A Cross-layer Dropping Attack in Video Streaming over Ad Hoc Networks [*]

Min Shao[†], Sencun Zhu[†], Guohong Cao[†], Tom La Porta[†] and Prasant Mohapatra[‡]

[†] Department of Computer Science & Engineering
The Pennsylvania State University
Email: {mshao,szhu,gcao,tlp}@cse.psu.edu

[‡] Department of Computer Science
University of California at Davis
Email: prasant@cs.ucdavis.edu

ABSTRACT

Significant progress has been made to achieve video streaming over wireless ad hoc networks. However, there is not much work on providing security. Is existing security solution good enough for securing video streaming over ad hoc networks? In this paper, we discover a cross-layer dropping attack against video streaming. We first identify a general IP layer dropping attack and then reveal its destructive impact by leveraging the application layer information (e.g., video streaming). Through simulations, we quantify the impact of this attack as a function of several performance parameters such as delivery ratio, hop number and the number of attackers. The surprising result with this attack is that with a 94% delivery ratio, the receiver still cannot watch the video! We also propose several possible solutions to address the dropping attacks. Due to the unique characteristics of this attack, as long as malicious nodes exist, the network will suffer from this dropping attack.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Data communications*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Applications*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; K.6.5 [**Computing Milieux**]: Management of Computing and Information Systems—*Security and Protection*

General Terms

SECURITY, PERFORMANCE, ALGORITHMS

Keywords

Cross-layer, Dropping Attack, Security, Video Streaming, Ad Hoc Networks

1. INTRODUCTION

With popular web sites like Youtube, Yahoo, and many news servers, more and more people are used to watch video through the Internet. Recently, most of these web servers start to offer video services to people on the move. Due to the limitations of 3G wireless networks such as high cost, low bandwidth, researchers [18] propose to use a hybrid of ad hoc networks and 3G wireless networks, where the ad hoc network can provide much higher speed and is much cheaper. Further, ad hoc network is more flexible since it does not rely on the wireless infrastructure such as 3G, and can be used in many areas. For examples, soldiers can form an ad hoc network and share the real time video of the battlefield. Fire fighters can obtain the real time video during disaster recovery. Passengers in different vehicles on the road can play video games or share video clips through a vehicular ad hoc network [11].

There are many technical challenges for supporting video streaming over wireless ad hoc networks. Due to the mobility of the wireless nodes, the topology of the ad-hoc network may frequently change. In some cases, this topology change may break the established routing path between the source and destination, resulting in packet losses and reducing the quality of the video. Other issues such as high error rate of the wireless link can also increase the packet loss rate and reduce the video quality. In the past several years, many researchers proposed various solutions to address these problems and video streaming in ad hoc networks is becoming more and more practical ([22, 10, 17, 12]).

Although it is technically feasible to support video streaming in ad hoc networks, there are many security issues, especially for applications such as battlefield and disaster recovery. Security issues at the routing layer and medium access control layer have been well studied in ad hoc networks ([14, 13, 15, 8, 23, 7]), but we have not seen security attacks leveraging the characteristics of the application layer protocols. Indeed, an attacker may create much damage by exploiting the application layer knowledge.

Most video streaming is based on MPEG [4], which defines different packet formats such as I, P, B frames. To save bandwidth, the P and B frames are encoded based on the I-frame and thus they are smaller than the I-frames. This feature can be exploited by the attackers to launch more serious attacks. For example, if the attacker drops the I-frame, the receivers cannot decode the received P and B frames, which can significantly reduce the video quality. Here the objective of an attacker is to maximize the reduction of video quality without being identified. The dropping attack is usually at the network layer, which is most likely based on IP. At the network layer, it is hard to precisely attribute a packet to an I, P, or B frame. However, based on the application knowledge, the attacker can identify the I, P, B frames by measuring the packet size. Further, the attacker can reduce the video quality without increasing the number of packet drops by exploiting the IP fragmentation

[*] This work was supported in part by Army Research Office (W911NF-05-1-0270 and W911NF-07-1-0318) and the National Science Foundation (CAREER-0643906).

SecureComm 2008, September 22 - 25, 2008, Istanbul, Turkey
Copyright 2008 ACM 978-1-60558-241-2 ...$5.00.

knowledge. The I-frame is usually very large, and may have to be cut into several IP packets. The attacker can exploit this knowledge by only dropping one IP fragment instead of the complete I-frame. Without the dropped IP fragment, the receiver cannot reassemble the I-frame and hence cannot play the video.

In this paper, we show that the attacker can launch various packet dropping attacks by exploiting the application layer and network layer knowledge without creating abnormal behavior. Through extensive simulations and analysis, we show that the attackers can significantly reduce the video quality without increasing the packet dropping rate. We also propose several possible solutions to address the dropping attacks on video streaming.

The rest of the paper is organized as follows. The next section introduces some background knowledge on video streaming. Section 3 presents various dropping attacks and Section 4 evaluates the performance of the network under attacks. Section 5 talks about how to diagnosis the IP dropping attack and analytical model. Related work is discussed in Section 6 and section 7 concludes the paper.

2. BACKGROUND

In this section, we present some background information on video compression and video streaming.

2.1 Video Compression

MPEG ([4]) has been developed and widely used for storing and streaming videos. With compression, it reduces the bandwidth required to transmit digital video. Based on the original video data, an MPEG encoder produces a coded bit stream representing a sequence of encoded pictures. There are three types of encoded pictures/frames: I (intracoded), P (predicted), and B (bidirectional).

