ABSTRACT

SHAH, NIPUL JAYVANT. Preventing Denial of Service Attacks on Reliable Multicast Networks. (Under the direction of Dr. Douglas Reeves.)

Multicast is finding a lot of application in modern day networks and the Internet. There are various existing protocols that support the wide range of requirements demanded by these applications. If all the receivers in a multicast group are required to get all the packets at more or less the same time (i.e. synchronized reliable receiving), then the transmission rate of the source ends up being controlled by the rate of the slowest receiver in this group. Although, this is a requisite in some applications, it poses as a serious threat to the group. In other words, if one or more receivers were to artificially create a packet loss, then the source would be busy sending repairs and will consequentially slow down the overall transmission rate. This leads to a Denial of Service attack on the other group members.

The goal of this thesis is to suggest a mechanism to deter, if not prevent, the hostile receiver(s) from causing such an attack. We first study the problem with respect to a specific reliable multicast protocol, viz. Pragmatic Generic Multicast (PGM), by conducting experiments, which prove that PGM is also affected by the ‘slowest receiver problem’. If the source can work out an optimum transmitting rate, we may be able to reduce the repair requests in the network and have a more stable system. To achieve this, we look at the possibilities and advantages of using an auction-based mechanism, such as the Generalized Vickrey Auction (GVA) to compute the optimum rate, based on the rate requests from the various

participating receivers. We implement our mechanism in PGM and conduct experiments in order to compare its performance to that of the existing PGM protocol. Our results prove that for a network having malicious members, an appropriate auction-based mechanism complemented with policing stabilizes the source transmission rate and hence prevents a Denial of Service attack on other group members.

PREVENTING DENIAL OF SERVICE ATTACKS
ON
RELIABLE MULTICAST NETWORKS

by

NIPUL SHAH

A thesis submitted to the Graduate Faculty of
North Carolina State University
In partial fulfillment of the
requirements for the Degree of
Master of Science

COMPUTER ENGINEERING

Raleigh
Dec 2002

APPROVED BY:

_____
Dr. Douglas Reeves
Chair of Advisory Committee

_____
Dr. Peng Ning
Committee Member

_____
Dr. Peter Wurman
Committee Member

BIOGRAPHY

Nipul Shah was born on November 03, 1978 at Ahmedabad in Gujarat, India. He had been residing in Bombay, India since birth. He completed his schooling at Fatima High School, Bombay (1982-1994) and higher secondary at Swami Vivekanand's Junior College, Bombay (1994-1996). He obtained a Bachelor's degree in Electrical and Computer Engineering from Vivekananda Education Society's Institute of Technology, affiliated to Bombay University, Bombay (1996-2000).

In search of a career in Computer Networking, he came to study at NCSU, Raleigh, North Carolina, USA. He got a Master of Science degree in Computer Engineering from NCSU (2000-2002).

ACKNOWLEDGEMENTS

This thesis was made possible by loads of inspiration, motivation, guidance and patience from my advisor, Dr. Douglas Reeves. I would like to extend many thanks to him.

I would also take this opportunity to thank my family and friends for their endless support and understanding. And also, special thanks to:

* Ashish Sureka, for helping with Auction-based mechanism
* Sherlia Shi, for help with the PGM module in ns-2
* My committee members: Dr. Peng Ning and Dr. Peter Wurman.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACK	Acknowledgment: A message sent by the receivers to the source on receiving the data.
AWD	Advance with data: One of the flow control mechanisms used by the source in PGM.
AWT	Advance with time: Another flow control mechanism used by the source in PGM.
DLR	Designated Local Repairer: An element in the local network that responds to the repair requests instead of the source.
GSRM	Generic Scalable Reliable Multicast: A reliable multicast transport protocol.
GVA	Generalized Vickrey Auction: An efficient and incentive-compatible auction mechanism.
IP	Internet Protocol: A network layer protocol.
ISP	Internet Service Provider: An ISP is a company that provides access to the Internet.
MFTP	Multicast File Transfer Protocol: A reliable multicast transport protocol.
NAK	Negative Acknowledgment: A message sent by the receiver to the source on detecting lost data.
NCF	NAK Confirmation: A message sent by the NE to the downstream NE or to the receiver from which it received a NAK.
NE	Network Element: Switches, routers, firewalls, etc.
NLA	Network Layer Address: The address of the interface of the network element, e.g. IP address, IPX address, etc.
NNAK	Null Negative Acknowledgment: A message sent by the DLR to the source every time it sends out a repair data.
ODATA	Original Data: Data transmitted by the source as part of the multicast session.

OSPF	Open Shortest Path First: A routing protocol used in networks to carry out routing updates.
PGM	Pragmatic Generic Multicast: A reliable multicast protocol defined by RFC 3208.
RDATA	Repair Data: Data transmitted by the source or DLR in response to a repair request.
RIP	Routing Information Protocol: A routing protocol used in the networks to carry routing updates.
RMP	Reliable Multicast Protocol: A reliable multicast transport protocol.
RMTP	Reliable Multicast Transport Protocol: Another reliable multicast protocol.
SPM	Source Path Message: A message transmitted by the source, interleaved between data to maintain state information in the NEs and receivers.
SPMR	SPM Request: A PGM option used by the receiver to request a SPM transmission from the source.
TSI	Transport Session Identifier: A unique ID for each multicast session.

1. INTRODUCTION

1.1. Overview of thesis:

Reliability and Synchronization are two of the many attributes of a multicast session that may be desired based on the requirements of an application. If the application desires synchronized reliable receiving, the source will need to transmit at a constant rate to all the receivers for synchronization and utilize some bandwidth for providing repairs for reliability. Thus, if the repair bandwidth increases, the overall throughput of the transmission will reduce if reliability is to be maintained. In other words, the slowest receiver controls the source transmission rate in a reliable multicast session. This is what we call as the Slowest Receiver Problem.

In order to conduct experiments, we work with a specific reliable multicast protocol, viz. Pragmatic Generic Multicast (PGM). We study how the flow control mechanism in PGM leads to the slowest receiver problem. To demonstrate this, we conduct several tests on PGM using a simulator. The results show that if a single receiver is sending NAKs, the source is forced to slow down to a rate driven by this slowest receiver. Now, if multiple receivers are dropping packets and sending repair requests (NAKs) or sending false NAKs even though they have received the corresponding data packet, then reduction in throughput is of much greater magnitude. To resolve this problem, we suggest the use of an auction-based mechanism such as the Generalized Vickrey Auction (GVA), used along with some rate policing done either at the source or network elements to select the receivers that are to participate in the reliable multicast session. We then propose how this can be implemented in PGM

and present results after conducting some experiments on our mechanism. Results from our experiments show that an auction-based mechanism, used in conjunction with some kind of rate policing, allows synchronized reliable receiving for all complying receivers at an unchanged throughput.

1.2. Advantages of multicast over unicast:

Communication between hosts in a computer network can be divided in to three categories: unicast, multicast and broadcast.

Unicast implies a one-to-one communication between hosts, broadcast refers to a one-to-all communication between hosts and multicast corresponds to a one-to-many (but not all) communication between hosts. The difference between broadcasting and multicasting is that in broadcasting, packets are delivered to all the hosts in the given network, while in multicast, packets are delivered to some specific group of hosts which have subscribed to receive them.

One may consider multicast to be similar to many unicast connections at one time. But that is exactly how multicast not only differs from unicast, but also has a big advantage over it. If unicast were used to serve one-to-many connections, the sender would have to operate many connections and send the same packet over all the connections, which leads to inefficiency. On the other hand, if multicast is used, the sender sends just one packet to the group. This packet is duplicated by the network elements (switches, routers, firewalls, etc) as and when required. Thus the two main advantages of using multicast over unicast are reduction in the bandwidth used and a decrease in the source load.

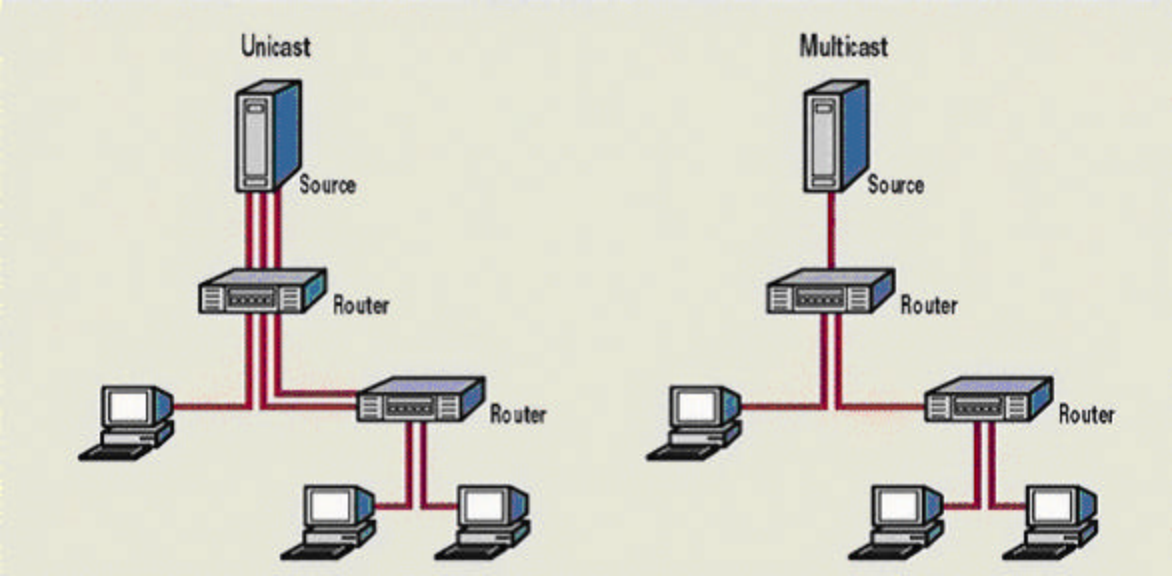

Fig 1: Unicast vs. Multicast: Unicast sends a separate stream to each receiver, while multicast sends one stream that is separated at the multicast routers on the way to the destinations [4].

1.3. Multicast applications:

Multicast is useful because it allows the construction of truly distributed applications, and provides important performance optimizations over unicast transmission. There is currently an experimental Multicast Backbone, called the Mbone[23], which is exploring applications of IP multicast. The most widely used multimedia conferencing applications are the MBone freeware applications that provide audio, video, and electronic whiteboard and session directory services [7].

Some other multicast applications:

- News/sports/stock/weather updates
- Distance learning
- Routing updates (OSPF, RIP etc)
- Pointcast-type “push” apps
- Videoconferencing, shared whiteboards
- Distributed interactive gaming or simulations

- Email distribution lists
- IPv6 over IPv4
- Voice-over-IP
- Database replication
- Distribution of broadcast TV channels

1.4. Research in reliable multicast:

Certain group communication applications, such as interactive audio/video conferencing, use unreliable multicast transmission. This is because such applications can tolerate some data loss, but cannot tolerate the delays caused by waiting for the retransmission of missing data. Other applications, such as situation awareness and replicated file servers, require reliable multicast transmission [1]. Many different reliable multicast protocols have been developed to meet the differing needs of their applications. Most reliable multicast protocols are optimized for performance and are robust to common faults, such as lost packets and failure of one or more group members.

The Transmission Control Protocol (TCP) meets the general requirements of reliability and ordered packet transmission for unicast. However, no such general purpose protocol exists for multicast since the different multicast applications have varied reliability requirements. As such reliability techniques can be divided into 2 basic categories:

Sender-initiated reliability:
Sender assumes the role of loss detection and expects to receive an ACK (positive acknowledgement) from every receiver for every packet it sends. Any missing ACK

corresponds to a lost packet and the sender takes appropriate action. Thus, the sender requires knowledge of all the receivers. The frequency of ACKs is fixed at at-least the rate at which the transmit window is advanced, and usually more. ACKs primarily determine transmit buffer management. When using multicast, the problem of reliability is not as easy to solve as it is in the unicast case, where the sender of the data keeps track of how much data it has sent and the receiver reports back by way of acknowledgments, how much it has received. This procedure unfortunately does not scale to a large number of receivers, as the number of acknowledgments sent back to the original sender would increase linearly with the number of receivers. This is called the ACK-implosion problem [5][22]. It keeps the ACK-based reliable multicast protocols from scaling well.

Receiver-initiated reliability:
The responsibility of detecting lost packets is left up to the receiver. It generates a NAK (negative acknowledgement) on detecting a lost packet and on receipt of a NAK the sender takes appropriate action. The sender no longer requires to maintain information about the multicast group members. The frequency of NAKs is a function of the reliability of the network and the receiver's resources (and so, potentially quite low) [2]. NAKs primarily determine repairs and reliability. Unfortunately, a NAK implosion is also possible if the group is large and the packet is lost near the original sender, as this would result in a large number of packets being sent almost simultaneously to the original sender. In such cases where correlated losses occur, suppression mechanisms can be used to minimize the number of duplicate NAKs produced. Similar suppression mechanisms can also be used to prevent a flood

of retransmissions when any member with the appropriate data may respond to a NAK.

