

Replication of Partitioned Media Streams in Wireless Ad Hoc Networks

Shudong Jin

Department of Electrical Engineering and Computer Science
Case Western Reserve University, Cleveland, OH 44106-7071

jins@cwru.edu

ABSTRACT

Media streaming in wireless ad hoc networks is challenging due to the stringent resource restrictions and the decentralized architecture. To support long and high-quality streams, one viable approach is *divide-and-conquer*. A media stream is partitioned into segments, and then the segments are replicated in a network and served in a peer-to-peer fashion. It alleviates resource requirements on light-weight devices, improves load balancing, and provides an opportunity for fine-grain replication, among others. This paper describes a peer-to-peer service model using this approach, and in particular, studies replication strategies for the segments. We exploit topological properties of the underlying networks, and exploit correlation of streaming access. Several strategies are described and evaluated. A novel strategy uses adaptive and selective replication. It infers end-host clustering from hop-distance, and selectively replicates media segments to avoid starving any of them. Preliminary simulation study demonstrates its effectiveness in minimizing the cost to discover and retrieve media data.

Categories and Subject Descriptors: C.2.1 [Computer Communication Networks]: Network Architecture and Design—Network Topology; C.2.5 [Computer Communication Networks]: Internet; G.2.2 [Discrete Mathematics]: Graph Theory—Network problems.

General Terms: Design, Performance.

Keywords: Multimedia Streaming, Data Replication, Topology.

1. INTRODUCTION

An ad hoc network [4] is a collection of wireless mobile hosts forming a temporary network without a centralized administration. The mobile hosts or pervasive end-user devices are small and light enough to carry them around. They have wireless network interfaces, but due to the limited range of their transmissions, a multi-hop network must be constructed and the hosts need cooperate to perform communication tasks. Media streaming in such networks is attractive, ranging from short messages to medium-length news clips, and to long high-quality movies. Media streaming is also challenging due to stringent resource restrictions at the mobile hosts, dynamic network connectivity, and potentially high loss rate.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'04, October 10-16, 2004, New York, New York, USA.
Copyright 2004 ACM 1-58113-893-8/04/0010 ...\$5.00.

To support media streaming applications with limited resources, one approach is divide-and-conquer: partitioning a media stream into segments and managing them in a peer-to-peer fashion. For example, a ten-minute clip is partitioned into ten segments of equal size and distributed into an ad hoc network. If any end-user wants to display it, the device first sends queries to its neighbors to ask for content. Then from multiple sources, the device receives the segments and *continuously* plays out the entire clip. The primary advantage of this approach is that it eliminates the requirement on large memory space in the devices. This is important if we want to support long and high-quality media streams. Additional advantages include more balanced load among mobile hosts. Certainly, this approach brings us complexity on searching for multiple segments and reassembling them.

This paper describes an ongoing study. We focus on replication strategies for media segments in complex networks such as peer-to-peer and ad hoc wireless networks. Our goals are to develop, analyze, and implement replication strategies that exploit network topological properties and streaming access characteristics. This paper describes and evaluates several strategies. The novel *adaptive selective replication* strategy leverages network hop-distance to infer end-host clustering, and replicates media segments to avoid starving any of them, for any streaming access patterns. Preliminary results demonstrate its effectiveness in reducing cost to search and retrieve media data in a peer-to-peer fashion. The remaining of this paper is organized as follows. Section 2 briefly describes our service model and illuminates the importance of network topology and streaming access patterns. Section 3 describes several replication strategies and Section 4 compares them using simulation study. We summarize the work and brief future work in Section 5.

2. SERVICE MODEL AND PROBLEM

In the service model, a media stream is partitioned into M segments of equal size. The size of segments is determined such that each node in the network can keep one segment in its memory space. Here the nodes are assumed to be uniform and have comparable memory space, although this assumption can be relaxed. During a streaming access by node P , there are two phases: query phase and streaming phase. In the query phase, P sends queries to its neighbors and its neighbors forward the queries, and so on. The queries can be flooded into the network, with a time-to-live (TTL) value. The TTL value is used to limit the scope of flooding. Whenever a query is forwarded to the next hop the TTL value decreases by one, until it is equal to zero. The TTL may start from one, and it increases if there are still segments not found. Each node, upon receiving a query for segments, returns a message to the previous hop. Query results from multiple nodes are cumulated in a single message. Node P eventually receives messages indicating the sources

of segments and knows the hop-distance of each replica. In the streaming phase, node P determines from which node to retrieve each segment. A simple solution is to pick a source node within the least number of hops. This service model aims at minimizing the total network hop-distance. Finally, node P continuously receives data from the sources according to the schedule. Note, this schedule may change due to the dynamics of the network. It is possible node P has to send out new queries.

