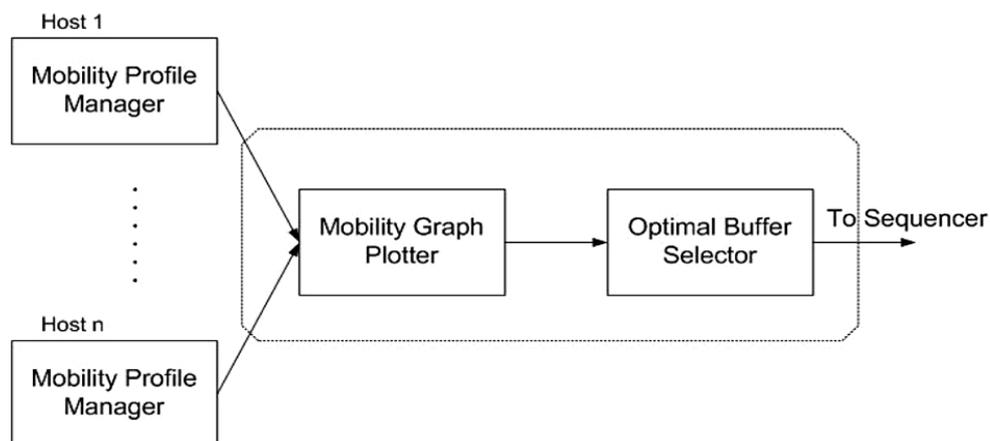


Figure 6. Multicast manager.

mobility information of the corresponding mobile node and informs the mobility graph plotter of the corresponding multicast manager in the MSC. The mobility graph plotter constructs the velocity vector graph. In the graph, mobile nodes moving towards a common point are identified and grouped. When multicasting is to be done, the multicast managers of the corresponding mobile nodes are contacted. The optimal buffer selector identifies the group to which the mobile nodes belong. If all the mobile nodes belong to the same group, then the corresponding BTS is chosen as the buffer location; otherwise the MSC is chosen as the buffer location. In this way, our approach provides a better solution for multicasting.

#### 4. Performance Analysis

The performance of our system mainly depends on the following factors:

- Efficiency of the genetic algorithm
- Handover prediction accuracy

*Genetic Algorithm:* The most attractive feature of the genetic algorithm is that it improves with generations. Since GE is based on the survival of the fittest principle, only individuals with better fitness function value are passed on to the next generation. In addition, the use of mutation prevents the algorithm from getting trapped in local optimum. Also, GE does not waste time in finding the best paths; instead, it finds paths that satisfy the fitness function. Thus, in terms of time complexity, GE is efficient too.

*Handover Prediction:* Normal handover predictors predict only on the basis of the current received signal strength indicator (RSSI) values. Instead, our handover predictor tracks the mobility pattern of the mobile host periodically. By maintaining the mobility profile, our handover predictor makes more accurate predictions. In addition, the prediction threshold also includes a Hysteresis threshold to account for the zig-zag motion.

*Simulation Results:* We simulated the performance of our architecture using NS – 2. We simulated using a network having 20 wired nodes, 5 base stations and 5 mobile nodes. The graph in Figure 6 shows the performance of our architecture under differing number of paths between the source and destination. We took a

mobile node as the client and a wired node as the server and plotted the average packet loss. The graph in Figure 7 shows that when only a single path is available, our architecture produces a performance better than the single path architecture. When there are sufficient paths available, the packet loss rate is considerably reduced to around 2%. This means that our architecture produces about three times better performance than existing architectures.

Another key point to note is that as the number of paths is increased above 5, the packet loss does not improve considerably. Figure 8 shows the trace of packet loss per frame for two substreams in SPT and our proposed MPT. The results show that our proposed architecture significantly reduces the per-frame packet loss compared to SPT, thus producing better quality streaming.

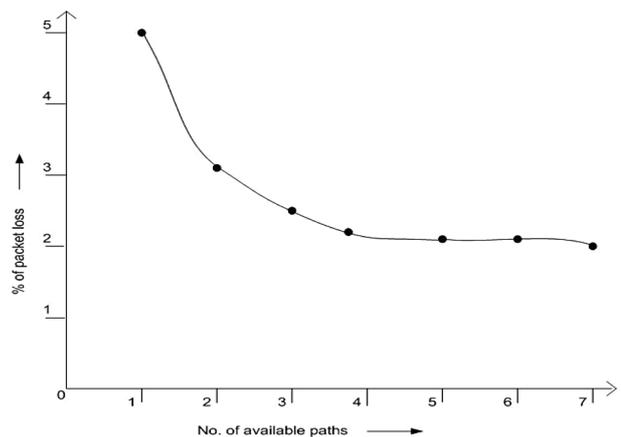


Figure 7. Graph showing the performance of the proposed architecture under different values of available paths.

