

The Snowball Effect: Detailing Performance Anomalies of 802.11 Rate Adaptation

Michael Loiacono, Justinian Rosca
Siemens Corporate Research
755 College Rd. East, Princeton, NJ 08540
Email: {michael.loiacono, justinian.rosca}@siemens.com

Wade Trappe
Wireless Information Network Laboratory (WINLAB)
Rutgers University
Email: trappe@winlab.rutgers.edu

Abstract—Current rate adaptation schemes for 802.11 exhibit sudden and severe drops in throughput in real application scenarios. Although there have been several propositions to remedy such rate adaptation failures, there has not been a thorough analysis of the causes that lead to such a “Snowball Effect.” This paper provides an analysis of the factors that lead to the poor performance of current rate adaptation schemes in real environments. We show that current rate adaptation schemes fail because they do not differentiate between poor channel conditions and collisions as the source of transmission failures, and consequently invoke improper responses that cascade to dramatic throughput degradation. We support the analysis through experimentation with real data from a wireless video surveillance application, and provide recommendations for the next generation of WiFi rate control schemes.

I. INTRODUCTION

Commercial wireless technologies, such as 802.11, have led to significant enhancements in consumer access and connectivity. Unfortunately, the current generation of 802.11 equipment can be easily demonstrated to fail when subjected to adverse conditions associated with many real applications (especially voice/video over WLAN). One notable flaw of current local area wireless technologies is their use of simplistic rate adaptation mechanisms, which can cause a wireless application to experience sudden, severe and often irrecoverable drops in throughput. This effect, which throughout this paper we shall call the “Snowball Effect” (due to its characteristic behavior witnessed in laboratory experiments), is a serious hurdle for the deployment and adoption of local wireless technologies for a broad range of applications and, consequently, a thorough understanding of this effect and the performance of current 802.11 rate adaptation schemes is necessary in order to remedy this issue.

Rate adaptation refers to techniques for dynamically and adaptively choosing modulation schemes according to channel conditions. The modulation schemes that yield high PHY rates (e.g. 54Mbps) are fragile and susceptible to corruption from interference. On the other hand, more resilient modulation schemes can be employed at the expense of lower PHY rates. There is clear need for such techniques in order to use the wireless channels effectively.

In order to understand the problems associated with rate adaptation schemes, it is important to realize that transmission failures occur for two reasons, collisions and poor channels, and that the current generation of rate adaptation algorithms typically address one or the other of these issues, and generally

not both. A consequence of such a design is that the responses taken may not be suitable for alleviating the actual cause of transmission failure and might, in fact, worsen the problem. In particular, an algorithm that assumes errors are due to poor channel conditions will severely malfunction if the errors are actually due to collisions, and vice-versa.

Indeed, the seriousness of this problem is indicated by several recent papers (e.g. [1], [2], [3]) where the authors discuss ways to differentiate between the two types of errors. Although there are several papers that propose such solutions, there are few if any papers that diagnose the cause of the problem itself. We believe that in order for rate adaptation schemes to overcome this problem, it is necessary to thoroughly understand the details of the problem itself. In this paper, we set out to thoroughly understand these details, and ultimately arrive at the principal requirements of next generation rate adaptation schemes.

Our analysis is based on both simulation studies and experimental evidence. Throughout our analysis, we have sought to explore this phenomena under real application conditions, and thus we have setup a multi-camera video surveillance application in our laboratory for the purpose of diagnosing the Snowball Effect. Through our analysis, we have found that the underlying cause of the Snowball Effect is the fact that the current generation of rate adaptation schemes do not distinguish between the causes for transmission errors. In particular, when an adaptive algorithm assumes that all transmission errors are due to poor channel conditions when the true cause is collisions, the algorithm will invoke an improper response that leads to a chain reaction of events that ultimately causes the system throughput to drop into a downward spiral. Under certain conditions, the system may recover from the downward spiral on its own, and additionally, the failure may be predicted using a statistic that we derived.

The rest of this paper is organized as follows: Section II defines the Snowball Effect through a complete real-world example. After that, Section III gives a formal analysis of the rate control problem. In Section IV, both simulations and real experiments are used to support the claims made in the Formal Analysis, and also to provide additional perspectives of the Snowball Effect. Then, Section V discusses related work. Finally, we conclude the paper in Section VI.

