# MobiVoD: A Video-on-Demand System Design for Mobile Ad hoc Networks*

Duc A. Tran
Department of Computer Science
University of Dayton
Dayton, OH 45469
duc.tran@notes.udayton.edu

Minh Le, Kien A. Hua
School of EECS
University of Central Florida
Orlando, FL 32816
{mle, kienhua}@cs.ucf.edu

## Abstract

*We present a design for a system that provides video-on-demand (VOD) services to mobile ad hoc clients. Such a system allows the clients to access video information anytime anywhere. MobiVoD, the proposed solution, overcomes many difficulties currently challenging video streaming in a mobile ad hoc network. The new environment includes a three-tier architecture, in which the mobile VOD system employs a periodic broadcast protocol to achieve maximum scalability; and the clients leverage an ad hoc network caching technique to minimize the service delay. This system can sustain client failure and mobility, and provide true VOD services to most clients.*

## 1. Introduction

As people tend to work beyond their office desk, we can expect the next generation of wireless communication networks to include rapid deployments of independent mobile users. With the emergence of wireless technologies such as IEEE 802.11 [12] and Bluetooth [4], mobile users are enabled to connect to each other directly without any networking infrastructure such as the Internet and infrastructure-based wireless LANs. In other words, the users form a *mobile ad hoc network* (MANET) [7]. A MANET is an autonomous network of mobile nodes that communicate with each other only via direct wireless links. No fixed topology exists in such a network and nodes can join and leave freely.

MANETs can find many applications due to its appealing working environment. Popular examples include establishing survivable, efficient, dynamic communication for emergency/rescue operations, disaster relief efforts, and military networks. In this paper, we focus on another kind of application that can run in a MANET: *we design a technique that provides* video-on-demand *(VOD) services to mobile users*. A VOD system is an interactive multimedia system working like cable television, the difference being that the client can select a movie from a large video database stored at a distant video server. Individual clients in an area are able to watch different programs when they wish to, making the system a realization of the video rental shop brought into the home.

VOD has been on a rise on the Internet in recent years. According to an In-stat/MDR[1] report in 2002, the market for VOD would reach 1.9 billion USD and the number of online VOD users would reach 17 million by 2006. As VOD becoming an integrated part of an increasing number of applications and wireless networks emerging to dominate the communication environment of the future, it is interesting and worthwhile to design a VOD system for MANETs. Such a system has many practical applications. For instances, airlines could provide VOD services in airport lounges to entertain passengers on their own PDA (personal digital assistant) while they are waiting for a flight; a museum could provide video information on the exhibits on demand over the wireless network; in education, a university could also install such a system on campus to allow students to watch video recorded earlier from lectures they were not able to attend.

Despite the promising future of a system for mobile VOD, implementing it, at least, needs to answer the following main questions:

1. What should be the architecture for a mobile VOD system?
    - Challenge: The coverage of wireless transmission is limited. An 802.11-enabled host can only reach other devices within 100m of its radius, while that radius is 10m for Bluetooth. These

* This research is partially funded by the National Science Foundation under grant No. ANI-0088026.

1 http://www.instat.com/

limits set a threshold on the communicability between two mobile hosts. If two mobile hosts are too far away from each other, they cannot communicate directly but must go through multiple intermediate hosts. In the case of video transmission, if a video goes through multiple hops from the source to the destination, significant amounts of bandwidth and energy of the intermediate mobile hosts are consumed. The architectural design should allow the system to cover clients within a distance long enough from the video server so the effectiveness of the system is increased.

2. What should be the communication protocol for a client to download a video from the video server?
   - Challenge: The wireless bandwidth is limited whereas a video is typically large. 802.11g, 802.11b, and Bluetooth provide maximum bandwidths of 54Mbps, 11Mbps and 1Mbps, respectively. A video server enabled with 802.11g, therefore, would not be able to deliver more than 36 1.5Mbps MPEG1 video streams simultaneously to its wireless clients. The situation is worse for 802.11b, which can only support at most seven concurrent such video streams. Consequently, how to transmit a video via wireless to a large number, say 100, of clients is challenging. We need a communication paradigm for video dissemination so the system can scale well with the size of client population.
3. How to ensure that a client can receive a video *truly* on demand?
   - Challenge: Under a significantly constrained network like a MANET, where many clients compete for limited wireless bandwidth, it is an impossible mission to satisfy all client requests instantly. However, keeping the service delay close to zero is always desired by any VOD system. A traditional way to shorten service delay in computer systems is use *cache*. The question then is what is cached, where to cache, and how to use cache, which we need to address.

It is also desirable to address other technical issues such as "is the designed system open to future wireless technologies?" and "is the system extendable to to support mobile clients with heterogeneous capabilities?" In this paper, we propose *MobiVoD*, a design for mobile VOD systems, which takes into account the aforementioned issues. Specifically, we propose to use: (1) a three-tier architecture for mobile VoD, which allows deployment of the system in a wide area, (2) a broadcast protocol based on Periodic Broadcast as the way to disseminate videos to clients, and (3) client selective-caching schemes, called *Dominating-Set Cache* and *Random-cache*, which allow clients to play a video on demand by exploiting nearby caches.

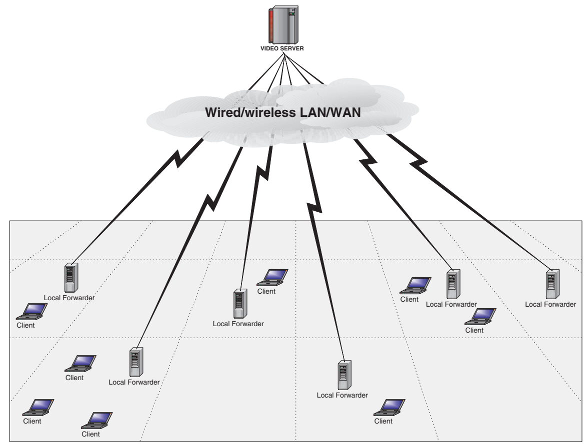


**Figure 1. MobiVoD System Architecture: Server, Local Forwarders, and Clients**

The remainder of this paper is organized as follows. The architectural and protocol details of MobiVoD are presented in Section II. Section III provides the results drawn from our simulation study on the performance of MobiVOD. Section IV proposes how this technique is extended to handle client heterogeneity and how future work is carried out. The paper is concluded in Section V.

## 2. MobiVoD: Proposed Solution

We propose in this section MobiVoD, a solution for providing VOD services to mobile users with ad hoc connections. Firstly, we describe the system architecture of MobiVoD. Next, we present how the server disseminates videos to the clients. We lastly investigate the advantage of client caching in guaranteeing every client a zero service delay. Hereafter, the term "bandwidth" alludes to "wireless bandwidth" without specified otherwise.

