An Adaptive Strategy for Mobile Ad Hoc Media Streaming

Min Qin and Roger Zimmermann, Senior Member, IEEE

Abstract—Mobile devices are increasingly popular and many of them are capable of handling multimedia content. Users enjoy the ability to access their collection of media objects anywhere. Wireless connectivity is often integrated into these handhelds, therefore providing the opportunity to stream multimedia content among mobile and ad hoc peers. An important consideration is to transfer a multimedia object in its entirety. This is often challenging since the transmission time for such an object can be considerable due to an unfavorable combination of a large object size and limited available bandwidth. We previously introduced a novel strategy to improve the probability of success to stream a video sequence based on studying the minimum buffer size. This strategy takes advantage of layered video encoding schemes such as Scalable Video Coding (SVC) or Multiple Description Coding (MDC). The technique adaptively selects the number of layers to be streamed to deliver more frames before the wireless link disconnects while keeping the video quality high. In our current study we simplify this strategy by using the streaming probability alone to dynamically adjust the number of layers to be delivered. Our proposed technique improves the prediction accuracy by incorporating the 802.11 Auto-Rate Fallback (ARF) scheme along with two popular mobility models: the random waypoint and the random walk mobility models. While ARF—which steps down the sending rate when consecutive transmission errors occur—is implemented in all hardware that follows the popular IEEE 802.11 standard, it is not commonly modelled in existing work. In addition, our approach can re-transmit missing layers if peers reconnect after a link break, hence improving the rendering quality. We have performed extensive simulations to validate our technique and the results show an improvement in streaming probability as well as the number of layers that are transmitted.

Index Terms—Content distribution, mobile environments, media streaming, wireless ad hoc communication.

I. INTRODUCTION

Streaming audio or video content between mobile ad hoc devices is an interesting application, especially with the widespread deployment of smartphones and other handhelds. As an example, consider an urban environment where users are watching short video clips that are transmitted to their devices via mobile TV (e.g., the Iseng ISDB-T service in Japan; many mobile devices include a Iseng tuner which operates in the licensed, wide-area frequency spectrum from 470 to 770 MHz). Some city areas, say a park, are quite unobstructed at close range but may experience difficulties to receive mobile TV signals because of shielding by high-rise buildings in the surrounding areas. This issue has been acknowledged for China’s upcoming DTTB mobile TV service as a challenge in its metropolitan cities with tall and compact buildings [1]. In such situations, users in weak TV reception zones might be interested in watching what others, who are just entering the weak zone from a strong reception zone, have previously recorded. By allowing sharing of video content via MANETs (mobile ad hoc networks) the effects of “blind spots” can be diminished and the overall experience for many users improved (Fig. 1). In our study we focus on a scenario where the users who request the video clips would like to experience minimal startup delay and smooth, timely playback. Exemplar systems that implement such media sharing ideas include commercial devices such as Microsoft’s Zune and Apple’s iPhone and iPod Touch devices, and experimental systems such as Bubbles [2] and MStream [3]. Many of the recent handhelds can operate via wireless IEEE 802.11 networks, which provide broadband-level bandwidth and a communication range of hundreds of feet. This allows a user to move freely when she is streaming video content from others who are within her communication range. Therefore, video streaming among mobile ad hoc peers is feasible and enables interesting new applications. This trend is further driven by content providers who are increasingly interested to reach mobile users. As an example, in the United Kingdom the BBC has started to stream live TV for compatible handsets in the spring of 2009.

There are a number of challenges when streaming video content among mobile ad hoc peers. First, a primary concern is to ensure the complete delivery of a media object—which is often sizable—across a wireless link whose quality may be constantly changing. For example, the wireless bandwidth may diminish or the link may even disconnect as the distance between two mobile ad hoc peers increases. Therefore, the probability whether a video sequence can be successfully streamed before the ad hoc link breaks is of special interest to mobile users. Second, in many situations receiving all the video frames is of more importance than achieving the best video quality. For example, with a news video clip some of the quality can be sacrificed as long as the complete story is
conveyed. Our approach makes use of layered video coding (SVC or MDC) to achieve a high probability that lower layers are received in a timely manner. A third challenge is posed by the modelling of a realistic wireless environment; a task which can be highly complex. For this reason many previous studies assume the network bandwidth to be constant when analyzing the streaming performance between two mobile ad hoc peers. Furthermore, most existing techniques are concerned with link or path duration ([4], [5]) when the network has reached its stationary regime [6]. Finally, traditional approaches often do not utilize the locally available information of a node in predicting future link status.

In our prior work we have investigated some key issues pertaining to wireless ad hoc streaming. We devised an algorithm for calculating the probability of successfully streaming a video sequence between two mobile ad hoc peers over a wireless link in the foreseeable future [3], [7]. We extended the work to support the 802.11 Auto-Rate Fallback (ARF [9]) scheme [8], which is vital for a realistic model of wireless communication and has not been considered in other studies. Additionally, frames can be labelled with priorities to reflect their importance in the entire sequence to achieve a better viewing experience. Our strategy takes advantage of the multilayer structure of recent video encoding schemes, such as Scalable Video Coding (SVC) [12] and Multiple Description Coding (MDC) [13], [14]. By regularly monitoring the available buffer size at the receiver side, our streaming strategy adaptively selects the number of layers to be streamed to ensure the delivery of the maximum possible number of video frames [8].

![Diagram](image)

**Fig. 1.** Example scenario where some urban residents in strong radio signal reception zones capture mobile TV news clips and then share them with others in weak reception zones via ad hoc networking. The dashed circles indicate the ad hoc transmission ranges.