Although consequences of the Snowball Effect are clear

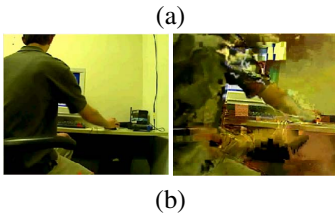
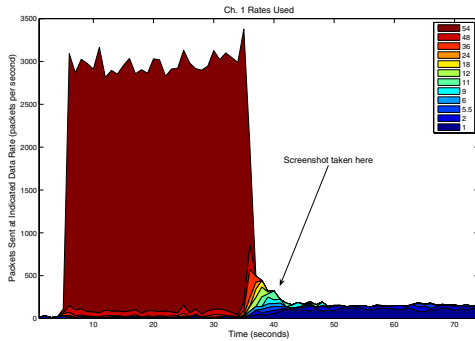


Fig. 1. Video streaming application. (a) Prior to $t=35$, the system accommodates ≈ 3000 packets per second, with nearly all packets being transmitted at 54Mbps. At $t=35$, the system begins to crash. Note the selection of decreasing PHY rates, and also the steep drop in the number of packets per second. (b) Screen-shot of normal (left) and Snowball-degraded (right) video streams.

(severe and sustained drops in system throughput, possibly unrecoverable system failure, and loss of functionality for the end user), the dynamics of this anomaly are not so clear.

II. COMPLETE EXAMPLE - VIDEO STREAMING IN A MULTI-CAMERA ENVIRONMENT

In this section, we define and illustrate the Snowball Effect in a real-world setting. We have used hardware from different vendors for transmission of voice, video, and best effort data traffic over wireless LAN, and in all cases, the voice and video applications suffered seriously because of manifestations of the problem. For clarity, we only focus on the video streaming example.

A. Experimental Setup

Our real-world experimental setup is based on a wireless, real-time, surveillance/security system. We consider the following scenario: Six to ten Axis 207w¹ cameras stream video (3Mbps Constant Bitrate MPEG4/RTP) wirelessly (802.11g) to the central security center (a Dell OptiPlex GX280) via a wireless access point (Linksys WRT54G). The GX280 processes the video in real-time to perform actions such as: face detection, person tracking, forbidden zone detection, etc. Such systems are commonly used to secure airports, subway systems, and other civil infrastructure.

In our lab, the wireless signal is exposed to regular interference from nearby APs, lights, microwaves, etc., but overall, the channel is good enough for 54Mbps transmission. Since there are so many contending wireless stations, we expect the

¹Interestingly, the Axis cameras' release notes caution of a potential downfall when trying to simultaneously use a large number of cameras, but no remedy is suggested. The problems we describe manifest with many other hardware components and configurations, and are not specific to this particular setup or vendor.

TABLE I
NETWORK CONFIGURATION

Layer	Parameter	Value
Application	Codec	MPEG4
	Traffic Direction	One-way (Uplink)
	Source Rate	3 Mbps
	Total Packet Size (bytes)	1536
Transport	Protocol	IP/UDP/RTP
	Header (bytes)	20+8+12
Link/MAC	Protocol	802.11g
	Header (bytes)	8+24 (link+MAC)
	Link Adaptation	Depends on experiment
	Retry Limit	3
	Queue Length	1000 packets + expiration logic
	Fragmentation	Disabled
	RTS/CTS	Disabled
Physical	Protocol	802.11g
	Header (bytes)	8 byte preamble
	Max Rate (bps)	54 Mbps
	Base Rate (bps)	1 Mbps

TABLE II
802.11G PHY/MAC PARAMETERS

Parameter	Value	Comments
Slot time	9 μs	Idle slot time (σ)
SIFS	10 μs	Single Inter-Frame Spacing time
DIFS	28 μs	$SIFS + 2 \cdot \sigma$
CW_{min}	16	Minimum contention window
m	8	Backoff stages
CW_{max}	1024	$2^m CW_{min}$
ACK Packet Size	14 bytes	Size of an ACK

main cause of transmission failures to be collisions, not poor channel conditions. Given this setup, our system is at high risk for the Snowball Effect (since the rate adaptation mechanisms will react assuming that the failed transmissions are due to a bad channel while they are really due to collisions).