### 2.1. System Architecture

Illustrated in Figure 1, our mobile VOD system consists of three components: *video server*, *clients*, and *local forwarders*. The video server stores video files. Clients are the mobile users (devices or the people who use them), who subscribe for the VOD service provided by our system. Because the only way to communicate with the clients is via wireless transmissions, it is not possible for the video server to transmit a video to clients located in a wide geographic range. Therefore, we propose to install a scatter of local forwarders $\{LF_1, LF_2, .., LF_k\}$. A local forwarder $LF_j$ is a stationary and dedicated computer and used to relay the service to $LF_j$'s transmission coverage area. This area is called a local service area.

The local forwarders and clients are referred to as "nodes". Each of the nodes is equipped with a wireless network interface card (wNIC) so they are able to form a mobile ad hoc network. Since the current Bluetooth technology's bandwidth is less than 1Mbps, too little for video streaming, it is recommended that a node run an IEEE 802.11 protocol. In any setting, the bandwidth capacity at local forwarders should be no less than that at clients.

Every local forwarder receives the video packets from the server. This local forwarder then broadcasts the packets via its wNIC. If a client is within the service area of a local forwarder, the former can receive the video packets broadcast from the latter.

The server and set of local forwarders form a service backbone, interconnected either via a wired WAN/LAN or via an infrastructure-based wireless network. The topology for disseminating the video packets from the server to every local forwarder can be a star rooted at the server or be any overlay topology connecting the server and all the local forwarders. Depending on what working environment the system is running in be the locations of the local forwarders determined. For instance, on a campus or at an airport terminal, the local forwarders should be geographically uniformly distributed. In a big building with closed-door halls, however, we should install a local forwarder in each hall. We let the issue of determining the locations and the topology of the local forwarders beyond the scope of this paper.

Our architecture is not to be against the access-points based architecture. The rationale for our MANET-based design is to allow clients to have multiple direct wireless connections so they can directly share and exchange video information, which is part of our client caching schemes to be presented later. If the local forwarders were installed as access points, such communication exchanges among clients would go through the local forwarders, making the latter severe bottlenecks. Therefore, we propose to install the local forwarders as just MANET nodes.

### 2.2. Server Broadcasting

Although many existing Internet VOD techniques [11, 8, 24, 9] are based on the client/server approach, we argue that this approach does not fit well for MANETs simply because wireless bandwidth cannot support many clients using separate connection channels. The peer-to-peer approach for VOD [19, 20, 25, 16] is not suggested either since transmitting a long video from a wireless node to another via more than one wireless hop is inefficient in terms of bandwidth and energy used. Therefore, MobiVoD employs broadcasting at the video server to disseminate videos to wireless clients; specifically, we propose to use Periodic Broadcasting [1, 10, 17, 21, 18]. Using this approach, a video is divided into several segments, each repeatedly broadcast on a separate communication channel. A client receives a video by tuning to one or more channels at a time to download the data. The broadcast schedule at the server and playback synchronization protocol at the client ensure that the broadcast of the next video segment is available to the client before the playback of the current segment runs out.

| Solution | Caching space | Bandwidth |
|---|---|---|
| Staggered [15] | 0% of video | $1 \times$ r |
| Skyscraper [10] | 10% of video | $2 \times$ r |
| Pyramid [21] | 75% of video | $\geq 4 \times$ r |
| Permutation-based [1] | 20% of video | $\geq 2 \times$ r |
| Pagoda [17] | 45% of video | $\geq 5 \times$ r |
| Harmonic [13] | 40% of video | $\geq 5 \times$ r |
| Fast [14] | 50% of video | $\geq 6 \times$ r |

**Table 1. Client requirements in Periodic Broadcasting solutions (r is consumption rate)**

A clear advantage of Periodic Broadcasting is that the system can accommodate any number of clients. However, Periodic Broadcast poses some caching and bandwidth requirements on each client as illustrated in Table 1. Except for *Staggered Broadcasting* [15], which is a Periodic Broadcasting technique, all periodic broadcasting techniques require the client to download data simultaneously from at least two broadcast channels and to reserve some space for caching purposes. Since MANET clients are bandwidth-limited in both sending and receiving, and since a video is typical long, we propose to use Staggered Broadcasting, which requires the client to join only a single channel. In the future when wireless bandwidth is improved, other periodic broadcasting schemes may also be applicable.

Staggered Broadcast works simply as follows. Since broadcasting of each video is independent from that of another, without loss of generality, we focus on a single video. This video is partitioned into $K$ equally sized segments. Given the video consumption rate $r$ (Mbps), we allocate a server bandwidth of $r \times K$ for the video[2]. This bandwidth allocation is divided into $K$ logical channels, each repeatedly broadcasting the video with a transmission rate equal to the consumption rate. The scheduling of these broadcasts is illustrated in Figure 2.

Each local forwarder joins all those broadcast channels and therefore receives every packet broadcast from the

2 We assume that the server has enough bandwidth for this allocation

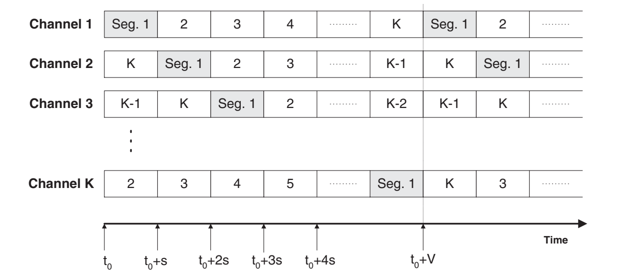


**Figure 2. Broadcasting video segments at the server/local forwarder: $s$ is the duration of a segment, $t_0$ the start time of server broadcast, and $V$ the duration of the video**

server. This local forwarder broadcasts, by radio waves, these packets to its service area. Its broadcast schedule is exactly the same as that of the server. As a result, a client in any service area can receive the broadcast packets. The playback procedure at a client follows the simple algorithm below:

Procedure ClientPlayback()
1. Detect a local forwarder $LF$
2. Find the channel from $LF$ that is going to broadcast the first segment soonest
3. Wait to join that broadcast
4. Download and play the video data from this channel
5. Quit this channel

Using Staggered Broadcast, a client joins only a single broadcast channel at all time, thus the client bandwidth requirement is no more than the playback rate. In addition, the rendering at the client only consists of receiving a video packet, decode, and display it, the computational complexity required for playback is minimal.