1.4.1. Flow-control in reliable multicast:

As described above, reliable multicast is either ACK-based or NAK-based. In general, the flow control schemes in these mechanisms can be described as:

- ACK-based flow control scheme: Here a window based flow control (as in TCP) can be applied. In a window based flow control scheme, the sender has a fixed size window that is not larger than any receiver's receiving buffer. When a receiver correctly receives a packet, it sends an ACK for the corresponding packet to the sender. The sender uses a sliding-window algorithm and slides the send-window when ACKs for a packet from all the receivers arrive at the sender.
- NAK-based flow control scheme: In this case the window flow control cannot be applied. A rate based flow control scheme is deemed more suitable. In a rate based flow control scheme, the sender adjusts its transmission rate based on NAKs it receives from receivers. Now, if the sender simply reduces its transmission rate whenever a NAK arrives, the transmission rate becomes too much regulated. Thus, a suitable algorithm should be used to select the NAKs that would affect the transmission rate.

A comparison of the performance of ACK-based and NAK-based flow control schemes for reliable multicast can be found in [16].

1.5. Reliable multicast protocols:

Reliable multicast transports have been a subject of research for a number of years already. As a result, there exist a significant number of protocol implementations (and their variants) already. Many of these multicast transports are very useful, operate well and have long-standing and successful operational records. In particular, MFTP[17] from the StarBurst Communications Corporation and the suite of reliable multicast protocols RMP[18], RMTP[20] and PGM[2] - offered by the GlobalCast Communications, Inc. (now owned by TIBCO software)- are used by companies around the world on their multicast enabled networks. A survey of a number of the existing reliable multicast protocols can be found at [5].

1.6. Application layer multicast:

Although it has been over a decade since IP multicast was proposed, it is still in limited use due to various reasons. Some of these are described in [8] as:

> First, IP Multicast requires routers to maintain per group state (and in some proposals per source state in for each multicast group). The routing and forwarding tables at the routers now need to maintain an entry corresponding to each unique multicast group address. However, unlike unicast addresses, these multicast group addresses are not easily aggregatable. This increases the overheads and complexities at the routers.
> Second, there is a dearth of experience with additional mechanisms like reliability and congestion control on top of IP Multicast, which makes the ISPs wary of enabling multicasting at the network layer. Although there exists proposals for such mechanisms over IP Multicast, the impact of these solutions on the wide-area Internet is not clear. Congestion control for multicast applications acquires far greater importance than the unicast case, and therefore, needs to be well understood before wide scale deployment.
> Third, the pricing model for multicast traffic is not yet well defined.

Recently there has been some research done in moving the multicasting functionality from the network layer to the application layer since it is easier to deploy over existing networks. Some of the existing research work has been described in [8]. This paper does a comparative study of some of the existing Application Layer Multicast protocols.

1.7. Organization of thesis

In chapter 2, we describe a particular reliable multicast protocol that we work with in this thesis, viz. Pragmatic Generic Multicast (PGM). After describing the basic principles and concepts of PGM, we proceed to discuss in Chapter 3 the Slowest Receiver Problem and study its effect on PGM. Having understood the problem, we explain the principles of a solution to the Slowest Receiver Problem in Chapter 4. It covers the use of an auction-based mechanism in a reliable multicast scenario. Chapter 5 comprises of the implementation details of this solution as applied to PGM. This implementation is then validated experimentally and the results are presented in Chapter 6. An alternate solution, which allows a tradeoff to be made between throughput and reliability, is described along with other experimental results in Chapter 7. Chapter 8 contains the conclusions.

2. PRAGMATIC GENERIC MULTICAST

To understand the Slowest Receiver problem, we need to conduct some experiments on reliable multicast that we can analyze. And again, after proposing a solution, we need to validate it experimentally. To do so, we need to understand some basic principles of any one reliable multicast protocol that we shall be experimenting with. From the various existing reliable multicast protocols providing reliability on top of network-layer multicast, we describe Pragmatic Generic Multicast (PGM) below, as PGM has attracted industry interest, advocated by Cisco, TIBCO software and Talarian.

PGM is a reliable multicast transport protocol for applications which require ordered, duplicate-free, multicast data delivery from multiple sources to multiple receivers. It guarantees that a receiver in a multicast group either receives all the data packets from the transmissions and retransmissions, or can detect an unrecoverable data packet loss. Thus, PGM is intended as a solution for multicast applications with basic reliability requirements. It is network layer-independent.

Details on the working of PGM can be found in [2]. Below we mention some basic concepts that will be required for understanding the material that will be presented in chapters ahead.

2.1. Operation summary:

PGM runs over a datagram multicast protocol such as IP multicast. The source sends sequenced data packets (ODATA), and the receivers send selective negative acknowledgements (NAKs) of packets they deem to have been lost. Since NAKs

provide the sole mechanism of reliability, they are further acknowledged by the network elements by sending NAK confirmations (NCFs). On receiving NAKs, the source sends out the repair data with the corresponding sequence number (RDATA). To establish source path state in network elements, the source also sends Source Path Messages (SPMs) periodically.

2.1.1. Terms and concepts:

Before we explain the procedures of various elements involved in PGM, this sub-section details the terms and concepts that will be used ahead.

Transport Session Identifiers (TSI):
TSIs are globally unique identifiers for each transport session. Every PGM packet is identified by a TSI.

Sequence Numbers:
These are used to identify and order ODATA packets. PGM uses a circular number space from 0 through (2^{32}-1) to generate them.

Transmit Window:
The source maintained transmit window corresponds to the amount of the transmitted data retained by the source for repair. The trailing edge of the transmit window represents the sequence number of the oldest data packet available for repair from a source, while the leading edge represents the most recent data packet transmitted by the source.

Window Increment and Increment Window (see Fig. 2):
The fraction of the transmit window by which the transmit window is advanced is called as the Window Increment. And the oldest such fraction or trailing fraction of the

transmit window itself is called as the Increment Window. In terms of sequence numbers, the Increment Window corresponds to the range of sequence numbers that will expire first when the transmit window advances.

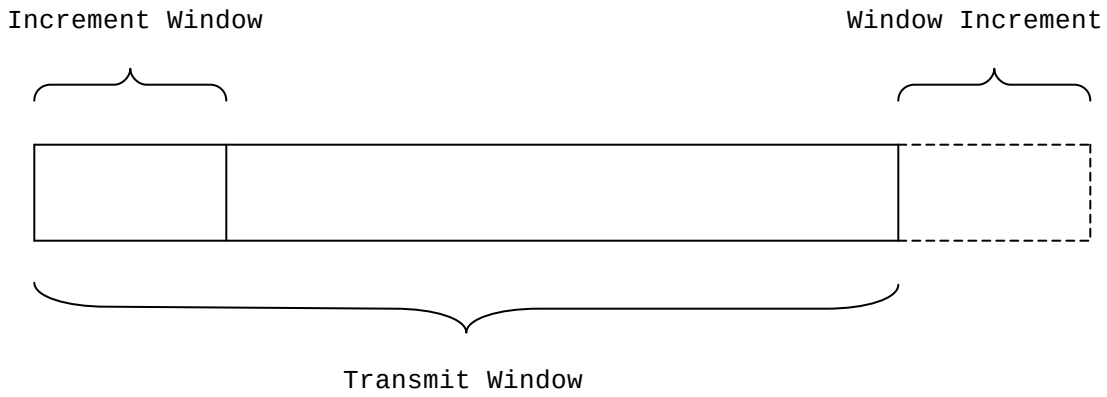

Fig 2: Windows in PGM: Transmit window is the range of packets retained by the source for repairs. Window Increment is the fraction of transmit window by which transmit window is advanced. Increment Window is the range of packets that first expire when transmit window advances.

Source Path State:

PGM network elements require the source path state to forward NAKs upstream on the reverse of the distribution tree. The source path state is simply the address of the upstream PGM hop on the reverse distribution tree.

Lost packets:

PGM receivers check the sequence numbers of ODATA packets received and by comparing them, they can detect any gaps in received data. These correspond to lost packets. The PGM receiver, on detecting lost packet(s) attempts to obtain them from the source by transmitting repair requests, i.e. NAKs.

Missed packets:

PGM receivers send NAKs for any lost packets that they detect. If however, on repeated repair requests, they do not receive the repairs and transmission window is advanced

beyond the sequence numbers of the missing packets, these packets are considered to be permanently lost, i.e. these packets are considered to be missed packets.

2.1.2. PGM packet types:

Source Path Messages (SPM):
These are transmitted by the source to maintain source path state in the PGM NEs and to provide transmit window state to the PGM receivers. SPMs are multicast to the group.
They contain the TSI, a SPM sequence number, the trailing edge of the source's transmit window, the leading edge of the source's transmit window and the network layer address (NLA) of the interface of the PGM NE on which the SPM is forwarded.

Original Data (ODATA):
They are the data packets, containing application data, transmitted by the source to the receivers. ODATAs are also multicast to the group.
They contain the TSI, the trailing edge of the source's transmit window and a data sequence number.

Repair Data (RDATA):
They are the repair packets that are transmitted by the source or the DLRs in response to the repair requests received. RDATAs are multicast to the group.
They contain the TSI, the trailing edge of the source's transmit window and the ODATA sequence number of which it is a repair.

Negative Acknowledgments (NAK):
These are transmitted by the PGM receivers upon detecting a missing data sequence number. They are sent to the source

to request repairs. NAKs are unicast PGM-hop by PGM-hop to the source.
They contain the TSI, sequence number of the missing ODATA, the unicast NLA of the source and the group multicast NLA.

Null Negative Acknowledgments (NNAK):
They are sent by the DLR to the source to provide the flow control feedback, for every repair requests they receive from receivers or NEs. NNAKs are unicast PGM-hop by PGM-hop to the source.
They contain the TSI, sequence number of the missing ODATA, the unicast NLA of the source and the group multicast NLA.

Negative Acknowledgment Confirmations (NCF):
They are transmitted by the network element and the source in response to NAKs received. NCFs are multicast to the group.
They contain the TSI, the sequence number present in the NAK that is being confirmed, the unicast NLA of the source present in the NAK and group multicast NLA present in the NAK.

2.1.3. Source functions:

- Data transmission.
  The source transmits ODATA packets only within the transmit window, at a rate no greater than the ‘maximum cumulative transmit rate’. Different transmission strategies define this maximum rate as being appropriate for the implementation. Also a source must strictly prioritize sending of pending NCFs first, pending SPMs second, and only send ODATA or RDATA when no NCFs or SPMs are pending. The priority of RDATA versus ODATA is application dependent.

- Source Path State.
  The source also transmit/multicast SPMs interleaved between ODATA and RDATA packets (Ambient SPMs), at a rate which is at least sufficient to maintain the source path state in the PGM NEs. In the absence of data to transmit, the source transmits SPMs at a decaying rate (Heartbeat SPMs) to maintain state information in the NEs and in the receivers.
- Negative reliability.
  A source must immediately multicast an NCF in response to any NAK it receives.
- Repairs.
  After multicasting an NCF in response to a NAK, a source must then multicast RDATA in response to any NAK it receives for data packets within the transmit window.
- Transmit window advance: Sources advance the trailing edge of the transmission window based on one of the many strategies. Some of these are described ahead in this chapter.

2.1.4. Receiver functions:

- Data reception.
  For a given transport session, the receiver accepts any ODATA or RDATA received within the receive window (the receive window is a copy of the source’s transmit window, maintained at each receiver). It discards any duplicates or packets outside the receive window.
- Source path state: Receivers use SPMs to determine the last-hop PGM network element for a given TSI to which to direct their NAKs. Also a receiver cannot initiate a repair request until it has received at least one SPM for the corresponding TSI.

- Data recovery.
  By comparing the sequence number of the most recently received ODATA or the leading edge value in the most recently received SPM, with the leading edge of contiguous data, a receiver can detect missing packets. If it does, the receiver initiates a NAK generation, for each missing packet, to the last-hop PGM network element. NAK initiation consists of setting up a repair state at the receiver and starting a back-off timer. If this timer expires without receiving any matching NCF or NAK (probably transmitted by an other receiver in the group), the receiver unicasts the NAK.
  On transmitting a NAK, the receiver activates 2 timers; one for a shorter period which waits for the corresponding NCF from the upstream NE/source (pending NAK state), and the other for a longer period which waits for the corresponding RDATA to arrive from the source (outstanding NAK state). Upon expiry of any of these timers, the receiver retransmits the NAK.
  Receipt of corresponding RDATA cancels the repair state for that sequence number.
  The receiver cancels NAK generation for any pending or outstanding NAKs on the advancing of the receive window.
- Receive window advance.
  Receivers immediately advance their receive windows upon receipt of any PGM data packet or SPM within the transmit window that advances the receive window.