- An **I-frame** is encoded as a single image, with no reference to any past or future frames.
- A **P-frame** is encoded relative to the past reference frame. A reference frame is a P-frame or I-frame. The past reference frame is the closest preceding reference frame.
- A **B-frame** is encoded relative to the past reference frame, the future reference frame, or both frames. The future reference frame is the closest following reference frame (I or P). The encoding for B-frames is similar to P-frames, except that it may refer to future reference frames.

The sequence of encoded frames is specified by two parameters: the distance between I or P-frames (denoted by M), and the distance between I frames (denoted by N). Thus, if M is 3 and N is 9, a typical sequence of encoded frames is shown in Figure 1.

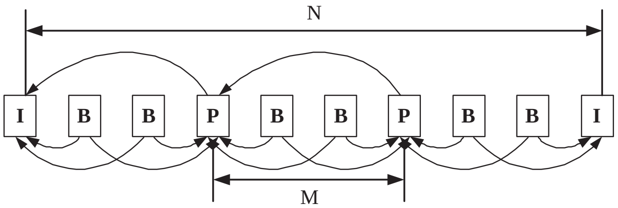


Figure 1: A typical sequence of encoded frames

The arrows represent the inter-frame dependencies. Frames do not need to follow a static IPB pattern. Each individual frame can be of any type. However, a fixed IPB sequence is used through the entire video stream for simplicity. In this paper, we assume a fixed IPB sequence is used.

2.2 Video Streaming

Datagram protocols, such as the User Datagram Protocol (UDP), can be used to transmit the media stream as a series of small packets. However, there is no mechanism within UDP to guarantee data delivery. It is up to the receiving application to detect packet loss or corruption and to do error recovery.

The Real-time Streaming Protocol (RTSP) [21], Real-time Transport Protocol (RTP) [20] and the Real-time Transport Control Protocol (RTCP) [6] were specifically designed to stream media over the network. RTP and RTCP are built on top of UDP and are commonly used together. RTP is used to transmit data and RTCP is used to control QoS. The structure of a RTP packet is shown in Figure 2. As shown in the figure, the real-time video that is being transferred forms the *RTP Payload*. The RTP header contains information related to the payload, e.g. the source, size, encoding type etc.

IP header	UDP header	RTP header	RTP payload

Figure 2: The RTP packet structure

MBZ	T	TR		N	S	B	E	P		BFC		FFC

Figure 3: MPEG video-specific header

An MPEG Video-specific header (Figure 3) shall be attached to each RTP packet after the RTP header. Here, field P indicates frame type. This value is constant for each RTP packet of a given frame. Value 000B is not used and 101B - 111B are reserved for future extensions to the MPEG ES specification. Value 001B, 010B and 011B indicate I, P and B frames, respectively.

3. SECURITY ATTACKS ON VIDEO STREAMING

In this section, we first show how the attacker can launch dropping attacks by exploiting the IP fragmentation and the application layer knowledge. Then we show how the attacker can obtain such knowledge and how he can drop the right packet.

3.1 Packet Dropping by Exploiting IP Fragmentation Knowledge

IP can only provide an unreliable (i.e., best effort) service, which means that the network cannot guarantee packet delivery. Thus, the received packets may be corrupted, out of order, duplicated, or lost. These issues will be addressed by the upper layer protocol. For example, to ensure in-order delivery, the upper layer may have to buffer the out-of-order packet and wait for the missing packet.

Data from the upper layer protocol is encapsulated into one or more IP layer packets. If the upper layer protocol does not fragment the the application data to the size of the Maximum Transmission Unit (MTU), the IP layer has to cut the data packet into smaller fragments so that the link layer can transmit them.

Most video streaming protocols rely on UDP, which may not fragment the application data. As a result, the IP layer has to cut the data into k IP fragments with sizes of MTU, MTU, ..., smaller than a MTU. As long as one fragment is lost, the receiver will not be able to reassemble the original packet. Thus, by exploiting the IP fragmentation knowledge, the attacker only needs to drop one fragment of the packet to achieve the same effect of dropping multiple fragments of the same packet.

The packet dropping attack has the most damage when the packet coming from the transport layer (mostly UDP) is very big. For example, a UDP datagram can be up to 65535 bytes long. When it's passed down to network layer, it can be fragmented into 29

packets. Dropping the 29th packet (the 29th packet is only half-full) only means 1.7% of the whole datagram, but the receiver has to discard the remaining 98.5% datagram.

3.2 A Layered Model for Dropping Attack

The packet dropping attack can become much worse if the attacker exploits other application layer knowledge such as the I, P, B frame information. For example, by dropping the I-frame, the received P and B frames are useless.

Packet dropping by exploiting the fragmentation knowledge is only useful if the transport layer does not fragment the data packet; otherwise, this attack will not be effective. Similarly, packet dropping by exploiting MPEG knowledge is only useful if the network layer can identify the I frames. Therefore, the attacker needs to detect and identify if the network has the specific vulnerability to exploit. The process is referred to as *sensing*. In this paper, we show how the attackers can sense and exploit the vulnerabilities at the network layer and application layer.

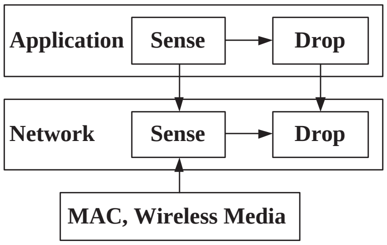


Figure 4: Cross layer dropping attacks

As shown in Figure 4, the application layer senses application types and targets on specific application for dropping attack. It also sets higher level dropping rules that define when dropping attacks should take place and what packets in the victim network should be dropped. The network layer interacts with the corresponding MAC, IP, TCP, and UDP protocols. This layer senses packet sizes and types which can then be exploited for dropping attacks. By exploiting such cross-layer knowledge, the attacker can launch more serious attacks.

