

# Server Replication Techniques for Display of Continuous Media in Mobile Ad Hoc Networks

Shahram Ghandeharizadeh, Shyam Kapadia, Bhaskar Krishnamachari  
Computer Science Department  
University of Southern California  
Los Angeles, CA 90089, USA  
{shahram,kapadia,bkrishna}@usc.edu

## Abstract

Continuous media, audio and video clips, are characterized by their large size and pre-specified bandwidth requirement. If a clip is not delivered at its pre-specified rate then its display might suffer from frequent disruptions and delays termed jitters. This paper explores jitter-free display of clips with a mobile ad-hoc network of devices. Our motivating application is in-vehicle entertainment systems for both vans and luxury cars (currently realized using DVD players). We envision a car-to-car, peer-to-peer (C2P2) network of devices that collaborate with one another to support jitter-free displays. The primary contributions of this paper include: a taxonomy of techniques for continuous display and a case study evaluation of replication and switching strategies using simulation studies and analysis. Our results show that replication of content across multiple servers can significantly improve quality-of-service and the bandwidth consumption within the network.

## 1 Introduction

During the past few years, automobile manufacturers have been marketing and selling vehicles equipped with entertainment systems. These systems typically consist of a DVD player, a fold-down screen, a video game console, and wireless headphones. In its present form, storage and content are tied together. This limits the number of available titles to those DVDs and CDs in the vehicle. We envision separation of storage and content where content might be staged on demand across the available storage for previewing. This provides passengers with a large number of titles that might be either a) pre-recorded, e.g., audio or video on demand, b) time-shifted programming, e.g., previewing of live TV or radio broadcast where the display is lagging behind broadcast, or c) live broadcast. With this vision, vehicles are equipped with car-to-car, peer-to-peer (C2P2) devices. Each C2P2 is equipped with abundant amount of storage, a processor, and a wireless networking card (802.11a or b). It might integrate into the navigation system of a vehicle and its existing network for delivery of data to the on-board fold-down screen or wireless headphones. Different C2P2 devices might store different clips and exchange these clips with one another to support on-demand delivery of continuous media.

The principle characteristics of continuous media is their sustained bit rate requirement. If a system delivers a clip at a rate lower than its pre-specified rate without special precautions (e.g., pre-fetching), the user might observe frequent disruptions and delays with video and random noises with audio. These artifacts are collectively termed *jitters*. For example, CD-quality audio (2 channels with 16 bit samples at 44 kHz) requires 1.4 Megabits per second (Mbps). Digital component video based on the CCIR 601 standard requires 270 Mbps for its continuous display. These bandwidths can be reduced using lossy compression techniques due to redundancy in data. For example, with the MPEG standard, the bandwidth requirement of CD-quality audio can be reduced to 384 Kilobits per second. MPEG-1 reduces the bandwidth requirement of a video clip to 1.5 Mbps. With some compression techniques such as MPEG-2, one can control the compression ratio by specifying the final bandwidth of the encoded stream (typically ranging from 3 to 15 Mbps). The average bandwidth of a DVD quality video (MPEG-2) is typically quoted at 4 Mbps.

Compression schemes are categorized into Constant Bit Rate (CBR) and Variable Bit Rate (VBR) techniques. With both techniques, data must be delivered at a pre-specified rate. The VBR schemes have the advantage that, for the same average bandwidth as CBR, they can maintain a more constant quality in the delivered image by utilizing

more megabits per second when needed, e.g., when there is more action in a scene. Jitter-free delivery of VBR encoded clips can be conceptualized as a sequence of CBR delivery schedules [11]. To simplify discussion, the rest of this paper assumes CBR encoded clips.

C2P2 devices must collaborate to deliver a clip to a C2P2 device that is displaying this clip. This is challenging for several reasons. First, the size of a compressed video clip is large. For example, a two minute MPEG-2 encoded cartoon with an average bandwidth requirement of 2 Mbps is 30 Megabytes in size. A two hour MPEG-2 encoded video clip with an average bandwidth requirement of 4 Mbps is 3.5 Gigabytes in size. (Without loss of generality, this paper focuses on video due to its significant size and bandwidth requirements that are higher than audio.) Second, the C2P2 devices are mobile with a fixed radio range. A displaying client might temporarily become disconnected from all other C2P2 devices. The duration of such an event depends on the geographical location of the C2P2 devices, their radio range, velocity, and trajectory of movement. Third, the amount of delay observed from when a client requests a clip to the onset of clip display, termed startup latency, must be minimized. A naive approach that materializes the entire clip prior to initiating its display is undesirable. Fourth, resources must be managed intelligently in order to maximize the number of simultaneous clip displays to different clients. Ideally, with

