Improving Mobile Ad-hoc Streaming Performance through Adaptive Layer Selection With Scalable Video Coding

Min Qin
Department of Computer Science
University of Southern California
Los Angeles, CA 90089
mqin@usc.edu

Roger Zimmermann
School of Computing
National University of Singapore
Singapore 117543
rogerz@comp.nus.edu.sg

ABSTRACT
Numerous types of mobile devices are now popular with end users, who increasingly use them to carry multimedia content on the go. As wireless connectivity is integrated into many handheld devices, streaming multimedia content among mobile ad-hoc peers is becoming a popular application. In this paper, we first introduce a mathematical model for calculating the probability of successfully streaming a multimedia object between two mobile ad-hoc peers. Unlike previous techniques that assume a constant wireless bandwidth or fixed node location, our work supports the 802.11 Auto-Rate Fallback scheme along with two popular mobility models: the random waypoint and the random walk mobility model. When delivery of the whole video content is of crucial importance, we introduce a novel streaming strategy to improve the probability of successfully streaming a video sequence based on our proposed mathematical model. This strategy takes advantage of the Scalable Video Coding scheme to adaptively select the number of enhancement layers to be streamed to the receiver. Simulation results show that our strategy can improve the probability to stream a media object by a maximum of 60% while keeping the video quality relatively high.

Categories and Subject Descriptors
C.2.4 [Computer and Communication Networks]: Distributed Systems—distributed applications; C.4 [Performance of Systems]: Reliability, availability, and serviceability, modeling techniques

General Terms
Algorithms, Design, Reliability

Keywords
Link availability, Streaming, Mobile ad-hoc networks, Mobility models, Scalable video coding

1. INTRODUCTION
With the widespread use of SmartPhones and other handhelds, streaming audio or video content between mobile ad-hoc devices is becoming a popular application. For example, patrons visiting a museum can watch the video clips taken by others during their tour to get better learning experiences. Exemplar systems that implement such ideas include Microsoft Zune and MStream [11]. Most of the recent wireless handheld devices can operate via wireless 802.11 networks, which provide broadband-level bandwidth and a communication range of hundreds of meters. This allows a user to move freely when he is streaming multimedia content from others that are within his communication radius. Therefore, media streaming among mobile ad-hoc peers is feasible and enables many new applications.

One challenge in streaming multimedia content among mobile ad-hoc peers is to deliver the content, usually large in size, over a wireless link whose quality is constantly changing. For example, the wireless bandwidth may drop or the link may even break as the distance between two mobile ad-hoc peers increases. Therefore, the probability to successfully stream a multimedia object before a link breaks is of special interest to mobile users. It can help users evaluate the smoothness of the streaming process and improve mobile streaming experiences.

Because of the high complexity involved in modeling a realistic wireless environment, many previous studies assume a constant network bandwidth in analyzing the streaming performance between two mobile ad-hoc peers. Additionally, most existing techniques are concerned with link or path duration when the network has reached its stationary regime [14]. They often do not utilize the present node information in predicting future link status. In this paper, we introduce a mathematical model for calculating the probability of successfully streaming a multimedia object in the foreseeable future. Compared with other studies, our work supports the 802.11 Auto-Rate Fallback (ARF) [6] scheme along with two popular mobility models: the random waypoint [19] and the random walk mobility model [9]. It does not assume the presence of GPS equipment, since GPS is not always available and it may not work indoors.

When delivery of all video frames is of the highest priority, this paper introduces a new mobile streaming strategy based on our proposed mathematical model. For example, users are often more interested in the general content of a video sequence rather than its picture quality. In such cases, our strategy can deliver more frames to the user before a link is broken while keeping the video quality rela-
tively high. It takes advantage of the multi-layer structure of the Scalable Video Coding (SVC) [12] scheme. SVC has been widely adopted by many video encoding methods, including the latest extension to the popular H.264/MPEG-4 AVC [3] codec. The method divides a video sequence into a base layer and several enhancement layers. The base layer has a lower bitrate compared to the original stream. It 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. By regularly monitoring the available buffer size at the receiver side, our streaming strategy adaptively selects the number of enhancement layers to be streamed to ensure delivery of the entire video stream. Therefore, the goal of this strategy is to allow users to watch the entire video by minimally sacrificing the video quality. An alternative to SVC is the Multiple Description Coding (MDC) [17]. MDC divides the original video stream into several correlated bitstreams. Unlike SVC, each stream in MDC can be decoded independently. Because our work only utilizes the multi-layer feature of SVC, it can be easily extended to work with MDC.