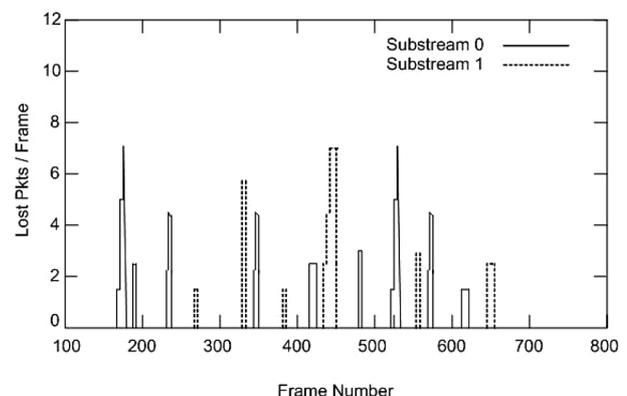


Figure 8a. Graph showing the packets lost per frame for two sub-streams in proposed MPT.

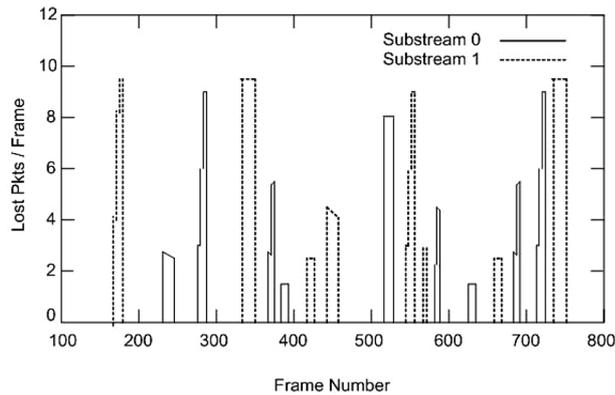


Figure 8b. Graph showing the packets lost per frame for two sub-streams in SPT.

## 5. Conclusion and Future Work

Our proposed multipath architecture for multimedia streaming in infrastructure-based wireless networks provides robust streaming service under varying network conditions. The robustness of the architecture comes from the design features like adaptive encoding, multipath routing and proactive buffering. Unlike existing architectures that suffer severely due to link failures, the multipath nature of our architecture provides some extra degree of tolerance to link failures. In addition, the use of mobile proxies enables early prediction of handover so that data can be proactively prefetched and migrated to the desired access point in advance. The mobile agents also provide scalability.

Our architecture is designed specifically to infrastructure-based wireless networks. With some minor modifications it can be adapted to adhoc environment. While adapting to adhoc networks, we can exploit the mesh topology of adhoc networks which inherently provides multiple paths between nodes. While extending to adhoc networks, we also need to take care of network partitioning. Also, our architecture does not propose any specific optimizations for multicasting. Another way of improvement may be caching. In our architecture, caching schemes are not discussed. Caching comes handy for multicast multimedia streams. By incorporating caching into the architecture, frequently accessed multimedia content can be served in a more efficient manner. But we need to handle the cache coherency problems.