One key issue is how to replicate the segments in the network. Replication can be done in different ways, *i.e.* via reactive pulling and proactive pushing. This paper considers the case of reactive pulling. During the streaming phase, node P determines which segment to keep in its own memory space, or not to keep any at all. We should point out that the strategies described here can also be adopted in proactive pushing systems.

Before describing several replication strategies, we first realize that the replication problem is non-trivial, since the general k -cache location problem [8, 6] is already NP-hard. The problem here is not only to determine the number of replicas for each segment, but also to determine the location of each replica. We have the following claim (proof is omitted due to space limitation): The problem of determining the optimal number of replicas for each segment and the location of each replica in a general network is NP-hard. It means even in a relatively small network with only tens to hundreds of nodes, it is computationally infeasible to find the off-line optimal solution. Therefore, in large networks we should seek heuristics to find good sub-optimal solutions.

To develop heuristics for replication solutions, we need first understand the characteristics of the networked system. This work exploits two categories of system characteristics: network topological properties and workload characteristics.

2.1 Topology of Complex Networks

Many networks exhibit complex behavior that is much different from regular graphs and random graphs. One social network model is small-world graph [10]. Small-world graphs exhibit connectivity properties that are between random and regular graphs. Like regular graphs, they are highly clustered; yet like random graphs, they have typically short distances between arbitrary pairs of vertices. Recently, small-world graph was proposed to model various networks in the computer science society, including the Web graph, the Internet connectivity, peer-to-peer networks, and potentially wireless ad hoc networks. In particular, Helmy [2] showed if long-range connections are established in a *spatial* ad hoc network, then it is likely to become a small-world. In other words, there is an incentive to create remote connections to decrease the diameter of the network for efficient searching.

Naturally, a question is, can we exploit such small-world properties to develop heuristics and improve the performance of replication strategies. For example, the nodes in a local-area sub-network can be highly clustered and well connected. Obviously, we hope to migrate media segments into the nodes of a sub-network such that these nodes store distinct replicas to be shared by the community. Intuitively, since a network (such as an ad hoc network) has strong clustering, there is a good opportunity to do so, and it is more important for us to develop heuristics aware of the topological properties.

2.2 Characteristics of Streaming Access

The second category of system characteristics is the unique streaming access patterns. Media streaming is different from simple file downloading. A streaming access is a process with richer properties and the end-user fully participates in the process [7, 1, 9]. For

example, a streaming access can be incomplete, *i.e.*, partial access due to the end-user's interruption. This may happen if there is a lack of interests at the user side, or if the quality of service (provided by the networks) is simply not acceptable. The second case is more likely in wireless ad hoc networks where packet loss rate can be high and may vary over time. In addition, there are also user inter-activities during a streaming access session. For example, a user is more interested in a specific part of a movie and jumps around frequently.

These access patterns have two implications on fine-grain replication of a media stream. First, there is a non-uniform probability of requesting each segment. For example, with partial access, the earlier segments are more frequently requested. Let p_i denote the probability to access the i -th segment, where $1 \leq i \leq M$. Then $p_1 \geq p_2 \geq \dots \geq p_M$. Second, requests for different segments are correlated. Consecutive segments are often requested together. In the partial access case, request to the j -th segment implies the i -th segment is also requested, given $i \leq j$. To design replication strategies, we should consider these properties. Although a single media stream has been partitioned into multiple segments, we should be aware of that they are correlated and should be replicated in a coordinated way.

3. REPLICATION STRATEGIES

This section describes several heuristic-based strategies and illustrate the intuition behind them.

3.1 Exploiting Topological Properties

In a fully decentralized system, it is desirable for a node to make a decision independently, without consulting any other nodes. One approach is to make random choices:

STRATEGY 1. *Random replication.* A randomly chosen segment is stored in a node.