The rest of paper is organized as follows. In Section 2, related work is introduced. The mathematical modeling of the probability for successfully streaming a video sequence is presented in Section 3. We describe in detail how to calculate the streaming probability by using both the random walk and the random waypoint mobility models. Section 4 describes a novel streaming strategy to improve streaming probabilities in mobile ad-hoc environments and Section 5 presents the performance of our scheme. Finally, we summarize our contributions and provide suggestions for further research in Section 6.

2. RELATED WORKS

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 [13] with good accuracy [7]. The two-ray ground model can be expressed by the following equation:

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

(1)

Here \( p_r \) is the transmitted signal power. \( G_t \) and \( G_r \) are the antenna gains of the transmitter and the receiver, and \( h_t \) and \( h_r \) are the heights of the transmit and receive antennas, respectively. 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 several link availability models [9, 8, 11] for calculating the probability that two mobile ad-hoc nodes are connected over time. In our previous work [11, 10], we introduced a model to calculate the probability \( L(t) \) that a link between two mobile ad-hoc peers can continuously last for \( t \) seconds. Compared to other approaches, this model is more suitable for streaming applications due to its continuous nature. However, our prior study 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 memory size, which is often no more than, for example, 32 MB. Also, wireless bandwidth may vary as the distance between the two peers changes. In our new design we address these real world constraints.

For IEEE 802.11 protocols, ARF is used to adapt to changes in link quality. When consecutive transmission errors occur, the sender will step down its transmission rate. And conversely, if there are no errors in a short period of time, the transmission rate will be increased. Most 802.11 protocols support multiple modulation techniques and receivers have varying sensitivities when operating with these modulations. For example, Table 1 shows the modulation techniques and receiver sensitivities for the Orinoco 802.11b wireless card [4]. Consequently, 802.11 protocols have different communication radii when operating with different maximum bandwidth. As shown in [2], the goodput of 802.11 protocols drops suddenly as the distance between two ad-hoc nodes increases above a certain threshold. 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 1 shows the relative zone distribution of the 802.11b protocol with settings of Table 1. Many studies (e.g., [18]) 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. To the best of our knowledge, there has been no study on combining these two subjects.

3. MODELING STREAMING PROBABILITY FOR 802.11 NETWORKS

Table 2 lists all the terms that are used in this study. For simplicity, we assume that the requested media stream is encoded at a constant bitrate (CBR) \( br \) and satisfies \( br < bu_M \). Since most media streams of handheld devices use maximum

\[ \text{Table 1: Orinoco 802.11b wireless card specification} \]

<table>
<thead>
<tr>
<th>Modulation scheme</th>
<th>BPSK</th>
<th>QPSK</th>
<th>CCK5.5</th>
<th>CCK11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theoretical data rate(Mbps)</td>
<td>1</td>
<td>2</td>
<td>5.5</td>
<td>11</td>
</tr>
<tr>
<td>Receiver sensitivity(dbm)</td>
<td>-94</td>
<td>-91</td>
<td>-87</td>
<td>-82</td>
</tr>
<tr>
<td>Transmission power(dbm)</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Bandwidth zone distribution of 802.11b with Auto-Rate Fallback
bitrates of less than 400kbps (e.g. DivX 5), it can be supported by the lowest bit rate modulation scheme (1Mbps) of the 802.11b protocols. In Table 2, \( bw_i \) is the average bandwidth of the ith zone. It is lower than the maximum nominal bandwidth of that zone. This is because 802.11 protocols use collision avoidance schemes such as CTS/RTS, which significantly reduce channel utilization. For example, when operating with CCK11 modulation, the actual usable bandwidth for UDP is around 6Mbps rather than the theoretical 11Mbps. Additionally, bandwidth may be even lowered due to the presence of obstacles and interference. Therefore, we assume 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 initial distance between them is approximated by \( d_0 \) (\( d_0 \leq R_M \)). An accurate estimation of \( d_0 \) can be obtained through signal strength [15], acoustic sensing [16] 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.

