Supporting Guaranteed Continuous Media Streaming in Mobile Ad-hoc Networks with Link Availability Prediction

Min Qin

Department of Computer Science
University of Southern California
Los Angeles, CA 90089
mqin@usc.edu

Roger Zimmermann

Department of Computer Science
University of Southern California
Los Angeles, CA 90089
rzimmerm@usc.edu

ABSTRACT

As wireless connectivity is integrated into many handheld devices, streaming multimedia content among mobile ad-hoc peers is becoming a popular application. In this paper, we introduce a new QoS criterion called *path-availability-based service coverage* to mobile streaming applications. Based on this QoS criterion, we propose a dynamic service replication strategy for providing guaranteed continuous streaming service to all nodes in mobile ad-hoc networks (MANETs). Unlike previous approaches that rely on remaining link duration to determine when service replication should be invoked, in our strategy, a client requests service replication when the path availability between the client and a server instance drops below a certain threshold. This strategy is based on our previous work on continuous link availability, which proves to be helpful in achieving smooth streaming performance in MANETs. Simulation results show that our service replication algorithm can quickly increase the service coverage to 100% while reducing the chance of link breakdowns during service replication.

Categories and Subject Descriptors

C.4 [Performance of Systems]: Reliability, availability, and serviceability; C.2.4 [Computer and Communication Networks]: Distributed Systems—*distributed applications*

General Terms

Algorithms, Design, Reliability

Keywords

Link availability, Streaming, Mobile ad-hoc networks, Mobility models, Service replication

1. INTRODUCTION

With the widespread use of PDAs and other handhelds, streaming multimedia content among mobile ad-hoc peers is becoming a popular application. In our previous work, we introduced the MStream [6] architecture that implements such ideas. However, mobile ad-hoc devices are constrained by their limited communication range. As a result, a single server instance may not be able to provide streaming service

to all the mobile users in a large area, even if intermediate nodes can forward multimedia streams. When provisioning of continuous multimedia services is the major objective of Quality of Service (QoS), it is necessary to replicate streaming services to multiple nodes so that each mobile client can find at least one streaming server instance in its vicinity.

To achieve good coverage, some approaches [2] simply introduce multiple predetermined data replicas into a MANET. However, the network topology is constantly changing when users are mobile. Additionally, a path that is currently connecting a client and a server may break in the immediate future. Therefore, introducing pre-determined data replicas may not cover all the users as the network topology changes. On the other hand, dynamically replicated and created data services [3] can easily adapt to the changing network topology. For example, in Figure 1, client C holds the multimedia content and serves as a server for all the remaining nodes. However, client D is about to leave the communication area of B and the path from C to D is likely to break. In order for D to access the streaming service after the network partition, C needs to replicate the multimedia content to D before the partition occurs.

In [3], the authors introduced the NonStop protocol to provide guaranteed continuous multimedia streaming service across a MANET. Their work is based on the Reference Velocity Group Mobility (RVGM) model [7], which classifies mobile users into multiple mobility groups. Each mobility group has a logical center and group members move with a similar velocity as the logical center. In order to provide streaming services to all the mobile hosts, one node in each mobility group functions as the server and provides streaming service to the remaining nodes. When a network partition is about to take place among several mobility groups, a server predicts the remaining link duration and prepares to replicate the service to a client that belongs to the groups with no server instance. One drawback related to the NonStop protocol is that it predicts the remaining link duration by assuming that all nodes do not change their current velocities until the network partition takes place. However, the RVGM model is based on the random walk mobility model [4], which makes this assumption inaccurate if the streaming time is long. In [6, 5], we introduced a mathematical model for predicting the possibility that a link can last for t seconds. Compared to the remaining link duration, our model proves to be more accurate in representing link reliability. As we have shown in [6, 5], our model can improve the mobile ad-hoc streaming performance by reducing the number of link breakdowns and choosing better media

Figure 1: Dynamic service replication before partition takes place

sources. Another problem related to the RVGM model is the requirement to identify the mobility group for each mobile user. This requires a node to constantly monitor the distance between itself and its neighbors. However, group membership in the past often cannot represent group membership in the future, especially when nodes are highly dynamic. In many situations, each user moves independently and the NonStop protocol will try to replicate the streaming service to all the nodes in the system.

