

Group-Server Scheduling for Continuous Multimedia Streaming in MANETs

Kazuya Sakai
Department of CSSE
Auburn University
Auburn, AL 36849, USA
sakaika@auburn.edu

Wei-Shinn Ku
Department of CSSE
Auburn University
Auburn, AL 36849, USA
weishinn@auburn.edu

Roger Zimmermann
Dept. of Computer Science
National University of Singapore
Singapore 117417
rogerz@comp.nus.edu.sg

Abstract—At present mobile devices are prevalent with end users and continuous media streaming services in mobile ad-hoc networks (MANETs) support popular applications. One of the requirements for applications that stream isochronous media is that the network link has to be continuously available. In this study, we introduce two group-server scheduling schemes to improve link continuity: *static group-server scheduling* and *dynamic group-server scheduling*. With our solution, if one of the current links between a client and a server instance breaks, the client can still download the multimedia content from another scheduled server peer. The simulation results show that the proposed schemes significantly improve the effective link duration in MANETs.

I. INTRODUCTION

Multimedia streaming among mobile users is becoming a popular application with the widespread availability of PDAs, cell phones, and other handheld devices. Continuous media streaming applications have been implemented in fashionable consumer electronics such as Microsoft's Zune and Apple's iPhone. Most of the recent mobile devices employ IEEE 802.11x as one of their wireless communication protocols, as it provides high bandwidth and a transmission range of up to two hundred meters. With this technology, a mobile user is able to download multimedia content from other mobile users even if they are moving during the transmission. One of the major challenges in multimedia streaming is to continuously stream a media file without any interruptions. In other words, the link between two peers, a client and a server, has to be continuously available during the streaming process. Thus, continuous link availability is crucial to the success of multimedia streaming. There exist a number of studies [5]–[7], [9], [10] that have analyzed continuous link availability based on the current user location and mobility. However, all previous research assumes that a peer streams multimedia files from a single server instance. In MANETs, it is not straightforward to find a server instance which supports a long link duration. For example, the typical size of a MP3 file is 3 MB (or about 3 minutes of music). If the transmission rate of a mobile device is 512 kb/s, a link between two peers has to be continuously available for 48 seconds to transmit the complete file. This is not always achievable if most users have high mobility.

To address this issue, we introduce a group-server scheduling technique and a multi-hop peer search that allow a client

to download multimedia files from scheduled group-server instances and to calculate the future link availability over multi-hop peers. Even if the link with the current server instance breaks, a client can still continuously download from another scheduled server instance without any interruption and hence achieve smooth media streaming. The contributions of this research are as follows:

- We introduce the Static Group-Server Scheduling (SGSS) technique to increase link continuity. In the proposed scheme, even if the link with the current server instance breaks, a client can still stream from another scheduled server instance.
- We extend SGSS to Dynamic Group-Server Scheduling (DGSS), in which the scheduling of group-server instances is dynamically configured during the streaming process.
- We formalize the expected link duration over multi-hop peers in contrast to existing models, which merely estimate the probability that a link will be available for a given period of time.
- We conduct extensive simulations to evaluate our two group-server scheduling techniques. The experimental results show that our proposed schemes significantly improve the link duration as well as the media streaming success rate between a client and server instances.

II. RELATED WORK

A. Link and Path Availability Models

A number of link and path availability models have been proposed in the past. In [10], a numerical method on link and path durations was investigated for different mobility models. Han *et al.* [4] validated the results presented in [10] by using Palm calculus [1]. In these studies, the network behavior was analyzed when it reached the stationary regime. In other words, the authors analyzed the average link or path durations. However, they did not predict the future link availability based on the current user location and velocity. As a result, such models are not sufficient for advanced multimedia streaming applications.