Because both nodes of a link will keep their movement for a certain amount of time after the link breaks, it is very unlikely for two nodes to get reconnected before the connection times out. In order to guarantee seamless delivery of a multimedia object, 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 ith 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 \( \sum_{i=1}^{M} t_i bw_i \). Because \( t_i \) is generally very small, the amount of data that can be consumed by the receiver during time period \( t \) is determined by \( br \times (t - t_i) \). Therefore, the receiver buffer is saturated if \( \sum_{i=1}^{M} (bw_i - br) t_i + br \times t_i > B \). Since the maximum amount of data that can be buffered is \( \min(B, \sum_{i=1}^{M} (bw_i - br) t_i + br \times t_i) \), the total amount of data streamed to the receiver is given by \( \min(B + br \times (t - t_i), \sum_{i=1}^{M} bw_i t_i) \). To guarantee delivery of the multimedia object before the link is broken, the following inequation has to be satisfied:

\[
\min(B + br \times (t - t_i), \sum_{i=1}^{M} bw_i t_i) \geq S \tag{2}
\]

According to the above inequation, when the buffer size is adequate, the probability to stream a multimedia object is largely determined by the time period \( t_i \) that node \( n \) spends in each zone of \( m \). On the other hand, a small buffer size results in a longer streaming time \( t \), which largely lowers the streaming probability due to node mobility. To calculate \( t_i \), we divide our discussion to consider two commonly used mobility models: the Random Waypoint Mobility Model and the Random Walk Mobility Model.

### 3.1 Random Waypoint Mobility Model

The random waypoint model is one of the most commonly used models for mobile ad-hoc network simulations. In this model, a node is initialized with a certain starting position. Then it randomly selects a destination on the map and moves towards that destination with a speed randomly chosen between \([0, v_{max}]\). After reaching the destination, the node selects a new random destination and continues the above process.

Let \( v \) and \( \theta \) denote the relative speed and direction of node \( n \) with respect to node \( m \). When the simulation area is large, the distance between two consecutive destinations is usually very long, assuming uniform distribution. As a result, neither node changes its velocity before the link between them breaks. Therefore, we can assume that \( v \) and \( \theta \) do not change during the streaming process. As shown in Figure 2, the distance node \( n \) can travel within a circle of radius \( R_i \) is given by

\[
r_i(\theta) = \begin{cases} 
\sqrt{R_i^2 - d_0^2\sin^2\theta} - d_0 \cos \theta & d_0 \leq R_i \ 
2\sqrt{R_i^2 - d_0^2\sin^2\theta} & d_0 > R_i, \cos \theta \leq \tau 
0 & \text{otherwise}
\end{cases}
\]

where \( \tau \) is defined as \( \tau = -\sqrt{1 - \frac{R_i^2}{d_0^2}} \). Since the relative
velocity does not change, the time \( t_i \) node \( n \) spends in zone \( i \) is given by:

\[
t_i = \frac{r_i(\theta) - r_{i-1}(\theta)}{v}
\]  

(4)

To simplify our discussion for the case when \( i = 1 \), we define \( r_0(\theta) = 0 \) for all theta values. Combining Equations 2 and 4, the relative speed \( v \) has to satisfy the following condition in order for the streaming process to finish successfully:

\[
v \leq \min \left( \frac{r_M(\theta)BR}{S - B} \cdot \frac{\sum_{i=1}^{M} bw_i(r_i(\theta) - r_{i-1}(\theta))}{S} \right)
\]

(5)

Define

\[
v(\theta) = \min \left( \frac{r_M(\theta)BR}{S - B} \cdot \frac{\sum_{i=1}^{M} bw_i(r_i(\theta) - r_{i-1}(\theta))}{S} \right)
\]

(6)

The probability to successfully stream the media object before the link breaks is thus given by \( \int_{0}^{2\pi} \text{prob}(v \leq v(\theta))d\theta \). By the law of cosine, we have

\[
\text{prob}(v \leq v(\theta)) = \frac{1}{\pi} \cos \left( \frac{\pi}{2} \frac{v^2 + v^2_{\text{max}} - v^2(\theta)}{2v_{\text{max}}^2} \right)
\]

(7)

where the \( \cos \) function is defined as

\[
\cos(x) = \begin{cases} 
0 & x > 1 \\
\cos^{-1}(x) & -1 \leq x \leq 1 \\
x & x < -1
\end{cases}
\]

(8)

Figure 3 shows the streaming probability when using the wireless card model from Table 1 and both nodes are moving in accordance with the random waypoint mobility model. The transmission loss is set to 10 dbm. In Figure 3(a), we can see that the buffer size has an immense impact on the streaming probability. When the buffer is small, the probability to successfully stream a multimedia object drops significantly if the size of the media object becomes larger. Our first observation is that maintaining a large buffer size is crucial for mobile ad-hoc streaming applications.