In the following section, we will use video streaming as an example to show how the attackers can launch attacks.

3.3 Dropping Attack by Exploiting MPEG

As illustrated in Figure 1, the P-frames and B-frames depend on the closest preceding I-frame. If the I-frame is lost, all the following P-frames or B-frames before the next I-frame become useless. Thus, if an attacker can sense the I-frame, he is able to launch dropping attacks.

To see the damage of this dropping attack, we choose a 60min video trace as an example (This video trace is extracted from movie: *Star Trek - First Contact* [3].) It has 89998 frames, among which, there are 7500 I frames. It has an IPB pattern with $N = 12$ and $M = 3$; i.e., if an I-frame is dropped, all the following 11 frames will become useless. If many continuous frames are lost, video may pause for some time at the receiver. Normally, considering vision persistence, when the video pauses for more than 50ms due to packet loss, it will be noticed and counted; otherwise, the user will not notify the packet loss problem.

To increase the dropping impact, the attacker may only drop the last IP fragment which belongs to the I-frame. Thus, I-frame dropping actually refers to drop the last IP fragment in the rest of the paper.

We randomly choose a certain number of I frames to drop and look at its impact on video streaming. As shown in Figure 5, with I-frame dropping, the video can be paused for 32minutes by dropping 3% of the total packets. With the same dropping rate, random dropping can only achieve 6 minutes of effective pausing time. This demonstrates that dropping attack using cross-layer knowledge can cause much more damage to video streaming.

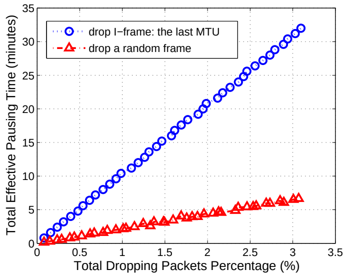


Figure 5: Impact of Dropping Attacks

3.4 Sensing in Ad-hoc Networks

In order for the attacker to launch the dropping attack, he has to be able to identify the I frames. In this section, we show how the attackers can achieve this.

In Section 2.2, we show that an attacker can sense packets by checking the corresponding field P in RTP header which is embedded in an IP packet if it is not encrypted. However, encryption may be used to protect the packet content. In this paper, we assume that the entire packet is encrypted and only packet size and packet timing information can be measured.

Sensing based on the packet size: Due to IP fragmentation, the packet size at the network layer includes multiple full MTU size (F-packet) and a not-full MTU size (N-packet). If several F-packets are observed and one N-packet at network layer, they are most likely fragments of the same packet, and their packet sizes are added to get the original datagram size. If several N-packets are observed, they belong to different datagrams. These rules are valid in most situations except when a datagram is fragmented into only F-packets or a N-packet of a datagram is lost. Later, we will show that the proposed sensing algorithm can tolerate such errors. Next, we first show how a sensing algorithm can identify different kinds of frames without considering packet loss, and then considering packet loss.

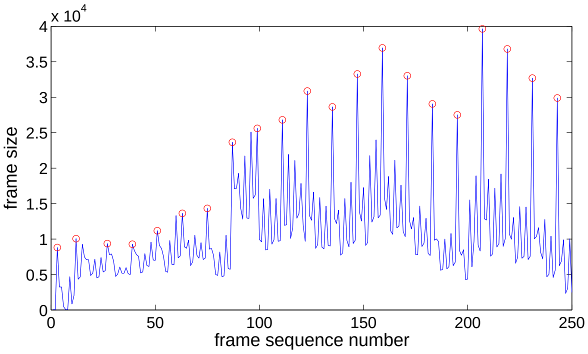


Figure 6: Local maximum with N=12

Sensing without packet loss: After collecting all the datagram sizes, the attacker needs to find out which datagram actually contains an I-frame. Figure 6 shows a trace of 250 continuous frame sizes, based on which there are several observations:

1. I-frame is the largest (local peaks) compared with the following P and B frames;
2. P-frame size varies between the size of the neighboring I-frames and B-frames.

Based on these observations, local maximum can be used to mark all the peaks shown as circles in Figure 6.

Algorithm 1 Look for N

```
Input: an array of datagram sizes, $s$;
Output: $N$;
Procedure:
for $neighborsize = 2$ to 20 do
    peaks=localmaximum(s, neighborsize);{localmaximum finds all the peaks' indexes on $s$ within $neighborsize$}
    $diffpeaks = \{peaks[2] - peaks[1], peaks[3] - peaks[2], \dots, peaks[n] - peaks[n-1]\}$; {find the difference between every neighboring peaks' indexes}
    c=the total number that $diffpeaks_k \neq neighborsize$;
    if $c/n < \epsilon$ then
        $N = neighborsize$;
    end if
end for
return $N$;
```

Based on Local Maximum, Algorithm 1 can be used to look for N. In the algorithm, the $neighborsize$ is checked from 2 to 20 which covers most commonly used N and there are several observations:

- When $neighborsize < N$ (Figure 7(a)), peaks are marked within a smaller neighborhood. Some of the peaks marked by Local Maximum ($peaks$) are P-frames. So the difference between neighboring peaks ($diffpeaks$) is mostly close to M.
- When $neighborsize = N$ (Figure 7(b)), all the peaks marked by Local Maximum are I frames, because within N neighbors, the I-frame always has the largest size. So the difference between neighboring peaks is N.
- When $neighborsize > N$ (Figure 7(c)), I and P-frames both have chances to be marked as peaks because Local Maximum are evaluated within a bigger neighborhood and the I-frame might be smaller than the I-frame or P-frame in the next cycle. Thus, the difference between neighboring peaks may not be equal to $neighborsize$.