Several continuous link availability models are given in [5], [6], [8], [9]. In [6], the authors estimated a time interval T

during which the current link will be available if two peers do not change their current velocities and directions. Then, a statistical method was applied to calculate the probability that the link is continuously available for the period of T , denoted as $L(T)$. The research in [5] improved the link availability model proposed in [6]. To evaluate the reliability of a link, the author used $T \times L(T)$ as a metric when the value of T is very small. Qin *et al.* [9] introduced an iterative algorithm to predict the probability that a link is continuously available given an initial distance of d between two peers and a duration t required by a media streaming process denoted as $L(d, t)$. Furthermore, they analyzed the future link availability in the case that the current velocities and directions of two peers are unknown. In [8], the authors provided the path availability model based on [9]. The path availability is obtained as the product of each link probability in a path:

$$\prod_{\forall k \in path} L_k(d, t) \quad (1)$$

However, none of these aforementioned solutions considered group-server scheduling.

B. Nodal Mobility Models

A number of nodal mobility models for MANETs were broadly investigated in [2], [3]. Among them, the mobility models which are most commonly used include the random waypoint model and the random walk model [3]. In the random waypoint model, a node repeatedly generates random destinations and pause periods. When the node arrives at a destination, it waits for a random period of time. After the pause time expires, the node again produces a new destination and moves towards it.

In the random walk model, a node moves from its current location to a new location by randomly selecting a direction and speed. When a node arrives in its destination, a new speed and direction are again randomly chosen. A nodal movement occurs either during a time interval called epoch time or across a distance to travel.

III. SYSTEM DESIGN

A. Motivation

Based on the link availability models discussed in Section II, a mobile peer can determine the best neighbor from which to continuously download multimedia content. However, previous link availability models focus on streaming from a single peer-server instance. If the link between two mobile peers or one of the links among a path is disrupted, the media streaming process fails. In MANETs, it is difficult to find a server peer which supports a long duration for streaming large multimedia content, due to the frequent changes in the topology. Hence, we are interested in solutions that prolong the continuous link duration.

B. Assumptions and Definitions

We denote S to be the size of a multimedia file and B to be the transmission rate. In this paper, we consider the streaming of continuous media where the transmission must not be interrupted during the whole streaming process. To complete a multimedia streaming task, the duration of the link between two mobile peers has to be longer than S/B seconds and the link should not experience any disconnections. When the link between a client and a server instance breaks up, media streaming fails unless another scheduled server instance can be found within the transmission range (the system architecture is described in the following sections). If the complete media content is transmitted, the multimedia streaming task is successfully completed.

C. Design Goals

Figures 1 and 2 depict two snapshots of a MANET. In these figures, a black circle represents a mobile peer, a solid line denotes a trajectory, a dashed line represents a link between two peers, and a dashed circle indicates a transmission range. Before describing the proposed schemes, we introduce two concepts: single/multi-hop peer search and single/group-server streaming.

In Figure 1, client i first finds the best server instance j_1 among its single-hop peers since links are available with only single-hop neighbors at time t_0 . Here, the best server instance is defined as the server instance which has the longest expected connection duration with the client (the details are described in Section III-D). If the current server instance j_1 moves out of i 's transmission range (Figure 2), the client is no longer able to download the multimedia content from j_1 . Thus, i needs to stream from other server instances and we name this method *group-server streaming*. Since j_1 is supposed to provide the longest connection duration, it is highly possible that no original single-hop peer will be in the communication range when the link between i and j_1 breaks. To set up group-server streaming, i has to search its neighbors over multiple hops, hence perform a *multi-hop peer search*. Based on the current velocities and moving directions, the link between i and j_1 is expected to break at time t_1 and the corresponding topology at t_1 is shown in Figure 2. Server instance j_2 is expected to have a link with client i at time t_1 . Therefore, i can schedule the media streaming from group-server instances $\{j_1, j_2\}$ as illustrated in Figure 3. Let us assume that it takes 60 seconds to finish the streaming task, and i will have a link with j_1 from 0 to 37 seconds and with j_2 from 32 to 63 seconds. Consequently, client i sets up a schedule such that it streams the media file from j_1 for the first 35 seconds and from j_2 for the following 25 seconds.