Contrary to our previous outcome in [11], results in Figure 3(b) show that the initial distance does have a big impact on the streaming probability even if the buffer size is adequate. This is because our previous work assumes a uniform transmission rate. Therefore, the streaming time is fixed and the probability does not vary too much since both nodes can either move towards or away from each other. When 802.11 ARF is considered, node \( n \) may be located in different zones of node \( m \) if the initial distance changes. Consequently, the stream is transmitted at a different rate and the time required to stream the media object can be greatly affected.

### 3.2 Random Walk Mobility Model

The random walk mobility model was introduced and studied in [9]. Compared with the random waypoint model, the random walk mobility model does not suffer from speed decay and non-uniform node distribution [14]. Let \( n \) and \( m \) denote two mobile nodes of a given ad-hoc link. According to [9], the random walk mobility model 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 with mean \( \lambda^{-1} \). During each mobility epoch, a node moves with a constant speed and direction. The speed is a random variable uniformly distributed between \([0, v_{\text{max}}]\) and the direction is uniformly distributed over \([0, 2\pi]\). Given a larger mean epoch length, a node is more likely to go straight for a long period of time. \( v_{\text{max}} \) and the mean epoch length can be obtained from previous statistics, which is beyond the scope of this paper.

The random walk mobility model is very hard 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 \( bw_1 \). Let the mobility vector \( \vec{R}_i(t) \) represent the distance and direction of node \( i \)'s position at time \( t_0 + t \) relative to its position at time \( t_0 \). According to [9], the magnitude of the relative mobility vector \( R_{mn}(t) = R_n(t) - R_m(t) \) is approximately Raleigh distributed with a phase uniformly distributed between \([0, 2\pi]\). We have

\[
P(\{R_{mn}(t) < r\}) = 1 - e^{-\frac{r^2}{\lambda m + \lambda n}}(9)
\]

where \( \lambda_i = \frac{2\lambda}{\lambda_i^2 + \mu_i^2} \). Here \( \lambda_i^{-1} \) is the mean epoch length of node \( i \). \( \delta_i \) and \( \mu_i^2 \) are the mean and variance of node \( i \)'s speed. Therefore, given that the magnitude of \( R_{mn}(t) \) is \( r \) at time \( t \), the probability that node \( n \) is located within zone \( 1...i \) after \( t \) seconds is given by

\[
P'(d(t) \leq R_i | R_{mn}(t) = r) = \int_{0}^{R_i} f(r, t) dr
\]

(10)

From Equation 9, the probability density of \( r \) is given by

\[
f(r, t) = \frac{2r}{\alpha_m + \alpha_n} e^{-\frac{r^2}{\alpha_m + \alpha_n}}
\]

(11)

According to [11], 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 \( 1...i \) after \( t \) seconds can be approximated by

\[
P(d(t) \leq R_i | d(t) \leq R_M) = \frac{P(d(t) \leq R_i)}{P(d(t) \leq R_M)} (12)
\]

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_i(t) \leq
The expected maximum bandwidth after $t$ seconds is thus given by

$$bw(t) = \sum_{i=1}^{M} bw_i P_i(t)$$ (14)

**Lemma 3.1.** Given that $d_0 < R_1$, $bw(t)$ is a monotonic decreasing function.

**Proof.** According to the Rayleigh distribution, the probability density of $|R_{ma}(t)|$ as a function of $r$ and $\theta$ is given by

$$f(r, \theta, t) = \frac{r}{\pi(\alpha_n + \alpha_n)} e^{-\frac{r^2}{\alpha_n + \alpha_n}}$$ (15)

Given $r_1$ and $r_2$ ($r_1 < r_2$), define

$$G(t) = \frac{f(r_1, \theta, t)}{f(r_2, \theta, t)} = \frac{r_1^2 r_2^2}{r_2^2 - r_1^2} e^{-\frac{r_1^2}{\alpha_n + \alpha_n}}$$ (16)

as a function of $t$, we have

$$G'(t) = -\frac{r_1 r_2^2}{r^2 (\alpha_n + \alpha_n)} \left(2 e^{-\frac{r^2}{\alpha_n + \alpha_n}} - e^{-\frac{r_1^2}{\alpha_n + \alpha_n}}\right) < 0$$ (17)

Therefore, function $G(t)$ is a monotonic decreasing function.