We monitored a plethora of system statistics in real-time using AiroPeek SE [4] - a packet sniffing application which we customized to track features of interest using its SDK. The statistics were available in real time, but they were also archived for later use. Additionally, we recorded the received live video streams on the OptiPlex GX280.

The experimental procedure itself was as follows: We started 6 cameras, and simultaneously used AiroPeek to sniff the ether while the screen capture utility recorded the live video. We let the system run until the Snowball Effect occurred. We ran the experiment several times, varying several camera parameters like the video source rate. For the sake of brevity, we only present results from the experimental configuration detailed in Tables I and II.

Additionally, unless otherwise noted, we endorse the following facts/assumptions:

- 1) Prob[there is at least one packet in a station's queue] is near 1, but not exactly 1.
- 2) A wireless packet transmission can fail for two reasons: collision or bad channel.
- 3) There are no hidden/exposed nodes.

B. The Snowball Effect - Experimental Observations

When we operate the network as described in the previous section, it suffers a catastrophic failure after a short amount of time. This catastrophic failure is characterized by:

- Lower 802.11 PHY rates (Figure 1)
- Sudden, steep drop in throughput/goodput (Figure 2)

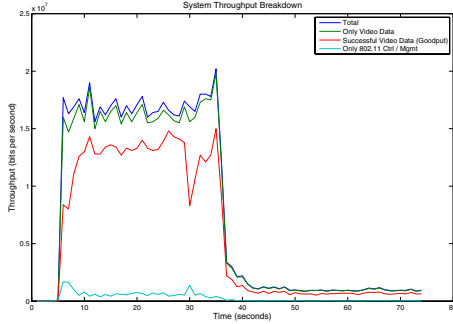


Fig. 2. Prior to $t=35$, the system obtains nearly 18Mbps total throughput. After $t=35$, the system crashes and throughput is driven nearly to zero. In addition to throughput, the figure shows the breakdown of throughput into CTRL/MGMT traffic and video traffic, and it also shows goodput.

- Long interruption of some of the video streams or voice calls
- Network may not recover on its own.

To the end user, this translates to severely degraded video quality (Figure 1(b)), dropped calls (in the case of voice), and loss of application functionality. We call this the Snowball Effect, and we formally analyze it next.

III. FORMAL ANALYSIS

The previous figures show that the system suddenly experiences serious performance degradation - goodput and PHY rates drop abruptly, video quality becomes very bad. What causes this sudden change?

Throughout the formal analysis presented below, we assume that rate adaptation algorithms trigger to lower PHY rates based on some combination of conditions on packet statistics such as increased delay, decreased goodput, increased interval between successfully transmitted packets, or increased number of retries. Of course, there are many types of rate adaptation algorithms, but they all fundamentally rely in some way on these “trigger statistics” [3].

We will show that when a station in the BSS lowers its PHY rate, the global probability of a collision, $P[\text{collision}]$, increases, which in turn, causes the aforementioned rate adaptation “trigger” criteria to be satisfied. Therefore the rate adaptation mechanism lowers the PHY rate further, even though the channel conditions remain adequate for 54Mbps transport.

A. Normal Operation

Consider an initial set of parameters that characterizes each of N stations in a basic service set (BSS) under normal operation:

- λ_i (arrival rate for station i): Assume λ_i is deterministic and identical for all i ($\lambda_i = \lambda, \forall i \in N$) due to constant bitrate video
- μ_i (service rate for station i): General Distribution

For this generic queue structure (i.e. D/G/1/1000), these two parameters yield the probability that a newly arriving packet in station i finds the queue empty, call it $q_i, 0 < q_i < 1, i \in N$. For now, let all stations transmit at the highest rate (i.e. 54Mbps), so $\mu_i = \mu, \forall i \in N$. Furthermore, denote one of these N stations as station j , so $\mu_j = \mu_i = \mu$, and $q_j = q_i = q$.