Initially, a node may not have any segment. After it requests segments, it randomly picks one of them to keep. It is possible that a new segment replaces an existing segment. The intuition behind this strategy is, random decision tends to have uniform number of replicas for all segments (assuming that the segments are requested with the same probability). In addition, random replication does not require any auxiliary information about the segments.

However, this random strategy may not work well in complex networks. Given that many networks exhibit clustering behavior, a good replication strategy should exploit these properties. In other words, we need *topology-aware* or *clustering-aware* solutions that can adapt to current demands. To achieve this goal, we can use hop-distance as a heuristic, *i.e.*, replicate segments which can not be discovered in neighboring nodes. The hop-distance is a handy piece of information obtained in the query phase. We consider:

STRATEGY 2. *Adaptive replication.* A segment with the longest path to retrieve is stored in a node.

3.2 Exploiting Access Patterns

A subtle problem is streaming access has unique characteristics. The segments have varying *popularity*; the requests to them are correlated. One access pattern is partial access in streaming applications. In many applications, it is possible that the requesting node aborts its access. Replication strategies should consider the popularity of the segments. We distinguish two random replication strategies:

STRATEGY 3. *Random on-demand replication.* A segment randomly chosen from the m requested segments is stored in the node.

STRATEGY 4. *Random selective replication.* With probability $\frac{m}{M}$, a segment randomly chosen from the m requested segments is stored in the node.

With random on-demand replication, the node always picks a segment with probability $\frac{1}{m}$ to store. That is, the node *faithfully* replicates a segment to its local memory. However, our analysis (omitted here) show this strategy may give strong preference to popular segments, and may deprive less popular segments of their chance. That is, less popular segments will be *starved*. On the contrary, with random selective replication, the node may choose not to replace an existing segment. Our analysis (omitted here) show that, for any segment popularity distribution, this random selective strategy leads to a nice property: the expected number of replicas of a segment is proportional to its popularity. This is true even if requests to different segments are correlated.

Correspondingly, for adaptive replication, there are also two distinguishable strategies:

STRATEGY 5. *Adaptive on-demand replication.* Among the m requested segments, a segment with the longest path to retrieve is stored in a node.

STRATEGY 6. *Adaptive selective replication.* With probability $\frac{m}{M}$, among the m requested segments, a segment with the longest path to retrieve is stored in a node.

With adaptive on-demand replication, the node always picks each of the m requested segments with probability $\frac{1}{m}$ to store. Similar to random on-demand replication, this strategy may overemphasize the popularity of more popular segments. A good thing is that it considers hop-distance in replacement. When there are too few replicas of a less popular segment, likely this segment will be retrieved from a remote source and hence it is likely to be replicated. In this way, the problem of starvation is alleviated. Among all four strategies, adaptive selective replication appears to be the best. Selective replication guarantees that less popular segments will not be starved, while the use of hop-distance in replication decision minimizes the total cost.

4. PERFORMANCE COMPARISON

To evaluate difference replication strategies, we take the following methodology. First, a small-world network is generated using the model described and analyzed by Kleinberg [5]. Briefly, in this model, nodes are first distributed in a plane. Then links are created randomly between pairs of nodes, such that the probability of having a link between two nodes u and v is proportional to $d_{u,v}^{-r}$, where $d_{u,v}$ is their Euclidean distance and r is a constant. For the simulation study here, $r = 3.0$. Similar results were obtained when using a wide range of values. There are 10000 nodes in the network and their average degree is 3.0. Node degrees are the same as those in random graphs. We have calculated the average path length $L \approx 15.2$ between two nodes, and clustering coefficient $C \approx 0.08$, using the definitions of average path length L and clustering coefficient C in [10]. Note that, its average path length is slightly larger than that of a random graph, and clustering coefficient is much larger. According to [10], this network is a small-world graph.

Second, we generate workloads to drive the simulation. A long media stream is divided into 100 segments. Initially, one replica of each segment is randomly distributed in the network. Then streaming accesses are triggered at random points in the network. A streaming access may request the complete media stream, or a partial stream. In both case, the access starts with requesting the

first segment, and may stop in the middle. The popularity of the segments are skewed as a result of partial access. Earlier segments are more frequently requested as many nodes may abort the access in the middle. To model the popularity of the segments, a Zipf-like distribution is used. The probability that an access requests the i -th segment is proportional to $i^{-\alpha}$, where α is a constant and it is set to 0.5 in the simulation here.