For a given $\theta$, node $n$ is more likely to be distributed in areas where $r$ is larger as $t$ increases. Because node distribution is similar whether the link is continuously available or not [11], it is more probable for node $n$ to be within the outer zones of $m$ as $t$ increases. Therefore, $bw(t)$ is a monotonic decreasing function of $t$. When $t$ is infinitely large, $G(t)$ is equal to $\frac{1}{\lambda_n}$ by Equation 16. As a result, node $n$ is uniformly distributed over the simulation run a long time. This corresponds to the result from [1, 14] when the stationary regime of the random walk model has been reached.

Figure 4 shows the expected wireless bandwidth over time from Equation 14. Here we use the same wireless model as shown in Figure 4(a), and the actual bandwidth of each zone is about half of the maximum bandwidth of that zone. As shown in Figure 4(a), the expected bandwidth is constantly dropping if node $n$ is initially located within zone 1 of node $m$. However, when the initial distance $d_0 > R_1$, the expected bandwidth is no longer a monotonic decreasing function. In Figure 4(b), for example, node $n$ is initially located within zone 2. When both nodes start to move, it is possible for node $n$ to move into zone 1 after $t$ seconds. As a result, the expected bandwidth may increase temporarily since zone 1 has a higher bandwidth than other zones. Therefore, the expected bandwidth may fluctuate during the initial period of the simulation depending on the bandwidth distribution of each zone and the probability that node $n$ is located within a specific zone.

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

$$t_i = t \int_{s=0}^{t} P_i(s)ds$$ (18)

Combining Equations 2 and 18, we have

$$\min(B + br \times (t - t_i), t) \times \sum_{i=1}^{M} bw_i \int_{s=0}^{t} P_i(s)ds \geq S$$ (19)

Figure 4: expected bandwidth over time, $\lambda_1^{-1} = \lambda_2^{-1} = 30$ sec

The first half of the Equation 19 can be easily solved to get an upperbound of $t$. However, determining the upperbound of $t$ from the second half of Equation 19 is difficult. Here we introduce a novel recursive algorithm to approximate the upperbound of $t$. 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 from [11] 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 18. Based on such an integration, we can calculate 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 procedures of this recursive algorithm.

**Algorithm 1** Recursive algorithm for finding the ad-hoc streaming time

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

**Theorem 3.2.** If the expected bandwidth is monotonic decreasing over time, our recursive algorithm converges to the only value $t_0$ that satisfies

$$\sum_{i=1}^{M} bw_i \int_{s=0}^{t_0} P_i(s)ds = S$$ (20)
Because the function
\[ f(t) = \sum_{i=0}^{M} bw_i \int_{s=0}^{t} P_i(s) ds \] (21)
is a monotonic increasing function of \( t \), there exists only one value \( t_0 \) that satisfies \( f(t_0) = S \). Note that \( t' \)'s initial value \( S_{bw_M} < t_0 \). Without loss of generality, assume the current value of \( t = t_0 + \Delta t \) (\( \Delta t \geq 0 \)) is passed to our recursive algorithm. We have
\[
t_{\text{new}} = \frac{S(t_0 + \Delta t)}{\sum_{j=1}^{M} bw_j \int_{s=0}^{t_0+\Delta t} P_j(s) ds} = \frac{S(t_0 + \Delta t)}{S + \sum_{j=1}^{M} bw_j \int_{s=0}^{t_0+\Delta t} P_j(s) ds} (22)
\]
Since the expected bandwidth is decreasing over time, we have
\[
\sum_{j=1}^{M} bw_j \int_{s=0}^{t_0+\Delta t} P_j(s) ds \\
\leq \frac{\Delta t}{t_0} \sum_{j=1}^{M} bw_j \int_{s=0}^{t_0} P_j(s) ds = \frac{S\Delta t}{t_0} (23)
\]
Consequently,
\[
t_{\text{new}} \geq \frac{S(t_0 + \Delta t)}{S + S\Delta t/t_0} = t_0 (24)
\]
According to the definition of \( P_j(s) ds \), we have
\[
\sum_{j=1}^{M} \int_{s=t_0}^{t_0+\Delta t} P_j(s) ds = \Delta t (25)
\]
From Equations 22 and 25, we have
\[
t_{\text{new}} \leq \frac{S(t_0 + \Delta t)}{S + bw_M \Delta t} (26)
\]
So the difference between \( t_{\text{new}} \) and \( t_0 \) satisfies
\[
\left| \frac{t_{\text{new}} - t_0}{\Delta t} \right| = \frac{S - t_0bw_M}{S + bw_M \Delta t} < \frac{S - t_0bw_M}{S} < 1 (27)
\]
Therefore, after each iteration, \( t_{\text{new}} \) is getting closer to \( t_0 \) and the rate of convergence is smaller than \( \frac{S - t_0bw_M}{S} \). So our recursive algorithm converges to \( t_0 \).