We use c/n to filter those wrong $neighborsize$s. After the loop, the biggest $neighborsize$ is kept, which is N' (to differentiate it from the real N). In order to verify the detection accuracy of the above algorithm, we collect 666 video traces ([2]) with different Ns and different scenes, including movies, cartoons, sports events, tv shows, parking lot cameras and class lecture videos and run Algorithm 1 on each of them. If the value returned from the algorithm (N') is equal to the real N, it's counted as correct, and the result is shown in Figure 8.

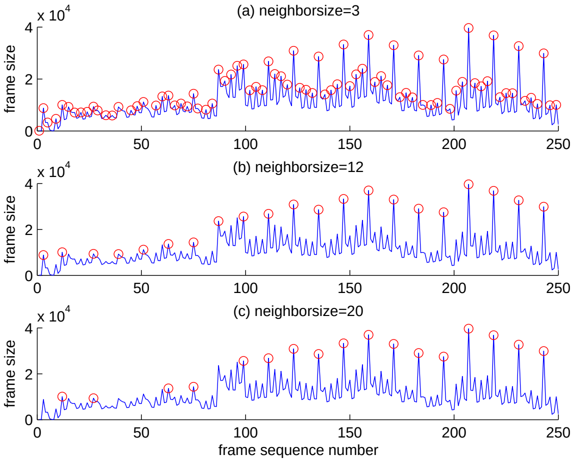


Figure 7: local peaks with different neighborsize

From Figure 8, we can see that when the threshold $\epsilon \leq 0.3$ and the number of frames checked $L \geq 400$, the detection accuracy is greater than 0.95. Considering the video is streaming at a speed of $40ms/frame$, 400 frames take $16s$; i.e., it only takes an attacker $16s$ to find out the correct N to launch the attack.

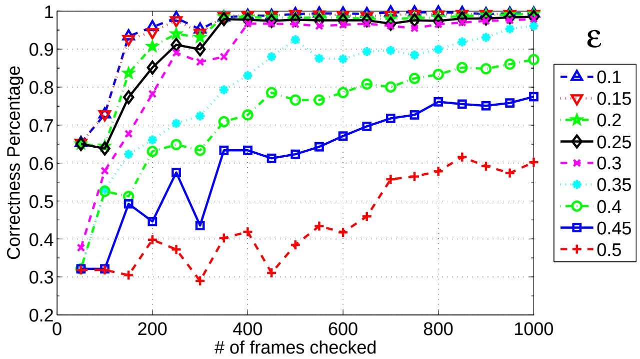
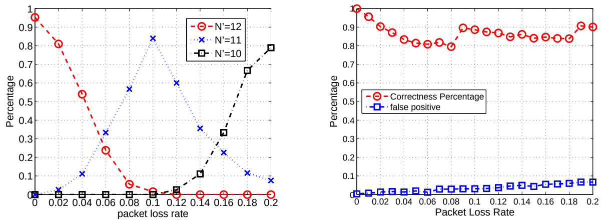


Figure 8: Correct Detection Ratio

Sensing with packet loss: MAC layer packet loss is common in wireless ad hoc network and it may affect the sensing accuracy. Intuitively, packet loss decreases N'. For example, with a $N = 12$ video streaming, when packet loss rate is 0.1, N' would be 11 instead of 12 and N' will decrease to 10 when the packet loss rate increases to 0.2. Based on this observation, we perform the following simulations. Assume package loss rate ranges from 0.01 to 0.20. On a set of video traces with $N = 12$, we check 400 frames in each video trace and set $\epsilon = 0.3$ to sense N. The result is shown in Figure 9(a).

The figure shows the percentage of video traces with certain N' under different packet loss rate. When the packet loss rate is around 0.1, the probability of $N' = 11$ is about 85% and the probability of sensing $N = 12$ is only 2%. When the packet loss rate increases to 0.2, 79% of the video traces appear to have $N' = 10$. To overcome the impact of packet loss, the attacker can check more frames to increase the sensing accuracy.



(a) The Impact of Packet Loss on N (b) Correct Detection Ratio with Packet Loss

Figure 9: Packet Loss

3.5 Dropping the Right Packet

After having N, the attacker can drop the N-packet belonging to the I-frame. The simplest way is to count N datagrams and drop an N-packet. However, this assumes that all the packets are forwarded without loss, which may not be the case in wireless ad hoc network. In case of packet loss, if the attacker still count to N, a wrong packet will be dropped. For example, in a $N = 12$ video stream, if a B-frame is missing, the cycle will become $N = 11$. Therefore, by counting to N, the dropped packet will not be an I-frame packet and this chain effect will continue and affect the following packets.

To overcome the packet loss problem, Algorithm 2 is used which is based on the Local Maximum and the N' sensed in Section 3.4. This algorithm is to test if the current packet is a local maximum when $neighboursize = N'$. If so, the packet belongs to an I-frame and should be dropped; otherwise, it should not be dropped.

Algorithm 2 Does current packet belong to I-frame?

```
Input: $N$;
 an array of datagram sizes, $s$;
 size of the datagram size array, $size$;
Output: $TRUE/FALSE$;
Procedure:
peaks=localmaximum(s, N);{localmaximum finds all the peaks' indexes on $s$ within $N$}
if $peaks[n] == size$ then
    return $TRUE$;{Does the last peak index point at the current packet (which belongs to the last datagram)? }
end if
return FALSE;
```

To show the effectiveness of the algorithm, we perform simulations based on Algorithm 2 on a video trace ($N = 12$) to find out all the fragments which belong to the I frames. Comparing the value returned from the algorithm with the actual (I,P,B) value, the results are shown in Figure 9(b).