In this paper, we propose a new service replication strategy inspired by our previous mathematical model on continuous link availability. Unlike the NonStop protocol, service replication is invoked when the continuous path availability between a server instance and a client drops below a certain threshold. Our model can work with the random walk mobility model. Therefore, there is no need for each client to determine its current group membership.

2. PATH-AVAILABILITY-BASED SERVICE COVERAGE

Let us Assume that the requested streaming service requires t seconds to complete. Then, for a given path p , we define continuous path availability $L_p(t)$ as the probability that the path will be continuously available during $[t_0, t_0 + t]$ given that it is available at t_0 . In [6, 5], we proposed a mathematic model for calculating continuous link availability between two mobile hosts. Here we extend it to compute continuous path availability over multiple hops. Let $l_1, l_2, \dots, l_k$ denote the links that constitute a path p . According to [1], the correlation between two adjacent ad-hoc links is weak in a MANET. As a result, $L_p(t)$ can be approximated by

$$L_p(t) = L_{l_1}(t)L_{l_2}(t) \dots L_{l_k}(t) \quad (1)$$

Our path-availability-based service coverage specifies that a client is covered by a streaming server instance if the continuous path availability $L_p(t)$ between the client and the server instance satisfies $L_p(t) \geq \gamma$, where $0 < \gamma \leq 1$ is an adjustable threshold to determine path robustness. The implication behind this QoS criterion is that if the client begins to stream the multimedia object at the moment, the chance for the streaming process to complete without any interruption is larger than γ . As a result, a higher γ value implies a smoother streaming performance for each mobile client.

Due to node mobility, a client that is currently covered by a streaming server instance may leave that server's cov-

Algorithm 1 Dynamic service replication based on path availability analysis

```

1:  $i = NodeID$ ,  $currentavai = 0$ ,  $nextavai = 0$ 
2: if  $node(i).type = server$  then
3:    $currentavai = 1$ ,  $nextavai = 1$ ,  $path = \emptyset$ 
4:   broadcast  $updatemsg(i, currentavai, nextavai, path)$ 
    to nearby hosts
5: else
6:   while there are messages received from a neighbor  $j$ 
    do
7:      $flag = false$ 
8:     if  $L_{i,j}(t) \times node(j).currentavai > currentavai$ 
    then
9:        $currentavai = L_{i,j}(t) \times node(j).currentavai$ 
10:       $path = node(j).path + \{j\}$ ,  $flag = true$ 
11:    end if
12:    if  $L_{i,j}(t + \Delta t) \times node(j).nextavai > nextavai$  then
13:       $nextavai = L_{i,j}(t + \Delta t) \times node(j).nextavai$ 
14:       $flag = true$ 
15:    end if
16:    if  $flag = true$  AND  $currentavai \geq \gamma$  then
17:      broadcast  $updatemsg(i, currentavai, nextavai, path)$ 
    to nearby hosts
18:    end if
19:  end while
20: end if
21: if  $currentavai \geq \gamma$  AND  $nextavai < \gamma$  then
22:   Replicate the service from the given  $path$ 
23: else
24:   do nothing,  $currentavai$  is still higher than  $\gamma$ 
25: end if

```

erage after a period of time. To provide guaranteed service coverage to all mobile hosts at all times, our dynamic service replication strategy regularly calculates the continuous path availability between a client and a server and replicates the service when a client is about to leave the coverage of a server. Algorithm 1 shows the procedure of path availability calculation and partition prediction.

In Algorithm 1, each node calculates its new continuous path availability when an update message is received from its neighbors. A client can process several update messages at a time before sending out its new path availability information. To reduce the number of messages to be passed, a client does not need to send update messages to its neighbors if the path availability between that client and a server instance is less than γ . Also, a hop limit can be introduced so that messages cannot travel too far and flood the network.