A displaying C2P2 device may prefetch a portion of a referenced clip in order to minimize jitters. This increases the startup latency because it delays the display. The C2P2 device may consider factors such as mobility in order to control the amount of prefetched data. Another factor is the number of replicas for a clip  $X$ . With a larger number of copies for clip  $X$ , once the displaying C2P2 device encounters a path break to a C2P2 device that is transmitting  $X$ , it is more likely for this C2P2 to locate another candidate C2P2 containing  $X$ . In this paper, we analyze the number of replicas by assuming a random assignment of replicas to the C2P2 devices. Extensions of this work to analyze alternative replication strategies [3], e.g., based on a uniform, proportional or square-root of a clip's frequency of access, block versus clip level replication [6], and alternative placement of data is a future research direction.

There exists an extensive body of excellent studies in both Mobile Ad hoc NETWORK (MANET) and data delivery techniques in support of jitter-free display of continuous media with wired networks. To the best of our knowledge, this paper is a first study that explores the intersection of these two disciplines at an application level. In the following, we provide a brief overview of each discipline, starting with MANET.

In the 1990s, a large number of network-layer routing techniques have been proposed and developed for MANETs [10]. This already extensive body of MANET routing literature leaves few open problems in purely network-layer issues. However, there are still significant higher-layer design challenges in the context of highly demanding applications such as the display of continuous media. This is the subject of our paper.

In parallel, extensive research was conducted on jitter-free display of continuous media [4, 2], admission control and scheduling, e.g., Group Sweeping Scheme [14], data placement techniques, e.g., Staggered striping [1], scalable encoding techniques, e.g., Sub-band coding [13], configuration planners [5], network caching strategies [8, 12], etc. This list of prior research is not intended to be exhaustive as there are numerous excellent studies that are not cited due to space limitations. At the same time, this identified prior research must be adapted in support of jitter-free display of continuous media with MANET.

The primary contribution of this study is a taxonomy of techniques in support of jitter-free display of continuous media using MANET. We focus on one branch of this taxonomy, namely, staggered non-redundant data delivery, with DSR [7] when a single display is active. We use simulation studies to analyze the startup latency, percentage of jitter-free displays, and the bandwidth required to support this display as a function of block size, data delivery rate, number of clip replicas, and mobility characteristics. Our results demonstrate that clip-level replication of content across multiple servers can significantly improve application quality in terms of startup latency (by 75% ) and delivery ratio as well as reduce the bandwidth consumption within the network (by about 25% in our studied scenarios). Analytical models of Section 3.3 formalize the observed simulation results.

The rest of this paper is organized as follows. Section 2 presents a taxonomy of jitter-free display techniques. Section 3 focuses on one branch of this taxonomy and presents simulation results from a case study using DSR. Our conclusions and future research directions are contained in Section 4.

## 2 A Taxonomy

A C2P2 device may act as either a displaying client (termed client), a data producing server (termed server), an intermediary that routes packets from a server to a client (termed router), or idle waiting to serve in one of these roles. A C2P2 device might simultaneously be a client that displays clip  $X$ , a server that produces a fraction of clip  $Y$  for

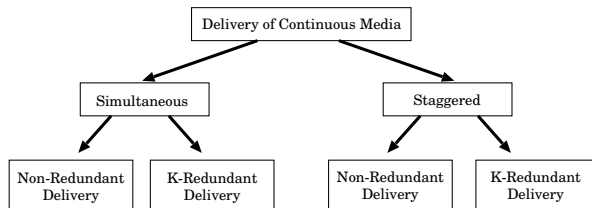


Figure 1: A taxonomy of data delivery techniques

another client, and a router for clip  $Z$ . The bandwidth of its wireless connection is the limiting factor.

Techniques to deliver blocks can be categorized into on-demand and delayed approaches. With an on-demand approach, a server delivers a block as soon as possible. With a delayed approach, a strategy dictates a schedule of block delivery at some point in the future. A delayed approach may consider the current geographical location of a client and a server, velocity, trajectory of movement, etc. This enables a client and server to anticipate whether they are either approaching, moving away, or maintaining an approximately fixed distance. Using this knowledge, a server may deliver a block to a client once they are in radio range.

An on-demand block delivery technique might be scheduled by either a client or a server. In the following, we describe each in turn, starting with client driven paradigms.

## 2.1 Client driven paradigms

With a client-driven approach, a client requests the delivery of a block from a server. Given the  $z$  blocks of a clip  $X$ , i.e.,  $X_0, X_1, \dots, X_{z-1}$ , a client identifies the candidate servers that contain a block  $X_i$ ,  $0 \leq i < z$ . This is termed the discovery process. Next, it schedules the delivery of  $X_i$  from either one or more servers. This might employ concurrent (parallel) retrieval of blocks. With simultaneous schedules, a client may request delivery of blocks at the same time from one or more servers. With staggered schedules, a client may request delivery of blocks in a time staggered manner. With simultaneous block retrieval,  $k$  simultaneous transmitters (either routers or servers) might compete for the available wireless network bandwidth to a client, reducing the link bandwidth. With staggered block retrieval, the client may control the schedule of retrieval to prevent such contention.