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 shown in Figure 5. However, when the initial distance \( d_0 > R_1 \), the above recursive algorithm is not guaranteed to converge. This is because the expected bandwidth is not monotonically decreasing over time and Inequation 23 may not be satisfied. It is possible that the \( t \) values fluctuate around \( t_0 \). In order to solve such situations, when our iterative algorithm returns a \( t_{\text{new}} < t_0 \), we calculate the new \( t \) value as \( \frac{t_{\text{new}}}{1+\epsilon} \) and then compare \( \sum_{i=0}^{M} bw_i \int_{s=0}^{t_0} P_i(s) ds \) with \( S \). If the new \( t \) value is larger than \( t_0 \), we keep this \( t \) value and start the recursive algorithm again. Otherwise, we keep on updating \( t \) by using the above linear interpolation method. However, this situation rarely happens as we experimented with different \( d_0 \) values that are larger than \( R_1 \).

Figure 5 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 by using ns-2 [5]. The result of ns-2 are shown in lines marked with a cross. We use the same wireless card model from Table 1.

Similar to that of Figure 3, buffer size has a huge impact on mobile streaming performances when \( S \) is large. In Figure 5, when \( d_0 \) 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 \( d = 200m \), 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 6 shows the confidence interval of the actual streaming time when \( B = 20MB \). 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.

4. IMPROVING VIDEO STREAMING PROBABILITY FOR MOBILE DEVICES

4.1 Calculation of Minimum Buffer Size

From Figures 3 and 5, we observe that the buffer size has a very significant impact on the probability to successfully stream a media object when \( S \) is large. If \( B \) is small, the buffer saturates quickly and a large portion of the wireless bandwidth is wasted afterwards. It takes a long time for the multimedia content 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 multimedia object if the link can last long enough. Therefore, we observe that determining the minimum buffer size is very important for improving mobile streaming applications.

From Equation 2, the minimum buffer size can be determined by solving the equation

\[ B_{\text{min}} + b r \sum_{i=1}^{M} t_i = \sum_{i=1}^{M} b w_i t_i = S \]  

(28)

For the random walk mobility model, from Equation 19, we can first calculate the streaming time \( \sum_{i=1}^{M} t_i \) by assuming the buffer size \( B \) is infinite. The minimum buffer size is then determined by \( S - b r \times (\sum_{i=1}^{M} t_i - t_1) \). Note that the expected time for nodes \( n \) and \( m \) to be continuously connected before the link breaks is given by

\[ t_e = \int_{t=0}^{S/bw_{\text{max}}} t L(t) \, dt \]  

(29)

The minimum buffer size can thus be determined by

\[ B_{\text{min}} = S - b r \times (\max(t_e, \sum_{i=1}^{M} t_i) - t_1) \]  

(30)

The calculation of \( t_e \) is introduced in [10].

For the random waypoint mobility model, from Equation 5, different \( \theta \) values result in different \( B_{\text{min}} \) values. For a given \( \theta \), let \( t(\theta) \) denote the amount of time that both nodes must be continuously connected in order for the multimedia object to be successfully streamed. We have

\[ t_i = \frac{(r_i(\theta) - r_{i-1}(\theta)) t(\theta)}{r_{M}(\theta)} \]

From Equation 2, we have

\[ t(\theta) = \frac{r_M(\theta) S}{\sum_{i=1}^{M} (r_i(\theta) - r_{i-1}(\theta)) b w_i} \]  

(31)

Since the maximum relative speed is smaller than \( v_{1\text{max}} + v_{2\text{max}} \), for a given \( \theta \), \( B_{\text{min}}(\theta) \) satisfies

\[ B_{\text{min}}(\theta) = S - b r \times (\max(\frac{r_M(\theta)}{v_{1\text{max}} + v_{2\text{max}}}, t(\theta)) - t_1) \]  

(32)