B. Initial Entry into the Downward Spiral

Now, assume that an event happens (e.g. burst of interference, etc.) which forces one of the N stations (say station j) to move to a lower PHY rate, say 36Mbps. We now have a new set of parameters for the low-rate station:

- $\lambda'_j = \lambda_i = \lambda, \forall i \in N$ (arrival rate is unchanged)
- μ'_j (new service rate for slow station), $\mu'_j < \mu_j$

The arrival rate does not change, but the decreased PHY rate causes the slow station’s service rate to decrease to $\mu'_j < \mu_j$, and consequently, there is a reduced probability, $q'_j < q_j$, that a newly arriving packet (in the slow station) will find the queue empty.

C. Global Effect of One Station Switching to a Lower PHY Rate

It is tempting to assume that this lower μ'_j corresponds only to the station operating at a reduced rate. However, in [5], it was shown that fast stations transmitting at 54Mbps obtain the same throughput as slow stations at 36Mbps² due to the throughput-fairness property of the 802.11 DCF. Therefore, this new μ'_j applies globally to *all* of the stations. Consequently, the reduced q'_j also applies to all stations. Thus, we have:

- $\lambda_i = \lambda, \forall i \in N$ (unchanged)
- $\mu_i = \mu'_j, \forall i \in N$ (global service rate reduction)

Theorem. *Decreasing the global service rate, μ increases $P[\text{collision}]$.*

Proof. We will use some of the results in [6], which is an extension of Bianchi’s results [7] for the case of unsaturated networks. Denote \tilde{C} as the number of *contending* stations (i.e. the number of stations with at least one packet in their queue) in a discrete time step. Initially, assume all stations transmit at their maximum rate (54Mbps). Furthermore, denote \tilde{C}_{avg} as the *average* number of contending stations in each time step:

$$\tilde{C}_{avg} = N - \sum_{i=1}^N q_i \quad (1)$$

Now, let one of the N stations (station j) switch to a lower rate. Then, by our previous arguments, there is a global reduction in service rate (i.e. μ_i is reduced to $\mu'_j, \forall i$, and hence, q_i is reduced to $q'_j, \forall i$). Then, by Equation 1, this means that \tilde{C}_{avg} (and hence \tilde{C}) is likely to be larger now than it was when all stations were transmitting at 54Mbps and obtaining q_i .

Recall the formula for calculating the collision probability under non-saturated conditions [6], $p(\tilde{C}) = 1 - [1 - \tau(\tilde{C})]^{\tilde{C}-1}$, where $\tau(\tilde{C})$ is the probability that one of the contending stations transmits in a given time step. By inspection of this equation, we see that the collision probability grows with \tilde{C} . But we just showed that \tilde{C} grows with the number of low-rate stations in the BSS, and hence with decreasing service rate. \square

²This effect will be amplified in 802.11n, where there is a wider range of PHY rates to choose from.

When the collision probability grows, we argue that there is an increased likelihood that the “trigger statistics” satisfy their criteria. For one example, consider the “retry” “trigger statistic”. We just showed that lowering the PHY rate increases $P[\text{collision}]$. But increasing $P[\text{collision}]$ implicitly implies that there are more failed transmissions, on average. When transmissions fail, they are retransmitted until they either succeed or expire [8]. Thus, increasing $P[\text{collision}]$ also increases the average number of retries.

As a second example, consider the “delay” “trigger statistic”. The delay for a station in an unsaturated 802.11g BSS is comprised of the following components: $D = D_{IFQ} + D_{MAC} + D_{TX}$, where D_{IFQ} is time spent waiting in the interface queue, D_{MAC} is time spent in the MAC (delay due to DCF contention, backoff, etc.), and D_{TX} is transmission time. Lowering the PHY rate obviously increases D_{TX} . Furthermore, as we just explained, the corresponding increase in $P[\text{collision}]$ implies an increase in the average number of retries. When a station suffers a failed transmission attempt, that station must increment its backoff stage before the transmission may be reattempted [8]. Thus, as the number of retries increases on average, we see a corresponding shift to higher backoff stages being used more frequently. Consequently, a packet spends more time doing backoff, thus increasing D_{MAC} . In fact, lowering the PHY rate of a station in the BSS hurts doubly bad since it will also cause the medium to be sensed busy more frequently by other stations, hence causing their backoff counters to take longer to reach zero (and increasing D_{MAC} even further). The sharp increases in D_{TX} and D_{MAC} mean that it takes longer for a packet to leave the transmission stage. Since a packet cannot leave the IFQ for the transmission stage until the previous packet leaves the transmission stage, this means that D_{IFQ} also increases.