Note that while the existing solution by Qin *et al.* [9] focuses on single-hop peer search and single-server streaming, our proposed scheme is based on multi-hop peer search and group-server streaming. We assume that prevalent content is available among nearby mobile peers and we do not consider streaming across multiple hops in this research. Only the search process is expanded beyond a single hop with the

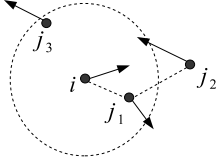


Fig. 1. Topology at time t_0 .

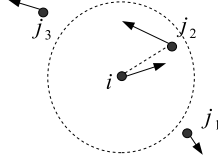


Fig. 2. Excepted topology at time t_1 .

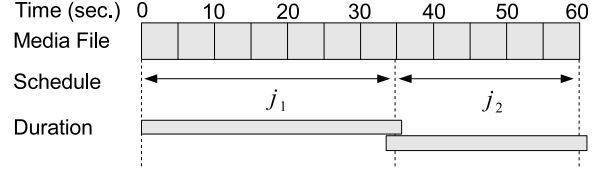


Fig. 3. The group-server scheduling.

goal of identifying servers that will move within transmission range by the time they are needed. Overall, the main goal of this paper is to improve the effective link duration by using group-server scheduling. Depending on the time at which a client schedules group-server streaming, two novel scheduling schemes, Static Group-Server Scheduling (SGSS) and Dynamic Group-Server Scheduling (DGSS), are described in the following sections.

D. The Expected Link Duration

The link availability models proposed in earlier work [5], [6], [9] provide the probability that a link is continuously available for a period of time as required for streaming a media file. However, in our schemes, we use multiple server instances to stream media content. Therefore, our foci are when a link between a client and a group of server instances will be available and the expected duration of the link. In this section, we formulate the expected link duration for both the random waypoint model and random walk model.

1) Link Duration with the Random Waypoint Model:

Assume the initial distance between two mobile peers, i and j , is d_0 . Each peer is moving in an arbitrary direction with the velocities v_i and v_j , respectively. The velocity and the direction of a peer is uniformly distributed between $[0, v_{max}]$ and $[0, 2\pi]$, respectively. The relative velocity v can be obtained by Equation 2.

$$v = \sqrt{v_i^2 + v_j^2 - 2v_i v_j \cos \theta} \quad (2)$$

Here $\theta \in [0, 2\pi]$ is defined as the angle between the two peers i and j , as shown in Figure 4. In the case that the space is large, the distance between consecutive points of a peer, i.e., the starting and ending locations of a movement segment, is usually very long. Thus, as an approximation we can assume two peers do not change their current velocities and directions during the link duration. For a given distance d , the expected link duration between peers i and j can be obtained by:

$$T_{ij}(d) = \begin{cases} \frac{\sqrt{R^2 - d^2 \sin^2 \phi} - d \cos \phi}{v} & \text{if } d \leq R \\ 0 & \text{if } d > R \end{cases} \quad (3)$$

Here, ϕ is defined as the angle of the relative velocity vector \vec{V} as shown in Figure 4. Note that the expected link duration E_{ij} is the same as T_{ij} since we assume peers do not change their velocities and directions during the link duration. Therefore, the link from i to single-hop peer j is available between:

$$[t_0, t_0 + E_{ij}] \quad (4)$$

Next, we discuss the future link availability between two peers i and j whose initial distance is $d_0 > R$, i.e., the future link availability with multi-hop peers. To calculate the future distance at time t , denoted by d_t , we introduce Equation 5.