Therefore, if \( \theta \) is known, we can use Equation 32 to calculate the minimum buffer size. When \( \theta \) is unknown, we can calculate the average \( B_{\text{min}} \) value by

\[ B_{\text{min}} = \frac{1}{2\pi} \int_{\theta=0}^{2\pi} B_{\text{min}}(\theta) \, d\theta \]  

(33)

Figure 7 shows the distribution of \( B_{\text{min}} \) as a function of \( S \). As indicated in Figure 7, different mobility models do not have much impact on buffer size requirements. Also, \( B_{\text{min}} \) increases linearly with \( S \). According to Equation 28, the difference between \( B_{\text{min}} \) and \( S \) is mostly affected by the stream rate \( b r \). A smaller \( b r \) value results in a higher buffer size requirement for the multimedia object to be successfully streamed.

4.2 Improving Streaming Probability through SVC

For most handheld devices, \( b r \) is often limited by their available physical memory. Therefore, given a video sequence of size \( S \), we can determine whether there is enough buffer space based on our previous study of \( B_{\text{min}} \). Note that when \( B < B_{\text{min}} \), \( S \), \( b r \), and \( B \) all affect the streaming probability. On the other hand, when \( B > B_{\text{min}} \), only the stream size \( S \) affects the streaming probability. Therefore, for a fixed \( B \) value, we can either reduce the video stream size \( S \) or the bitrate \( b r \) to improve the probability of successfully delivering the stream by carefully studying the value of \( B_{\text{min}} \).

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. To deliver more video frames, we introduce a new streaming strategy by incorporating SVC. For SVC, it is possible to lower the bitrate \( b r \) to improve the probability of successfully delivering the stream by discarding enhancement layers. Therefore, the streaming probability can be increased by sacrificing the video quality for a certain number of video frames. To achieve this, we can calculate the number of enhancement layers to be discarded to adapt to present buffer space limitations. Algorithm 2 shows this new streaming strategy.

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. For each enhancement layer \( i \) to be added to the stream, we calculate the bitrate \( r_{i}(\theta) \) of the stream and the remaining stream size \( S_{i} \). Note that the current buffered data at the receiver may have a different average consumption rate than \( r_{c}(\theta) \). Therefore, we need to adapt to such inconsistency by calculating the average consumption rate \( a b r_{i} \) in the next \( t \) seconds, where \( t \) is the maximum of the expected remaining link duration or remaining streaming time.
Assuming the currently buffered video sequence requires $t_b$ seconds to play back. If $t_b > t$, then $abr_i$ is the average bit rate of the first $t$ seconds of the buffered video. Otherwise, data consumed in these $t$ seconds include all the video frames that are currently buffered and the first $t - t_b$ seconds video of bitrate $rate_i$. Thus $abr_i$ can be calculated as

$$abr_i = \frac{(t - t_b) \times rate_i + (B_{\text{max}} - B)}{t}$$

(34)

where $B_{\text{max}}$ is the maximum buffer size at the receiver. For each resulting $S_i$ and $abr_i$ pair, $B_{i,\text{min}}$ can be calculated from Equation 30 or 32 depending on the mobility models. If $B$ is smaller than all $B_{i,\text{min}}$, it is unlikely for the sender to deliver the whole video stream before the link breaks if she streams any enhancement layers. On the other hand, if the available buffer size $B$ is larger than some $B_{i,\text{min}}$, there are two possible cases:

- If $t_i$ or $\frac{R_{\text{avg}}(t)}{\gamma_{\text{avg}}(t)}$ is smaller than the expected streaming time $\sum t_i$, streaming enhancement layer 1 to layer $i$ does not deliver the whole video stream, though the buffer will not saturate before the link is broken.

- If $t_i$ or $\frac{R_{\text{avg}}(t)}{\gamma_{\text{avg}}(t)}$ is larger than the expected streaming time $\sum t_i$, the whole video stream can be successfully delivered given that we only stream enhancement layer 1 to layer $i$.

Therefore, $B > B_{i,\text{min}}$ does not guarantee that the video sequence can be successfully streamed if we start streaming enhancement layer 1 to layer $i$. We need to check the remaining stream size $S_i$ to see if it can be streamed before the link breaks. As shown in Algorithm 2, when $B > B_{i,\text{min}}$, we do not stream enhancement layer $i$ if the probability $P_i$ 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 whole video stream. By reducing the number of enhancement 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. Unlike traditional SVC based streaming applications which use SVC to adapt to heterogeneous networks or ensure fault tolerance, our streaming strategy utilizes SVC to ensure delivery of the video sequence by adapting to node mobility and buffer size limitations at the receiver.