The purpose of this study is to build upon and extend our prior work, refine the probabilistic method and combine the different aspects of the ad hoc media streaming strategy into a coherent, self-contained approach. Specifically, the contributions of this study are as follows:

- We derive a more general equation for modeling the streaming probability. As in our prior work, the method does not require any simplifying assumptions such as uniform streaming bandwidth, fixed node locations, or infinite buffer size.
- We introduce a novel layer re-transmission scheme that further improves the streaming probability — and hence the video quality — in the case when peers re-connect after a link break.
- Extensive experimental results have been conducted and included based on ns-2 simulations, while our earlier work was purely based on theoretical analysis.

The rest of paper is organized as follows. In Section II the prior and related work is described. The calculation of the probability for successfully streaming a video sequence is reviewed in Section III. Section IV describes the streaming strategy to improve streaming probabilities in mobile ad hoc environments and Section V presents a detailed performance evaluation of our scheme. Further improvements of the streaming strategy are presented in Section VI. Finally, we summarize our contributions in Section VII.

## II. PRIOR AND RELATED WORK

When obstacles are present, the coverage area of an 802.11 wireless device is no longer a perfect circle. Therefore, it becomes very difficult to analyze the link quality due to the irregular shape of the wireless coverage. For simplicity, we focus our discussion on an open environment, where there are very few obstacles. In such an environment, the signal strength of an 802.11 protocol can be approximated by the two-ray ground model [15] with good accuracy [16], [17]. Note that we do not model interference among different wireless devices due to the inherent complexities of a dynamic environment. The two-ray ground model can be expressed through the following equation:

\[
p_r(d) = \frac{(h_t h_r)^2}{d^4} G_t G_r p_t
\]

Here \(p_r(d)\) is the received signal strength and \(p_t\) is the transmitted signal power. \(G_t\) and \(G_r\) represent the antenna gains of the transmitter and the receiver, \(h_t\) and \(h_r\) are the heights of the transmit and receive antennas, respectively, which are positioned at a distance \(d\) apart. Therefore, given a certain transmission power and receiver sensitivity, we can easily calculate the communication radius \(R\) of a wireless ad hoc device.

Given a known communication radius, there exist many link estimation models [3]–[5], [11], [18] for studying the link duration or the probability that two mobile ad hoc nodes are connected over time. Some of these works [4], [5] are based on the random waypoint mobility model. They simply assume two mobile nodes move in a straight direction before the link breaks up. This may work well for vehicular streaming applications since vehicles tend to move in a straight direction. According to Yoon et al. [31], the random waypoint model is considered harmful because it suffers from speed decay (i.e., the expected speed during steady state is lower than the initial speed) and non-uniform node distribution. The random
walk mobility model, on the other hand, divides a node’s movement into a sequence of intervals called mobility epochs. Each epoch is a random period of time that is exponentially distributed. During each mobility epoch, a node moves with a constant speed and direction. The speed is a random variable uniformly distributed between $[0, v_{max}]$ and the direction is uniformly distributed over $[0, 2\pi]$. Compared with the random waypoint mobility model, the random walk mobility model does not suffer from speed decay and non-uniform node distribution [6]. As a result, future link availability is not affected by a host’s present location. Such location independence makes it possible to introduce a mathematical framework to model the continuous link availability between two ad hoc peers. However, the random walk mobility model has not been widely analyzed due to its complexity.

In our previous work [3], [7], we introduced an algorithm to calculate the probability $L(d_0, t)$ that a link between two mobile ad hoc peers can continuously last for $t$ seconds given their initial distance $d_0$. Note that Table II lists all the terms that are used in this study. According to Jiang [18], the process that the link between two moving nodes experiences velocity changes is a Poisson process. Therefore, the probability that there are $k$ changes of velocity to this link within $t$ seconds is given by

$$C(k, t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!} \quad (2)$$

where $\lambda$ is the mean duration between velocity changes. Let $L_k(t)$ denote the probability that a link is continuously available from $t_0$ to $t_0 + t$ given that the link has experienced $k$ changes in velocity. According to Equation 2, we have

$$L(d_0, t) = \sum_{k=0}^{\infty} L_k(t) C(k, t) \quad (3)$$

Our previous work [3] used an iterative algorithm to calculate $L_0(t), L_1(t), ... L_k(t)$. Experimental results show that the initial distance $d_0$ does not have a big impact on $L(d_0, t)$. This is because both nodes can either move towards or move away from each other at the beginning. Therefore, we can simplify $L(d_0, t)$ with $L(t)$. Compared to other link availability models [11], [18], which studies the availability of a link even if it experiences disconnections, our recursive algorithm is more suited to streaming applications due to its continuous nature. However, similar to most other studies, our prior work assumes a uniform streaming bandwidth for all distance settings and an infinite buffer size at the receiver. For real world situations, the buffer size of most handheld devices is limited by their physical main memory size, which is often no more than several dozen megabytes. Furthermore, ARF is used to adapt to changes in link quality for IEEE 802.11 protocols. When consecutive transmission errors occur, the sender will step down its transmission rate by using another modulation scheme. Consequently, wireless bandwidth may vary as the distance between the two peers changes. In order to address these issues, we extended our design [8].

Most 802.11 protocols support multiple modulation techniques and receivers have varying sensitivities when operating with these modulations. For example, Table I shows the modulation techniques and receiver sensitivities for the Orinoco 802.11b wireless card [20]. Consequently, 802.11 protocols have different communication radii when operating with different maximum bandwidth. As shown in [19], the goodput of 802.11 protocols drops suddenly as the distance between two ad hoc nodes increases above a certain threshold, as illustrated in Figure 2. As a result, the communication range of the 802.11 protocol can be classified into a number of zones. Different zones are associated with different mean goodput.