The value of $K$ is determined depending on the bandwidth limit of local forwarders. Suppose that $N$ is the number of videos broadcast from the server and $B$ (Mbps) the bandwidth capacity of a local forwarder. If every video is transmitted with a rate $r$ (Mbps), $K$ is chosen according to the following relationship: $r \times K \times N \leq B$. If $N$ = 5, $B$ = 54 (IEEE 802.11g), $r$ = 1.5 (MPEG-1), $K$ should be a number less than 7.

A disadvantage of Staggered Broadcasting is a high service delay. If a client requests to join MobiVoD in the middle of a broadcast of segment 1, this client already misses the already-broadcast packets belonging to segment 1 and therefore must wait until the next broadcast of this segment. For instance, in Figure 2, if a client requests at time $t_0 + s + \delta$ ($0 < \delta < s$), it must wait until time $t_0 + 2s$ to start downloading segment 1. Hence, the service delay is $s$ - $\delta$; in the worst case, it is $s$. Supposing $K$ = 5 channels, which is likely to be the case with IEEE 802.11g and MPEG-1, broadcasting a 60-minute video results in a worst-case delay of $s$ = $V/K$ = 60/5 = 12 minutes. In the next section, we propose a caching scheme in which clients may share their first segment to erase this delay.

### 2.3. Client Caching

In MobiVoD, a client may have the following two buffers:

- Reusable Buffer: This buffer is used to cache the first segment of the video. Therefore, the size of this buffer is that of the first segment. A client needs a reusable buffer if it is selected to cache the first segment.
- Prefetched Buffer: This buffer is used for video playback. The size of this buffer is that of the already-broadcast portion that the client misses. A client needs this buffer if it opts to make use of the first segment cached at a nearby client's reusable buffer.

Firstly, let us assume that every client in service caches the first segment. We will relax this assumption later as we introduce our selective-caching schemes. Let us consider a new arriving client $X$ who detects that it already misses the current broadcast of the first segment. Instead of waiting for the next broadcast of the first segment, this client looks for an existing client $Y$ in its transmission range, who has a cache of the first segment in the reusable buffer. If such $Y$ exists, $X$ can download and play the missing portion from $Y$, and, at the same time, store the packets broadcast from $X$'s local forwarder into the prefetched buffer. Once $X$ finishes playing the missing portion, it switches to play the data in the prefetched buffer. In this case, though $X$ misses the current broadcast of the first segment, $X$ still manages to watch the video immediately.

Since the clients are bandwidth limited, we should not let an existing client forward the cached data to more than one other client at the same time. Therefore, in choosing $Y$, we skips clients that have been forwarding its cache to some other client. The playback procedure for the new client is summarized below:

Procedure ClientPlaybackWithCache()
1. Select the first client $Y$ such that:
    $Y$ is in $X$'s transmission range
    $Y$ holds segment 1 in reusable buffer
    $Y$ currently is not forwarding cache to other clients
2. If such $Y$ does not exists
3. Run ClientPlayback()
4. Else
5. Run the two following tasks in parallel:
6. Task 1:
7. Detect a local forwarder $LF$

```
        Find the channel from LF that is broadcasting
            the first segment soonest
        Wait to join this broadcast
        Download and cache packets into prefetched buffer
        Quit this channel
    Task 2:
        Download/Play the missing portion from Y's
            reusable buffer
        Play data in prefetched buffer
        Note: Segment 1 is cached into reusable
            buffer during the previous two steps
```

Clearly, by caching the first segment at every existing client, we significantly increase the chance subsequent clients can join the service instantly. The tradeoff of this "cache-everywhere" strategy, however, is the amount of storage space needed for caching. We propose to use selective caching, in which only a number of selected clients need to cache. The advantage of selective caching is the saving on caching space, the disadvantage being with a higher service delay. Our purpose is to find out what selective-caching algorithm results in better service delay given the same saving on caching space.

The simplest selective-caching algorithm is *Random-cache*, which lets a client cache the first segment with some probability. The disadvantage of random caching is that even though a new client may find some existing clients in its transmission neighborhood, these existing clients may not keep any cache. To avoid this situation, we propose an alternative caching approach, called *Dominating-Set Cache* (DSC), in which the following steps are taken:

1. Compute a set of clients $DSet$ so that for each client $Y$ outside $DSet$, there exists a client $X \in DSet$ such that $X$ and $Y$ are within the transmission range of each other. $X$ and $Y$ are said to be a "neighbor" of each other and the set $DSet$ is called a dominating set of the clients.
2. Only if a client belongs to $DSet$, this client caches the first video segment.

The use of a dominating set of mobile hosts was proposed in many MANET works, mostly in broadcasting/routing protocols [23, 2, 22]. We use the concept of dominating sets for caching purposes in our mobile VOD system. Provided that caching at existing clients follows DSC, we handle a new client $X$'s arrival as follows (illustrated in Figure 3):

- Case 1: $X$ is in the transmission range of a client $Y \in DSet$ and $Y$ is not currently forwarding cache to any other client
    1. $X$ makes use of the cache at $Y$ and immediately plays the video as explained in Procedure ClientPlaybackWithCache() above.
- Case 2: $X$ is in the transmission range of a client $Y$ outside $DSet$ and not in the transmission range of any client in $DSet$
    1. $Y$ finds a neighbor $Z \in DSet$ such that $Z$ currently is not forwarding cache to any other client.
    2. $Y$ downloads the broadcast portion that $X$ misses from $Z$ and forwards it to $X$. $X$ can play the video immediately as explained in Procedure ClientPlaybackWithCache() above.
- Case 3: Neither case 1 nor case 2 holds. In this case, $X$ waits until the next broadcast of segment 1, when it will follow Procedure ClientPlayback() to play the video.

DSC guarantees that if a new client can reach an existing client, there is always a cache of the missing portion within two hops. Although multi-hop transmission is not suggested for streaming video in MANETs, our scheme should work well because two hops is short and the size of the missing portion is small, thus the cache downloading is quick.

A failure may occur while a new client is downloading its missing portion from an existing node; for example, when the existing node moves far away or quits the system. The new client detects this failure by observing that it has been waiting for the next packet for some period long enough. In this case, the new client can repeat the cache search above. However, if a new cache is found, the new client just needs to download part of the missing portion, which has not been downloaded from the previous cache. Again, since the cache is less than two hops away and the missing portion is short, we expect a small probability that a client needs to switch to a new cache.