2.1.5. Network element (NE) functions:

- Source path state.
  NEs use SPMs to establish source path state for the corresponding session. They then forward them on each

outgoing interface, and while doing so include the NLA of the outgoing interface in the corresponding SPM.

- NAK reliability.
  For every NAK received, NEs immediately multicast a NCF on the interface on which the NAK was received and maintain a repair state, viz. the sequence number of NAK, the input interface on which NAK was received, the session identifier, etc.
- Constrained NAK forwarding.
  NAK forwarding rules are very similar to those used for the receivers. The differences lie in:
  NEs do not backoff. They immediately forward the first NAK to the upstream PGM NE.
  NEs do not retry NAKs on expiry of the no-RDATA timer, if the NAK has already been confirmed by the upstream PGM hop. They rely on the receivers to re-attempt the repair request.
  ODATA cannot cancel NAK state in NEs as in the receivers, since ODATA are switched by the NEs without transport layer intervention.
- NAK elimination.
  Two NAKs having the same session identifier and the same sequence number are considered to be duplicates. NEs discard any duplicate NAKs received if a repair state already exists for that NAK, i.e. if that NAK has already been forwarded upstream.
- Constrained RDATA forwarding.
  From the NAKs received, NEs maintain a repair state, which consists of a list of interfaces on which a NAK was received. When RDATA is received, it checks this list of interfaces for the corresponding NAK and forwards the RDATA only to these interfaces. Thus, the repairs are constrained only to the interested subset of the network.

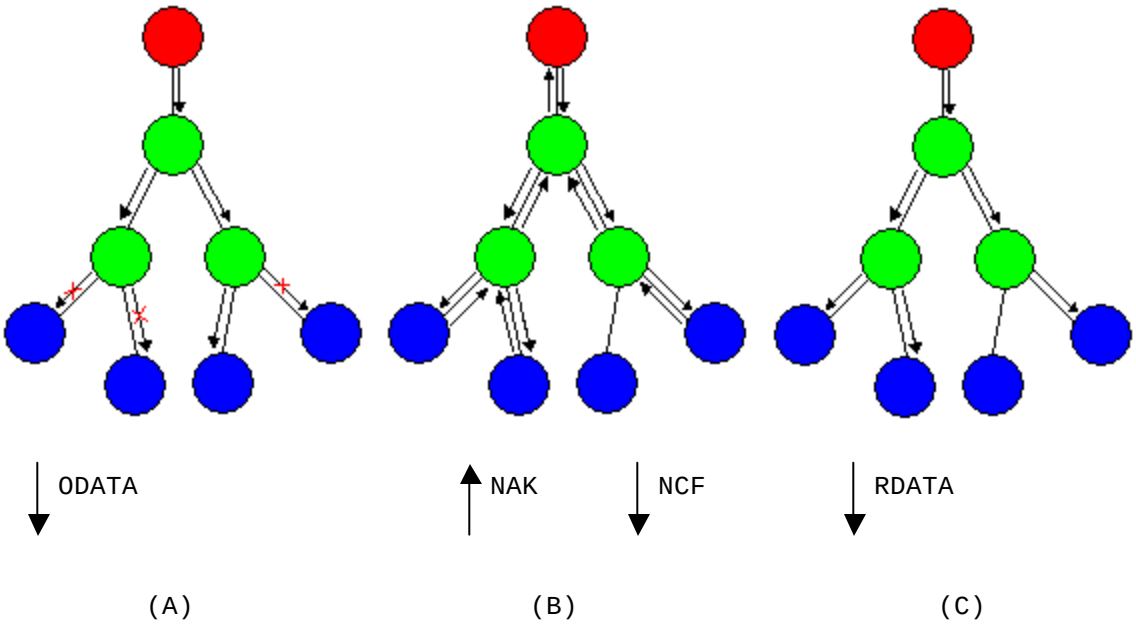

Fig 3: Data transmission in PGM: Fig(A): ODATA is multicast by the source. 3 of the ODATA packets are dropped/lost during transmission. Fig(B): On detecting a missing packet, the 3 receivers unicast a NAK to the source, PGM-hop by PGM-hop. The NE uses constrained NAK forwarding, i.e. only the first NAK received is forwarded upstream. All similar NAKs (for same sequence number) are constrained. However, the NEs and the source, multicast a NCF immediately on the interface on which the NAK was received. Fig (C): On receiving a NAK, the source multicasts a corresponding RDATA to the group. The NEs use constrained RDATA forwarding, i.e. RDATA is forwarded only on the interfaces that have a repair state.
Red: PGM Source, Green: PGM Network Elements, Blue: PGM Receivers.

2.2. Flow control mechanisms:

The PGM protocol does not constrain the strategies that a source uses to advance the transmit window. However, the related RFC suggests two mechanisms that are described in brief below:

2.2.1. Advance with time (AWT):

The transmit window is advanced in real-time. All the timers are real-time based. The maximum transmission rate of the source is calculated from SPMs and ODATA only, while the repairs consume any extra bandwidth that may be

available. E.g., if the available bandwidth for the source is 1 Mbps, while the maximum transmission rate is 0.5 Mbps, then a combined rate of ODATA and SPMs is maintained at 0.5 Mbps. The RDATA and NCFs use the remaining 0.5 Mbps.

The AWT strategy is suggested only for real-time streaming applications where receiving data on time is more important than completeness. NAKs from receivers do not affect the data transmission rate, thus there is no feedback. Such a mechanism is not affected by the ‘slow receiver problem’ (described in the next chapter), as source maintains a constant data throughput irrespective of the number of NAKs received, at the cost of lower reliability. From our example above, a constant data rate of 0.5 Mbps is maintained by the source with SPMs interleaved with the data packets. And the source can only send the RDATA and NCFs at a maximum rate of 0.5 Mbps. Thus, if the source receives too many repair requests, the source buffer may soon overflow leading to losses.

2.2.2. Advance with data (AWD):

In this strategy the maximum transmit rate of the source is calculated from the SPMs and from both the ODATA and RDATA. Any excess available bandwidth is used only for NCFs. The timers are not real-time as in ‘advance with time’, but are data-driven. Using the same example as above, i.e. if the available bandwidth for the source is 1 Mbps, while the maximum transmission rate is 0.5 Mbps, the combined rate of SPMs, ODATA and RDATA is not to exceed 0.5 Mbps. The excess 0.5 Mbps is used for transmission of NCFs. SPMs are always sent with a higher priority. But the priority in sending ODATA and RDATA is left to the application.

Unlike the ‘advance with time’ mechanism described above, NAKs received for any ODATA sequence number that lies within the increment window resets the transmission window advance interval, i.e. the transmission window advance timer is reset. Thus, NAKs received by the source affect the source rate if they are for packets in the earlier part of the transmission window. If the source receives NAKs in the increment window, this implies that one or more receivers are lagging quite a bit since increment window is the oldest fraction of the transmission window. Thus, the source rate is too high for one or more receivers. If the source receives too many NAKs in the increment window, the advance of the transmission window is delayed accordingly. Thus, flow control is provided by the AWD mechanism, which allows the source to take the receiver’s receiving capabilities into consideration. The maximum transmission rate used by the source to send ODATA, RDATA and SPMs is not actually reduced. Thus, rate control is obtained indirectly by delaying the transmit window advance, rather than by directly affecting source transmission rate. Strong reliability can be maintained by this mechanism if RDATA is given higher priority over ODATA while transmitting.

This strategy is intended for non-real-time, messaging applications based on the receipt of complete data at the expense of delay.

2.3. Local repairs:

The PGM protocol specifies various procedures and functions for the source to provide repairs in response to the NAKs. The protocol also specifies options and procedures that permit designated local repairers (DLRs) to announce their availability and to redirect NAKs to themselves rather than to the source. This allows for distributed repair

capabilities. The reader is referred to [2] for more details. Local repair capabilities are not considered in the remainder of this thesis.

Using these concepts of PGM, we need to understand how the Slowest Receiver Problem pertains to PGM. In the next chapter we explain the Slowest Receiver Problem, how it impacts reliable multicast, specifically PGM and present results obtained by conducting tests on the two flow control techniques of PGM, as specified in [2].

3. SLOWEST RECEIVER PROBLEM

In the earlier chapter we covered the basic principles and the operation summary of a specific reliable multicast protocol, viz. Pragmatic Generic Multicast (PGM). In this chapter, we now explain the Slowest Receiver Problem in the case of a reliable synchronized multicast and how it relates to PGM. We then present the results of various experiments demonstrating the effect of the Slowest Receiver Problem on PGM.

3.1. Slowest receiver problem in Reliable Multicast:

A multicast session is similar to having a number of unreliable unicast sessions running at the same time, along with a lot of advantages. When reliability is introduced in the picture, not only does this operation get a lot more complicated, but it also introduces some vulnerabilities and performance issues such as scalability, congestion-control, etc. One specific issue, which is the motivation for this thesis, is the ‘slowest receiver problem’. This problem has been identified and there has been some research to improve this. [10] compares the performance of IP unicasting with IP multicasting in such scenarios, and based on the sender delay chooses to transmit unicast or multicast. [11] And [12] use another approach of excluding the slow member from the group. Yet another approach consists in using a communication protocol with a relaxed reliability criterion thus accepting that some messages are lost [13].

If we use a different rate of transmissions for different groups, we lose out on synchronization between the receivers. What we need is to be able to solve the problem without losing reliability or synchronization, the source

being able to decide an optimum rate to transmit. We introduce pricing of receivers as a solution.

But initially, lets understand the ‘slowest receiver problem’ with regards to the PGM protocol and in the later chapters we explain our mechanism.

In an attempt to attain reliability, the sender sends repair data for every such request that it receives. Sending repairs uses the transmitting bandwidth and depending on the transmission mechanism affects the overall transmission rate of the actual packets. Thus, if reliability is to be achieved, the sender keeps responding to the repair requests from the slowest receiver (along with those from other receivers) and thus the sender moves only as fast as the slowest receiver. While this is a requirement in many applications, this also becomes a major drawback in other multicast applications as faster receivers are forced to wait and accept a low data rate, even though they are willing for the application to send data at a much higher rate.

While this is truly a drawback, a receiver could have varied reasons for being the ‘slowest receiver’. The receiver may be connected by using a slower link, or the receiver may be having a smaller receiving buffer size. Another possibility is for a receiver to take undue advantage of this vulnerability of a reliable multicast session. If a receiver was to generate packet losses by dropping packets intentionally or create a similar attack, it would be sending out many repair requests (NAKs). Thus it is possible for a receiver to overwhelm the source in a reliable multicast session with large number of NAKs and effectively reduce the transmission rate of the source. This leads to a Denial of Service attack to the other

members of the reliable multicast group. If more than one receiver were to create a similar sort of attack, the threat is increased manifold. A larger group of such receivers, sending NAKs at even a low rate, may be capable of reducing the data throughput to a standstill.

We conducted experiments to investigate the impact of different NACK rates on the transmit speed of PGM. These experiments are described below.

3.2. Experimental validation of ‘slowest receiver problem’ in PGM:

The experiments have been conducted using the Berkeley based software, Network Simulator, ns2.1b2. [14], and the original PGM patch used with the simulator was developed by [15].

The original version of PGM in the simulator implemented the ‘advance with time’ transmission window mechanism. After conducting some experiments with that, we implemented the ‘advance with data’ window mechanism. We compare the results from the two mechanisms in the section ahead.