Each node has memory space just enough for one segment (there is no loss of generality, since with larger memory space in each node, we can divide the stream into fewer segments). In addition, in the simulation described here we do not consider “free loaders”, which do not provide storage and service to other nodes. Even when free loaders are considered, the relative performance of all described replication strategies does not change much.

The main performance metric is the *number of hops* in order to discover and retrieve a replica of a segment. A good strategy tries to minimize this hop-distance. This metric is positively related to *the number of queries* flooded into the network. Another performance metric is the number of replicas of each segment in the network. A good replication strategy, may also balance *the number of replicas* of each segment even when the access to the segments is highly skewed. This is important, since more replicas in a network generally mean shorter hop-distance and better reliability (if there is a high packet loss rate), and balanced replication generally means stable quality of the entire media stream.

Figure 1 shows the average number of hops to reach out in order to discover and retrieve a segment. The random on-demand replication strategy has the worst performance. It takes clearly more hops to discover less popular segments, although few most popular segments appear otherwise. The random selective replication strategy works reasonably well; however, compared with the adaptive selective strategy, it is still worse. We have also compared the number of flooding queries injected to the network. Random on-demand replication needs about 15 times more queries than adaptive selective replication, and needs about 7 times more queries than random selective replication. In addition, two adaptive replication strategies have another advantage. We observe that with adaptive replication, the number of hops to discover different segments is more stable.

To further understand the behavior of these replication strategies, Figure 2 shows the number of replicas of each segment in steady-state (a snapshot), for the four strategies, respectively. Both x-axis and y-axis are in logarithmic scale. Random on-demand replication places the most popular segment on over 40% of all nodes. Almost as bad is adaptive on-demand replication. These two strategies clearly have a biased preference for popular segments. As a result, less popular segments are much less likely to be replicated in the network. We observe that the random selective replication indeed replicates the segments based on their popularity. In the simulation, the most popular segment is accessed 10 times as often as the least popular segments, and it is replicated about 10 times as often. On the contrary, the adaptive strategy replicates segments based on cost, *i.e.*, the number of hops to reach a segment. As a result, a less popular segment have a good chance to be replicated. This is consistent with our discussion in the previous section, *i.e.*, in a small-world network with strong clustering phenomenon, it is beneficial to replicate data in an adaptive fashion.

To summarize, adaptive replication is effective in minimizing the average hop-distance to discover the requested segments, and in reducing the variation of hop-distances for different segments. The use of selective replication avoids bias against less popular segments and guarantees they are not starved. The combined use of adaptive replication and selective replication achieves the best performance when the segments have varying popularity.

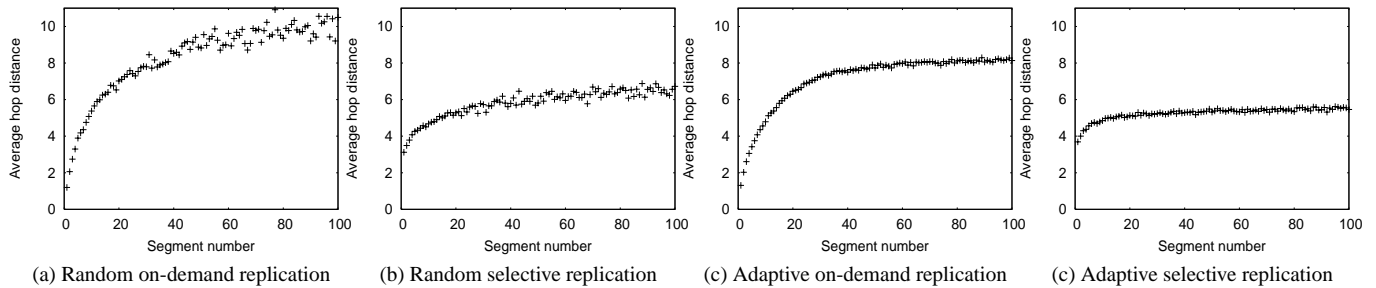


Figure 1: Average number of hops to discover a replica of a segment, as a function of the segment number.