Table 1: Simulation settings of our service replication strategy

Parameters	Value
Simulation area	$[0, 1000m]^2$
Communication range of all nodes	100 m
Mean epoch length of each client [4]	60 sec
Path availability calculation interval	10 sec
Initial number of servers	3
Maximum velocity of each node	3 m/s

Figure 2: Service coverage ratio as a function of time, $N = 200$

When a mobile node detects that it is about to leave the coverage of a server and cannot find another server nearby, it should immediately contact its current server and request the streaming service to be replicated. After service replication, the mobile host can serve as a new streaming server instance. To save energy, a mobile node does not need to request for service replication if another node in its vicinity is already replicating the service.

3. SIMULATION RESULTS

To study the effects of our dynamic service replication strategy, we conducted a number of simulations with the settings shown in Table 1 by using MATLAB. Initially, three randomly chosen mobile hosts provide the streaming service to all the remaining nodes. Each node's velocity is uniformly distributed between 0~3 m/s and it does not make any stops. Here we assume each client has enough power to stream media content and calculate link availability between its neighbors. When the simulation starts, hosts that are about to leave the coverage area of these three nodes request the service to be replicated to them. Figure 2 shows the service coverage ratio as a function of time, where the service coverage ratio is defined as $\frac{\text{number of nodes covered}}{\text{total number of moving nodes}}$.

In Figure 2(a), we set γ to 0.7. Because continuous link availability increases as the replication time decreases, nearly 100% of the nodes are covered at all times if replication time is less than 10 seconds. When service replication time increases, it takes more time for an uncovered node to be covered due to the longer replication time. Also, the chance for a path to break during service replication is higher. Therefore, the service coverage ratio is always lower if the replication time becomes longer.

Figure 2(b) shows the service coverage ratio with different γ values when the replication time is 20 seconds. According to the definition, a higher γ value should result in a higher QoS for each client. However, when the γ value increases, the service coverage area of a single server is reduced. As shown in Figure 2(b), the initial service coverage ratio is lower when γ is larger. However, it takes about the same amount of time for our service replication strategy to distribute the content to cover all the hosts in the network. This is because when γ is small, the chance that the service cannot be successfully replicated due to a link breakdown is also high. Therefore, a node that is replicating the service may suddenly lose its connection to the server instance and

Figure 3: Number of server instances as a function of time

Figure 4: Comparison between path-availability-based and duration-based service replication strategy, $N = 250$

need to wait a long time before it encounters another server that can provide it with streaming service.

Figure 3 shows the number of server instances as a function of time when the replication time is equal to 20 seconds. If we replicate the service to all the nodes in the network, the service coverage ratio is naturally 100%. However, doing this is very expensive and unnecessary. In Figure 3(a), we set γ to 0.7 and start the simulation with different node densities. As shown in Figure 3(a), the total number of server instances largely depends on the node density. When N is small, more server instances are required to cover all the mobile hosts since nodes are more likely to be isolated. In contrast, when the node density is high, all the mobile hosts are likely to be connected. It takes only a few replications to cover all the hosts in the network. As a result, our dynamic service replication strategy is most effective when node density is high. Figure 3(b) shows the number of server instances with varying γ values. When γ is large, a single server instance covers fewer mobile clients. Therefore, it requires more server instances to cover all the mobile nodes.

To evaluate our service replication mechanism, we compared it with a similar strategy that is based on the remaining path duration. To achieve this, we extend the work of [3] by allowing service replication through multi-hop paths. We let each node regularly calculate the remaining link duration between itself and its neighbors based on their current locations and velocities. Thus for a given path between a server instance and a client, the remaining path duration is determined by the minimum duration of all links along

Figure 5: Service replication with hop limit, $N = 250$, $\gamma = 0.7$

the path. A client requests for service replication when it finds that the maximum remaining duration of all paths to a server is equal to the service replication time. We assume that each node moves independently and there is no need for each node to identify its group membership. Figure 4 shows the results of using these two different strategies when the service replication time is equal to 30 seconds.