Similar arguments, omitted for brevity, reveal that increasing $P[\text{collision}]$ causes goodput to decrease, and the interval between successfully transmitted packets to increase.

D. Completing the Circle

According to the fundamental fact stated in the beginning of this section, rate adaptation algorithms trigger to lower PHY rates based on some combination of conditions on the trigger statistics. We just showed that when a station lowers its PHY rate needlessly, it increases $P[\text{collision}]$, and hence, it promotes the trigger statistics to satisfy their criteria, hence causing stations’ rate adaptation mechanisms to lower PHY rates even further. Therefore, we are back where we started, but at a lower PHY rate. This cycle continues until nearly all stations are using their lowest rate. At this point, catastrophic failure is imminent.

IV. EXPERIMENTAL ANALYSIS

In this section, both simulations and real experiments are used to support the claims made in the previous section. Furthermore, we provide additional perspectives of the Snowball Effect by using results from real experiments.

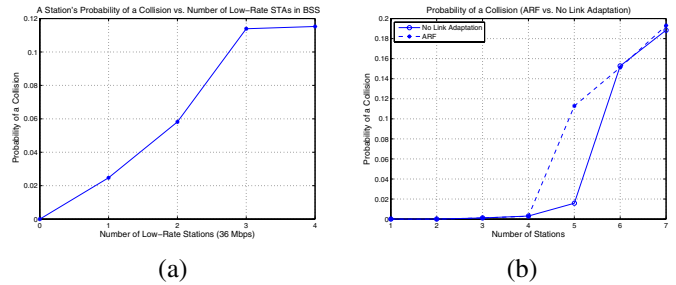


Fig. 3. (a) Probability of collision versus number of low-rate stations; (b) Probability of collision for ARF link adaptation vs. no link adaptation.

A. ns2 Simulation

Our ns2 [9] simulation environment is designed to replicate the parameters of our real experiments (as shown in Table I). There are no interferers, and the SNR is perfect.

The goal of our first ns2 experiment is to validate the aforementioned theorem by studying the relationship between $P[\text{collision}]$ and the number of low-rate stations. We expect $P[\text{collision}]$ to increase with the number of low-rate stations. Indeed, Figure 3(a) shows $P[\text{collision}]$ for a station in the BSS vs. the number of low-rate (36Mbps) stations for a fixed number (5) of total stations in the BSS. We increased the number of low-rate stations from 0 to 4, and reported $P[\text{collision}]$ for the station that remained at a high rate (54Mbps). It is clear from the figure that increasing the number of low rate stations increases $P[\text{collision}]$, thus validating our hypothesis.

The goal of our second experiment is also to validate the aforementioned hypothesis, but in a way that better resembles real-life rate adaptation. Namely, we set out to study the probability of a collision for a station in a BSS that uses ARF [10], and in a BSS that does not use any rate adaptation. The results, Figure 3(b), clearly show that for a given number of stations, when ARF is used, $P[\text{collision}]$ is higher than if ARF is turned off and all stations transmit at a fixed rate of 54Mbps. Since our simulation was configured with no interferers and a perfect SNR, the increase in $P[\text{collision}]$ is entirely due to ARF being fooled: it treats collisions as channel errors, and lowers the PHY rate needlessly - confirming our previous claims.

B. Snowball effect in real-lab WLAN video streaming

Multiple real-world experiments with video/voice over WLAN confirm the “Snowball” behavior. We first describe in more detail the results already shown in Figures 1 and 2. After that, we examine the time needed to recover from the “crashed” regime when one or more clients are intentionally shut down so as to free up bandwidth and ease contention.