Figure 2 shows an example of the relative zone distribution of the 802.11a protocol. Many studies (e.g., [21]) have verified that bandwidth changes abruptly rather than smoothly when an ad hoc device moves across the zone boundary of another device. Therefore, to accurately model streaming probabilities in mobile ad hoc networks, both continuous link availability and non-constant wireless bandwidth need to be considered. In our previous work [8], we introduced an algorithm for calculating the streaming probability $P$ with both the random walk and random waypoint mobility model. Our results show that the receiver buffer size $B$ has a significant impact on the probability to successfully stream a large-sized video sequence $S$. If the buffer $B$ is small, it saturates quickly and a major portion of the wireless bandwidth is wasted afterwards. Therefore, it requires a long time for the video sequence to be successfully streamed and the streaming probability may be greatly lowered. In contrast, a large buffer size ensures that we can use the maximum possible bandwidth to stream the video sequence if the link can last long enough.

NonStop [22], [23] is another approach for enhancing media streaming performance in MANET environments. This work is based on the Reference Velocity Group Mobility (RVGM) model [32], which classifies mobile users into multiple mo-

![Fig. 2. Example bandwidth zone distribution of IEEE 802.11a with Auto-Rate Fallback (ARF) [19]. Note that ARF refers to the Adaptive Auto-Rate Fallback technique.](image-url)
bility groups. Each mobility group has a logical center and group members move with a similar velocity as the logical center. When a network partition is about to take place between two mobility groups, a server predicts the remaining link duration and replicates the service to another group. One drawback related to the NonStop protocol is that it predicts the remaining link duration by assuming that all nodes do not change their current velocities until the network partition takes place. This assumption may work well when the streaming time is relatively short. However, the RVGM model is based on the random walk mobility model. This makes the above assumption inaccurate if the streaming time is long. Therefore, our work analyzes link availability between individual mobile clients. Additionally, identifying mobility groups is very difficult.

Many recent video encoding schemes are based on a multi-layered structure to adapt to the heterogeneous nature of modem computer networks. For example, SVC divides a video sequence into a base layer and several enhancement layers. The base layer can be decoded independently to reconstruct the original video stream with a lower resolution or frame rate. Enhancement layers can be added to the base layer to improve video quality. To be error resilient, base layer packets may be given higher priority for error protection. MDC, on the other hand, divides the original video stream into several correlated bitstreams and each stream can be decoded independently. Previously we proposed a new streaming strategy by utilizing the multi-layer feature of these encoding schemes [8]. This strategy dynamically adjust the number of layers to be streamed to the client. It allows users to watch the entire video by minimally sacrificing the video quality in a mobile peer-to-peer streaming environment. Note that some characteristics of SVC and MDC are different, for example error resilience and coding overhead. Additional techniques, such as unequal error protection for different layers in SVC can be combined with our approach. The reader interested in system-related aspects of MDC is referred to the work of Mayer-Patel [33]. Our results show that the receiver buffer size has a very significant impact on the probability to successfully stream a video sequence when the video size is large. If the buffer is small in size then it saturates quickly and a large portion of the wireless bandwidth is wasted thereafter. Hence, a long time is required for the video sequence to be successfully streamed and the streaming probability may be greatly lowered. In contrast, a large buffer size ensures that we can use the maximum possible bandwidth to stream the video sequence if the link can last long enough. In this paper, we reduce the complexity of our previous approach by omitting the calculation of the minimum buffer size. Instead, we directly use the streaming probability to adaptively adjust the number of layers to be sent. This new technique results in a simpler algorithm that is nonetheless capable of the same performance.

Other work exists in the area of transcoding data for transferring web data (images, etc.) and on hand-off of connections to mobile clients. Such techniques generally address somewhat different problem statements and their solutions are therefore not directly applicable to improve video streaming with ad hoc wireless networks.

### III. MODELLING STREAMING PROBABILITY FOR 802.11 NETWORKS

For simplicity, this work assumes that the maximum bitrate of the multi-layer stream is smaller than \( b_{wM} \). Since most media streams of handheld devices use maximum bitrates of less than 400 kbps (e.g., DivX 5) because of their small display dimensions, it can be supported by the lowest bitrate modulation scheme (1 Mbps) of the 802.11b protocol. In order to dynamically change the number of layers to stream and adjust the video bitrate, the client needs to constantly monitor the buffer information. We denote with \( fbr \) the video bitrate that the server will be using to stream the video content from the current time. For example, if the server decides to only stream the base layer according to the current network and buffer conditions, \( fbr \) would be equal to the bitrate of the base layer. However, the client’s current playback rate may not be equal to \( fbr \). The reason is that the client may have already buffered \( t_b \) seconds of video and the buffered frames may include multiple layers. Let \( br(t) \) denote the video playback rate on the client at the future time \( t \). If \( t_b > t \), the client will still be consuming the currently buffered frames at time \( t \). If \( t_b < t \), after consuming all its buffered content in \( t_b \) seconds, the client will utilize a playback rate of \( fbr \) in the next \( t - t_b \) seconds. Therefore, \( br(t) = fbr \) if \( t > t_b \).

In Table II, \( bw_i \) denotes the average bandwidth of the \( i \)th zone. Because of the MAC (media access control) layer overhead and different modulation schemes, a realistic \( bw_i \) is lower than the maximum nominal bandwidth of that zone. For example, when operating with CCK11 modulation, the actual usable bandwidth for UDP is around 5.5 Mbps, rather than the physical-layer maximum of 11 Mbps. Additionally, 802.11 collision avoidance schemes such as CTS/RTS may also reduce channel utilization. Other factors that can affect the effective bandwidth include obstacles and interference. The precise modeling of wireless bandwidth in real world environments including all these factors would be very complex as it depends on the details of the environment. In
our study we assume that the average goodput of each zone can be measured, either by monitoring it from time-to-time or by studying previous statistics. To analyze the streaming probability, we assume the following properties between two mobile ad hoc peers:

- Data can be streamed from one node to another as long as the distance between them is smaller than $R_M$.
- A link is considered “broken” if the distance between two nodes is larger than $R_M$.