In Figure 9(b), the *Correctness Percentage* shows the ratio of dropped I frames to the total number of I frames. The dropped I frames include two types: I frames dropped due to the dropping attack and those due to packet loss. Overall, more than 80% of I frames are actually dropped. As shown in the figure, there are two sudden increases of the correctness percentage when the packet loss rate changes from 0.08 to 0.09 and from 0.18 to 0.19. This is due to the changing of N' as shown in Figure 9(a).

We also evaluate the *false positive* which is the percentage of N-packets belonging to P-frame or B-frame, but being misclassified as belonging to I frames and being dropped. From the attacker's point of view, a low false positive rate indicates that the attacker will not significantly decrease the delivery ratio and still be able to maintain the same pause time when dropping those misclassified N-packets. But a high false positive may significantly decrease the attacker's pause time since he drops too many misclassified N-packets instead of correct ones. As shown in the figure, the false positive is very low. For example, it is at most 0.07 when packet loss rate is 0.2.

Multiple attackers along the same routing path: If there is only one attacker on the routing path, it simply drops the I frames according to the N' it senses. If there are more than one attackers, the attackers other than the first one will treat it as packet loss.

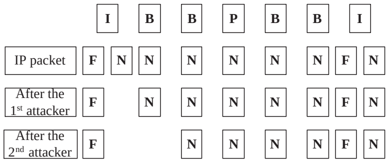


Figure 10: The effects of multiple attackers along the same routing path. F means a full MTU packet and N means a not-full MTU packet.

As shown in Figure 10, when a packet arrives at the first attacker, the attacker drops the N-packet in the I-frame. If there is another attacker on the path, the attacker will think F-packets and N-packets both belong to the I-frame and drop the N-packet. However, the second drop is not necessary since it belongs to the B-frame. Similar cases exist if there are more attackers available in the routing path. However, if the first attacker did not drop the N-packet in the I-frame, the next attacker on the same routing path still has chance to capture and drop it.

In summary, if the prior attacker drops the right packet with a low probability, the following attackers will have a high chance to drop the right N-packet in the I-frame. However, if the prior attacker drop the right packet with a high probability, or there are too many attackers on the same routing path, the following attackers will have less chance to drop the right packet.

4. PERFORMANCE EVALUATIONS

In this section, we use simulations to quantify the effects of dropping attacks on the system performance.

4.1 Metrics and Simulation Setup

The dropping attach can affect the system performance in different ways. To measure these effects, we use the following metrics.

- *Delivery Ratio:* The ratio of the received bytes to that sent out by the source. It is related to the packet dropping rate. With a large packet dropping rate, the delivery ratio will be smaller, and vice verse. From the attacker point of view, he should use a smaller packet dropping rate (i.e., high delivery ratio) to avoid being detected.
- *Video Pause Time:* It shows how long the video pauses. The attacker tries to increase the pause time without reducing the delivery ratio.

Two types of attackers are considered, i.e, the *dumb* attacker and the *smart* attacker. The dumb attacker randomly drops IP packets with a certain dropping probability. The smart attacker is the special attacker described in Section 3 which tries to drop the last fragment of the I-frame and drops it with a certain dropping probability.

The simulation is based on GlomoSim [1]. Each simulation uses a 20min video. The source node sends out streams of video packets to the destination node every 40ms. We consider a small system in which 9 nodes are placed 300m away from each other in a line. Nodes use IEEE 802.11 MAC with a node receive range of 376.782m. The channel capacity is 2 Mb/s. The first node (node 1) communicates with the last node (node 9). The other 8 nodes route packets without generating any traffic. Attackers are compromised nodes among these 8 nodes. We consider the effects of the dropping probability and the number of attackers on the system performance.

4.2 The Effects of the Dropping Probability

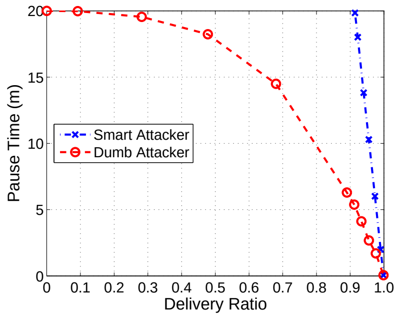
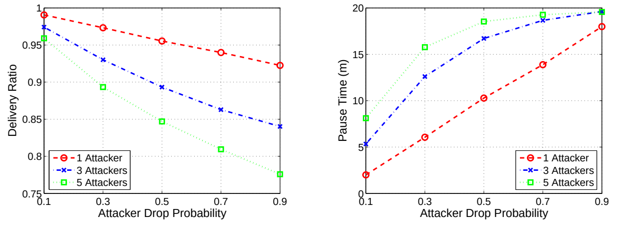


Figure 11: The effects of the delivery ratio on the video pause time

Figure 11 shows the effects of the delivery ratio on the video pause time. Intuitively, the pause time increases as the delivery ratio drops. However, the deliver ratio has different effects on the pause time for the dumb and smart attackers. With the same dropping probability, the dumb attacker drops each packets with the same probability, but the smart attacker only drops the I-frame. By dropping the I-frame, other P and B frames cannot be used, and hence become useless and increase the pause time.