Many distributed algorithms were proposed to solve the dominating set problems in MANETs, such as in [23, 2, 6]. Since mobile hosts may move or fail, these algorithms allow a mobile host to change status from "not in dominating set" to "belong to dominating set". Our situation is different. We need to decide if a client belongs to $DSet$ as soon as it joins the system. If it is in $DSet$, it will cache the first segment. If DSC decides a client is not in $DSet$, this client will not hold any cache and therefore will never belong to this set in the future. Therefore, we just use the following policy to decide whether a new client is going to cache: *Initially, there is no client and $DSet$ is empty. A new client belongs to $DSet$ if and only if no client in $DSet$ is within the transmission range of the new client.* To implement this policy, the new client $X$ broadcasts a request and any client $Y$ who intercepts the request will send a reply back to $X$ if $Y$ holds a cache. If $X$ receives at least a reply, the new client decides that it will not cache (which also means $X$ is not in $DSet$). If $X$ does not receive any reply, $X$ decides that it will cache (which also means $X$ belongs to $DSet$).

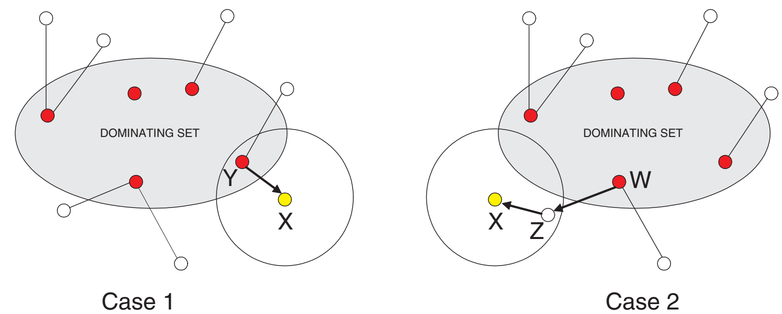


**Figure 3. If a new client can reach an existing client, the former always finds a cache within two hops**

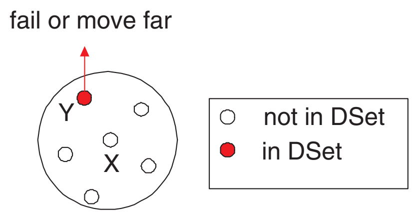


**Figure 4. $X$ is currently not in $DSet$. After $Y$ fails or moves far away, $X$ cannot be reached by any caching node. If a new client arrives close to $X$, the new client cannot make use of any cache and therefore has to wait until the next broadcast of the first segment from its local forwarder.**

We may find a case where $DSet$ becomes not a dominating set of clients, such as illustrated in Figure 4. However, this would also happen to the existing algorithms for constructing dominating sets in MANETs if we employed them to build $DSet$. The next section gives some simulation results on how occurrences of such case affect the performance of MobiVoD.

## 3. Performance Evaluation

We evaluated the performance of MobiVoD through a simulation study. We present the results of the study in this section.

### 3.1. Simulation Model

Without loss of generality, we focus on a service area surrounding a local forwarder with radio transmission radius $D$ = 100m. Of course, only clients who locate in this area are able to access the broadcast channels from the local forwarder. We assume that the service area is a 2-D circle and the local forwarder's location is at the center (0, 0). Client locations are generated in random within this circle. Furthermore, a client can contact any other client within a distance of $d$ = 20m, which is a safe distance to achieve an effective bandwidth close to the nominal IEEE 802.11b bandwidth.

Each simulation runs for $T$ = 24 hours, during which clients join the service area with arrival times following a Poison distribution with a rate $r_{arrival}$ (clients/minute). To model client failure, we assume a client failure rate $r_{fail}$ (clients/minute) that means at every second, on average $r_{fail}$ clients fail. The failed clients are randomly selected among the existing clients. We use the following model for client mobility in our system: at each second, a client is moving with a probability $p_{move}$ to a random location within a distance $d_{move} \in (0, MoveMax]$. Although several other mobility models were proposed for MANETs [5], we believe our proposed mobility model suits VOD clients well because of the clients' interest in watching video and of the scope of our service area. We investigate broadcasting only one video of $V$ minutes, whose segments are broadcast on $K$ = 5 channels according to the Staggered Broadcasting algorithm. We believe five channels is feasible with current 802.11 standards. Table 2 summarizes the input parameters and their domain in our study.

We assess the performance of MobiVoD under three alternatives: (1) *All-cache*: every client is caching, (2) *Dominating-Set Cache (DSC)*, and (3) *Random-cache*: a client is caching with a probability which is chosen so that the number of caching clients equals the dominating set size found in DSC. We collect the results on the following output metrics:

- Service delay: The average period of time a client waits until serviced. This metrics illustrates how true on-demand the service is.
- Caching storage occupancy: The average storage occupancy for a caching client is computed as the ratio between the total caching space (in % of video size) used by caching clients to the total number of clients.

| Parameter | Notation | Default value | Value range |
|---|---|---|---|
| Service area radius | $D$ | 100m | N/A |
| Client transmission radius | $d$ | 10m | N/A |
| Simulation period | $T$ | 24 hours | N/A |
| Client arrival rate | $r_{arrival}$ | 6 clients/min | {2, 4, 6, 8, 10} |
| Client failure rate | $r_{fail}$ | 20% $\times$ $r_{arrival}$ | {0%, 10%, 20%, 30%, 40%} $\times$ $r_{arrival}$ |
| Moving probability | $p_{move}$ | 0.2 | {0.0, 0.1, 0.2, 0.3, 0.4} |
| Maximum moving distance | $MoveMax$ | 1m | N/A |
| Video length | $V$ | 60 minutes | {30, 60, 90} |
| Number of broadcast channels | $K$ | 5 | N/A |

**Table 2. Simulation Parameters**

We should reduce this caching occupancy because mobile devices have limited resources.

- Bandwidth requirement: When a client is receiving from or sending video packets to a mobile node (either local forwarder or client), the bandwidth needed is valued 1 (i.e., equal to consumption rate). The client bandwidth requirement is computed as the average bandwidth needed by a client. In our system, a high bandwidth requirement also results in a high energy consumed.

- Cache distance: A client has a cache distance of 1 or 2 if it can get a cache in 1 hop or 2 hops, respectively. The cache distance is zero if the client cannot get a cache anywhere and has to wait until the next broadcast of the first segment. We want to compute the average value of cache distance. Since transmitting video wirelessly via multiple hops is inefficient, we should keep cache distance small.

- Startup overhead: If a client is downloading the portion it misses from the current broadcast and the sending client of this portion fails or moves away, the receiving client incurs an overhead of finding a new cache holder. The startup overhead for a client is computed as the average number of times a new client needs to find a new cache due to failure in downloading the current cache. The startup overhead is zero for clients who cannot obtain a cache. It is desirable to keep the startup overhead as small as possible.