Figure 1 shows a real-world example of catastrophic failure in a video over WLAN experiment. Figure 1(a) shows the rapid rate at which the network crashes with respect to the decreasing PHY rates selected by the rate adaptation mechanisms. The total system packets per second indicates that system throughput plummets as lower PHY rates are selected, and hence, video quality is degraded. Figure 2 shows

the relationship between throughput (total number of bits that are physically put on the air), broken down into video data and 802.11 Ctrl/Mgmt packets, and the corresponding goodput (subset of video data packets that are successfully acknowledged by the intended receiver). It is clear that after the network crashes (at $t=35$), the goodput is relegated to less than 1Mbps.

1) *Recovering from Catastrophic Failure:* Once the network suffers a catastrophic failure, what does it take to get the system back at a high rate? To investigate this question, we performed the following experiment:

- 1) Configure 6 Axis cameras according to Table I.
- 2) Start all 6 cameras simultaneously.
- 3) When the network crashes, stop all but 1 of the cameras.
- 4) Start a timer and see how long it takes for the network (i.e. the 1 remaining camera) to recover back to its original PHY rate of 54Mbps, and source throughput of 3Mbps.

Then, we repeat this exact experiment, but instead of stopping all but one camera, we stop all but two cameras, etc. Figure 4 details our results. The captions under each figure show the number of active cameras after the crash and the average recovery times for each experiment. As expected, the recovery time for each experiment increases as we try to recover more and more cameras. In fact, apparently, we cannot recover four or more cameras (c.f. Figure 4(d)).

2) *Predicting when the regime of operation is transitioning into a dangerous state:* It is important to predict when the system state approaches catastrophic failure. In this subsection, we present one feature that highlights the evolution of the system state, and which furthermore, could be used to predict the imminent state.

We expect drops in goodput to indicate potential problems. Therefore, we propose the following feature, $0 < gpGap_i(t) = \text{goodput}/\text{video data throughput} < 1, \forall i \in N$. This feature is measured each second, and it is an indicator of how efficiently a station is using the medium. *Master gpGap* is defined as the same ratio, but it uses total system goodput and video data throughput instead of each stations' goodput/throughput.

In most experiments, one can visually notice a dip in the *Master gpGap* feature around the time of the crash. For example, Figure 5(a) shows such a dip 10 seconds before the network crashes. In some cases, the dip is narrow and deep, but in other cases, it is wide and shallow. We track the correlation between the *gpGap* statistics for three cameras at a time. The origin in this space indicates when *gpGap* is jointly zero for all 3 stations, so normal operation should ideally be as far as possible from the origin. We are interested in the minimum distance point to the origin, representing in 3-space, the lowest point in *gpGap* for all 3 cameras. Similarly, we look at the maximum derivative points of the *gpGap* statistic. We expect there to be two maximum derivative points - one highly negative point (formation of the dip) and one highly positive point (termination of the dip).

We show that *joint gpGap* statistics clearly indicate the evolution of the system state. In Figure 5(b) and (c), we present the *joint gpGap* statistics for cameras 2, 3, and 4, and their derivatives. The color bar to the right of each plot indicates the time dimension. In this experiment, we add a new station to the network every 10 seconds, so $0 < t < 10$ corresponds to one active camera, $10 < t < 20$ corresponds to two active cameras, etc.

Figure 5(b) shows three distinct clouds of operation: normal, prediction zone (a crash may be imminent), and crashed. The crashed state corresponds to the red points after the sixth camera becomes active, and the Snowball Effect takes its toll. In Figure 5, there are a series of points on both the left and right sides of the color bar. On the left side, we have marked points in time that are within $\epsilon = 0.075$ of the minimum distance point. Notice that the black minimum distance points are distributed most densely near the "prediction cloud" region marked on the color bar. On the right-hand side of the color bar, we have marked the points that are within $\epsilon = 0.40$ of the maximum derivative. One highly negative derivative point appears early in the prediction phase, indicating the formation of a dip. A highly positive maximum derivative point appears at the end of the prediction phase, completing the formation of a dip in the *gpGap* plot. Overall, the transition of the *jointgpGap* statistic in the scatter plot indicates if the system is operating in a healthy regime.