5. SIMULATION RESULTS

To analyze the benefits of our streaming strategy, we simulated it in ns-2 with the settings shown in Table 3 and the radio model from Table 1. For simplicity, the video sequence we used contains only one enhancement layer of 256 kbps. The stream size $S$ is 4.7MB for the base layer only or 23.4 MB for both layers. For each setting of $B$, we conduct 10000 independent simulations and calculate the average results. Figure 8 shows the probability to successfully deliver the video stream by using our streaming strategy compared with using only the base layer or both layers.

In Figure 8, 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

![Algorithm 2 Selective transmission of enhancement layers](image)

![Figure 8: Comparison of streaming probability with different enhancement layer selection strategies](image)
Table 3: Simulation settings

<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>64kbps</td>
</tr>
<tr>
<td>bitrate of the enhancement layer</td>
<td>256kbps</td>
</tr>
<tr>
<td>Maximum velocity</td>
<td>5m/s</td>
</tr>
<tr>
<td>Mean epoch length of each client</td>
<td>30 sec</td>
</tr>
</tbody>
</table>

Figure 9: Average percentage of base layer streamed

The buffer size increases, our strategy can achieve a comparable streaming probability to that of only delivering the base layer if $P_{min}$ is large. 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.

It is interesting to note that increasing the buffer size does not always increase the streaming probability. As shown in Figure 8, when $B$ is larger than 15 MB, the actual streaming probability is decreasing if $P_{min}$ is large. 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 $P_{min}$ is large, our algorithm is very conservative in such cases and only delivers the base layer if the predicted streaming probability is smaller than $P_{min}$. In contrast, if $P_{min}$ is small, then 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 $P_{min}$ is small.

Figure 9 shows the average percentage of the base layer that is streamed to the receiver for all the simulations. Similar to Figure 8, the performance of our streaming strategy is close to that of using the base layer only. Therefore, the receiver can capture most of the video content in a slightly lower or equal resolution.

Figure 10 shows the average percentage of the enhancement layer streamed. Figure 10 shows the average percentage of the enhancement layer that is streamed to the receiver. This metric describes the quality of the video that is received by the recipient. Note that streaming the base layer only always results in 0 percent of enhancement layer to be streamed. Compared with the strategy of streaming both layers, our streaming strategy makes a tradeoff between the video quality and the streaming probability. If the buffer size is small, a very small portion of the enhancement layer will be streamed. As $B$ increases, a higher percentage of the enhancement layer will be delivered. When $P_{min}$ is 0, our streaming strategy exhibits the same performance as the strategy of streaming both layers. Therefore, tuning $P_{min}$ adjusts the streaming performance for different scenarios. 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 movement. For example, in Equation 28, using $v_{1max} + v_{2max}$ as the upper bound of the current relative speed is very conservative. The probability for the relative speed to be equal to $v_{1max} + v_{2max}$ is very small. Therefore, if we know the current relative speed between two mobile nodes, we can obtain a more accurate estimation of the buffer size.

Figure 11 shows the simulation results when the relative velocity between two mobile ad-hoc nodes is accurately estimated by using the random waypoint mobility model. As shown in Figure 11, with known relative velocity information, our streaming strategy can still ensure the maximum achievable streaming probability along with higher...
video quality compared to that of Figure 10.

6. CONCLUSIONS

In this paper, we first introduced a mathematical model for calculating the probability of streaming a media object in mobile ad-hoc environments. Compared to previous techniques, our work does not assume uniform streaming bandwidth, or fixed node locations, or infinite buffer size. Based on this mathematical model, we introduced 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% with reasonably high video quality.

We plan to extend our algorithm in multiple directions. First, our current model does not work very accurately for the random walk mobility model. Instead of calculating an approximate streaming time, we plan to study its distribution to get better results. Second, since 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

This research has been funded in part by NSF grants EEC-529152 (IMSC ERC), CMS-0219463(TTR), IIS-0534761, NUS AcRF grant WBS R-252-050-101/133, and equipment gifts from Intel Corporation, Hewlett-Packard, Sun Microsystems and Raptor Networks Technology.

7. REFERENCES

http://ip.hhi.de/imagecom_G1/savce/index.htm