We investigate MobiVoD under the effects of client request rate, failure, mobility, and video length. For each case where an input parameter varies while the others stay fixed, we run our simulation several times. We have found that the results collected for those runs varied slightly and almost unnoticeable. Therefore, we pick one set of results for such a case and present them in the following sections.

### 3.2. Effect on service delay (Figure 5)

The average service delay without caching would be a half of the duration of the first segment, or $V/K/2$ = 60/5/2 = 12/2 = 6 minutes for 60-minute videos. It is obviously shown that caching helps reduce this delay substantially. When the client population is sparse ($r_{arrival}$ = 2), all the three caching techniques' delays are less than 90 seconds, which is 4 times better than without caching. These improvements are even more notable as the request rate increases. This is because as the client population is denser a client has a better chance to find a nearby cache, thus reducing the service delay. All-cache almost provides true on-demand services, as its offered delay is less than 10 seconds in most scenarios. DSC always outperforms Random-cache by about 10 seconds. When mobility and failure are more prone in the system, the service delay increases slowly. For instance, DSC's delay is 17 seconds when no client moves, while is only 40 seconds when 40% of the clients move every second. These results exhibit that MobiVoD's performance is stable under high dynamics of the system.

### 3.3. Effect on cache storage occupancy (Figure 6)

All-cache requires every client to cache 20% of the video, while both Random-cache and DSC requires 4 times less. Not all clients need to cache in Random-cache and DSC, and an average client needs to cache only 5% of the video in the reusable buffer, which we believe is feasible with clients with mobile devices. Given that these two techniques provide good enough service delay as we discuss earlier, they look more desirable than the All-cache scheme, especially when the request rate is higher than 4 clients/minute.

### 3.4. Effect of bandwidth requirement (Figure 7)

The bandwidth requirement does not vary significantly as the system experiences different request rate, failure rate,

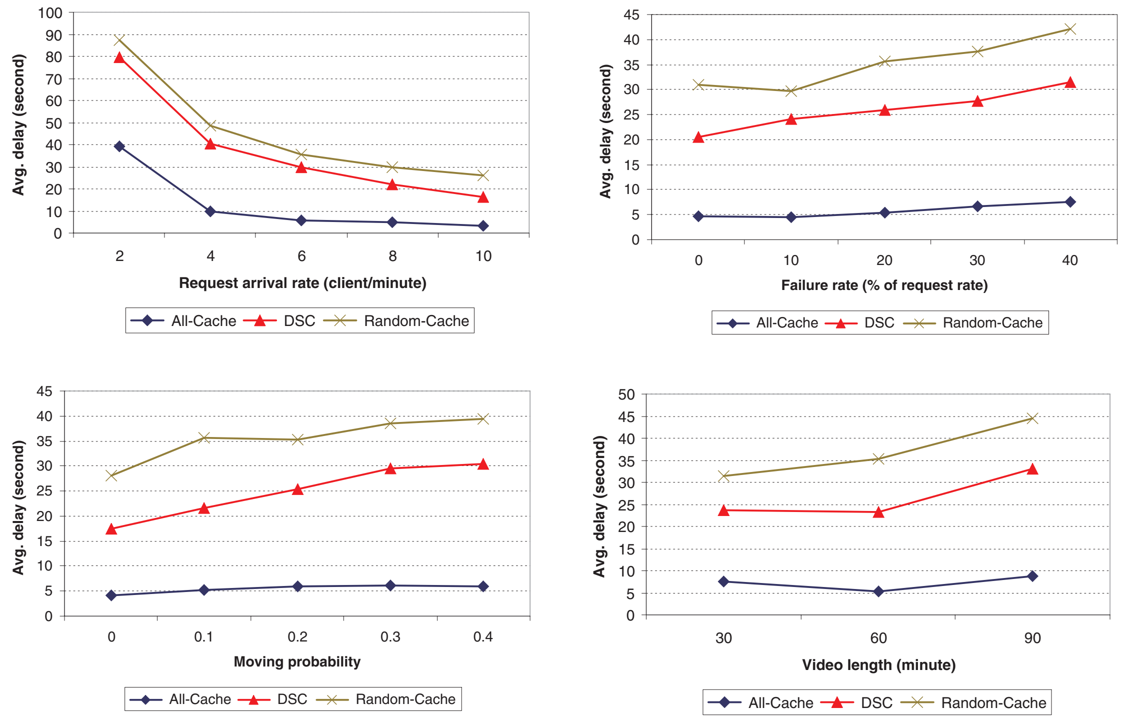


**Figure 5. Effect on service delay**

moving rate, and video length. In any case, the average bandwidth required by a client is less than 1.3 times the consumption rate (or playback rate). In contrast, as shown in Table 1, convention VOD broadcasting techniques other than Staggered Broadcasting always require bandwidths of at least 2 times the playback rate. Therefore, MobiVoD is more feasible for mobile clients enabled by current wireless technologies. There may be times when a MobiVoD uses a bandwidth of 3 times the playback rate; for instance, when this client is receiving packets from a server broadcast channel, receiving packets from a nearby cache, and forwarding the cached packets to a destination client. However, this situation lasts a short period that equals the duration of the already-broadcast data portion that the destination client misses.

We also note that DSC requires more client bandwidth than Random-cache does. However, the difference is tiny. For instance, when all the input parameters are set by default, an average DSC client needs a bandwidth of 1.185 times the playback rate while an average Random-cache client needs 1.18 times the playback rate. The bandwidth difference here is just 0.005 times the playback rate.

### 3.5. Effect on cache distance (Figure 8)

All-cache's cache distance is the shortest since a client never gets a cache more than one hop away while in the other two techniques a client may get a cache from two hops away. It is also understandable that Random-cache's cache distance is less than that of DSC. This is because DSC is more effective in using cache than Random-cache is, which is already substantiated in Figure 5 that DSC's delay is less than that of Random-cache. However, on average, a client using either scheme does not need to download a cache from more than 1.15 hops away.

When the request rate increases, the density of clients in the service area also increases and therefore the chance is higher for a client to reach a cache within two hops. This is consistent with the results shown in Figure 8 that the cache distance increases as the request rate increases.