A client's DSR connection may request either a block or a fraction of a block (termed a fragment) from a server. A fragment is specified as: (block-id, offset, length) where 1) offset  $\leq S_b$ , and 2) length  $\leq S_b$ . For example, given a one megabyte block  $X_1$ , if a client misses bytes 1024 to 1028 of  $X_1$  then it may request  $(X_1, 1024, 4)$  from the server containing  $X_1$ . The concepts described in this section apply to both retrieval of blocks and fragments.

Both staggered and simultaneous block retrievals may schedule redundant streams of the same block (fragment) in order to compensate for transient network connections between a server and a client. Given that a client  $C_1$  identifies  $m$  servers  $(S_0, S_1, \dots, S_{m-1})$  containing a referenced block  $X_i$ , it may invoke  $\ell$  servers to produce  $X_i$ ,  $\ell \leq m$ . The value of  $m$  is dictated by the placement of  $X$ , degree of replication for  $X_i$ , and the network connectivity. The value of  $\ell$  is dictated by the strategy that might be deployed either at a client or a server.

A client may re-compute a schedule of block retrieval for a block  $X_i$  every time its connection to a server containing  $X_i$  is lost (due to mobility). Alternatively, this might be performed either immediately prior to invoking a staggered schedule after some delay, a fixed time-out, etc.

## 2.2 Server driven paradigms

With a server driven approach, one or more servers may employ schedules to deliver data to a client. These schedules are computed either by 1) a client referencing data blocks, or 2) one or more servers that receive a request for data blocks. In its simplest form, a client discovers those servers that contain the blocks of its referenced clip  $X$ . Next, it associates a time-out period  $\psi_i$  with the delivery of each block  $X_i$  from one or more servers containing  $X_i$ . One may use either a simultaneous or a staggered concurrent retrieval methodology to dictate the value of  $\psi_i$  for each block  $i$ . It is the responsibility of a server to either reject or accept this client request. If it accepts this request upon its arrival at time  $T_j$  then it must establish a DSR connection and deliver  $X_i$  prior to the expiration of the time-out,  $T_j + \psi_i$ . A more sophisticated version of this approach specifies a time-out period  $\psi_i$  with some boundary  $\delta_i$  for each block  $i$ . If

the server accepts this request at time  $T_j$  then it must deliver  $X_i$  sometimes during  $T_{j+\psi_i}$  and  $T_{j+\psi_i+\delta_i}$ . This way, the client staggers transmission of blocks in order to prevent contention for its bandwidth.

### 3 Case Study: Staggered, Non-redundant Retrieval from Replicated Servers

Our focus in this paper is primarily on data retrieval strategies for a single-client in a C2P2 ad-hoc network. In the taxonomy outlined in the preceding section, replication across multiple servers and the possibility of redundant retrievals have been identified as key techniques to improve the quality of continuous media streams across highly dynamic MANET topologies. We now investigate, as a case study, a specific technique from this taxonomy — staggered, non-redundant retrieval of data using multiple replicated servers.

Our scenario is as follows. A single client node in the C2P2 MANET initiates the display of a fixed size continuous media clip. The bandwidth requirement of this clip is 2 Mbps. We consider clip-level replication whereby the entire clip is replicated at  $S$  servers in the network. At initialization, the client initiates a resource discovery phase to identify which servers contain its referenced clip. The client then initially starts retrieving the information from the nearest server. Due to mobility this connection is likely to suffer a path break and would therefore be likely to deteriorate over time. The client therefore utilizes a staggered retrieval strategy to switch between servers. We consider two versions of switched-server retrieval: path-based switching and block-based switching.

With path-based switching, when the current routing path to a server fails due to mobility at time  $t$ , the client terminates the connection and opens a new connection to the nearest server. In block-based switching, a block-size  $B$  is pre-defined, and the client then periodically establishes a connection to the nearest server once every  $B$  bytes have been received. At this time either the connection is renewed to the same server (if it is still the nearest), or a “server switch” occurs.

Note that with block-based switching, particularly when  $B$  is large, the routing path to the current server may fail in between block-based server switches; in this case the underlying routing protocol will continue the connection by re-establishing a path to the same server. Intuitively, we would therefore expect that, particularly for low-mobility scenarios, block-based switching with large block-sizes resembles path-based switching. In our investigations, we restrict ourselves to block-based switching.