$$d_t = \sqrt{d_0^2 + v^2 t^2 + 2d_0 v t \cos \phi} \quad (5)$$

Clearly, peers i and j are single-hop peers at time t , if $d_t \leq R$. Based on these equations, the link between them is available for duration $T_{ij}(d_t)$ starting from t , if $d_t \leq R$. Note that two peers will become single-hop neighbors only when $\cos \phi \leq -\sqrt{1 - R^2/d_0^2}$. From Figure 5, we observe that the link is available between:

$$\left[t - \frac{2\sqrt{R^2 - d_0^2 \sin^2 \phi} - \sqrt{R^2 - d_t^2 \sin^2 \phi} + d_t \cos \phi}{v}, t + T_{ij}(d_t) \right],$$

where $d_t \leq R \wedge \cos \phi \leq \sqrt{1 - R^2/d_0^2}$ (6)

2) Link Duration with the Random Walk Model:

To compute the link duration with the random walk model, we assume that peers do not change their current velocities and directions. In the random walk model, nodal movements occur during an epoch time (or a certain distance). The epoch time for each peer varies according to a normal distribution centered around the mean epoch time. Thus, peers change their current velocities and directions more often compared with the random waypoint model. In order to improve the accuracy of the expected time duration, we consider the case in which peers change both their current velocity and direction.

We employ λ_i^{-1} and λ_j^{-1} to denote the mean epoch times of peers i and j , which are independent from each other. According to [6], the process during which the link between i and j experiences velocity changes can be considered a Poisson process and the probability that there are k velocity changes within t seconds is given by:

$$C(k, t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!} \quad (7)$$

Here, $\lambda = \lambda_i + \lambda_j$. In other words, the probability that two peers do not change their velocities is $e^{-(\lambda_i + \lambda_j)t}$. After a peer changes both its velocity and direction, its movement is no longer predictable. Thus, we need to consider either a statistical approach or the worst case link duration. The statistical approach proposed in [9] employs very complex calculations and such computational overhead becomes a burden for resource-limited mobile devices. Therefore, in this research we consider the worst case link duration.

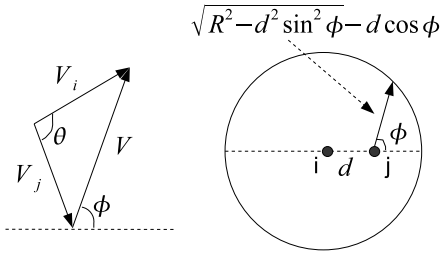


Fig. 4. Expected link duration of two peers.

We assume that a peer moves with the current velocity and direction for t_1 seconds and then changes its speed and orientation. Based on Equation 7, the expected duration until one of the peers changes its velocity and direction can be calculated as:

$$t_1 = \int_0^{T_{ij}(d_0)} t \cdot C(0, t) dt \quad (8)$$

After t_1 , the peer's movement is not predictable. The worst case occurs when two peers move from each other with the maximal speed v_{max} . Thus, the link duration in the random walk model is obtained by Equation 9:

$$E_{ij} = t_1 + \frac{R - d_{t_1}}{2v_{max}} \quad (9)$$

Equation 9 calculates the period of the link between nodes i and j starting from t_0 when they are single-hop peers. If the two nodes are not peers at t_0 , i.e., $d_0 > R$, the future link availability can be obtained from Equation 6 by replacing t with E_{ij} .

E. Static Group-Server Scheduling

With the static scheduling scheme SGSS, a client schedules group-server instances at the time when it requests media content based on peer information such as the current location, velocity, and direction. Consider a client i which requests a specific media content. The first server needs to be a single-hop peer of i , as the first server starts streaming at time t_0 . Thus, i selects the best server instance, i.e., the peer with the largest value of $E_{ij} \forall j \in N_{t_0}(i)$. Let us say this server is j_1 . Then, i schedules j_1 at time t_0 . If the expected time duration is less than the required media streaming period, i.e., $E_{ij_1} < S/B$, the client needs to find more server instances via multi-hop neighbors.