### 3.6. Effect on startup overhead (Figure 9)

While All-cache is the best, DSC is slightly better than Random-cache in that a DSC client switches cache holders less often than a Random-cache client does. We also found that as the moving probability increases or so does the video length, it is more often for a client to find a new cache because of the broken transmission of the old cache


0-7695-2070-7/04 $20.00 © 2004 **IEEE**

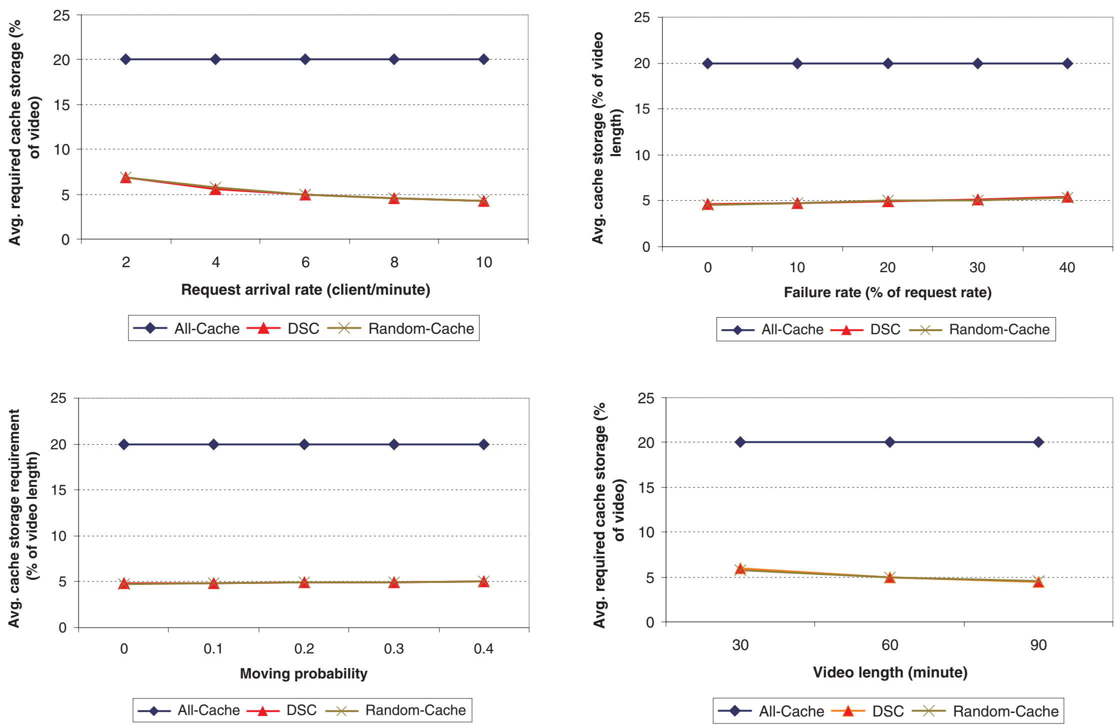


**Figure 6. Effect on cache storage occupancy**

(due to client failure or mobility). It is easily understandable for the case of video length because when it increases the size of the cache gets larger and the cache transmission is more prone to be broken. For the case of mobility, it is also explainable because more mobility means lost packets. There is also an increase in startup overhead when the failure rate increases, but this increase is slow. Overall, in the worst case, the average startup overhead is much less than two times of switching cache holders.

### 3.7. Simulation summary

We have investigated the performance of MobiVoD under three alternatives: All-cache, Random-cache, and DSC. All-cache provides almost true VOD services, however the storage for caching 20% of the video is a drawback that makes All-cache least desirable by current mobile clients. DSC and Random-cache, with much less caching space occupancies, perform similarly to All-cache in terms of bandwidth requirement, cache distance, and startup overhead. In addition, DSC and Random-cache offer service delays much better than without caching. In deed, in most scenarios, they are more than 9 times better than without caching. Between DSC and Random-cache, DSC is more preferable in that its service delay is shorter than that of the latter. On the other hand, the advantage of Random-cache is its flexibility in choosing the number of clients who will cache.

## 4. Enhancements to MobiVoD and future work

In the previous sections, we implicitly assume that clients are homogeneous, meaning they have same capabilities (ratio transmission radius, wireless bandwidth etc.). It is better to relax this assumption so the system is more accessible to different types of clients, especially those having bandwidth less than the video consumption rate. For this purpose, MobiVoD can be extended by employing a multi-resolution or layered video coding approach [3, 16]. In a heterogeneous system, two clients are considered neighbors if and only if they are in the transmission range of each other. Each video is encoded into several "layers", including a base layer and one or more enhancement layers. The base layer provides the version of least quality, while its combination enhancement layers provide better quality. Instead of broadcasting the entire video as in pure Periodic Broadcasting, the server, and thus every local forwarder, broadcasts all layers on separate channels. A new client selects a combination of layers that best match its resource constraints and only tunes in the corresponding channels to download such layers. As for the initial portion that the client misses from the cur-

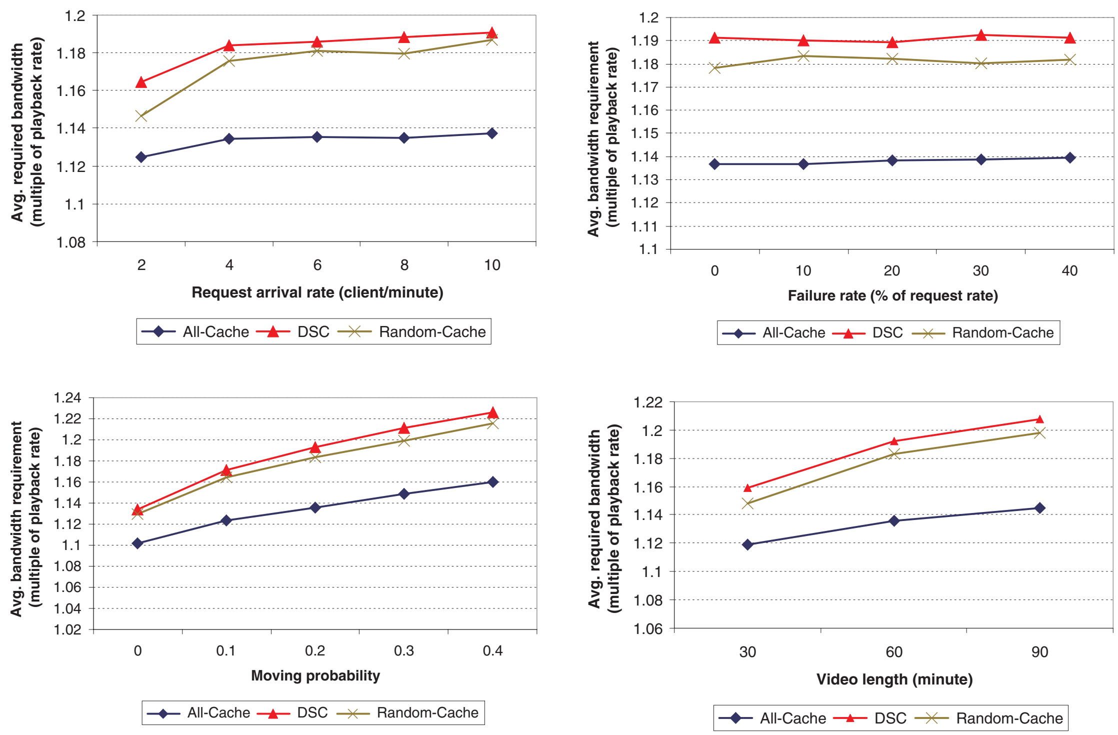


**Figure 7. Effect on bandwidth requirement**

rent broadcasts, it searches for a nearby client who caches a "version" of the first segment (a version is a combination of the base layer and one or more enhancement layers). If more than one such client are found, the client with the highest-quality version is selected.