We now describe the simulation setup and results from the simulation, showing how the bandwidth usage and startup latency for the clip playback can both improve significantly as the number of servers (i.e. the level of replication) is increased. We then formalize the problem through a first-order analytical model which agrees with our simulation results and provides generalizable insights.

#### 3.1 Simulation Setup

The simulation experiments were performed in ns-2. The scenario of interest had the following parameters: there are 20 nodes in a square area of size 200mx200m. The random way point model is chosen with a pause time of 0s, with node velocities chosen randomly from  $[0, V_{max}]$ , where  $V_{max}$  varied as 5m/s, 15m/s and 30m/s. All experimental results shown are averaged over 10 runs (10 random seeds), i.e. each plotted point represents the average of 10 runs.

For all the experiments we chose a single node as our client. The number of servers (no of nodes that have the video file of interest) were varied from 1 to 12. The client retrieves a 240s clip at a specified rate of 2Mbps. The clip is transferred through the network in sequential packets of size 512 bytes (CBR traffic).

Our simulator employs 802.11b as its MAC layer. The effective radio range is fixed at 100m. The bandwidth per link in the simulations was 2Mbps. Finally, the experiments were studied for different block sizes, namely, 1KB, 10KB, 100KB, 1MB.

The simulations show the results of performing block-level server-switching using dynamic source routing (DSR [7]) as an underlying protocol<sup>1</sup>. The primary source of disruption in data delivery are path-breaks due to mobility. We assume in-order sequenced delivery, so the path-repair latencies cumulatively result in increased startup latency for clip playback.

<sup>1</sup>based on experimental observations the non-local caching and snooping options in the ns-2 implementation of DSR were disabled

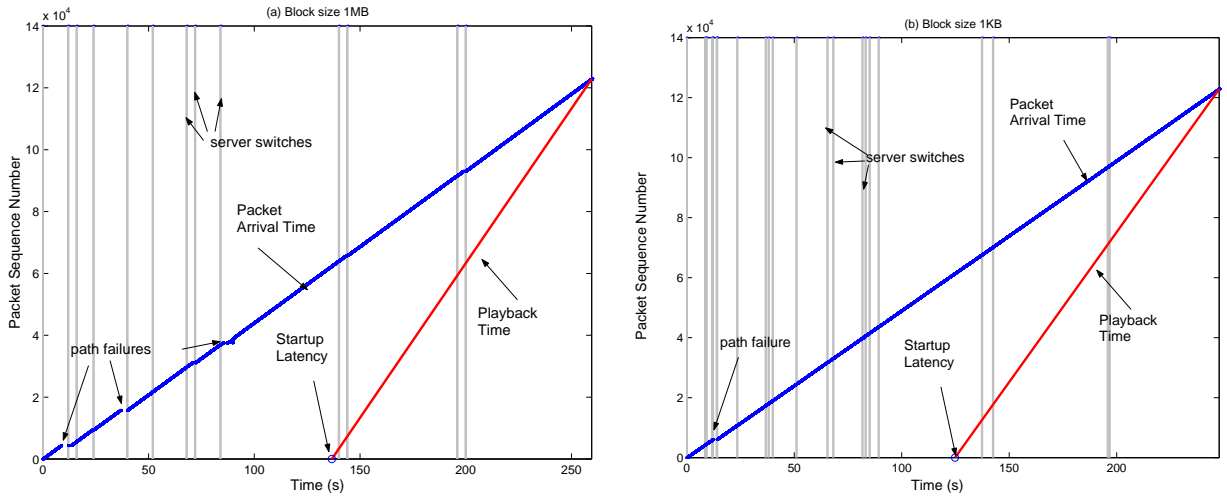


Figure 2: Simulation results of packet arrival times for a sample run involving block-based staggered non-redundant retrieval from 8 servers for (a) block size = 1MB and (b) block size = 1KB; maximum node velocity 30m/s; delivery rate 2Mbps and playback rate 4Mbps.

### 3.2 Simulation Results

Figure 2 shows sample runs of data delivery using the staggered non-redundant retrieval. The vertical lines in figure 2 (a) represent times when there is a server switch. There are two sloping curves. The one on top shows the packet arrival times, with some small gaps which correspond to path breaks due to mobility. The second sloping curve is a line which represents the playback time. For illustration we have chosen a playback rate of 4Mbps in this figure (all other simulation results are based on 2Mbps playback rate). The initial point of this playback time represents the startup latency. When there are greater numbers of path failures, the startup latency deteriorates. As shown in figure 2 (b), when a smaller block size (1KB) is used, there are more server switches. Since the client refreshes its server connection more frequently, there are fewer intra-block path failures in this scenario and as a result the corresponding startup latency is lower when the block size is low<sup>2</sup>.