We define t_x as the time when the link with j_x will break. Since the link with j_1 is expected to break at t_1 , the client needs to find a server instance which will provide a link at t_1 . To achieve this, the client broadcasts queries within H hops. From information obtained by the queries, the client can know the set of available server instances at time $t_1 - \delta t$ (δt is the required switching time), denoted as $N_{t_1 - \delta t}(i)$. By calculating the link duration, the client finds the best server, say j_2 , which has the longest link duration starting from $t_1 - \delta t$. Therefore, the client can switch the server instance from j_1 to j_2 at $t_1 - \delta t$. If $t_2 \geq S/B$, it can download the whole media content from j_1

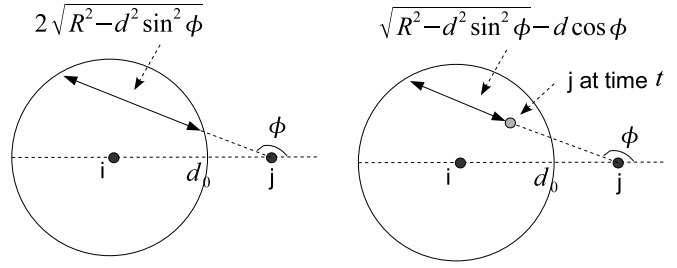


Fig. 5. Future link availability of two peers.

Algorithm 1 Static Group-Server Scheduling (H, S, B)

```

1: Scheduling:
2: Collect the information of server instances,  $N_{t_0}(i)$ 
3: Obtain  $j_1$  with  $\max_{j \in N_{t_0}(i)} E_{ij}$ 
4:  $t_1 \leftarrow$  the time that link with  $j_1$  will break
5: Schedule  $j_1$  at  $t_0$ 
6: if ( $t_1 \geq \frac{S}{B}$ ) then
7:    $i$  Starts the media streaming
8: end if
9:  $k \leftarrow 2$ 
10: while ( $t_k < \frac{S}{B}$ ) do
11:   Collect the information of server instances  $N_{t_k}(i)$  over  $H$ -hop
12:   if ( $N_{t_k}(i)$  is not empty) then
13:     Obtain  $j_{k+1}$  with  $\max_{j \in N_{t_k}(i)} E_{ij}$ 
14:      $t_{k+1} \leftarrow$  the time that link with  $j_{k+1}$  will break
15:     Schedules  $j_{k+1}$  at  $t_k - \delta t$ 
16:      $k++$ 
17:   else
18:      $i$  cannot schedule group-server instances
19:   end if
20: end while
21:  $i$  starts the media streaming
22: Media Streaming:
23:  $t_c \leftarrow t_0$  /* the current time */
24:  $k \leftarrow 1$  /* the index of the current server instance */
25:  $i$  starts the media streaming from  $j_k$ 
26: while ( $i$  is streaming) do
27:   if (the link with  $j_k$  is available) then
28:      $i$  is downloading the media content from  $j_k$ 
29:     if ( $t_c = t_k - \delta t$ ) then
30:       /*  $i$  switches the current server */
31:        $k++$ 
32:     end if
33:   else
34:     The media streaming fails
35:   end if
36: end while

```

and then j_2 . Otherwise, it again finds a server instance until $t_k \geq S/B$. If the client cannot find group-server instances $\{j_1, j_2, \dots, j_k\}$ to satisfy $t_k \geq S/B$, the media streaming fails. The complete algorithm of SGSS is formalized in Algorithm 1.

F. Dynamic Group-Server Scheduling

In the dynamic scheduling scheme DGSS, the group-server scheduling is dynamically configured during the media streaming process. The scheme improves the performance of the static case, as it is relatively easier to predict near future link availability compared with the availability into the more distant future. In addition, a peer may move in a totally different direction from what is expected. Therefore, the static