The version of PGM implemented in the simulator supports all general PGM procedures, including at least the following:

Senders:

- Multiple PGM senders on the same network
- RDATA generation
- NAK reliability
- Source Path State generation
- Transmit and increment windows

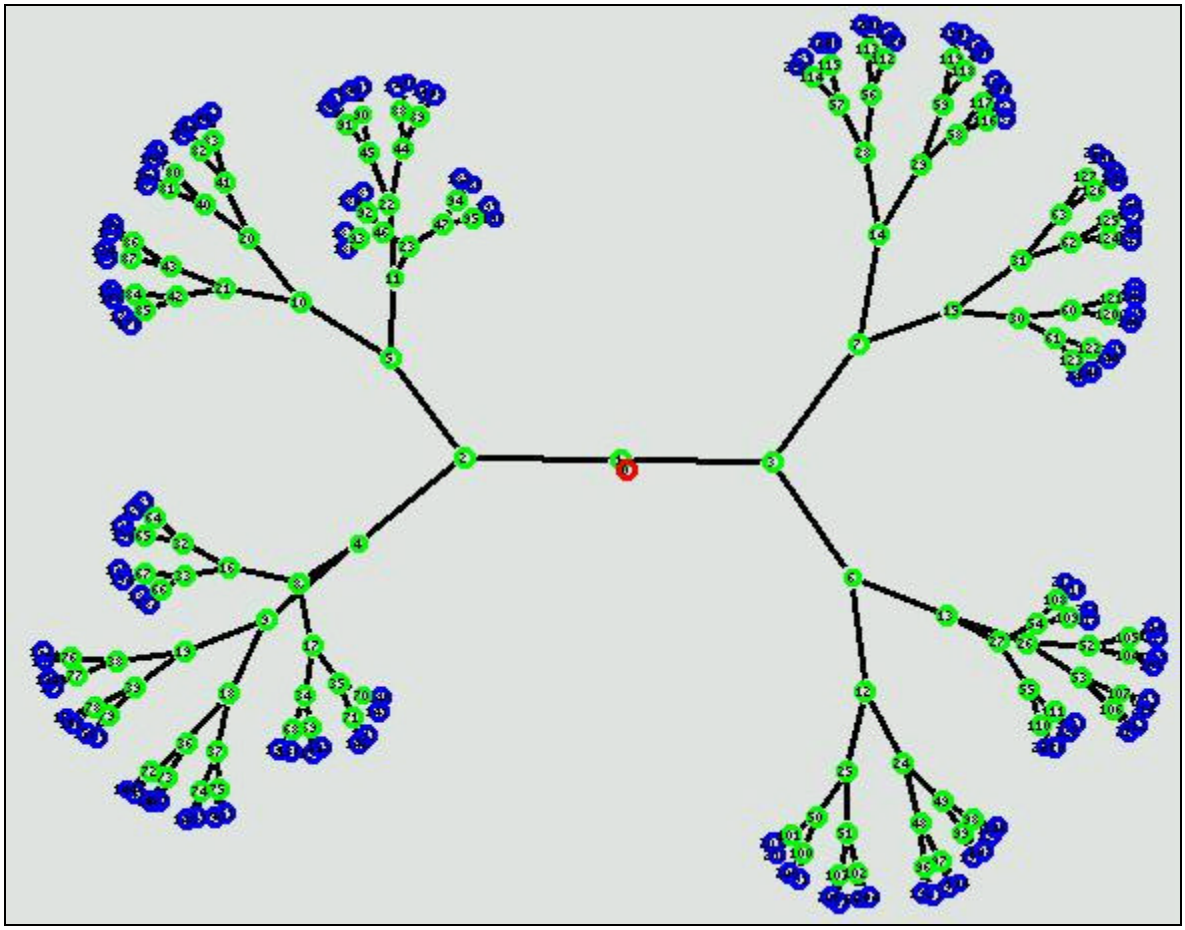

Fig 4: Topology used in all experiments for investigating the impact of NACK rate on the transmission speed of the source.
Blue - receivers, green - network elements, red - source

Network elements:

- Source Path State processing
- NAK reliability
- Constrained NAK forwarding
- NAK elimination
- Constrained RDATA forwarding
- NAK anticipation

Receivers:

- NAK suppression (with random back-off interval)
- NAK reliability
- Receive window

However, it does not support:

- PGM options
- Designated Local Repair (DLR) support

3.3. Experimental results:

The experiments were conducted using a 128 receiver network with a single sender. The topology used is shown in the fig. 4. These 128 receivers corresponds to a small size multicast network. Experimental results for larger networks can be expected to have similar characteristics. Each of the links in the network has a capacity of 1Mbps. It may be possible to have different links with different capacities. However, in order to give every receiver an equal opportunity to receive all the ODATA packets, we select each link of the same capacity. For all the experiments, malicious receivers send NAKs by dropping ODATA packets. For this, we employed a random drop with a pre-set probability. If the receivers would generate false NAKs, i.e. send NAKs though it has reliably received the corresponding data packet, the effect would be the same as far the other elements in the session are concerned. The experiments are conducted with receivers sending NAKs at different rates. These rates correspond to a ratio of data packets for which the receiver attempts/pretends to recover to the total number of ODATA packets sent by the source. We chose the maximum transmission rate of the source as 500 Kbps, thus half the bandwidth is to be used by the regulated rate and the other half, which is the excess bandwidth of the link, could be used for repairs as in AWT or NCFs as in AWD. The links use a drop-tail queuing mechanism, which uses FIFO scheduling and drop-on-overflow buffer management typical of most present day Internet routers, with the queue size for all links as 50 packets.

The source continues transmission until it has sent 5000 ODATA packets, so as to allow the experiments to run for sufficiently long periods of time.

3.3.1. Advance with time

The first set of experiments was conducted using the 'advance with time' transmission window mechanism. To show how different NAK rates from multiple receivers affect the performance of the AWT mechanism, we had 5 receivers dropping packets to generate NAKs. Malicious receivers may also generate false NAKs, i.e. they send NAKs though have reliably received the corresponding data packet. Both cases

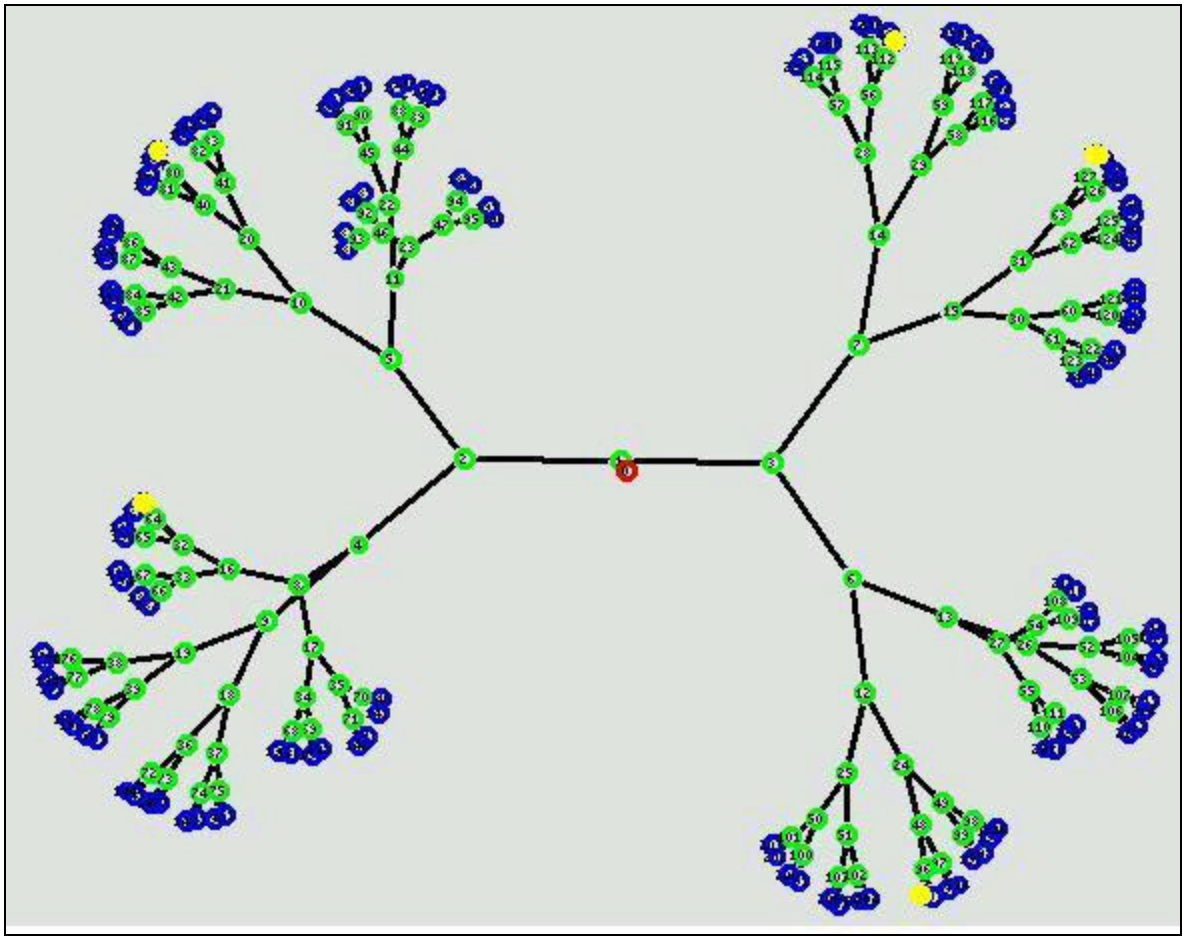

Fig 5: Topology showing the 5 malicious receivers used to test AWT. Blue - receivers, green - NEs, red – source, yellow –dropping receivers.

lead to the same effect on the source, as it is not possible to differentiate these NAKs and the source treats them equally. Thus, in both cases, NAKs have similar effect on the network performance.

All the other experiments conducted through the thesis are for 15 receivers dropping packets to maliciously generate NAKs. We attempted to perform tests with AWT with same number of receivers generating NAKs, but the lengths of the simulations were very long due to much more degraded performance of the network. Hence, in order to explain AWT, we use fewer receivers generating NAKs. We observed that even with much fewer receivers generating NAKs intentionally, the extent of damage done was very high. Fig. 5 shows these 5 receivers.

From Fig. 6 we see that the source data transmission rate remains the same for the various NAK rates, with 5 out of the 128 receivers generating NAKs. This is because the AWT mechanism uses the maximum transmission rate for sending only ODATA & SPM. RDATA & NCF use the available excess bandwidth of the link. Thus, for all the cases, it takes the source the same amount of time to transmit all ODATA packets, however due to excessive drops and source buffer overflow, reliability is very low for all receivers at higher NAK rates, as seen in Fig. 8 and Fig. 9.

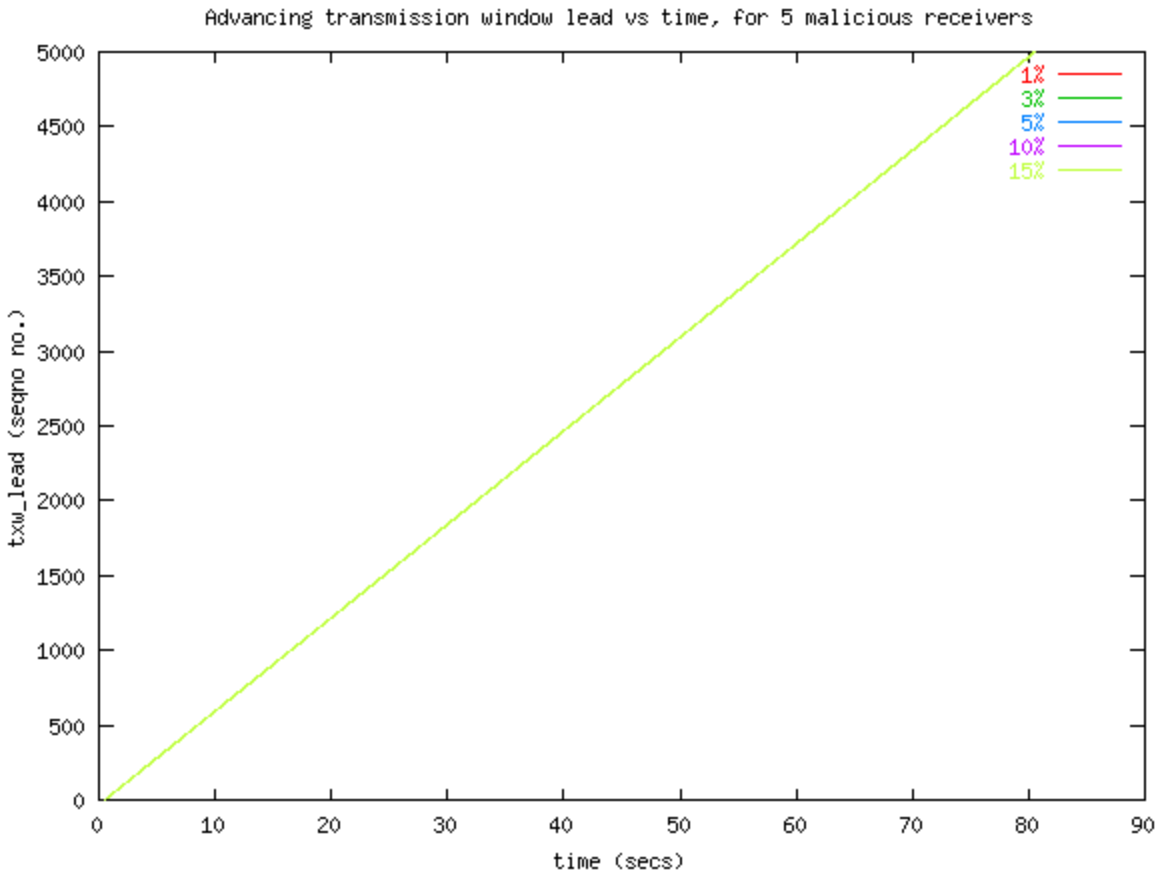

Fig 6: AWT: Advancing transmission window lead sequence no. (txw_lead) vs. time, for various NAK rates, with 5 malicious receivers generating NAKs.