In Figure 4, we execute our path-availability-based service replication strategy by setting γ to 0.7. Consequently, most of the mobile clients can find some server instances that are able to provide streaming service to them with a chance of 70% or more. As shown in Figure 4(a), our algorithm has a better service coverage ratio than the duration-based service replication strategy. Figure 4(b) shows the percentage of nodes that can immediately access the service over time. Unlike the service coverage ratio, a client is considered to be able to immediately access the service even if the path between that client and the server is about to break. Therefore, the service access ratio is identical to the service coverage ratio when setting γ to 0. As shown in Figure 4(b), clients that use our replication strategy can always find a path to a streaming server instance over time. This is due to the fact that our path availability model can reduce the number of link breakdowns [6, 5] during service replication and is thus more effective in replicating services. We found a total of 37 link breakdowns during the service replication time by using our strategy, compared to 50 link breakdowns by using the duration-based strategy. As a result, more service replication attempts ended in failure when using the remaining duration based strategy.

In a wireless environment, streaming multimedia object between ad-hoc peers that are many hops away may cause a lot of problems, including interference, high loss rate and low bit rate. To avoid these problems, we extend our service replication strategy by introducing a hop limit between a server instance and a client. In Algorithm 1, a client does not need to send update messages to nearby hosts if its hop count from a server instance reaches the limit.

Figure 5 shows the simulation results of our service replication strategy when the hop limit is enforced. A server instance covers fewer clients if the hop limit is smaller. As shown in Figure 5(a), the smaller the hop limit, the lower the initial service coverage ratio. However, our dynamic service replication algorithm can quickly increase the service coverage ratio to 100% even if multi-hop streaming is not allowed. Figure 5(b) shows the number of server instances over time. Because the number of clients that each server

instance can cover largely depends on the hop limit, more server instances are required to cover all the hosts if the hop limit becomes smaller. According to Equation 1, continuous path availability is affected by the number of hops in the path. It is less likely that a path with a lot of hops may have continuous availability larger than γ . Therefore, the impact on the number of server instances diminishes as hop limit increases.

4. CONCLUSIONS

In this paper, we introduced a dynamic service replication strategy based on our previous work on continuous link availability. We showed that our strategy can efficiently increase the service coverage ratio to 100% while keeping the number of replicas low when node density is high. Also, our work proves to be more advantageous compared to those that use the remaining path duration, such as [3].

We plan to extend our work in multiple directions. First, we intend to study the energy efficiency of our service replication strategy. Second, we will incorporate other mobility models that can benefit greatly from our service replication strategy, such as the RVGM model. Finally, for large multimedia objects, we have to split them into several small pieces in order to for our strategy to work. Therefore, how to determine each piece's size is an interesting future topic.

5. REFERENCES

- [1] Y. Han, R. J. La, and A. M. Makowski. Distribution of Path Durations in Mobile Ad-hoc Networks - Palm's Theorem at Work. In *Proceedings of the 16th ITC Specialist Seminar*, September 2004.
- [2] G. Karumanchi, S. Muralidharan, and R. Prakash. Information Dissemination in Partitionable Mobile Ad Hoc Networks. In *SRDS '99: Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems*, October 1999.
- [3] B. Li and K. H. Wang. Nonstop: Continuous Multimedia Streaming in Wireless Ad Hoc Networks with Node Mobility. *IEEE Journal on Selected Areas in Communications*, 21(10):1627–1641, December 2003.
- [4] A. B. McDonald and T. F. Znati. A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1466–1487, August 1999.
- [5] M. Qin and R. Zimmermann. An Enhanced Link Availability Model for Supporting Multimedia Streaming in MANETs. Technical Report 06-878, Computer Science Department, University of Southern California, June 2006.
- [6] M. Qin, R. Zimmermann, and L. S. Liu. Supporting Multimedia Streaming Between Mobile Peers with Link Availability Prediction. In *Proceedings of the 13th annual ACM International Conference on Multimedia*, pages 956–965, November 2005.
- [7] K. H. Wang and B. Li. Group Mobility and Partition Prediction in Wireless Ad-hoc Networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, April 2002.