V. RELATED WORK

Although the IEEE 802.11 standard allows flexibility regarding the selection of transmission parameters, such as PHY rate, it does not explicitly specify when and how to use specific settings. Early link adaptation techniques like [11] select the corresponding PHY rate according to a goodput maximization criteria. Alternatively, the Auto Rate Fallback (ARF) algorithm [10], and subsequent approaches [12] adapt rates depending on the frequency of contiguous transmission failures or successes.

Several authors have recently observed flaws in the rate adaptation mechanisms implemented in present hardware ([1], [2]). If a rate adaptation algorithm assumes that failures are due to a bad channel when they are really due to collisions, then it might unnecessarily lower the PHY rate, which causes diminished throughput, among other serious problems (as in ARF, [10]). Since detecting *channel problems* (e.g. path-loss, interference) is difficult in practice, [1] proposes a mechanism to detect *collisions*. In this mechanism, stations exchange transmission time information (piggybacked onto data packets) when transmission failures occur. However, the detection delay is too large to make this solution practical.

Another scheme, [2], uses RTS/CTS messages to differentiate between frame collisions and frame failures due to channel errors. In this paper, the authors recognize that an RTS frame transmission is not at all likely to fail due to a bad channel (because of its small size and robust transmission modulation), and hence use it as an indicator of collision vs. bad channel. Simulations showed throughput improvement, but

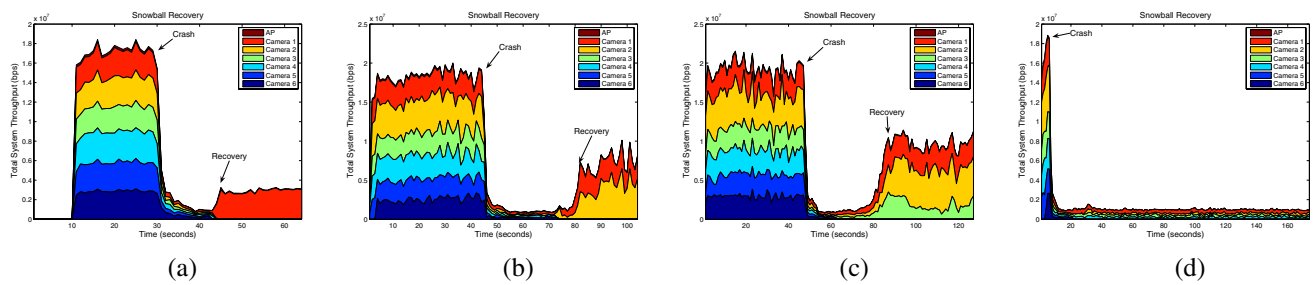


Fig. 4. Recovery from the Snowball Effect by intentionally stopping a subset cameras. (Recovery time is: (a) 12 sec. when 5 of 6 cameras are stopped after the crash; (b) 24 sec. when 4 of 6 cameras are stopped after the crash; (c) 54 sec. when 3 of 6 cameras are stopped after the crash; (d) System never recovers when 2 of 6 cameras are stopped after the crash.