Fig. 7 shows the cumulative retransmissions sent by the source over the period of the ODATA transmission. Once the ODATA transmission is complete, the source can use the entire bandwidth for sending the repairs, or stop and drop all retransmission requests, based on application. Thus, the plot only shows the retransmissions during the ODATA transmission.

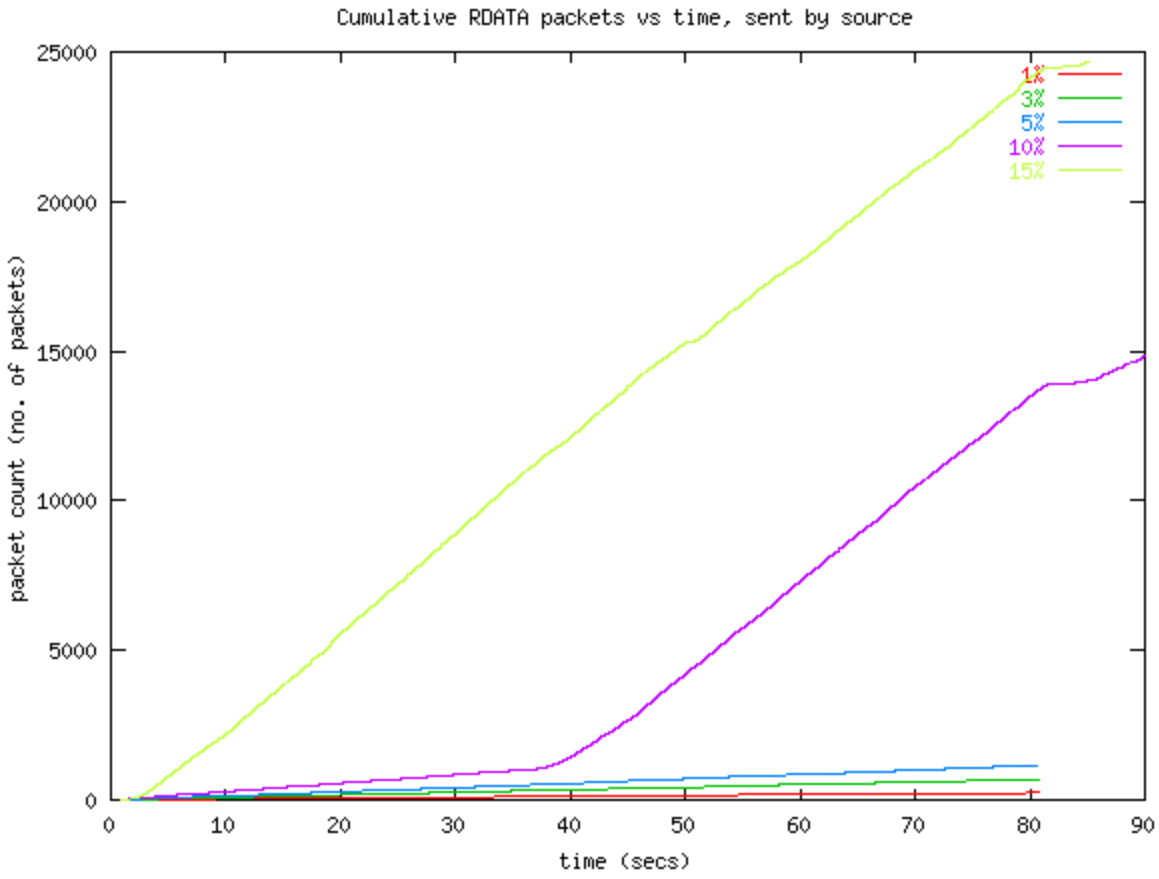

Fig 7: AWT: Cumulative retransmissions sent by the source vs. time, for various NAK rates, with 5 malicious receivers generating NAKs.

From the above graph we see that as the NAK rate increases, the rate of retransmissions also increase. For NAKS corresponding to 10% and 15% of total ODATA packets, the number of retransmissions or repairs sent by the source is several times the actual data. Since the source uses only the excess bandwidth to send the RDATA, the rate of RDATA stays steady after reaching a maximum. The consequence of this is that not all repair requests get a response. This leads to certain packets being lost permanently, i.e. missed, by the dropping receivers. This is shown in Fig. 8 below. This plot shows missed packets for one of the five malicious receivers that were generating NAKs by dropping packets. Similar results can be expected of the other 4 receivers.

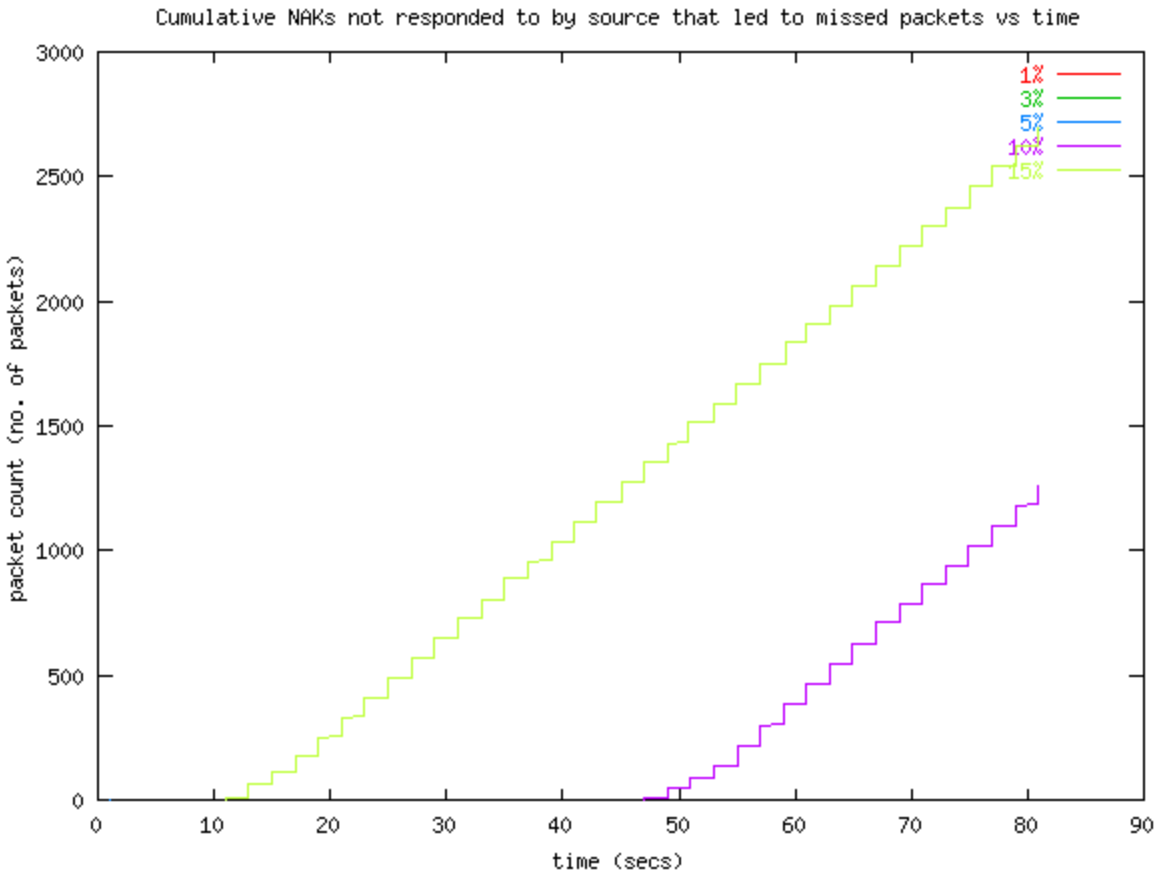

Fig 8: AWT: Cumulative NAKs not responded by source that led to missed packets for a certain malicious receiver vs. time, for various NAK rates with 5 total malicious receivers.

As the source attempts to respond to each repair request it receives, the source buffer overflows due to the high RDATA rate. This overflow leads to some RDATA being dropped from the source buffer. Now, if we assume an infinite buffer at source, then we would not see these losses. The limited buffer size also causes some of the ODATA to be dropped due to buffer overflow. This causes the non-malign receivers also to send repair requests, and thus also end up losing packets permanently as each RDATA is treated the same at the source.

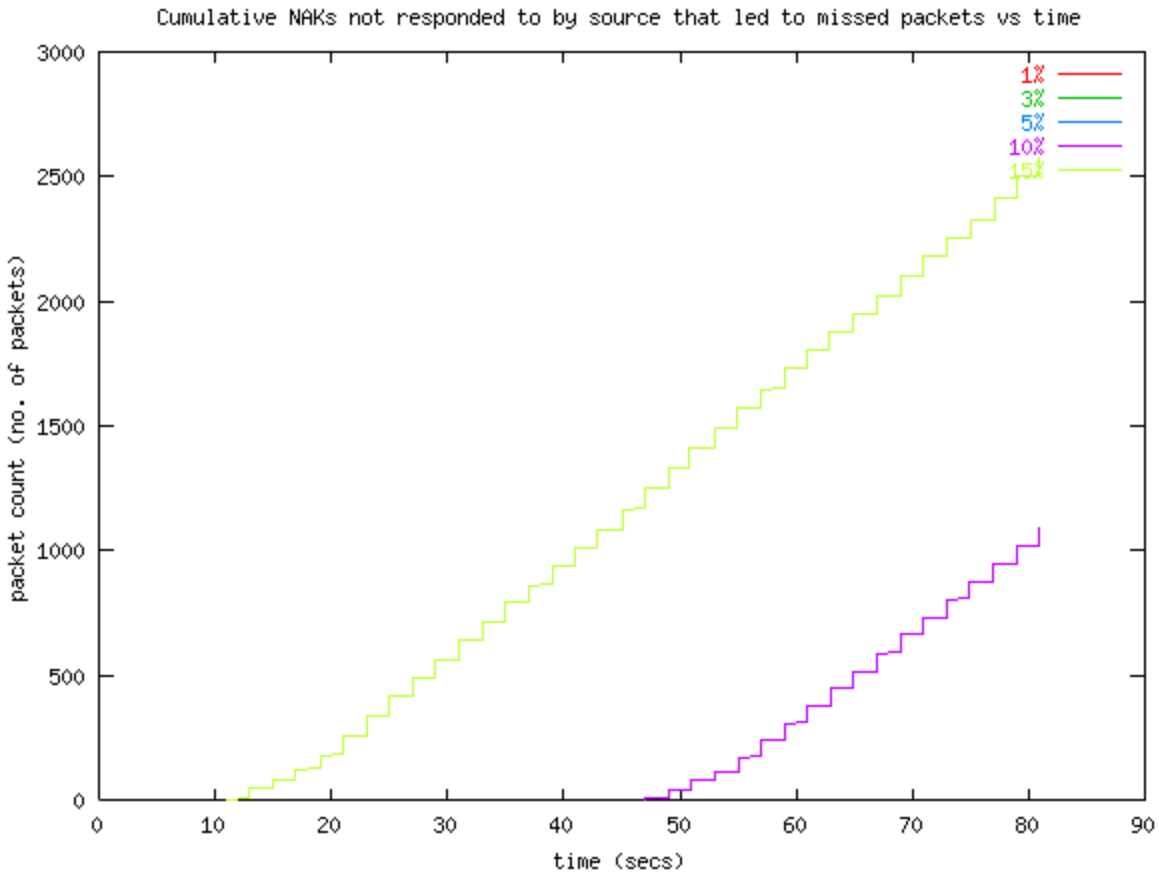

Fig 9: AWT: Cumulative NAKs not responded by source that led to missed packets for a certain non-malign receiver vs. time, for various NAK rates with 5 malicious receivers.

Thus, we observe that for the NAK rate corresponding to 15% of total ODATA packets, the number of missed packets, i.e. packets permanently lost, is almost half of the total number of data packets, and for the NAK rate equal to 10% of ODATA packets, it is almost 1/5 of the total number of data packets. Thus, we see this is no longer "reliable" multicast, and hence, advance with time and a fixed buffer size are unsuitable at these high dropping rates.

Fig. 9 shows the missed packets over the length of the simulation for a non-malign receiver. We can expect to see similar losses in the other non-malign receivers also.

3.3.2. Advance with data

Similar to the plots for the earlier mechanism, i.e. AWT, we have plots showing performance of 'advance with data' mechanism.