Algorithm 2 Dynamic Group-Server Scheduling (H, S, B)

```
1: Scheduling:
2: Collect the information of server instances  $N_{t_0}(i)$ 
3: Obtain  $j_1$  with  $\max_{j \in N_{t_0}(i)} E_{ij}$ 
4: Schedule  $j_1$  at  $t_0$ 
5:  $i$  starts the media streaming
6: Media Streaming and Scheduling:
7:  $t_c \leftarrow t_0$  /* the current time */
8:  $k \leftarrow 1$  /* the index of the current server instance*/
9:  $i$  starts the media streaming from  $j_k$ 
10: while ( $i$  is still streaming) do
11:   if (it is the query interval) then
12:      $i$  sends a query to  $j_k$ 
13:      $t_k \leftarrow$  the time that link with  $j_k$  will break
14:     if ( $t_k - t_c \leq$  threshold) then
15:       Collect information of server instances  $N_{t_k}(i)$  over  $H$ -hop
16:       if ( $N_{t_k}(i)$  is not empty) then
17:         Obtain  $j_{k+1}$  with  $\max_{j \in N_{t_k}(i)} E_{ij}$ 
18:         Schedule  $j_{k+1}$  at  $t_k - \delta t$ 
19:       end if
20:     end if
21:   end if
22:   /* downloading the media content */
23:   if (The link with  $j_k$  is available) then
24:      $i$  is downloading the media content from  $j_k$ 
25:     if ( $t_c = t_k - \delta t$ ) then
26:       /*  $i$  switches the current server */
27:        $k++$ 
28:     end if
29:   else
30:     The media streaming fails
31:   end if
32: end while
```

scheduling scheme may not work well in highly mobile environments.

With DGSS, a client i selects the first server j_1 among $N_{t_0}(i)$ to download a media content starting from t_0 . Then, i periodically sends queries to the current server instance to obtain its velocity and direction and updates the expected link duration. The query interval is determined by a threshold. If the expected link duration of the current server instance j_1 is less than the threshold, the client tries to find a new server instance by broadcasting queries within H hops. By querying the current server every interval, a client can switch server instances before the current link breaks. If the link with j_1 will break at t_1 , the client obtains the best server instance, say j_2 , among $N_{t_1 - \delta t}(i)$ peers. The client switches the server instance to download the media content at $t_1 - \delta t$ to keep the media streaming process continuous. The client repeats this process until the whole file is downloaded. In case no server instance can be found when the current link disconnects, the media streaming fails. The complete algorithm of DGSS is provided in Algorithm 2.

IV. EXPERIMENTAL VALIDATION

To evaluate the performance of the proposed schemes, we conducted the following simulations along with the single-server streaming scheme: SGSS with 2-hop peer search (SGSS w/ 2-hop), DGSS with 1-hop peer search (DGSS w/ 1-hop), and DGSS with 2-hop peer search (DGSS w/ 2-hop).

A. Simulation Configuration

In our simulations, the network topology is randomly generated by placing mobile peers in a 400m by 400m square field. The total number of peers is set to be 500 with a transmission range of 100m and a transmission rate of 512 kb/s. Among the peers, 100 (corresponding to 20% of peers) are server instances, which have content. The threshold used in DGSS is set to be 3 seconds. For a given configuration, 1,000 realizations of the network topology are generated. To simulate nodal mobility, the random waypoint model and the random walk model are used. Because of the space limit, we only show the simulation results with the random waypoint model. The trend of the simulation results with the random walk model is quite similar. Unless specified, the maximum pause time (used in the random waypoint model) is set to be 30 seconds, the maximal speed of mobile peers is set to be 5 m/s, and the size of a media file is set to be 3 MB (a general 3 minute MP3 file).

B. Evaluation Metrics

To evaluate the performance of multimedia streaming for different schemes, we employ three metrics, the link duration, the streaming success rate, and the control overhead. The link duration is defined as the time during which a link between a client and a server or a client and the scheduled group-server instances is continuously available. The streaming success rate is the ratio of successful multimedia streaming instances to the number of streaming requests from clients. If the streaming of a continuous media object is completed without any interruption, this media streaming process is counted as successful. The control overhead is counted as the number of queries to find server instances. In SGSS, a client sends a query to its peers or multi-hop peers when it requests a multimedia file. On the other hand, in DGSS, a client periodically sends a query to the current server instance every 3 seconds.