Let $m$ and $n$ denote the two mobile ad hoc nodes of a given link. Both $n$ and $m$ can move independently with regard to each other. The distance between them is approximated by $d_0$. An accurate estimation of the distance can be obtained through signal strength [25], Time of Arrival (ToA) [26], [27], acoustic sensing [28] or GPS, if applicable. To simplify our discussion, we fix node $n$’s frame of reference to $m$’s position and move $n$ relative to $m$. Therefore, the temporal wireless bandwidth between $n$ and $m$ is determined by the current bandwidth zone of node $m$ in which $n$ is currently located in.

When the simulation area is large, nodes tend to move away from each other after the link between them is broken. Therefore, it is very unlikely for two mobile nodes to get reconnected before the connection times out. In order to guarantee seamless delivery of a video sequence, the content has to be streamed to the receiver before the link is broken. Let $t_i$ denote the total time that node $n$ is within the $i$th zone of node $m$ before the link between them breaks. The total link duration is thus given by $t = \sum_{i=1}^{M} t_i$. According to the actual bandwidth distribution, the maximum possible amount of data that can be streamed between $m$ and $n$ is $\sum_{i=1}^{M} t_i b_{wi}$. Meanwhile, the amount of data that can be consumed by the receiver during time period $t$ is determined by $\int_{s=0}^{t} b_r(s) ds$. Therefore, the receiver buffer is saturated if $\sum_{i=1}^{M} t_i b_{wi} - \int_{s=0}^{t} b_r(s) ds > B$. Since the maximum amount of data that can be buffered is $\min(B, \sum_{i=1}^{M} t_i b_{wi} - \int_{s=0}^{t} b_r(s) ds)$, the total amount of data that can be streamed is $\min(B, \int_{s=0}^{t} b_r(s) ds, \sum_{i=1}^{M} t_i b_{wi})$. To guarantee delivery of the video sequence before the link breaks, the following inequality has to be satisfied:

$$\min \left( B + \int_{s=0}^{t} b_r(s) ds, \sum_{i=1}^{M} t_i b_{wi} \right) \geq S \tag{4}$$

By calculating the $t$ and $t_i$ constraints from Equation 4, the streaming probability is determined by the smaller of the two between $L(d_0, t)$ and $L(d_0, \sum t_i)$. From the first part of Equation 4, we can either reduce the video stream size $S$ or the bitrate $b_r(t)$ to improve the streaming probability for a fixed value of $B$. Since $b_r(t)$ is known, the client can calculate the approximate time $t$ from:

$$B + \int_{s=0}^{t} b_r(s) ds = S \tag{5}$$

After solving $t$, we can use the results detailed in [29] to calculate the streaming probability $L(d_0, t)$. However, the second part of Equation 4 is hard to solve since the probability to stream a video sequence is largely determined by the time period $t_i$ that node $n$ spends in each zone of $m$. In our previous work [8] we introduced a method for calculating the streaming probability with both the random walk and random waypoint mobility models. For the random waypoint model, because nodes do not change their velocities before the link breaks, we can easily calculate the link duration based on their initial velocities and distance. The streaming probability can thus be determined based on the duration a node spent in each zone of the other node. For the random walk model, it is very challenging to analyze due to its randomized nature. Here, we simplify our discussion to the case when $d_0 \leq R_i$. Hence, initially node $n$ can stream the multimedia content from $m$ with bandwidth equal to $bw_i$. In order to estimate the probability to stream a media object, we introduce a recursive algorithm for calculating the approximate streaming time $t$.

Let $R_{mn}(t) = R_n(t) - R_m(t)$ represent the relative mobility vector between $n$ and $m$. Given that the magnitude of $R_{mn}(t)$ is $r$ at time $t$, the probability that node $n$ is located within zone $1 \ldots i$ after $t$ seconds is given by

$$P\{d(t) \leq R_i| R_{mn}(t) = r \} = \left\{ \begin{array}{ll}
1 & \cos^{-1} \frac{d_0^2 + R_i^2 - r^2}{2d_0R_i} \\
\frac{1}{\pi} & \cos^{-1} \frac{d_0^2 + R_i^2 - r^2}{2d_0R_i}
\end{array} \right\} \quad r \leq R_i - d_0 \quad r > R_i - d_0 \tag{6}$$

According to [3], the node distribution is similar whether the link has to be continuously available or not. Consequently, by requiring node $m$ and $n$ to be continuously connected, the conditional probability that node $n$ is located within zone $1 \ldots i$ after $t$ seconds can be approximated by

$$P\{d(t) \leq R_i| d(t) \leq R_M \} = \frac{P\{d(t) \leq R_i \}}{P\{d(t) \leq R_M \}} \tag{7}$$

To simplify our representation, let $P_i(t)$ denote the probability that node $n$ is within zone $i$ after $t$ seconds, given that $m$ and $n$ are continuously connected. Define $P\{d(t) \leq R_0| d(t) \leq R_M \} = 0$ for all $t$ values and we have

$$P_i(t) = P\{R_{i-1} \leq d(t) \leq R_i| d(t) \leq R_M \} = P\{d(t) \leq R_i| d(t) \leq R_M \} - P\{d(t) \leq R_{i-1}| d(t) \leq R_M \} \tag{8}$$

Given a time period of $t$ seconds, the expected total time that node $n$ is within the zone $i$ of node $m$ is given by

$$t_i = \int_{s=0}^{t} P_i(s) ds \tag{9}$$

Combining Equations 4 and 9, we have

$$\min \left( B + \int_{s=0}^{t} b_r(s) ds, \sum_{i=1}^{M} t_i b_{wi} \int_{s=0}^{t} P_i(s) ds \right) \geq S \tag{10}$$

To determine the upper bound of $t$ from the second half of Equation 10, we introduced a recursive algorithm to approximate the value in [8]. Note that we are studying the case when $m$ and $n$ are continuously connected. By obtaining a rough estimation of the streaming time $t$, we can use our link availability algorithm to approximate the streaming probability as $L(t)$, which is the probability that $m$ and $n$ are continuously connected for $t$ seconds.