Figure 3 and figure 4 show how the startup latency varies as a function of the number of servers for different mobility scenarios and block sizes. The first observation is that the startup latency is reduced significantly as the number of servers (i.e. the degree of replication) is increased. We can see reductions as high as 75% in some cases (particularly in the high mobility scenario), by increasing the number of servers from 1 to 12. The curve falls off quite steeply initially, suggesting that even replication with 3-4 servers can be very useful. This demonstrates the main lesson of our work - replication strategies can significantly improve the quality of service for continuous media delivery in mobile ad hoc networks.

Another observation is that, particularly for the high-mobility scenario ( figure 3 (b)), the startup latency can be improved by decreasing the block size. Figure 4 shows that when the block-size is large, mobility has a significant impact on the startup latency (higher the maximum node velocity, higher the latency).

In our simulations we impose a final deadline of 260s for the delivery of all packets from the clip. Any deliveries that are not completed by this time are considered incomplete. Figure 5 shows how the percentage of completed deliveries varies with the number of servers for different block sizes. Again, we see that using replication improves the QoS for the clip delivery giving higher fraction of complete deliveries. We observed that the relatively smaller block sizes of 10KB and 100KB provide better performance than the large block size of 1MB.

Finally, another advantage to be gained from clip-level replication is that the total network bandwidth consumption can be reduced when more servers are employed. Intuitively this is because the average number of hops between the client and the nearest server decreases as the number of servers is increased. Hence the total number of packets sent in the network also decreases, since each packet must travel fewer hops. This is shown clearly in figure 6, which suggests

<sup>2</sup>Naturally, choosing a smaller block-size also increases the number of route setup messages required by the underlying DSR protocol, but this additional overhead is relatively insignificant compared to the large bandwidth application data

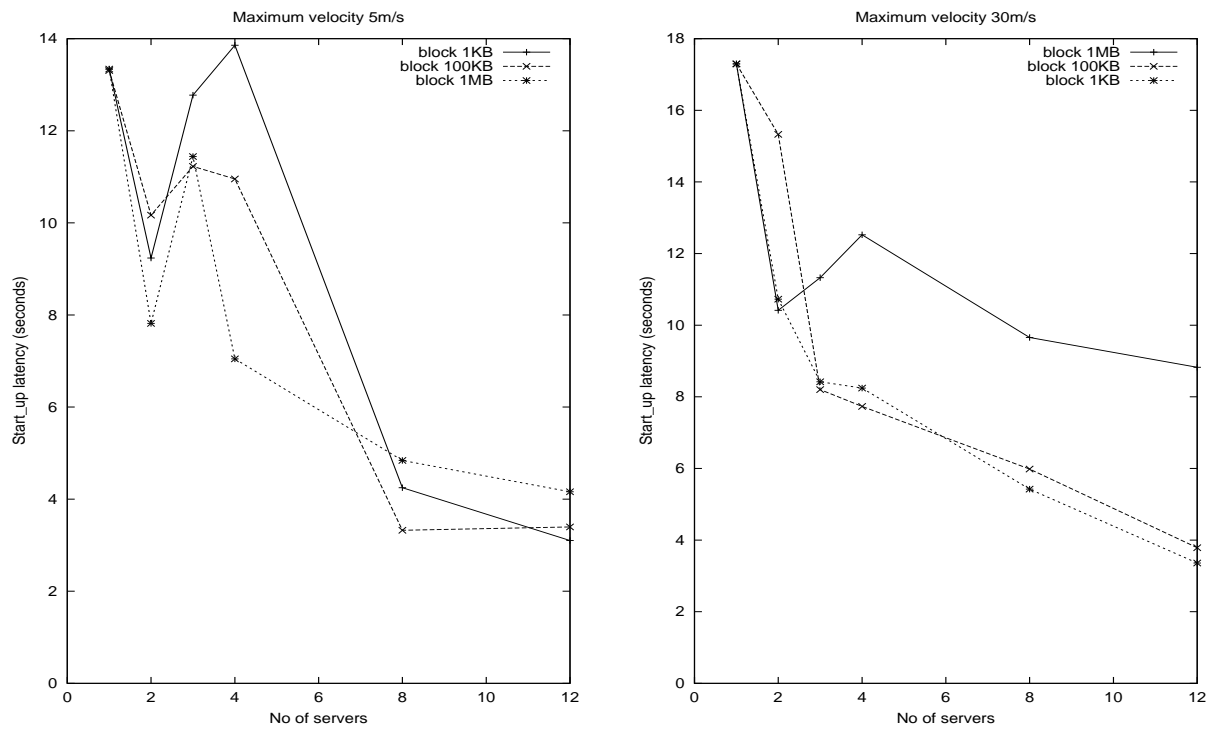


Figure 3: Startup latency for jitter-free delivery as a function of the number of servers with different block sizes for a) low-mobility scenario with maximum velocity of 5 m/s and b) high-mobility scenario with maximum velocity of 30 m/s.