As shown in the figure, the pause time decreases as the delivery ratio increases. However, the smart attacker can create more damage than the dumb attacker. With 91% delivery ratio, the smart attacker pauses the video for 19.848m and the dumb attacker only pauses the video for 5.381m. Note that when the deliver ratio is lower than 91%, the pause time of the smart attacker is out of the curve.

4.3 The Effects of Multiple Attackers



(a) Delivery Ratio

(b) Pause Time

Figure 12: Multiple Attackers

In this section, we evaluate the pause time under different number of smart attackers. With more attackers, the delivery ratio decreases and the pause time increases. From figure 12(b), we can see that the pause time is doubled from 1 attacker to 3 attackers when the packet drop probability is less than 0.5. As discussed in Section 3.5, when the drop probability is low, the later attackers may drop I-frame fragments, which caused the pause time to increase faster than the decrease of the delivery ratio (Figure 12(a)). When the drop probability increases, the prior attackers have dropped almost all the I-frame fragments and the later attackers start to drop P or B frames, and thus the delivery ratio decreases faster than the increase in the pause time. When the number of attackers continues to increase, the delivery ratio decreases faster than the increase of the pause time. This is because most of the drops in the 4th or 5th attacker are P or B frames.

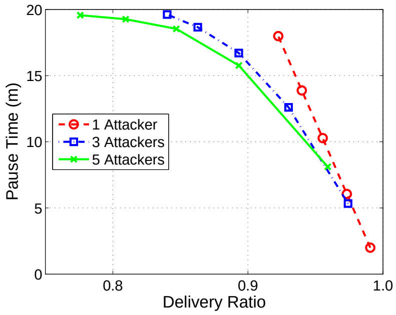


Figure 13: Delivery Ratio Vs. Pause Time under different number of attackers

Although more attackers can increase the pause time, it decreases the delivery ratio which makes the attacker easily detected. In order to better compare the gain and loss, we plot Figure 13. We can see clearly that with the same pause time, increasing the number of attackers will result in lower lower delivery ratio, because some of the drops are P or B frames. When the pause time is less than 5m, multi-attackers and single attacker almost have the same delivery ratio. The difference in delivery ratio becomes more obvious when the pause time is larger than 15m. For example, 3 attackers and 5 attackers have similar delivery ratio until the pause time increases to be over 18m. Therefore, if the attackers only want to pause the video pause for 5 or less, the number of attackers does not matter too much. On the other hand, if the attackers want to pause the video for 15 or more, there should only be one working attacker.

5. DISCUSSIONS

In this section, we discuss some possible ways to detect the dropping attacks, and some possible solutions. Also, we use a simple analytical model to show the difficulty of dealing with dropping attacks.

5.1 Attack Diagnosis

Nodes in the network under attack (i.e., the victim network) will notice the long video pause time. However, the video pause might be caused by various reasons, e.g., link transmission error or routing problem. In this subsection, we study two ways for the network to find out if it is under dropping attacks: One is at the destination node and the other is by the neighboring nodes.

5.1.1 Detection by the Destination Node

There may be various reasons for the destination node to experience long video pausing time.

1. Bad Signal: Data packets cannot reach the destination correctly due to a low signal to noise ratio (SNR).
2. Jamming: An attacker transmits signals that do not follow an underlying MAC protocol and severely interfere with the normal operation of wireless networks.
3. Congestion: A link or a node has too much data to send which results in long queuing delay and packet loss.
4. Network Disrupt: A route is broken or the network is partitioned.
5. Dropping Attack: The attacker selectively drops I frames as discussed in this paper.

Based on signal strength, delivery ratio and relative I-frame delivery ratio (i.e., the number of I-frame received divided by the total number of packets received), the destination node can decide if the video pause is caused by the dropping attack based on Table 1.

Table 1: Detection Table

Causes	Signal Strength	Delivery Ratio	I-frame Delivery Ratio
1	low	low	high
2	high	low	high
3	high	low	high
4	high	low	high
5	high	relatively high	low

There is a tradeoff between fast detection and false positive rate. At one extreme, the destination node could treat a single I-frame loss as an indication of a bad route. However, this may lead to high false positive, because congestion, channel fading, etc. can also lead to I-frame loss. On the other extreme, the destination node could wait to report the problem until a large number of I frames have been lost. Although the false positive rate is low, the created damage to the network is very high. Thus, the destination node should find a balance between these two extremes.

5.1.2 Neighbor Detection

Although the destination node knows that it is under dropping attack, it does not know where the attacker is. A better way to identify the attacker is through the collaborative detection of the neighboring nodes.

The watchdog protocol [19] was designed precisely for this purpose. The key idea of watchdog is to exploit the broadcast nature of the wireless medium. If node i sends a packet to k via j, i should overhear the subsequent transmission from (neighboring) j to k. If i cannot hear such transmission, it suspects that j drops the packets. Since a node may falsely accuse other nodes, other researchers [24] propose to use a group of neighbors instead of one neighbor to collaboratively detect packet dropping. With the cooperation of the neighboring nodes, they can find out if any node launches dropping attack.

Using neighbor detection also has its own limitation since neighbors have to keep monitoring other nodes, and hence power saving techniques are hard to deploy. Further, the communication among them also increases the control overhead, and it is hard to identify if the dropped packet is an I-frame or not. Similar to the attack detection by the destination node, there is a tradeoff between fast detection and false positives.

5.2 Dealing with Dropping Attacks

Once a routing path has been detected to suffer from dropping attacks, an alternate path should be established. To establish a new routing path, the source can send another routing request. After receiving the route reply messages, the nodes can build a routing path to exclude the malicious node if it can be identified. If the malicious node cannot be identified, the new routing path should differentiate from the old path as much as possible.