C. Simulation Results with the Random Waypoint Model

Figure 6 shows the link duration with respect to the number of hops. When the number of hops is more than 2, the results are better than when the number of hops is 1. However, no significant difference can be found when the number of hops is increased from 2 to 5. This is because a peer that is 3-hops (or 4-, 5-hops) away from a client has a low possibility to be a single-hop neighbor of the client after a period of time. Therefore, no matter how many hops a client searches beyond two hops, the link duration time does not improve.

Figure 7 presents the link duration as a function of the number of server instances. It is intuitively understandable that with more server instances in a network, a client will have a better chance to find a server instance with a lengthy duration. Thus, the link duration increases in proportion to the number of server instances in the MANET. Compared with SGSS, DGSS w/ 1-hop and w/ 2-hop result in longer durations.

Figure 8 demonstrates the link duration as the maximal velocity of mobile peers increases. As shown, high mobility environments decrease the duration of links between peers, and

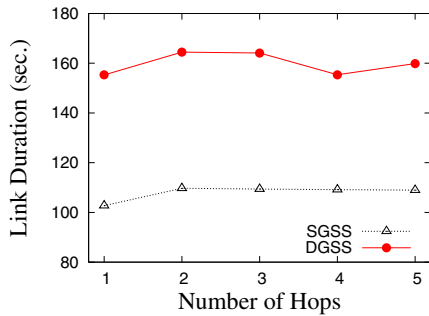


Fig. 6. Hop number.

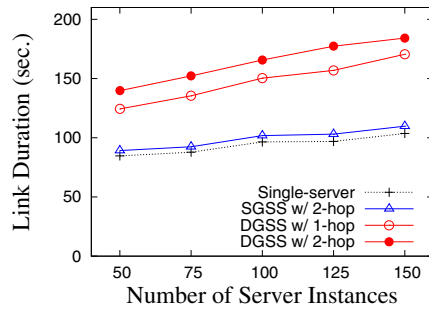


Fig. 7. Server instance.

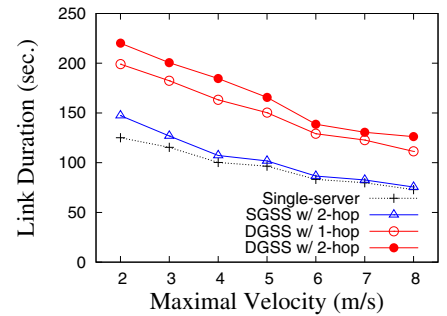


Fig. 8. Max. velocity.

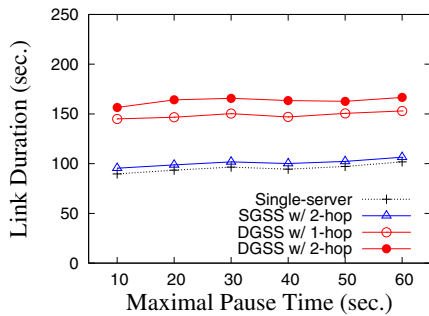


Fig. 9. Max. pause time.

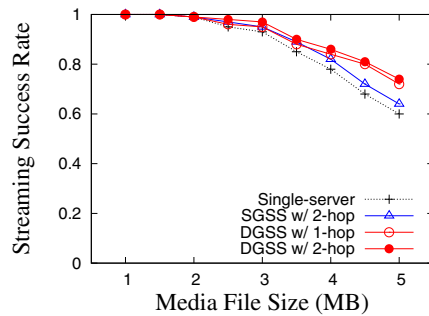


Fig. 10. Success rate.

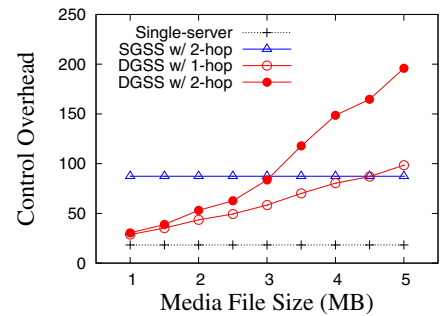


Fig. 11. Control overhead.

the link duration drops in proportion to the maximal velocity. DGSS w/ 1-hop and w/ 2-hop result in at least 40% longer duration than SGSS.