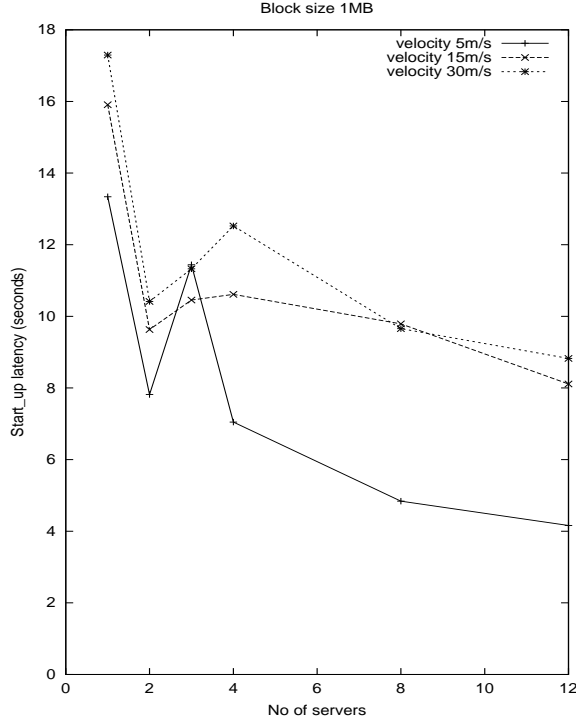


Figure 4: Startup latency for jitter-free delivery as a function of the number of servers with different mobility scenarios for a large block size of 1MB.

that nearly 25% savings in bandwidth consumption can be obtained with multiple servers. Thus our results suggest that replication significantly improves both quality of service and bandwidth consumption.

### 3.3 Analytical results

We now present a simple first order analytical model to show how the bandwidth usage decreases with multiple servers, as shown by our simulations. For the analysis, we consider  $n$  nodes placed in an operational area of 1 square units. Assume that the radio range of each node is  $R$  ( $R \geq 1$ ). We ignore edge effects, and assume that there is sufficient radio coverage and density so that all nodes within a distance  $kR$  are within  $k$  hops within the wireless network. Then  $n_k$ , the average number of nodes within  $k$  hops of an arbitrary client node is given approximately as  $n_k = \pi(kR)^2 n$ . Assume that  $S$  of the nodes are chosen randomly and independently to be servers.

$$P[\text{nearest server is } k \text{ hops away}] = P[\exists \text{ a server } k \text{ hops away and no server within } k-1 \text{ hops}] \quad (1)$$

$$= P[\text{exists a server } k \text{ hops away} | \text{no server within } k-1 \text{ hops}] \quad (2)$$

$$\times P[\text{no server within } k-1 \text{ hops}] \quad (3)$$

$$= 1 - P[\text{there is no server within } k \text{ hops} | \text{there is no server within } k \text{ hops}] \quad (4)$$

$$\times (1 - 2\pi((k-1)R)^2)^S \quad (5)$$

$$= \left(1 - \left(\frac{1 - 2\pi(kR)^2}{1 - 2\pi((k-1)R)^2}\right)^S\right) (1 - 2\pi((k-1)R)^2)^S \quad (6)$$

Hence the average number of hops between the nearest server and the client,  $H$  is a function of the number of servers and is given by the expression:

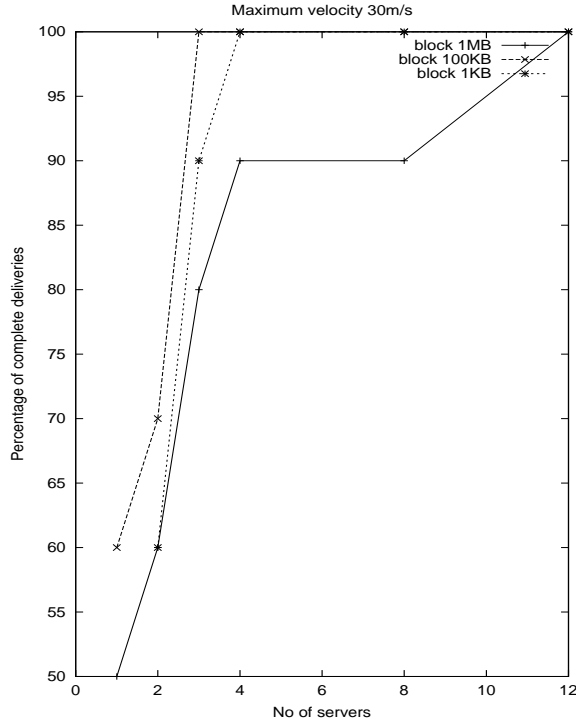


Figure 5: Percentage of completed deliveries as a function of the number of servers with different block sizes for a high-mobility scenario with maximum velocity of 30 m/s.