It may take a long time to establish a new route. An alternate solution is to employ multipath routing, which establishes multiple disjoint routing paths beforehand. If a routing path is under attack, the source can simply use another one. Also, reputation systems [8] can be used to help defend against the dropping attacks. To establish a new route, the source and the forwarding nodes only select well-behaved nodes to get around the attackers. In the next subsection, we develop a simple model to illustrate the time delay for establishing another routing path.

5.3 Analytical Model

Consider an ad hoc network with n nodes among which a nodes are malicious. Denote p as the probability that a randomly selected node is an attacker, so $p = a/n$. With a routing path of h hops, the probability that the path contains no attacker is $(1-p)^h$.

Before detecting the dropping attack, a number of delays are incurred. First, a duration T_{sense} is incurred for the attacker to sense the parameter N' in order to launch attack. Next, it will take the end node T_{detect} to detect the dropping attack. Finally, the node must wait to receive one or more route reply messages to establish a new route with a duration of T_{RR}. After these three phases, a node can transmit data using the new routing path. However, the new path includes at least one attacker with probability of $1-(1-p)^h$. If so, the node has to go through the three steps again. Based on the similar deduction method in [7], the time to find a good path is given by

$$E(T_u) = T_{RR} +$$

$$(E(T_{sense}) + E(T_{detect}) + E(T_{RR})) \times (\frac{1}{(1-p)^h} - 1) \quad (1)$$

We have several observations about Equation 1. When p approaches 1 or the route length is long, T_u is large. That is, it takes the node a long time to find a good path. On the other hand, when p is close to 0 or the route length is small, T_u approaches 0, which means that the node can find a good path quickly.

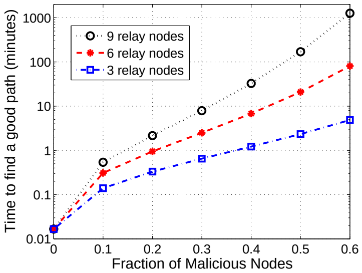


Figure 14: Time to find a good path

Based on Equation 1, Figure 14 shows the time to find a good path as a function of the percentage of attackers. The figure also compares three cases where the routing path includes 3, 6 and 9 relay nodes. With $T_{RR} = 1s$, when $N = 12$, the detection time is $16s$ as shown in Section 3.4. T_{detect} is selected to achieve a balance between detection time and false positive. We assume the attacker drops one I-frame in every 3 I-frames and when the destination node detects 10 I-frame missing, it reports dropping attack and starts to find a new path. Based on these data, $T_{detect} = 10 * 3 * 12 * 40ms/frame = 14.4sec$.

From the figure, we can see that without attacker, the network needs $1sec$ to re-establish a broken path. Suppose there are 6 relay nodes in the routing path (the middle line in the figure). With 20% of attacking nodes, it takes 1 minute to find a good path. With 40% of attacking nodes, the time increases to 6.8 minutes. The impact of the attacker will be more severe in large-scale networks where a longer routing path is more likely to include an attacker. For example, with 9 relay nodes, the time to find a good path increases to 2.2 minutes under 20% attacking nodes and 32.6 minutes under 40% attacking nodes.

5.3.1 *The Performance of Multipath Routing*

As discussed earlier, multipath routing can be used to deal with dropping attacks. Consider the best case where there always exists a good routing path. Then,

$$E(T_u) = T_{RR} + (E(T_{sense}) + E(T_{detect})) \times (\frac{1}{(1-p)^h} - 1) \quad (2)$$

Compared to Equation 1, it only reduces the duration by $E(T_{RR})$ which is small compared to the other two delays. Therefore, multipath routing doesn't really help to defend against dropping attack.

5.3.2 *The Performance of Reputation Systems*

Reputation systems ([8, 9, 5, 16]) can also be used to deal with dropping attacks. In this section, we evaluate the capability of such systems to defend against the dropping attack. In order to abstract from the technical details, we assume that the reputation system can be modeled as a black box with two parameters:

- False positives (f_p): This is the rate at which the reputation system reports well-behaved nodes as being malicious.
- False negatives (f_n): This is the rate at which the reputation system reports a malicious node as being well-behaved.

When establishing a new route, the source and the forwarding nodes try to select well-behaved nodes only. However, they may mistakenly choose (with probability f_n) a bad node as a well-behaved node. Assume that the system has a proportion (g) of good nodes. The probability that a randomly selected node is an attacker is $(1-g)f_n$, and we have:

$$E(T_u) = T_{RR} +$$

$$(E(T_{sense}) + E(T_{detect}) + E(T_{RR})) \times (\frac{1}{(1-(1-g)f_n)^h} - 1) \quad (3)$$

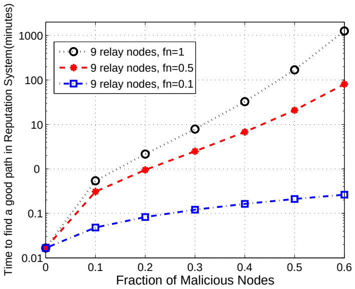


Figure 15: Total duration to find a good path in a reputation system

Figure 15 shows that using a reputation system with $f_n = 0.1$ can significantly reduce the time to find a good path compared to system without using reputation ($f_n = 1$). On the other hand, the false positives rate f_p has a negative impact, since the source and the forwarding nodes will avoid the real good nodes (g) with a bad reputation (f_p), during route establishment. This reduces the number of possible paths by a factor of $g(1-f_p)/g = 1-f_p$ with respect to a system that does not use any reputation mechanism. Therefore, when $f_p = 1$ (i.e. all good nodes are judged to be bad), $1 - f_p = 0$ and no route can be established. When $f_p = 0$, the reputation system does not mislead the route establishment process, and the performance is similar to that without reputation mechanism.