Because the maximum possible amount of time to stream an object of size $S$ is $\frac{S}{bw_M}$, it is chosen as the initial value
of \( t \) in our recursive approach. However, the actual streaming time should be much smaller than \( \frac{S}{bw_M} \) since the streaming bandwidth is not always \( bw/M \). Given that \( t = \frac{S}{bw_M} \), we can calculate the expected time \( t_i \) that the streaming bandwidth is \( bw_i \) by integrating node \( n \)'s distribution in zone \( i \) over the \( t \) seconds time period from Equation 9. Based on such an integration, we obtain a new average bandwidth and a new streaming time \( t_{new} \). The above procedure continues until \( t_{new} \) is within a certain threshold from its previous value. Algorithm 1 shows the procedure of this recursive algorithm. Proof of the correctness of the algorithm is given in [8].

Algorithm 1 Recursive algorithm for finding the ad hoc streaming time.

1: \( t_{new} = \frac{S}{bw_M} \), \( t = 0 \)
2: while \( |t - t_{new}| \geq \epsilon \) do
3: \( t = t_{new} \)
4: \( i = 1 \)
5: while \( i \leq M \) do
6: \( E(t_i) = \int_0^t P_i(s)ds \)
7: \( i = i + 1 \)
8: end while
9: \( newbw = \sum_{i=1}^{M} \frac{bw_i E(t_i)}{t} \)
10: \( t_{new} = \frac{S}{newbw} \)
11: end while

The actual convergence rate of our recursive algorithm is very fast. When \( \epsilon \) is chosen as 1, for example, it takes only 3 to 4 iterations for our recursive algorithm to finish for all different \( S \) values.

Figure 3 shows the streaming probability if nodes are moving according to the random walk mobility model. To verify the accuracy of our recursive algorithm, we simulated the streaming process with ns-2 and MATLAB. The simulation results are shown in lines marked with a cross. We use the same wireless card model from Table I. As shown in Figure 3, the buffer size has an immense impact on mobile streaming performances when the video size \( S \) is large. Note that the simulation results obtained from ns-2 are an almost perfect match with the MATLAB results and hence two curves each for \( B = 1 \) MB, \( B = 10 \) MB, and \( B = 20 \) MB overlap in Figure 3(a). For \( B = 40 \) MB there is a slight divergence for video sizes larger than 30 MB. In Figure 3(b), for example, there is a sharp drop of the streaming probability when the video size increases above 20 MB. In this case, the receiver's buffer will quickly get saturated since the sender can stream the video content using the full wireless bandwidth. After the buffer is saturated, the sender can no longer stream faster than the receiver is able to consume. As a result, it takes a much longer time to stream the remaining content and the streaming probability may be greatly affected due to node mobility.

Next we investigated the streaming time (Figure 4). When the initial distance \( d_0 \) is small, the simulation result is nearly identical to the result from our recursive algorithm. However, as \( d_0 \) increases, the actual streaming time \( t \) has wider variations. Therefore, approximating the streaming time by a single \( t \) value and then calculating \( L(t) \) as the streaming probability may not be very accurate. When \( d = 200 \) m, for example, there is a maximum of 15% difference between the simulated and predicted results. Consequently, our algorithm is more accurate for smaller \( d_0 \) values. Figure 4 shows the confidence interval of the actual streaming time with buffer size \( B = S = 20 \) MB. We only include those streaming simulations that are successfully completed before the link breaks. When \( d_0 \) is small, most streaming processes take about the same amount of time to complete. As \( d_0 \) increases, there is a big variance of the actual streaming time. However, the mean streaming time and the predicted results do not differ too much. Hence most successful streaming processes still finish within a short period of time since it is very unlikely for a link to last for a long time. Our recursive algorithm provides similar results as the mean streaming time. Therefore, it captures the approximate time that a successful streaming process may take.

Note that the above presented methodology can also be applied to scenarios where a mobile node communicates with a stationary second node, which may in effect be a basestation. In this case one of the node’s locations would simply be fixed.

IV. IMPROVING STREAMING PROBABILITY THROUGH ADAPTIVE LAYER SELECTION

Because users are often more interested in the general content of a video sequence, delivery of all video frames is of greater importance than video quality in many situations.
To deliver more video frames, the streaming probability can be used as a criteria to determine the potential success of the strategy. Here we revise our previous streaming strategy utilizing layered video encoding schemes such as SVC and MDC. For layered encoding techniques, it is possible to lower the bitrate of the video stream by discarding several of the layers. Therefore, the probability to stream the video sequence can be increased by sacrificing the video quality for a certain number of frames. To achieve this, Figure 5 shows how the sender can calculate the number of layers to be discarded to adapt to the present buffer space limitations. As shown in the flowchart, the sender calculates the streaming probability based on the number of layers it chooses. If the streaming probability falls below a certain threshold, the sender can discard a number of layers to increase it. Since the host movement can change at any moment, the receiver needs to communicate with the sender regularly to update the distance and buffer space information. How the receiver monitors its current buffer space is beyond the scope of this paper. After getting the updated information, the sender can change the number of layers to be streamed by applying the above strategy again. Algorithm 2 illustrates this new streaming strategy.