$$H(S) = \sum_{k=1}^{\lceil \sqrt{\frac{1}{2\pi}} \frac{1}{R} \rceil} k \left( 1 - \left( \frac{1 - 2\pi(kR)^2}{1 - 2\pi((k-1)R)^2} \right)^S \right) (1 - 2\pi((k-1)R)^2)^S \quad (7)$$

For a per-link delivery rate of  $W$  packets per second, and delivery duration  $\tau$  the total number of packets transmitted within the network for the delivery of a single clip is  $L(S) = W \times H(S) \times \tau$ . This is plotted in figure 7. As we can see, the analytical curve agrees very well with the simulation results presented in 3.2 – the bandwidth consumption for data delivery improves (decreases) as the number of servers is increased. It can also be observed that the radio range has a significant impact on the bandwidth consumption.

## 4 Conclusions and Future Research Directions

To our knowledge, this is one of the first papers to substantially address the challenges involved in continuous media delivery over a mobile ad hoc network. We analyzed data delivery techniques that enable a collection of mobile C2P2 devices to collaborate in support of a jitter-free display of continuous media clips. Obtained results demonstrate that data replication and block-based server-switched retrieval significantly enhances startup latency and prevents the likelihood of a displaying C2P2 device starving for data.

There are many obvious future extensions to this study. First, the presented results are based on a non-redundant staggered delivery of data, see Figure 1. We intend to explore the other branches of our taxonomy. Preliminary results indicate redundant data delivery is not always advantageous. In particular, multiple devices might attempt to simultaneously deliver data to a displaying C2P2 device, resulting in congestion and reducing the available bandwidth to the displaying device.

Second, while our obtained results quantify the total amount of bandwidth consumed by one display, we intend to extend this study to analyze an environment where multiple devices display clips simultaneously. We intend to design

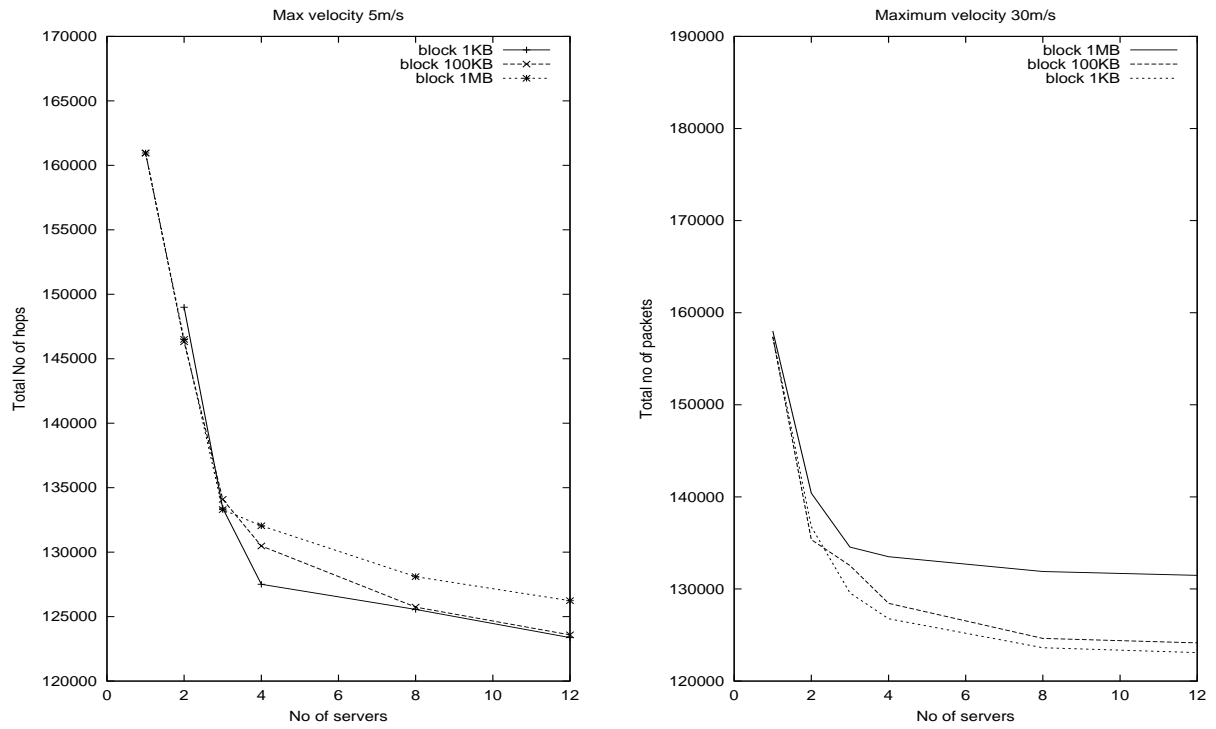


Figure 6: Total number of packets sent within the network for clip delivery as a function of the number of servers with different block sizes for a) low-mobility scenario with maximum velocity of 5 m/s and b) high-mobility scenario with maximum velocity of 30 m/s.