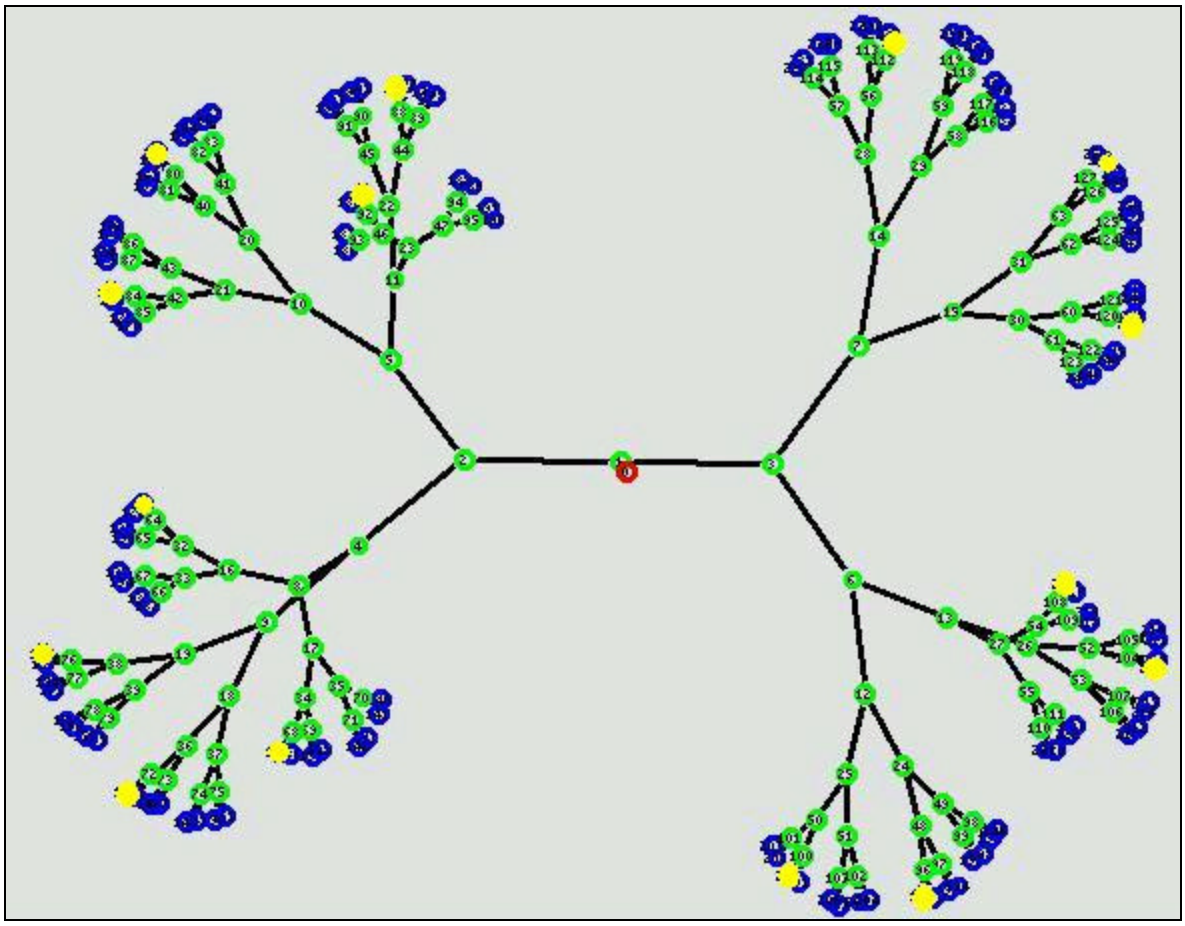

Fig 10: Topology showing the 15 malicious receivers used to test AWD. Blue - receivers, green - NEs, red – source, yellow –dropping receivers.

As explained earlier, all the tests ahead are conducted with 15 malicious receivers. Fig. 10 shows these receivers.

Fig. 11 shows decreasing ODATA transmission rate for increasing NAK rates. This is because as the NAK rate increases, the source receives more repair requests. In Advance with data (AWD), the maximum source transmission

rate is calculated from ODATA and RDATA. Thus, greater the time spent by the source sending RDATA, lower the ODATA rate. The excess bandwidth is used by the source to send the NCFs. Thus, because of the repair requests sent by the dropping receivers, the source is forced to slow down, in return sending slower transmissions to the other non-dropping receivers in the network. In this case, the only

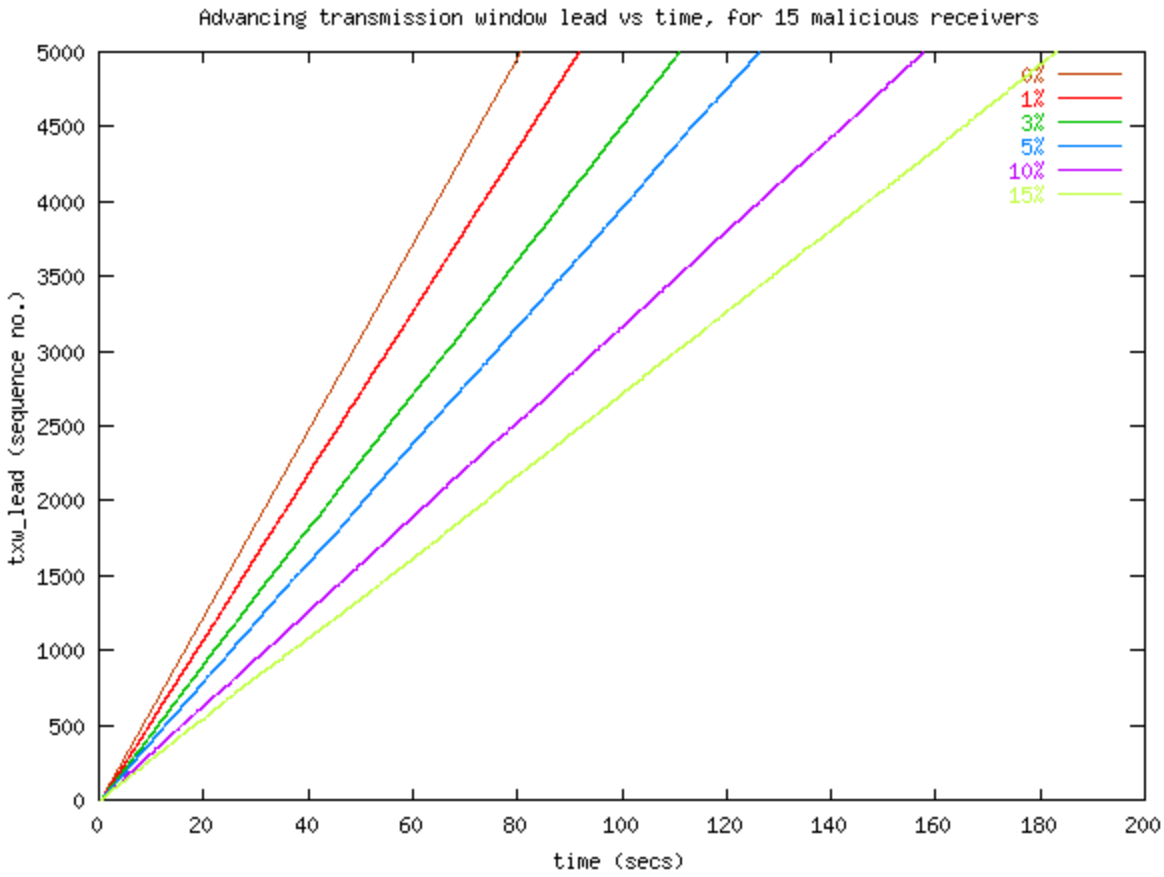

Fig 11: AWD: Advancing transmission window sequence no. vs. time for various NAK rates, with 15 malicious receivers generating NAKs.

possible way to improve performance without changing the mechanism would be to increase the maximum transmission rate of the source and the capacity of the links in the network.

From Fig. 11 above, we see that with 15 malicious receivers generating NAKs at rate corresponding to 1% of total ODATA

packets, the overall ODATA transmission rate has fallen by about 12%, and with a NAK rate corresponding to 15% of ODATA the transmission rate falls by about 55%.

The increasing number of retransmissions sent by the source can be observed in Fig. 12, which depicts the cumulative RDATA sent by the source over the period of the simulation for various NAK rates. With an increasing number of NAKs,

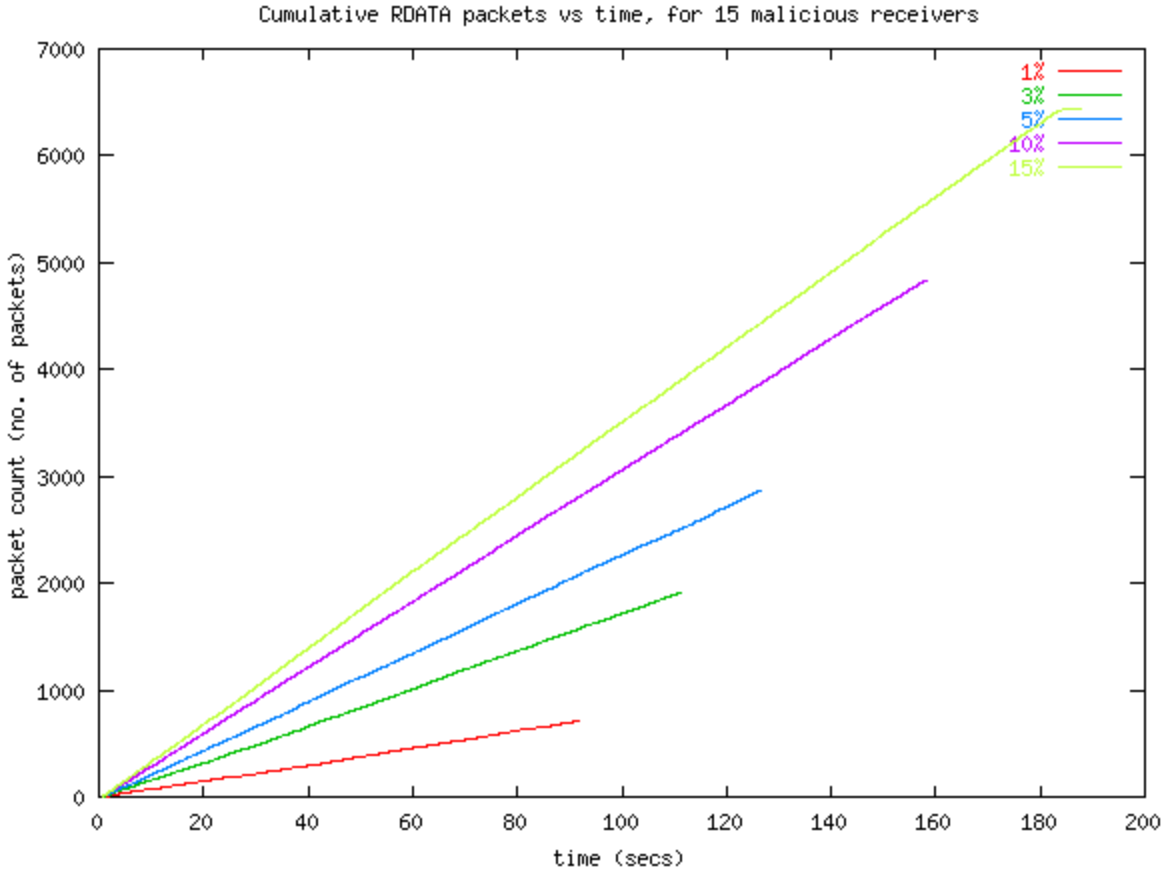

Fig 12: AWD: Cumulative RDATA sent by the source vs. time, for various NAK rates, with 15 malicious receivers generating NAKs.

multiple receivers send NAKs for the same sequence numbers. Since PGM NEs use constrained NAK forwarding, i.e. only the first NAK of the same sequence number is forwarded upstream, only one NAK of the same sequence number reaches the source. This controls the NAK implosion and also increases the efficiency of the PGM protocol.

The only limiting factor on the RDATA rate is the maximum transmission rate set for the source. Thus, the combined RDATA-ODATA rate will never be high enough to overflow the source buffer as long as this max rate is selected carefully. However, with increasing number of NAKs, source retransmission buffer may overflow creating permanent loss for some receivers. Fig 13. shows the cumulative packets missed for a specific malicious receiver.