6. RELATED WORK

According to [14], attacks on ad hoc networks generally fall into two categories: routing-disruption attacks and resource-consumption attacks. Much progress has been made in securing ad hoc networks against these attacks recently; however, none of them considers dropping attacks exploiting cross-layer knowledge.

Routing-disruption attacks include blackhole attack, wormhole attack, rushing attack, etc. Various solutions have been proposed to deal with these attacks. The Secure Efficient Ad hoc Distance vector routing protocol (SEAD) [13] was proposed to protect distance vector routing protocols (DSDV) against various attacks. SEAD makes use of one-way hash functions to authenticate the routing metric and the sequence numbers in the routing table. SEAD is robust against multiple uncoordinated attackers, but it does not address the problem of wormhole attacks. Ariadne [15] was designed to protect source routing protocols such as DSR. Ariadne relies on efficient symmetric cryptography and provides security against compromised nodes and attackers. The authors suggested two countermeasurements: passive acknowledgment and multi-path routing. They also suggested blacklisting poorly performing nodes to prevent them from being included in future routes, which has some similarity to the reputation-based systems [8].

Resource-consumption attacks include jamming, selective dropping, etc. The goal of this kind of attack is to consume the system resources such as memory, CPU or bandwidth as much as possible. Various techniques have been proposed to identify these attacks. In [23], Xu *et al.* explored four different types of jamming attack models and examined the capability of different measurements to classify the presence of a jammer. The measurements include signal strength, carrier sensing time, and the packet delivery ratio. However, it only detects the jamming attacks without finding the jammer. Aad *et al.* [7] introduces the Jellyfish attack against closed-loop flows such as TCP. In Jellyfish attack, the attacker selectively drops some packets to reduce the TCP throughput to almost zero.

Although the aforementioned research can secure ad hoc networks in some sense, none of them considers packet dropping attacks exploiting the IP fragmentation knowledge and the application layer video encoding knowledge.

7. CONCLUSION

In this paper, we studied a cross-layer dropping attack. By exploiting the application layer knowledge, the attacker can selectively drop I frames. Without these I frames, the receiver cannot play the video. As a result, the attacker can reduce the video quality without increasing the number of packet drops too much, and hence it is hard to be detected. We also proposed various ways to identify the I frames and studied such attacks in various settings.

We proposed several possible solutions to address the dropping attacks on video streaming. We also found that the victim network would always suffer from the dropping attack unless all forwarding paths were free of malicious nodes. As future work, we will investigate node mobility issues and study various solutions to deal with these cross-layer dropping attacks.

8. REFERENCES

[1] Glomosim, http://pcl.cs.ucla.edu/projects/glomosim/.

[2] http://trace.eas.asu.edu/tracemain.html.

[3] http://trace.eas.asu.edu/trace/pics/frametrace/mp4/verbose_firstcontact.dat.

[4] http://www.mpeg.org/.

[5] Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Trans. Parallel Distrib. Syst.*, 18(4):460–473, 2007. Member-Runfang Zhou and Fellow-Kai Hwang.

[6] Rtp control protocol extended reports (rtcp xr), Internet RFC 3611, 2003.

[7] I. Aad, J.-P. Hubaux, and E. W. Knightly. Denial of service resilience in ad hoc networks. In *MobiCom '04*.

[8] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the CONFIDANT protocol. In *MobiHoc '02*.

[9] S. Buchegger and J.-Y. Le Boudec. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In *P2PEcon 2004*.

[10] C.-O. Chow and H. Ishii. Enhancing real-time video streaming over mobile ad hoc networks using multipoint-to-point communication. *Comput. Commun.*, 30(8):1754–1764, 2007.

[11] M. Guo, M. H. Ammar, and E. W. Zegura. V3: A vehicle-to-vehicle live video streaming architecture. In *PERCOM '05*.

[12] M.-Y. Hsieh, Y.-M. Huang, and T.-C. Chiang. Transmission of layered video streaming via multi-path on ad hoc networks. *Multimedia Tools Appl.*, 34(2):155–177, 2007.

[13] Y.-C. Hu, D. B. Johnson, and A. Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *WMCSA '02*.

[14] Y.-C. Hu and A. Perrig. A survey of secure wireless ad hoc routing. *IEEE Security and Privacy*, 2(3):28–39, 2004.

[15] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Wirel. Netw.*, 11(1-2):21–38, 2005.

[16] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW '03*.

[17] S. Lin, Y. Wang, S. Mao, and S. Panwar. Video transport over ad-hoc networks using multiple paths. *IEEE International Symposium on Circuits and Systems*, 1:57–60, 2002.

[18] H. Luo, R. Ramjee, P. Sinha, L. Li and S. Lu. UCAN: A Unified Cellular and Ad-Hoc Network Architecture. In *MobiCom '03*.

[19] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00*, Boston, MA, USA.

[20] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications, Internet RFC 3550, 2003.

[21] H. Schulzrinne, A. Rao, and R. Lanphier. Real time streaming protocol (rtsp), Internet RFC 2326, 1998.

[22] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod. Cross-layer design of ad hoc networks for real-time video streaming. *Wireless Communications, IEEE*, 12(4):59–65, 2005.

[23] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *MobiHoc '05*.

[24] H. Yang, X. Meng, and S. Lu. Self-organized Network Layer Security in Mobile Ad Hoc Networks. In *ACM Wireless Security Workshop*, 2002.