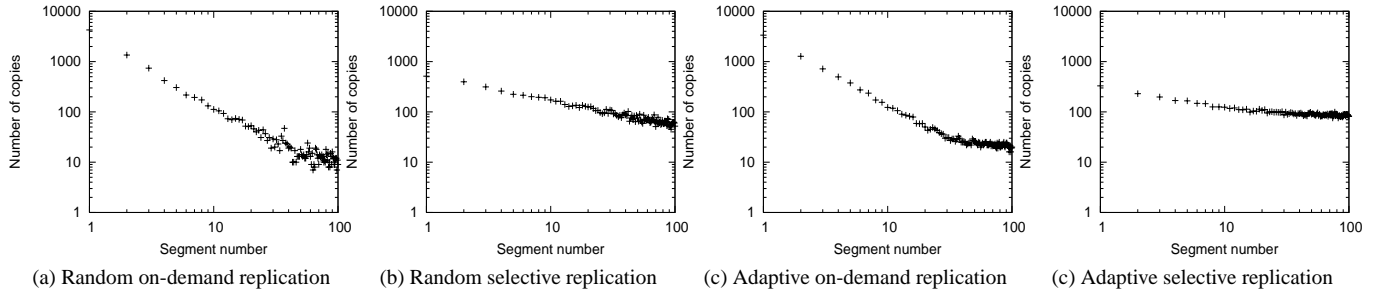


Figure 2: The number of replicas of a segment in steady-state, as a function of the segment number.

5. CONCLUSION AND FUTURE WORK

This paper has described a peer-to-peer service model for managing, searching, and streaming partitioned media objects. In particular, we have investigated and evaluated replication strategies for media segments. We first exploited network topological properties to optimize replication strategies. By doing so, our novel solution can adapt to network topology and use hop-distance to infer clustering behavior in complex networks. We have shown that capturing small-world behavior leads to noticeable performance gain. We also exploited the unique properties in streaming access. In particular, with selective replication technique, media segments are replicated without starvation.

Although the paper describes an ongoing work, the preliminary results have demonstrated that it is a very promising direction. We believe partitioning media objects is a viable approach to designing efficient and scalable streaming delivery mechanisms in decentralized systems such as wireless ad hoc and peer-to-peer networks. At this stage, we are systematically studying this approach in several directions. First, we are exploiting more topological properties, especially the small-world model described in [5], in order to design better decentralized searching algorithms [11]. In the context of this work, the searching algorithms should efficiently discover multiple correlated segments. Second, we hope to analytically study the behavior of the replication strategies for various streaming access patterns, including interactive real-time multimedia applications. We also believe there are other feasible and potentially more efficient replication solutions. Finally, we have been conducting a comprehensive evaluation of various replication strategies, using different network topologies (including real Internet and peer-to-peer network topologies), and using representative streaming workload generators [3]. Many interesting results are interpreted in an extended version of this paper.

6. REFERENCES

- [1] J. Almeida, J. Krueger, D. Eager, and M. Vernon. Analysis of educational media server workloads. In *Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, June 2001.
- [2] A. Helmy. Small worlds in wireless networks. *IEEE Communications Letters*, 7(10), October 2003.
- [3] S. Jin and A. Bestavros. GISMO: Generator of streaming media objects and workloads. *ACM SIGMETRICS Performance Evaluation Review*, 29(3), 2001.
- [4] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.
- [5] J. Kleinberg. The small-world phenomenon: An algorithm perspective. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 163–170, May 2000.
- [6] P. Krishnan, D. Raz, and Y. Shavitt. The cache location problem. *IEEE/ACM Transactions on Networking*, 8(5):568–582, October 2000.
- [7] J. Padhye and J. Kurose. An empirical study of client interactions with a continuous-media courseware server. In *Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, June 1998.
- [8] D. B. Shmoys, Éva Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *Proceedings of ACM symposium on Theory of computing (STOC)*, May 1997.
- [9] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin. A hierarchical characterization of a live streaming media workload. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2002.
- [10] D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 363:202–204, 1998.
- [11] H. Zhang, A. Goel, and R. Govindan. Using the small-world model to improve freenet performance. In *Proceedings of IEEE INFOCOM*, 2002.