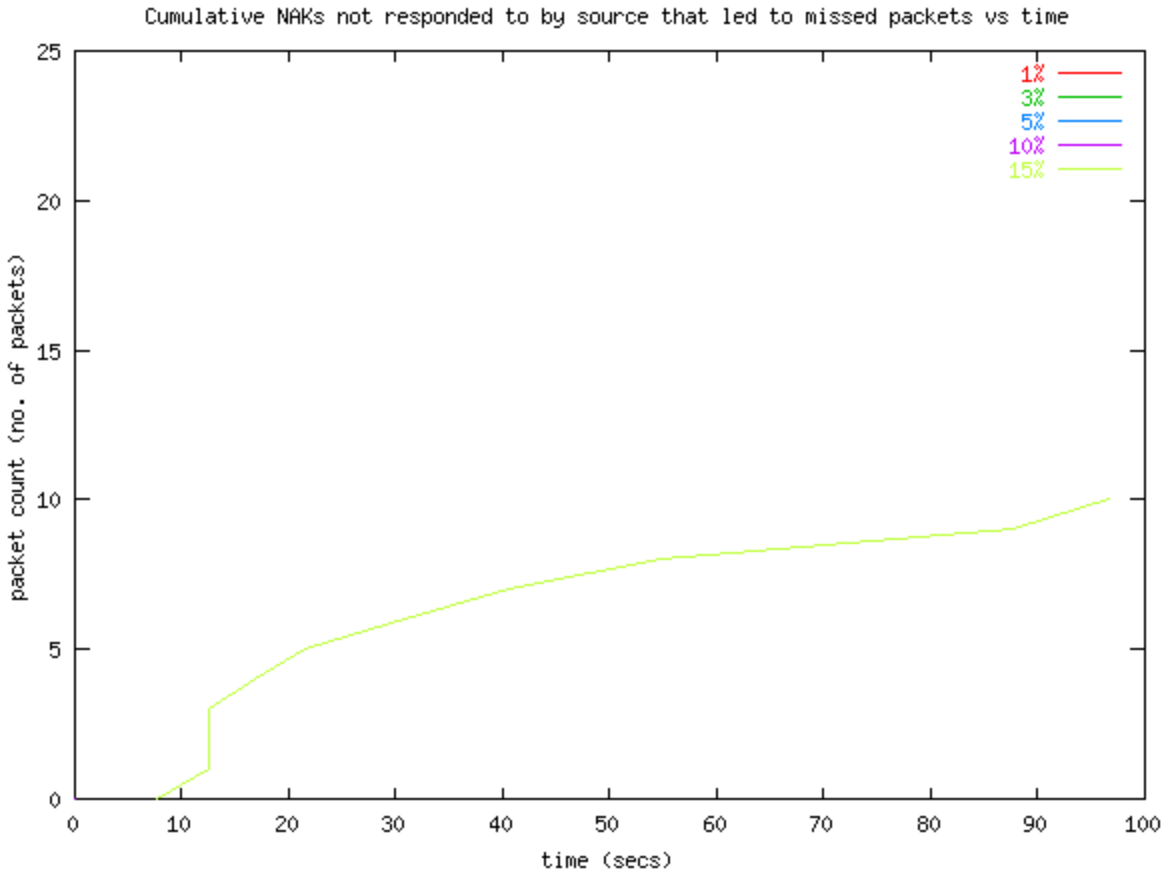

Fig 13: AWD: Cumulative NAKs not responded by source that led to missed packets for a malicious receiver vs. time, for various NAK rates, with 15 malicious receivers generating NAKs.

From this graph we see that only at very high NAK rates few of the malicious receivers, that are generating NAKs, undergo some permanent loss. We also observe that none of the non-malign receivers had any permanent packet losses.

Thus, AWD provides high reliability at the cost of throughput.

3.4. Overall analysis

From the above experiments, we see that for AWT mechanism, the source ODATA transmission rate is not really affected by the slower receivers, but leads to high permanent losses amongst these slow receivers. On the other hand, for AWD mechanism, while the slow receivers visibly affect the source ODATA transmission rate, the slow receivers have very high reliability. Also, in case of AWT, reliability is poor for all the receivers at higher NAK rates, while in the case of AWT, reliability is high for all the receivers even at higher NAK rates. Thus if throughput is maintained, reliability is lost. And if reliability is maintained, throughput is lost. These mechanisms were devised keeping in mind different applications.

One way to reduce the number of repair requests is to reduce the throughput as done by AWD. Another option is to reduce the amount of data being sent over the same transmission period. E.g. instead of sending a high quality video at T1 speed, video of lower quality at 56Kbps. Thus, the goal of the multicast session may be achieved, but at lower quality with lower throughput to permit slower receivers also to take part in the session.

If we are able to control which receivers join the session based on their receiving capabilities, the source would have a much better idea of what would be an ideal rate to transmit it. An optimum rate such as that would cause much less repairs being requested. This is possible to achieve through auction-based mechanisms. The application of such

auction-based mechanisms to reliable multicast is described in the next chapter.

4. PRINCIPLES OF A SOLUTION TO THE SLOWEST RECEIVER PROBLEM

In the earlier chapters, we described the principles of a specific reliable multicast protocol, viz. PGM, and then explained the Slowest Receiver Problem and demonstrated experimentally how it affects PGM. In this chapter we discuss how an auction-based mechanism, such as the Generalized Vickrey Auction (GVA) [21], can be used in a reliable multicast scenario to optimize the source transmission rate in an attempt to overcome the ‘slowest receiver problem’.

4.1. Why optimize?

A possible mechanism for providing rate control can be to optimize the transmitting rate. If we had a mechanism to steady the source transmitting rate, the sender is bound to receive a lot of repair requests, if this rate is very high, or if it is too slow it may not receive any repair requests. Which means that receivers are forced to wait and accept a low data rate, even though they are willing for the application to send data at a much higher rate. We need a mechanism by which the source adjusts to an optimum transmission rate.

The motivation behind this mechanism is to work out a scheme that prevents the receivers from generating false requests for retransmissions. If we assume that each receiver has a certain value for each session it wants to participate in based on the rate at which it receives the transmission, and if it was possible for the source to learn this true value for each receiver, the source can predict the rate requirements for each receiver and compute an optimum rate at which it can transmit. If this optimum rate turns out to be slow, then it means that the slower

receivers have a higher value for the multicast session and are willing to provide appropriate compensation for the slower transmission. If however there are receivers that do not provide the compensation and are yet attempting to affect the source throughput, appropriate action is taken such that other receivers are not affected. This control mechanism or policing action is described later in section 4.3 and 4.4.

Unlike the actual PGM protocol, which requires no knowledge of group membership, if we are to have some rate control based on the requirements set by the receivers, we need to have some information about group members. However very limited knowledge is required and needed only to perform some rate calculations.

Based on the application, we have two cases. For some applications, after the sender decides on an optimum speed, it doesn't require to change it, as there are no more receivers that are going to join in or leave during the session. However some applications are such that the receivers can join in or leave as they please during the multicast session. PGM is best suited for such applications. However, for testing purposes, we work only with the first case. Some suggestions have been made for the second case in section 8.2.

4.2. GVA and its application to PGM:

The rate control mechanism is based on two concepts:

- Every receiver has a value (maximum willingness to pay) for the session that it wants to participate in.
- Every receiver has an idea about its rate limitations.

If the sender has knowledge about these two variables of every receiver, it can decide on what is an optimum speed to transmit. Though, just having this information is not enough to ensure satisfactory reception speed to all receivers. There are some more issues that require to be considered.

- Receivers may quote some false value to influence a wrong decision by the sender
- Receivers may not keep up with the rate initially negotiated and request too many repairs to slow down the sender.

These issues are dependant on the choice of the auction-based mechanism used and can also be answered by policing the receivers. The policing mechanisms are described in detail later in section 4.3 and 4.4.

The motivation behind this technique is to use a pricing mechanism to utilize network resources more efficiently and at the same time be able to discourage/prevent slow receivers from affecting the data reception for other receivers.

Since this mechanism is based strongly on the rate messages from the receiver to the sender, it is important to ensure the reliability of these messages. If some messages carrying the rate information from the receivers to the source are lost, the optimum rate calculated by the source will be incorrect. If the message from the source to the receiver carrying the calculated rate information is lost, receivers will not be able to know at what rate transmission will occur. A bigger problem being that the network elements will not know how to police the receivers. Thus, reliability is an important issue for the polling request messages used by the source and polling responses

from the receivers. The reliability can be enforced by various known techniques, such as using some form of acknowledgements, repeated/periodic polling, etc. In this thesis, as described in Chapter 5, for the purpose of simplicity, we extend the functionality of NAKs so as to make use of their strong reliability in PGM, to carry the cost information from the receivers to the source. For downstream reliability, we rely on the periodic transmission of the Ambient SPMs, which also the carry the calculated cost information. If the initial cost SPM is lost and not received by certain receivers or NEs, then as mentioned above, NEs will not be able to police receivers. However, [2] specifies that a receiver cannot send any repair requests unless it has received at least one SPM. Receivers that want to request an SPM, incase they do not receive one before start of the session, can do so by unicasting a SPM Request (SPMR) message. The reader is referred to [2] for more details on how to use SPMR messages.

Another important issue here is the technique or mechanism being used by the source for calculating the optimum rate based on polling responses from the receivers. There are many desirable properties that such an algorithm may possess. Some of those properties are:

- Incentive Compatibility or truthful revelation
- Collusion proof
- Predictable pricing
- Efficiency
- Revenue/profit maximization
- Computationally feasible
- Lower complexity

One such auction mechanism has been used in [6], which examines the application of the Generalized Vickrey Auction to pricing reliable multicast.

The Generalized Vickrey Auction (GVA) is known as an efficient and incentive-compatible auction protocol for allocating multiple homogenous and heterogonous items in a distributed manner. In GVA, also called the Groves-Clarke "pivot mechanism", every bidder presents his/her value for every possible subset of licenses and the auctioneer chooses the final assignment according to a value-maximizing rule, specifying the payment to be made by every bidder. These payments create an incentive structure such that, for every participant, a dominant strategy is to reveal his/her true valuation. GVA satisfies individual rationality, Pareto efficiency, and incentive compatibility, when truthful bidding is the dominant strategy [9].

[6] assumes that the multicast source or the service provider has the ability to charge the subscribers and that the subscribers have a certain value for the session based on the rate at which they receive the transmission. The advantage of using the Incentive compatibility property of GVA in a reliable multicast scenario is explained as:

> We assume that the service provider will choose to admit anyone into the subscriber pool who has the capability to receive data at a rate at least as fast as the provider's chosen speed. However, the provider runs the risk of having a user overstate his reception rates in order to gain admittance to the subscriber pool. The task of admitting potential receivers into the subscriber pool and selecting a service speed that maximizes the social welfare is simplified greatly if the service provider can extract the true capabilities and values from potential subscribers. GVA has the property that it is a participant's weakly dominant strategy to truthfully reveal his value for, in this case, receiving data at various speeds and the limitations of his capabilities.

GVA computes a payment per individual, not a price per resource. Thus, it turns out two different receivers may pay a different price for the same resource. In the reliable multicast scenario, there is only one resource up for bids, since all receivers receive the same data at the same rate. Thus, if different receivers have different values for receiving transmission at different rates, GVA computes different costs for them. If they were to have equal values/bids for all rate options, GVA would compute equal costs for them, irrespective of the speed selected.

An Example:

Consider the network to be consisting of 3 multicast receivers, A, B & C. Fig. 14 shows the bids of these receivers for 3 rate options, viz. fast, medium and slow.

	Recv. (A)	**Recv. (B)**	**Recv. C**
Fast rate (F)	$v_A(F)$	$v_B(F)$	$v_C(F)$
Med. rate (M)	$V_A(M)$	$v_B(M)$	$v_C(M)$
Slow rate (S)	$V_A(S)$	$v_B(S)$	$v_C(S)$

Fig 14: The table shows various values (bids) for 3 receivers in a network, for 3 possible rate options.

GVA computes the optimum speed by first adding all the bids for each rate option, and then choosing the one with the largest sum. In our example, GVA computes the 3 sums as:

$$\Sigma\, v(F) = v_A(F) + v_B(F) + v_C(F)$$

$$\Sigma\, v(M) = v_A(M) + v_B(M) + v_C(M)$$

$$\Sigma\, v(S) = v_A(S) + v_B(S) + v_C(S)$$

If $\Sigma\, v(S)$ is the largest, then the *slow* rate is selected as the optimum rate for source transmission.

To compute the payment for a given receiver, GVA finds the sum of bids of other receivers, at the selected rate, in presence of the concerned receiver, and then subtracts it from the maximum sum of bids of other receivers, at a rate that would have been selected in absence of the concerned receiver. Thus, the payment value is basically the loss in value for the other receivers at the selected rate due to the presence of the concerned receiver. In our example, to compute the payment for receiver A at the selected slow rate:

Total value of other receivers at slow rate

$= v_B(S) + v_C(S)$

Max total value of other receivers in absence of A

$= v_B(F) + v_C(F)$

[We assume that $v_B(F) + v_C(F) > v_B(M) + v_C(M)$, i.e. in absence of A, a faster speed would have been selected]

Thus, payment for A

$P_A = [v_B(F) + v_C(F)] - [v_B(S) + v_C(S)]$

Similarly, we can compute payments P_B and P_C, for B and C respectively.

The actual payments for the receivers are less than or equal to their corresponding bids for that rate, i.e. receivers do not pay more than their bid for the selected rate.