Assume that the requested video sequence is composed of layers 0, . . . , L. Without loss of generality, layer 0 is selected as the base layer and is always streamed to the receiver. For layers 1, . . . , i, they are added in sequential order to the stream if Algorithm 2 finds that the streaming probability is larger than a certain threshold \( P_{\text{min}} \). In Algorithm 2, we regularly (each \( t_{\text{step}} \) seconds) monitor the current available buffer size \( B \) at the receiver side and the distance \( d \) between the two ad hoc peers. Constantly calculating the distance may drain the battery power of the mobile devices. In order to save energy, peers can calculate their distance from each other based on previous measurement and send update message to the each other only when their velocity changes. However, energy saving is beyond the scope of this paper. For each layer \( i \) to be added to the stream, we calculate the bitrate \( \text{rate}_i \) of the resulting stream and the remaining stream size \( S_i \). The streaming time \( t \) can thus be calculated from Equation 4 by substituting \( S \) with \( S_i \). After calculating \( t \), streaming probability \( P(\text{rate}_i) \), which is the probability to stream the lowest \( i \) layers of the whole video, can be calculated as \( L(d_0, t) \).

As shown in Algorithm 2, we do not stream layer \( i \) if the probability \( P(\text{rate}_i) \) to deliver a stream of size \( S_i \) is smaller than \( P_{\text{min}} \). Consequently, the algorithm tries to ensure that the receiver has a probability higher than \( P_{\text{min}} \) to receive the complete video stream. By reducing the number of extra layers to be streamed, we can increase the overall streaming probability. As a result, increasing the probability to deliver the video stream comes at the cost of degrading video quality for a number of frames.

Compared to our original approach, this new approach will produce a similar result. This is because in our previous approach, we need to first calculate the minimum buffer space. If the client does not have enough buffer space, the server will use lower video quality by discarding several layers. However, when client do not have enough buffer space, the streaming probability is also likely to be low. As a result, the server also needs to discard some layers to make the streaming probability higher than \( P_{\text{min}} \).

V. SIMULATION RESULTS

To analyze the benefits of our streaming strategy, we simulated it in the network simulator ns-2 with the settings shown in Table III and the radio model from Table I. For simplicity, we assume that the video sequence consists of two layers and is encoded with SVC. The base layer requires 64 kbps
and the enhancement layer is encoded at 256 kbps. Thus the stream size \( S \) is 4.7 MB for only the base layer or 23.4 MB for both layers. For each setting of \( B \), we conducted 10,000 independent simulations and calculated the average results. We have not included confidence intervals in the figures because we were able to ascertain that the individual results deviated by less than 5% from the mean values calculated. Figure 6 shows the probability to successfully deliver the video stream when using our streaming strategy compared with using only the base layer or both layers.

In Figure 6, when \( B \) is 1 MB, it takes more than 500 seconds to stream the video sequence even if the stream only consists of the base layer. Therefore, it is very improbable to deliver the whole video stream due to node mobility. When the buffer size increases, our strategy can achieve a comparable streaming probability to that of only delivering the base layer if \( P_{\text{min}} \) is large. We also experimentally applied our strategy with the random waypoint mobility model [30]. For most of the settings, both models yield similar simulation results. However, compared to the random waypoint mobility model, the random walk mobility model ensures a slightly higher chance to stream the video sequence. This is because in the random walk mobility model, a mobile node tends to move around its origin after the simulation starts. Therefore, the link can often last longer than with the random waypoint model. Our algorithm can reflect such differences by yielding a higher streaming probability for the random walk model. Based on the results of the two mobility models tested we conjecture that other models could also be used without fundamentally changing the outcome.

It is interesting to note that increasing the buffer size does not always increase the streaming probability. As shown in Figure 6, when \( B \) is larger than 15 MB, the actual streaming probability is decreasing if \( P_{\text{min}} = 0 \). This is because when our algorithm finds that there is enough buffer space for the remaining video stream, it will begin to stream both layers for the remaining video. However, this greatly increases the amount of data to be delivered, which also determines the streaming probability due to wireless bandwidth constraints and node mobility. When \( P_{\text{min}} \) is large, our algorithm is very conservative in such cases and only delivers the base layer if the predicted streaming probability is smaller than \( P_{\text{min}} \). In contrast, if \( P_{\text{min}} \) is small, both layers will be streamed as long as there is enough buffer space. Therefore, sufficient buffer space does not guarantee delivery of the whole video stream if \( P_{\text{min}} \) is small.

Figures 7 and 8 show the average percentage of the base layer and the enhancement layer that is streamed to the receiver in all the simulations. In Figure 7, the curve is very similar to that of Figure 6 because of the same reason we have discussed above. When the buffer size is sufficient, the sender is more aggressive and trying to stream both layers to the receiver if \( P_{\text{min}} = 0 \). On the contrary, when \( P_{\text{min}} = 0.9 \), the sender sends only the base layer to the receiver and is able to send more frames in a fixed amount of time. Therefore, the average percentage of the base layer received by the receiver is always higher when \( P_{\text{min}} = 0.9 \). The receiver can capture most of