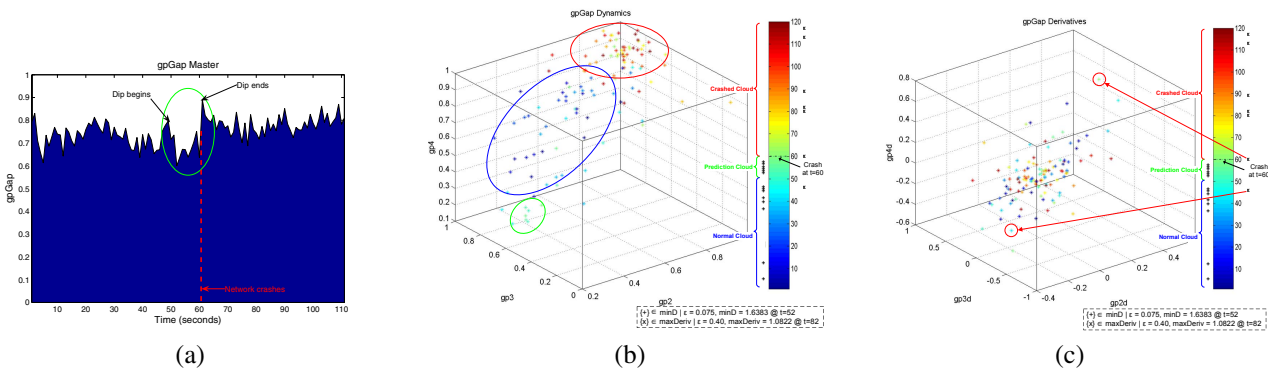


Fig. 5. The gpGap Statistic (a) may be a useful statistic for predicting catastrophic failure. Its time evolution (b), and derivatives (c) are shown in the plot.

the technique suffers from overheads, and the algorithm affects the fairness of the DCF mechanism itself.

A third adaptation scheme is SNR-based rate adaptation, where the algorithm estimates the SNR and adjusts the modulation scheme accordingly. Although this approach is impressive in theory, it faces serious practical shortcomings like long estimation delays and difficult exchange of signal information. Delays become particularly critical in the case of real-time voice/video streaming applications, where the link conditions rapidly fluctuate due to mobility or interference. To address this problem, [3] proposes a hybrid SNR-based rate adaptation mechanism. However, this approach is limited by the very accuracy of the estimation possible in a closed-loop, low delay system - especially in the absence of standardized channel condition measurements (e.g. 802.11k).

VI. CONCLUSION

The Snowball Effect is the result of an incorrect assumption made in current rate adaptation mechanisms. We showed that current rate adaptation schemes fail because they do not differentiate between poor channel conditions and collisions as the source of transmission failures, and consequently invoke improper responses that cascade to dramatic throughput degradation.

Based on our analysis, we recommend that a next generation rate adaptation algorithm should have the following requirements: (1) It should not modify the physical layer; (2) It should not react to *all* collisions or *specific* statistics at a fine granularity - this is overly complex and not needed; (3) Instead, the algorithm should track how close the system is to saturation, and adapt the rate according to the state the system

is currently operating at; (4) The algorithm should not attempt to exploit fine-grain IEEE 802.11 parameters.

The general problem of *when* such adaptation should take place, and in what way is the topic of some of our current and future research.

REFERENCES

- [1] J. Yun and S. Seo, "Collision Detection based on Transmission Time Information in IEEE 802.11 Wireless LAN," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW)*, 2006.
- [2] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs," in *Proceedings of the IEEE Conference on Computer Communications (Infocom)*, 2006.
- [3] I. Haratherev, J. Taal, K. Langendoen, R. Lagendijk, and H. Sips, "Automatic IEEE 802.11 Rate Control for Streaming Applications," *Wireless Communications and Mobile Computing*, vol. 5, no. 4, pp. 421–437, 2005.
- [4] "AiroPeek SE, WildPackets, Inc." <http://www.wildpackets.com/>.
- [5] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," in *Proceedings of the IEEE Conference on Computer Communications (Infocom)*, 2003.
- [6] M. Garetto and C.-F. Chiasserini, "Performance Analysis of 802.11 WLANs Under Sporadic Traffic," in *NETWORKING*, ser. Lecture Notes in Computer Science, R. Boutaba, K. C. Almeroth, R. Puigianer, S. X. Shen, and J. P. Black, Eds., vol. 3462. Springer, 2005, pp. 1343–1347.
- [7] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.
- [8] ANSI/IEEE Std 802.11, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.
- [9] "The network simulator - ns2," <http://www.isi.edu/nsnam/ns/>.
- [10] A. Kamerman and L. Monteban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, 1997.
- [11] D. Qiao and S. Choi, "Goodput enhancement of IEEE 802.11a wireless LAN via link adaptation," in *Proceedings of the IEEE Conference on Communications (ICC)*, 2001.
- [12] P. Chevillat, J. Jelitto, A. Barreto, and H. Truong, "A dynamic link adaptation algorithm for IEEE 802.11 a wireless LANs," in *Proceedings of the IEEE Conference on Communications (ICC)*, 2003.