Another extension of MobiVoD is to allow a client to download a cache more than two hops away. This enhancement would increase the chance of providing true on-demand services to clients, which is however suggested only when wireless bandwidth is more advanced than the current.

In our future work, we will conduct a simulation study on those extensions of MobiVoD. We also plan to implement MobiVoD in a real MANET environment, which includes further investigation into QoS issues (packet loss, radio interference, etc.). Although MobiVoD uses Staggered Broadcasting for disseminating videos to clients, other periodic broadcasting technique can theoretically work with our system. We will investigate how MobiVoD performs with such a technique in a realistic setup.

## 5. Conclusions

We presented MobiVoD, a framework for implementing a VOD system in mobile ad hoc network (MANET) environments. Given the increasing popularity of VOD services to computer users and the recent surge of MANETs, we believed that such a system, which enable people to easily access video information anywhere and anytime, would be beneficial to many application areas such as education, entertainment, and business.

To be feasible with the current MANET technologies, MobiVoD was designed to be simple yet efficient. MobiVoD uses periodic broadcasting as a way to disseminate videos to clients. A video is divided into segments, each broadcast on a separate communication channel. When a new client joins the system, it waits until the next broadcast of the first segment starts to download the first segment. After playing the first segment, the client immediately switches to the broadcast of the second segment to download it, and so on until all segments have been downloaded. Periodic broadcasting makes the system scalable with increases in the number of clients. However, the period a new client must wait before it starts the VOD service may be significant. In the worst case, this waiting period equals the size of the first video segment broadcast. We therefore proposed to use caches. We proposed two caching policies: Random-cache and Dominating-Set Cache (DSC). Using either schemes, when a new client requests the VOD service, instead of waiting for the next broadcast of the first segment, it joins the current broadcast immediately and downloads the video packets broadcast into a playback buffer. As for the be-

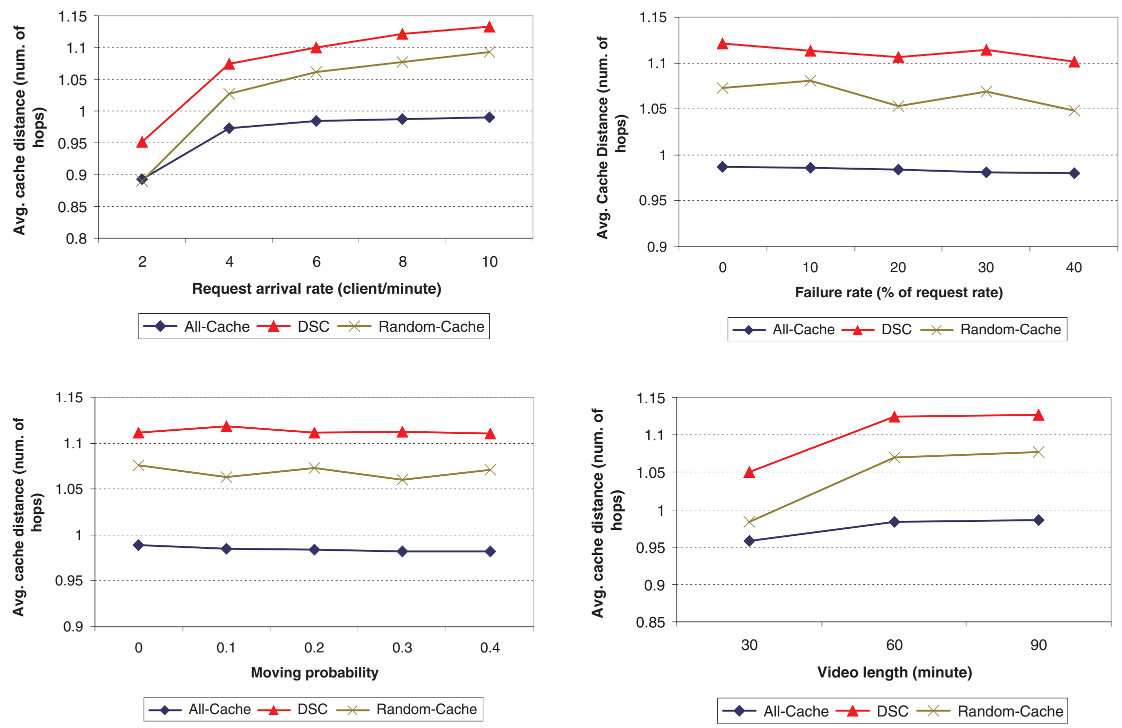


**Figure 8. Effect on cache distance**

ginning portion that was already transmitted by the current broadcast, the new client downloads and plays it immediately from a nearby cache, after which the new client switches to the playback buffer to play the rest of the video.

Our simulation-based performance study showed that MobiVoD could work well with the current wireless technologies. Unlike the conventional broadcasting techniques for VOD, which require every client to have a bandwidth of at least 2 times the playback rate, MobiVoD required most clients to have a bandwidth less than 1.3 times the playback rate. In addition, by using caches, MobiVoD improved the service delay by at least four times in comparison with the case of no caching. In most cases, a client waited less than a minute to start its requested service. We also found that though Random-cache is more flexible in controlling the number of caching clients, DSC provided true on-demand video services to more clients than Random-cache did.

MobiVoD is open to future wireless technologies and QoS enhancements. We will investigate such issues in our future work and further look at the performance of MobiVoD under realistic MANET environments.

## References

[1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. A permutation-based pyramid broadcasting scheme for video-on-demand systems. In *Proc. of the IEEE Int'l Conf. on Multimedia Systems'96*, pages 118–126, Hiroshima, Japan, June 1996.

[2] K. M. Alzoubi, P.-J. Wan, and O. Frieder. Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Lausanne, Switzerland, June 2002.