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video sequence length</td>
<td>600 sec</td>
</tr>
<tr>
<td>Bitrate of the base layer</td>
<td>64 kbps</td>
</tr>
<tr>
<td>Bitrate of the enhancement layer</td>
<td>256 kbps</td>
</tr>
<tr>
<td>Maximum velocity</td>
<td>5 m/s</td>
</tr>
<tr>
<td>Mean epoch length of each client</td>
<td>30 sec</td>
</tr>
</tbody>
</table>
the video content in a slightly lower or equal resolution by increasing the $P_{min} = 0.9$ value. Figure 8 shows the average percentage of the enhancement layer that is streamed to the receiver. This metric roughly describes the quality of the video that is received by the recipient. Note that streaming only the base layer always results in zero percent of the enhancement layer to be streamed. If the buffer size is small, a very small portion of the enhancement layer will be streamed by using our streaming strategy. As $B$ increases, a higher percentage of the enhancement layer can be delivered as long as the streaming probability is larger than the threshold. Compared with the strategy of streaming both layers, tuning $P_{min}$ allows the our streaming strategy to make a tradeoff between the video quality and the streaming probability. When $P_{min}$ is 0, our streaming strategy exhibits the same performance as the strategy of streaming both layers. More high quality video frames will be delivered to the recipient. However, this lowers the streaming probability quality and it becomes less likely to stream the whole video sequence. When $P_{min}$ gets larger, more video frames can be streamed to the receiver as video quality drops. Therefore, tuning $P_{min}$ adjusts the performance between video quality and streaming probability.

When $P_{min}$ is large, although our streaming strategy ensures a high streaming probability, the percentage of the enhancement layer that can be streamed to the receiver is comparatively low. However, the video quality can be improved if we have more detailed information of node movements. This is because the streaming probability is calculated as the average probability considering that both nodes can move in any direction. Therefore, if we know the current relative speed between two mobile nodes, we can obtain a more accurate estimation of the buffer size.

In our prior work we demonstrated that link availability prediction is more accurate if the relative velocity can be provided [3]. Relative velocity information can be obtained through changes in received signal strength even if GPS is not available. As we discussed earlier, the streaming probability is determined by the link availability. Thus given an accurate estimation on relative velocity, our streaming strategy is able to achieve better video quality by making better decisions on whether to stream the enhancement layers. Figure 9 shows the simulation results when the relative velocity between two mobile ad hoc nodes can be accurately estimated by using the random waypoint mobility model. As shown in Figure 9, when $B = 25$ MB and $d_0 = 0$, our strategy can deliver 95% of the enhancement layer to the receiver with known relative velocity, compared to that of 82% in Figure 8 without any relative velocity information. However, the streaming probability is hardly changed as it is determined by the minimum stream size. Therefore, our streaming strategy can ensure the maximum achievable streaming probability along with higher video quality if more detailed velocity information is available.

When a receiver’s buffer space is insufficient, our streaming strategy always streams the video sequence in low quality at the beginning. For example, Figure 10 shows the average bitrate during each second for those successfully delivered video streams with random waypoint mobility model. The first several hundred seconds of the video are always of lower video quality. This allows the receiver to hold more video content with the same buffer space. By conserving more buffer space for later frames, our strategy will find it feasible to deliver the rest of the video frames in both layers at a certain point of time. Therefore, the video quality is always poorer during the initial period of the playback time. This may not be very desirable for some video sequences. For example, some video frames may require higher quality in order for the viewer to notice some details in it, while other frames have lower requirements on video quality.

In order to solve this problem, we can label some of the video frames with high priority. Prioritized frames or video segments can be determined by viewers or producers of the video, which is beyond the scope of this paper. When these video frames are encountered, the sender always streams all the layers to the receiver regardless of the prediction results from our link quality model. However, doing so may not be very fair for video frames in the later part of the video.
sequence. For example, if all earlier frames are marked with high priority, they might consume all the buffer space quickly and make it improbable to deliver the rest of the video frames before the link breaks. To deliver as many video frames as possible, our strategy could ignore the high priority flag if it finds that the base layers of the remaining video sequence cannot be successfully streamed.

Figure 11 shows the average bitrate when the first and the last 100 seconds of the video sequence are marked with high priority. Note that when \( B = 5 \text{ MB} \), these 200 seconds of the video sequence cannot all be streamed with both layers. Therefore, after streaming both layers of the first 50 seconds video sequence, our streaming strategy finds it impossible to stream the all prioritized frames in both layers. As a result, it lowers the video quality to stream only the base layer. Later on, after the buffered high quality video is consumed and more buffer space is released, the bitrate is increased again for the last 100 seconds of the playback time.

VI. IMPROVING VIDEO QUALITY THROUGH LAYER RETRANSMISSION

When the simulation area is small, it is possible that two disconnected peers will quickly get reconnected. Therefore, the receiver may not have consumed all its buffered content yet when the link recovers. We can thus “repair” the buffered video stream by delivering all the missing data.

For traditional video streaming applications, it is only possible to resume streaming the video sequence from where it is left off after the connection is recovered. On the other hand, our streaming approach makes it possible to transmit the missing layers even if all the base layers have been delivered. Since the receiver is continuously consuming its buffered data, the sender only needs to stream all the extra layers of those video frames that are still in the buffer. According to the data that has been delivered before the link recovers, they can be classified into the following three categories:

- Some frames of the original video sequence (both base layer and extra layers) are still undelivered. In this case, the receiver cannot play back the entire video. Since part of the video content is missing, we need to continue streaming the remaining video frames when connection is reestablished. After all the frames have been delivered, the situation becomes the same as the second case.
- The base layer of all the frames has been successfully delivered while some frames in the buffer have their extra layers missing. In this case, we can selectively transmit the missing layers to the receiver.
- All frames of the original video sequence, with their base and extra layers, have been successfully streamed. In this case, there is nothing we need to do.