## References

- [1] P. BELLAVISTA, A. CORRADI, C. GIANNELLI. Mobile Proxies for Proactive Buffering in Wireless Internet Multimedia Streaming, *25th IEEE Conference on Distributed Computing Systems*, 06. – 10. June 2005, pp. 297–304.
- [2] S. MAO ET AL. Video Transport over Ad Hoc Networks: Multistream Coding with Multipath Transport, *IEEE Journal On Selected Areas In Communications*, Vol. 21, No. 10, Dec. 2003, pp. 1721–37.
- [3] S. MAO, S. LIN, Y. WANG, S. S. PANWAR, Y. LI. Multipath video transport over wireless ad hoc networks, *IEEE Wireless Communications, Special Issue on Advances in Wireless Video*, Vol. 12, No. 4, August 2005, pp. 42–49.
- [4] S. MAO, Y. T. HOU, X. CHENG, H. D. SHERALI, S. F. MIDKIFF. Multipath routing for multiple description video over wireless ad hoc networks, *Proceedings of IEEE INFOCOM 2005*, pp. 740–750, Miami, FL, March 13–17, 2005.
- [5] I. D. D. CURCIO, D. LEON. Application Rate Adaptation for Mobile Streaming, *Sixth IEEE International Symposium on a World of Wireless Mobile & Multimedia Networks (WoWMoM)*, 13. – 16. June, 2005, pp. 66–71.
- [6] S. MAO, Y. WANG, S. LIN, S. S. PANWAR. Video transport over adhoc networks with path diversity, *ACM Mobile Computing and Communications Review*, Vol. 7, No. 1, pp. 59–61, January 2003.
- [7] P. BELLAVISTA, A. CORRADI, C. STEFANELLI. Application-level QoS Control and Adaptation for Video on Demand, *IEEE Internet Computing*, Vol. 7, No. 6, pp. 16–24, Nov. – Dec. 2003.
- [8] B. LI, K. H. WANG. NonStop: Continuous Multimedia Streaming in Wireless Ad Hoc Networks With Node Mobility, *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 10, Dec. 2003, pp. 1627–1641.
- [9] D. SIDHU, R. NAIR, S. ABDALLAH. Finding disjoint paths in networks, *in Proc. ACM SIGCOMM*, Zurich, Switzerland, Sept. 1991, pp. 43–51.
- [10] S. J. LEE, M. GERLA. Split multipath routing with maximally disjoint paths in ad hoc networks, *in Proc. IEEE ICC*, Helsinki, Finland, June 2001, pp. 3201–3205.
- [11] P. BELLAVISTA, A. CORRADI. A Mobile Agent-activated Middleware for Internet Video on Demand, *IPSP Journal, Information Processing Society of Japan Press*, Vol. 43, No. 11, pp. 3301–3315, November 2002.

- [12] P. BELLAVISTA, A. CORRADI, R. MONTANARI, C. STEFANELLI. An Active Middleware to Control QoS Level of Multimedia Services, *Proceedings of the 8th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'01)*, Bologna, Italy, pp. 126–132, Oct. 31 – Nov. 2, 2001, *IEEE Computer Society Press*.
- [13] P. BELLAVISTA, A. CORRADI, A QoS Management Middleware Based on Mobility Prediction for Multimedia Service Continuity in the Wireless Internet, *Proceedings of the 9th IEEE International Symposium on Computers and Communications (ISCC'04)*, Alexandria, Egypt, pp. 531–538, June 29 – 31, 2004, *IEEE Computer Society Press*.
- [14] P. BELLAVISTA, A. CORRADI, C. GIANNELLI. Mobility Prediction for Mobile Agent-based Service Continuity in the Wireless Internet, *Proceedings of the 1st International Workshop on Mobility Aware Technologies and Applications (MATA'04)*, Florianopolis, Brazil, pp. 1–12, October 2004, *Sprinter-Verlag Lecture Notes on Computer Science*.
- [15] P. BELLAVISTA, A. CORRADI, L. FOSCHINI. Proactive Management of Distributed Buffers for Streaming Continuity in Wired-Wireless Integrated Networks, *Proceedings of the 10th IEEE/IFIP Network Operations & Management Symposium (NOMS'06)*, Vancouver, Canada, Apr. 2006, *IEEE Computer Society Press*.

Received: June, 2006  
Accepted: June, 2007

Contact addresses:

Arputharaj Kannan  
Department of Computer Science and Engineering  
College of Engineering, Guindy  
Anna University, Chennai-25, India.  
e-mail: kannan@annauniv.edu

Sundan Bose  
Department of Computer Science and Engineering  
College of Engineering, Guindy  
Anna University, Chennai-25, India.  
e-mail: sbs@cs.annauniv.edu

---

DR. ARPUTHARAJ KANNAN, the author of this paper has been working as a Professor in the Department of Computer Science and Engineering, College of Engineering, Guindy, Anna University, Chennai 600 025, Tamil Nadu, India since January 1991. He received his M. E. and Ph. D. degrees in computer science and engineering from Anna University, Chennai 600 025, Tamil Nadu, India. His areas of interest include network security, database security, artificial intelligence, data mining and temporal databases.

---

---

SUNDAN BOSE, the author of this paper has been working as a lecturer in the Department of Computer Science and Engineering, College of Engineering, Guindy, Anna University, Chennai 600 025, Tamil Nadu, India since January 2002. He received his B. E. and M. E. degrees in computer science and engineering from Madurai Kamaraj University, Madurai 625 021, Tamil Nadu, India. He submitted his Ph. D. thesis on September 2006 from Anna University, Chennai 600 025, Tamil Nadu, India. His areas of interest include network security, database security, artificial intelligence, data mining open source software and web technology.

---