[3] S. Bajaj, L. Breslau, and S. Shenker. Uniform versus priority dropping for layered video. In *ACM SIGCOMM*, pages 131–143, 1998.

[4] Bluetooth. http://www.bluetooth.com/.

[5] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483–502, 2002.

[6] Y. P. Chen and A. L. Liestman. Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Lausanne, Switzerland, June 2002.

[7] S. Corson and J. Macker. Mobile ad hoc networks (manet): Routing protocol performance issues and evaluation considerations. *RFC 2501*, January 1999.

[8] S. Gruber, J. Rexford, and A. Basso. Protocol considerations for a prefix-caching proxy for multimedia streams. In *Proc. of the 9th International WWW Conference*, pages 657–668, 2000.

[9] K. A. Hua, Y. Cai, and S. Sheu. Patching: A multicast technique for true video-on-demand services. In *Proc. of ACM*

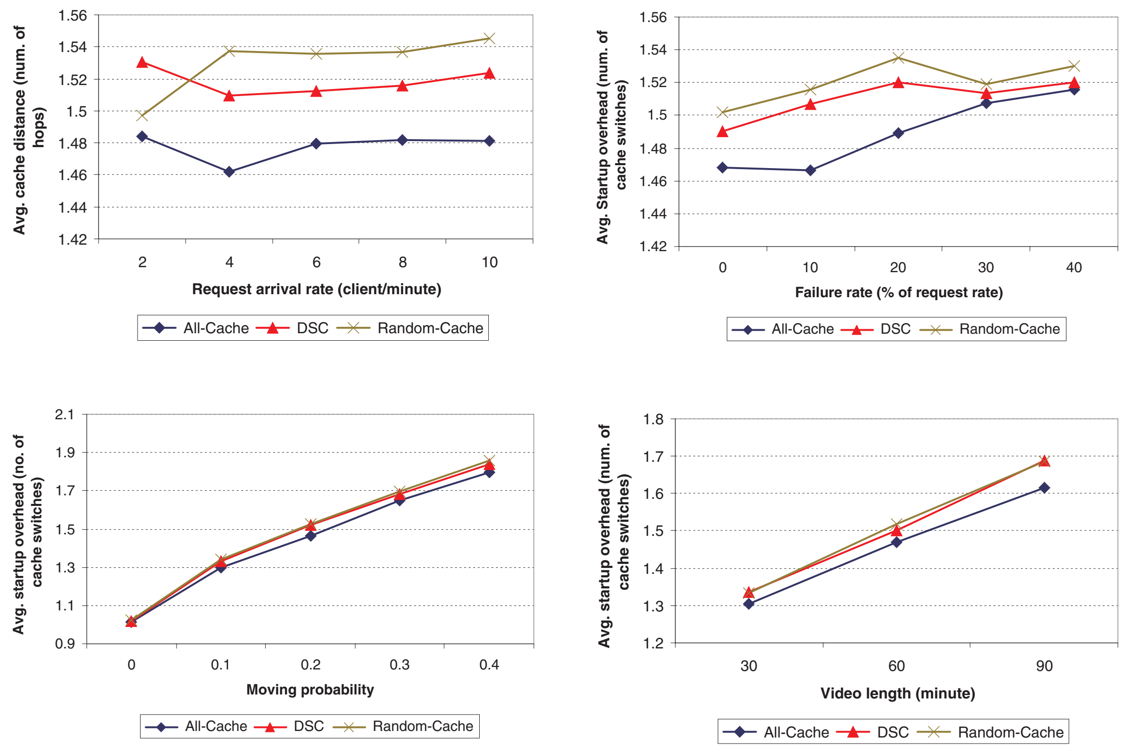


**Figure 9. Effect on startup overhead**


*MULTIMEDIA*, pages 191–200, Bristol, U.K., September 1998.

[10] K. A. Hua and S. Sheu. Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems. In *Proc. of the ACM SIGCOMM'97*, pages 89–100, Cannes, France, Sepetember 1997.

[11] K. A. Hua, D. A. Tran, and R. Villafane. Caching multicast protocol for on-demand video delivery. In *Proc. of the ACM/SPIE Conference on Multimedia Computing and Networking*, pages 2–13, San Jose, USA, January 2000.

[12] IEEE-Std-802.11. Wireless lan medium access control (mac) and physical layer (phy) specification. *1999 Edition*, 1999.

[13] L. Juhn and L. Tseng. Harmonic broadcasting for video-on-demand service. *IEEE Transactions on Broadcasting*, 43(3):268–271, 1997.

[14] L. Juhn and L. Tseng. Fast data broadcasting and receiving scheme for popular video service. *IEEE Transactions on Broadcasting*, 44(1):100–105, 1998.

[15] J. B. Kwon and H. Y. Heom. Providing vcr functionality in staggered video broadcasting. *IEEE Transactions on Consumer Electronics*, 48(1):41–48, 2002.

[16] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. In *ACM/IEEE NOSSDAV*, pages 177–186, Miami, FL, USA, May 12-14 2002.

[17] J. F. Paris, S. W. Carter, and D. D. E. Long. A hybrid broadcasting protocol for video on demand. In *ACM/SPIE Conference on Multimedia Computing and Networking*, 1999.

[18] S. Sen, L. Gao, and D. Towsley. Frame-based periodic broadcast and fundamental resource tradeoffs. In *IEEE Performance, Computing and Communications Conference*, pages 77–83, April 2001.

[19] D. A. Tran, K. Hua, and T. Do. A peer-to-peer architecture for media streaming. *IEEE JSAC, Special Issue on Advances in Service Overlay Networks*, To appear in 4rd Quarter, 2003.

[20] D. A. Tran, K. A. Hua, and T. T. Do. Zigzag: An efficient peer-to-peer scheme for media streaming. In *IEEE INFOCOM*, San Francisco, CA, March-April 2003.

[21] S. Viswanathan and T. Imielinski. Metropolitan area video-on-demand service using pyramid broadcasting. *ACM Multimedia systems Journal*, 4(4):179–208, August 1996.

[22] B. Williams and T. Camp. Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks. In *ACM Symposium on Mobile Adhoc Networking and Computing (MOBIHOC 2002)*, June 2002.

[23] J. Wu. Extended dominating-set-based routing in ad hoc wireless networks with unidirectional links. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):866–881, 2002.

[24] K.-L. Wu, P. S. Yu, and J. L. Wolf. Segment-based proxy caching of multimedia streams. In *Proc. of the 10th International WWW Conference*, pages 36–44, Hong Kong, 2001.

[25] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava. On peer-to-peer media streaming. In *IEEE Conference on Distributed Computing and Systems*, pages 363–371, July 2002.