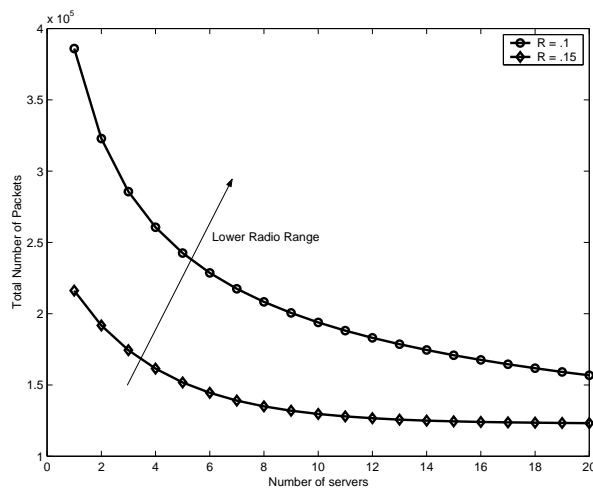


Figure 7: Total number of packets transmitted with respect to the number of servers for different normalized radio range settings

and evaluate a distributed admission control protocol which might reject a new request for a clip because its admission would cause currently active display(s) to suffer from jitters.

Third, the placement of data and the granularity of replication requires further investigation. The C2P2 devices may analyze the number of requests to a clip in order to replicate it proportional to its frequency of access [3]. Similarly, replication might be performed at the granularity of blocks. Moreover, the system may control the number of replicas for a block based on how urgently it is needed [6]. In [6], the authors replicate the first few blocks of a clip more frequently than its last few blocks because they are needed more urgently. (Mobility is not considered in [3, 6].) One may envision an environment where devices adjust the number of replicas dynamically based on future expected connectivity using current velocity and trajectory of movement.

## References

- [1] S. Berson, S. Ghandeharizadeh, R. Muntz, and X. Ju. Straggled striping in multimedia information systems. In *Proceedings of the ACM SIGMOD International Conference on Database Management*, 1994.
- [2] S. Ghandeharizadeh and R. Muntz. Design and Implementation of Scalable Continuous Media Servers. *Parallel Computing*, 24(1):91–122, May 1998.
- [3] E. Cohen and S. Shenker. Replication Strategies in Unstructured Peer-to-Peer Networks. In *Proceedings of the ACM SIGCOMM*, August 2002.
- [4] J. Gemmell, H. M. Vin, D. D. Kandlur, P. V. Rangan, and L. A. Rowe. Multimedia Storage Servers: A Tutorial. *IEEE Computer*, 28(5):40–49, 1995.
- [5] S. Ghandeharizadeh, S. Kim, and C. Shahabi. On Configuring a Single Disk Continuous Media Server. In *Proceedings of the ACM SIGMETRICS/Performance*, May 1995.
- [6] S. Ghandeharizadeh, B. Krishnamachari, and S. Song. Placement of Continuous Media in Wireless Peer-to-Peer Networks. Submitted for publication.
- [7] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.
- [8] R. Rejaie, M. Handley, H. Yu, and D. Estrin. Proxy Caching Mechanism for Multimedia Playback Streams in the Internet. In *The Fourth International Web Caching Workshop*, March 1999.
- [9] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. PATHS: analysis of PATH duration Statistics and their impact on reactive MANET routing protocols. In *Proceedings of Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2003.
- [10] C. E. Perkins. Ad Hoc Networking. isbn=0-201-30976-9. Publisher: Addison-Wesley Professional. 2001.
- [11] J. D. Salehi, Z. Zhang, J. F. Kurose, and D. Towsley. Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing. In *Proceedings of the 1996 ACM Sigmetrics Conference*, May 1996.
- [12] S. Sen, J. Rexford, and D. Towsley. Proxy Prefix Caching for Multimedia Streams. In *The Proceedings of the IEEE Infocom*, 1999.
- [13] T. Chiang and D. Anastassiou. Hierarchical Coding of Digital Television. 32:38–45, May 1994.
- [14] P.S. Yu, M-S. Chen, and D.D. Kandlur. Grouped sweeping scheduling for DASD-based multimedia storage management. *Multimedia Systems*, 1(1):99–109, January 1993.