Because the first case requires us to resume streaming from where it has left off, Algorithm 2 can successfully achieve this. However, for the second case, the situation is more complicated. First, the receiver needs to keep track of the missing layers to avoid wasting the buffer space and wireless bandwidth. Because for each \( \ell_{\text{step}} \) seconds that the sender uses the same bitrate to transmit the video stream, continuous frames are of the same video quality. The receiver can use such information to keep track of how many layers are streamed for each frame. It does not need to keep track of all the frames that have already been consumed. Only the missing layers for the buffered video frames should be requested by the receiver. Second, the missing extra layers need to be synchronized with their corresponding base layers before they are passed to the decoder. Because we focus on the streaming process, we hereby assume that decoders can be modified to accomplish both of these tasks, either with the help of sequence numbers or other mechanisms.

To show the benefit of this re-delivery scheme, we simulated our approach with the same settings shown in Table III. As long as the receiver has not consumed all its buffered video, it can ask the sender to resume streaming the unfinished content or transmit the missing enhancement layers if all base layers are delivered. In our experiments the simulation area is set to \( 1,000 \times 1,000 \text{ m}^2 \). One node is initially located at the center of the simulation area while the other node is \( d_0 \) distance away.
layer selection strategies.

Fig. 12. Comparison of streaming probability with different enhancement layer selection strategies.

Figures 13 and 14 show the average percentage of the base layer and enhancement layer that are streamed to the receiver, respectively. Similar to Figure 12, with the new strategy a higher percentage of the base layer can be received since the sender can resume streaming the video content. Note that in Figure 14, \( P_{\text{min}} \) does not have any impact on the average percentage of the enhancement layer that is received by each receiver. However, this does not contradict our results in Figure 8. Because our new strategy can deliver missing enhancement layers after reconnection, the percentage of the enhancement layer that can be delivered is largely determined by the amount of time that two nodes are connected with each other. Therefore, a smaller \( P_{\text{min}} \) does not guarantee that a higher percentage of the enhancement layer will be streamed. Combining the results of Figures 12, 13 and 14, we conclude that it is better to use a larger \( P_{\text{min}} \) value so that we enjoy a higher streaming probability along with better video quality.

Figure 15 shows the average bitrate of the video stream during each second for those successfully delivered video streams with the random waypoint mobility model. Compared to Figure 10, the video quality is not always higher with our new approach. For example, in Figure 10(a), the average bitrate is always 320 kbps when \( d_0 = 200 \) m and \( B = 25 \) MB, which is higher than the rate in Figure 15. This is because more nodes are able to receive the whole video stream in poorer quality as the sender can resume the streaming process. In Figure 10(a), only those receivers that are connected with their senders for a long time can successfully stream the whole video sequence. Because there is sufficient buffer space, all the layers will be streamed as long as the connection lasts long enough. However, in Figure 15, the sender may only transmit the base layer to the receiver if it finds that the link cannot last for a long time. Although the streaming process may still successfully complete if both nodes are reconnected after a period of time, the average video quality is lowered as the receiver has already consumed some of the frames that are previously delivered.

Figure 12 shows the probability to stream the entire video sequence. Compared to Figure 6, the new strategy has a 10% higher streaming probability since it can resume streaming the video sequence from where it has left off.

Fig. 12. Comparison of streaming probability with different enhancement layer selection strategies.

Fig. 13. Average percentage of base layer streamed.

Fig. 14. Average percentage of enhancement layer streamed.
Incorporating audio into our streaming strategy would be very useful to modify existing decoders to support the layer re-delivery shows an additional 10% increase in streaming probability.

In this paper, we first introduce an algorithm for calculating the probability of streaming a video sequence in mobile ad hoc environments. Compared to previous techniques, our work does not rely on simplifying assumptions such as uniform streaming bandwidth, fixed node locations, or infinite buffer size. Based on this more realistic mathematical model, we introduce a new streaming strategy by utilizing the SVC scheme. Simulation results show that our strategy can improve the streaming probability by a maximum of 60% without sacrificing much video quality. Further improvements through layer re-delivery shows an additional 10% increase in streaming probability.

Our work can be extended in a number of directions. First, this paper mainly focuses on the streaming process. It would be useful to modify existing decoders to support the layer re-delivery scheme after a connection is re-established. Second, our current streaming strategy only works for video sequences. Incorporating audio into our streaming strategy would be very useful. For example, the MPEG-4 Scalable Audio Coding can be used to change the bitrate of audio streams. For mobile ad hoc networks, constraints such as battery power are not addressed in this paper. We will extend our current research to further analyze these issues.

ACKNOWLEDGMENTS

The authors thank...


Min Qin received the B.E. and M.E. degrees in computer science and engineering from Shanghai Jiaotong University, Shanghai, China, in 1999 and 2002, respectively. He received the Ph.D. degree in computer science from the University of Southern California (USC), Los Angeles, USA, in 2007. He is presently working at Google Inc., CA. His research interests include video streaming and multimedia systems.

Roger Zimmermann (S’93–M’99–SM’07) received the Informatik Ingenieur HTL degree from the Höhere Technische Lehranstalt in Brugg-Windisch, Aargau, Switzerland, in 1986. He received the M.S. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, USA, in 1994 and 1998, respectively. He is currently an Associate Professor with the Department of Computer Science at the National University of Singapore (NUS) where he is also an investigator with the Interactive and Digital Media Institute (IDMI). Prior to joining NUS he held the position of Research Area Director with the Integrated Media Systems Center (IMSC) at the University of Southern California (USC).

His research interests are in the areas of distributed and peer-to-peer systems, collaborative environments, streaming media architectures, georeferenced video search, and mobile location-based services. He has co-authored a book, four patents and more than a hundred conference publications, journal articles and book chapters in the areas of multimedia and information management. He is an Associate Editor of the ACM Computers in Entertainment magazine and the ACM Transactions on Multimedia Computing, Communications and Applications journal. He is a member of ACM.