Figure 9 illustrates the link duration versus the maximal pause time. It can be seen that the link duration is stable even when the maximal pause time increases. Therefore, we can conclude that the link duration is independent of the maximal pause time.

Next, Figure 10 depicts the streaming success rate as a function of the media file size. A smaller-size media file requires a shorter link duration to complete the transmission, and thus the streaming success rate decreases inversely with the media file size. When the media file size is small there is no difference among all the schemes. However, when the media file size is more than 3.5 MB, DGSS w/ 1-hop and w/ 2-hop have a 10% higher success rate than SGSS w/ 2-hop and the single-server scheme. This is very important since the length of a MP3 file is generally 3 to 5 minutes, corresponding to 3 to 5 MB.

Figure 11 illustrates the control overhead as it relates to the media file size. In SGSS and the single-server scheme, a client launches queries only when it requests media content. Thus, the control overhead is stable against the media file size. Conversely, in DGSS, a client periodically sends out queries. Consequently, the control overhead increases in proportion to the time required to complete the streaming. Equivalently, the control overhead rises as the media size increases.

V. CONCLUSION

In this paper we propose two group-server scheduling schemes, Static Group-Server Scheduling (SGSS) and Dynamic Group-Server Scheduling (DGSS), for continuous media streaming among mobile ad-hoc peers. With our tech-

niques, a client downloads multimedia content from group-server instances, which allows the user to continuously stream media content even if the link with the current server instance breaks up. The simulation results show that our schemes, especially DGSS, significantly improve the link duration over the single-server scheme. In the future, we plan to investigate continuous media streaming in different types of networks, such as sensor and vehicle ad-hoc networks (VANETs).

ACKNOWLEDGMENTS

This research has been funded in part by the National Science Foundation grants CNS-0831502 (CT), CNS-0855251 (CRI).

REFERENCES

- [1] F. Baccelli and P. Bremaud. *Palm Probabilities and Stationary Queues*. Springer Verlag *Lecture Notes in Statistics*, 1987.
- [2] J.-Y. L. Boudec and M. Vojnovic. Perfect Simulation and Stationarity of A Class of Mobility Models. In *IEEE INFOCOM*, 2005.
- [3] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Comm. and Mobile Computing*, 2(5):483–502, 2002.
- [4] Y. Han, R. J. La, A. M. Makowski, and S. Lee. Distribution of Path Durations in Mobile Ad-Hoc Networks: Palm’s Theorem to The Rescue. *Computer Networks*, 50(12):1887–1900, 2006.
- [5] S. Jiang. An Enhanced Prediction-Based Link Availability Estimation for MANETs. *IEEE Trans. on Comm.*, 52(2):183–186, 2004.
- [6] S. Jiang, D. He, and J. Rao. A Prediction-based Link Availability Estimation for Mobile Ad Hoc Networks. In *IEEE INFOCOM*, pages 1745–1752, 2001.
- [7] A. McDonald and T. Znati. A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks. *IEEE Journal of Selected Area in Communications*, 17(8):1466–1487, 1999.
- [8] M. Qin and R. Zimmermann. Supporting Guaranteed Continuous Media Streaming in Mobile Ad-hoc Networks with Link Availability Prediction. In *ACM Multimedia*, pages 153–156, 2006.
- [9] M. Qin, R. Zimmermann, and L. S. Liu. Supporting Multimedia Streaming between Mobile Peers with Link Availability Prediction. In *ACM Multimedia*, pages 956–965, 2005.
- [10] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. PATHS: Analysis of PATH Duration Statistics and their Impact on Reactive MANET Routing Protocols. In *ACM MobiHoc*, pages 245–256, 2003.