$P_A = v_A(S), \quad P_B = v_B(S), \quad P_C = v_C(S)$

Thus, even though the receivers use the same resource, i.e. the slower rate transmission, their payments for using the slow rate are different, based on their actual bid values. A faster receiver may pay less for using the slower rate, while a slower receiver may be having a larger payment for using the slower rate.

Now let us assume that the slower receiver is not really slow, but is attempting to slow down the source

transmission. In order for it to affect the GVA rate decision, it would need to bid a higher value for a slower rate. By bidding a smaller value for the lower rate, the lying receiver is now not able to affect the GVA rate computation.

The paper describes both a centralized and distributed mechanism for computing costs using GVA. The distributed computing mechanism has several advantages as mentioned in [6]. However, in this thesis we limit our experiments to the centralized approach.

4.2.1. Implementation overview:

As mentioned earlier, each receiver has a value for a multicast session that it wants to participate in. Extending this a little further, each receiver has different values for each session based on the rate at which it receives or can receive the transmission.

At the beginning of the session, the source requests or polls each receiver for this cost information. In the request message the source may provide the receivers with the choices of various rates at which the source can transmit. In reply the receivers send back a poll response to the source containing a list of values/costs corresponding to the various choices it received. These values get aggregated as they pass upstream and finally, the source receives the total values corresponding to each of its choices. Using GVA, the source then calculates the optimum rate to transmit, and propagates this information back to the receivers.

This mechanism provides a means for providing the source with aggregated cost information for various rate options

so that the source may use a suitable algorithm to compute the optimum speed. The rate optimization method is not limited to this technique of passing pricing information. Any other mechanism can be used in its place that may provide the source with additional or different information that may be required by some other auction/pricing algorithm.

The actual details of the implementation of this mechanism are described in Chapter 5.

As discussed in this chapter, it is now possible for the sender to decide upon an optimum transmitting speed. This brings the issue of policing, addressed earlier. A difficult, but important, problem is to keep a check on the receivers and see that they honor their poll responses.

We discuss policing action with reference to PGM as the reliable multicast protocol and with ‘advance with data’ transmission window advance mechanism.

4.3. Source vs. Network-layer policing:

Policing can be achieved either by the source or by the PGM network element.

4.3.1. Source policing

The only feedback from the receivers to the source is in the form of NAKs. Hence, the source needs to accumulate information about receiver performance entirely on the basis of NAKs. If the source is able to obtain the receiver’s identity, it may be able to maintain some statistics about the behavior of that receiver. However, in PGM, for efficiency purposes, constrained NAK forwarding is

performed at network elements i.e. only the first NAK for a given sequence number is forwarded by the NE. Thus, the source is not able to maintain a correct account of individual receiver performance and the only policing action possible by the source is to make sure that the optimum transmitting rate is adhered to, maybe by just ignoring some NAKs, with reasonable leeway offered for NAKs due to regular network behavior.

As far as the bandwidth consumed at the source for sending the repairs is concerned, it does not matter to the source the order in which the NAKs are sent, or even if there are 1 or 3 or more receivers requesting the repair as long as there is only one NAK received by the source as required by PGM. Because of this constrained NAK forwarding, even if a large number of receivers request repair for the same ODATA, only one NAK reaches the source. All other NAKs are constrained by NEs and appropriate repair state is maintained.

Another parameter that can be considered by the source while deciding about the behavior towards the received NAK, is to check if the NAK lies in the increment window. Since NAKs in the increment window reset the transmission window advance timer, this is an important factor.

The advantage of policing at the source is that the entire process is implemented at the source and requires no support from the network layer.

The principal disadvantage is that without knowledge of which receivers are the cause for the NAKs, policing action can only be taken on NAKs in general and not against any specific or individual receivers. Thus, some policing

action may inadvertently affect other normal receivers also adversely.

4.3.2. Network-layer policing

If network-elements are able to peek into the pricing information propagated from the sender to the receivers, they can learn what is the optimum sender transmission rate. Using this information, the network elements can use any or a combination of the different performance measurements to control the amount of feedback sent to the source from the receivers. The implementation of such a technique of policing based on NAK rates in PGM is described in Chapter 5.

Thus added functionalities that are required by the NEs are:

- NEs will have to snoop into the packets exchanged during negotiation/cost-determination stage to learn about rate negotiation.
- NEs will have to check performance of each receiver. One of the possible ways is to check the NAK rate.
- Take policing action if receiver performance is found to be degrading. This action could be removing the receiver from session, or not forwarding any more NAKs from that receiver until receiver improves performance, or forwarding limited NAKs giving partial reliability.
- NEs may be required to reinstate receiver into the reliable session if the optimum rate changes or no more NAKs are sent by receiver.

Advantages of policing at the network layer are that it is possible to identify which interfaces are the cause of poor

functioning, thus marking out a smaller group of receivers. It is also possible to exactly identify the receivers that are not adhering to the initial state conditions. It is thus possible to take more specific action as compared to generalized policing in source policing.

The disadvantage is that extra functionality is required from all PGM network elements for the policing action.

4.4. Network-layer policing:

As receivers negotiate the optimum rate with the source, the network elements play a silent role of just forwarding the packets. However, once the session starts, and receivers send NAKs for missing packets, NEs play a larger role. They maintain state information for the NAKs received for each sequence number on each interface. An interface corresponds to the incoming link on the NE that connects a group of receivers, such as on a LAN, to the NE. Thus NEs would be in a position to police the entire group pf receivers connected to the interface as a whole or individually, if the NE had access to information exchanged during the early negotiation stage. This would allow the NEs to know what is expected from each receiver participating in the session and also to check their performance based on the rate of NAKs received from the receiver or some other criteria. With this we would be able to identify a certain misbehaving receiver or the group of receivers connected to the interface.

However, if the problem is because of a slow link either connecting receivers to NE interfaces upstream or between NEs, many receivers will send NAKs and they will be marked wrongly as misbehaving. Thus, it might be easier to police misbehaving interfaces, i.e. the entire group of receivers

connected to the interface as a whole, rather than the receivers alone and perform necessary policing action on the interface.

This mechanism can be generalized to all network elements. Once the network elements learn from the source what is the transmission rate, they can perform policing on all the interfaces, i.e. even on those interfaces that are only connecting other NEs and have no receiver connections. If, e.g., any of the interfaces are found to be requesting too many repairs, appropriate policing action is taken.

To conclude, in this chapter we have described how an auction-based mechanism, such as GVA, can be used along with policing to control the source transmission rate. In the next chapter we explain how this mechanism can be implemented in PGM.

5. IMPLEMENTATION OF A SOLUTION TO THE SLOWEST RECEIVER PROBLEM IN PGM

In the earlier chapters we described the potential threat to reliable multicast networks with PGM as an example and also suggested a pricing mechanism to reduce this threat. In this chapter we continue working with PGM and explain how our mechanism can be implemented in PGM. Chapter 6 shows some results obtained from running this new implementation of PGM in Network Simulator (NS-2).

5.1. Optimizing transmission rate:

5.1.1. Poll request phase (Collecting bids/costs)

Polling by the source can be done using some new type of packets or using the options field in PGM. In order to reduce complexity, we can also extend the functionality of available packet types.

We investigate the various downstream packet types available:

- ODATA: these packets are switched by NEs without transport-layer intervention.
- RDATA: they are multicast only on previously marked interfaces on which NAKs were received, i.e. constrained RDATA forwarding.
- NCF: the source sends NCFs only in response to NAKs received.
- SPM: these packets are transmitted by source either interleaved with the data packets (Ambient SPMs) or periodically in absence of data to transmit (heartbeat SPMs) and are used to maintain state information in the NEs and receivers.

From the above options, the SPM is the best-suitable packet type for use as cost-request messages, since minimal change is required in its operation. We now describe how SPMs can be used for polling all receivers.

Before the sender starts transmitting data packets for a session, it transmits a cost-request SPM (type 1) downstream. This SPM packet contains the various rate options that the source can transmit at.
NEs on receiving this forward it to the receivers (as per original PGM procedure).

5.1.2. Poll response phase

The only packet type that moves upstream in PGM is the NAK. Thus we use specially marked NAK packets to carry cost information back to the source. This also allows us to take advantage of the strong reliability provided for NAKs in PGM, by means of NCFs.

When receivers receive this type of SPM (type 1), they transmit (without any backoff) a cost-NAK upstream. This cost-NAK contains the bids/costs of the receiver for each of the rate options in the cost-request SPM.

On receiving any cost-NAK the NE immediately transmits back a NCF. Unlike the procedure specified for PGM where NCFs are multicast on the interface, NCFs will have to be unicast back to the cost-NAK transmitter. To do so, it will have to read the NLA of the receiver from the received NAK, and unicast a NCF back to that address. This NCF is not multicast, since the PGM protocol specifies that PGM receivers cancel their NAK generation on hearing identical NAKs. The NAK was to be multicast to prevent NAK implosion and improve network efficiency. But in our case, we want to

hear from each receiver and thus there is no advantage in multicasting it back to the interface.

When a NE receives the first cost-NAK it uses a timer in order to wait for other cost-NAKs from that and other interfaces. Costs from all cost-NAKs are aggregated and when the timer expires a single cost-NAK is forwarded upstream with the aggregated information. Any more cost-NAKs received after the aggregated cost-NAK has already been sent are immediately forwarded upstream. This mechanism allows a large number of cost-NAKs sent from a LAN or other group of receivers to be aggregated, while any late-comers or aggregated NAKs moving upstream do not have to wait at upstream NEs (as long as upstream NE has already sent it’s aggregated NAK).

5.1.3. Rate-information propagation

The source totals the cost from each cost-NAK received, and based on the size of the network the source will soon have received feedback from all existing receivers. The source then computes the optimum rate, as discussed in earlier chapters, and also calculates a threshold value (a maximum permissible NAK rate) that is to be used by NE for policing. This threshold value is based on certain leniency permitted. If all receivers are now receiving at the optimum rate, then ideally, no receiver should send any NAKs. However, considering normal network behavior some ODATA packets may be lost in transmission and hence it may be possible to receive some NAKs. Based on this factor, a maximum permissible NAK rate is calculated, which tell the NE to allow, say 5, NAKs in *x* interval of time on each interface. This information, i.e. the optimum rate and the threshold value, is put in to another SPM, cost-information SPM (type 2), which is sent downstream.

On receiving the cost-information SPM (type 2), NEs peek into the packet and obtain the threshold level that they will use for policing each interface.
When the receivers receive this SPM, they learn the optimum rate selected and based on their initial bids/cost-values, they are liable to pay some amount to participate in the session. Other receivers may choose to dropout or continue with no guarantee of reliability depending on application. If they chose to continue and send out NAKs, since they are operating at a lower rate, the NE policing the corresponding interface, will not forward the NAKs that exceed the threshold value. Thus, the slower receivers that chose to continue have limited reliability.

There needs to be some method for ensuring that the receivers pay the cost they had submitted earlier. The mechanism that may be used to enforce this is outside the scope of this thesis.

5.2. Network-layer Policing:

From the ***Rate information propagation*** stage, the NEs obtain the threshold value for policing each interface. Policing is based on the rate of NAKs received on each interface. To calculate the rate of NAKs on the incoming interface, the NE keeps a history of the last 5 NAKs received on each interface. When a new NAK arrives, it compares the time difference of the new NAK with the one that arrive 5 NAKs ago. From this information, it can compute the current NAK rate. Thus, for our experiments, we use this sample of last 5 NAKs to obtain the NAK rate. If the current NAK rate (calculated using the new NAK) is higher than threshold then the NAK is not forwarded, i.e. it is ignored/dropped.

Based on how the history is maintained, policing can lead to partial reliability or no reliability on an interface.

1. If a NAK that is ignored is stored in the history of that interface (i.e. even though the NAK is ignored, it is still considered while computing NAK rate) then ‘no’ NAKs from that interface will be forwarded as long as the NAK rate is above threshold. Thus, the receiver will experience zero reliability if it transmits too many NAKs.

2. If an ignored NAK is not considered while computing NAK rate, then NAK rate corresponds to only those NAKs that are forwarded by the NE. Thus, even if the receiver sends too many NAKs, some of the NAKs will be forwarded, giving partial reliability to a receiver/interface.

We have implemented the earlier method of policing that provides no reliability on interfaces that request too many repairs, unless their repair rate reduces within acceptable levels. It is also possible to have much stricter policing, wherein the NE may multicast a warning on the erroneous interface and then cut-off the interface for the entire length of the session. The actual technique of policing used by a network element may vary throughout the network allowing different degrees of policing to different groups of receivers. These techniques are thus application dependent and not restricted to the one we have implemented.

5.3. Capabilities of an adversary:

As described in Chapter 3, an adversary, in the form of a malicious receiver, can take advantage of the ‘slowest receiver problem’. The malicious receiver may drop packets to generate large number of NAKs or create false NAKs (i.e.

send NAKs even though it has not lost any packets), both having an identical effect of degrading source throughput. It may also take control of or cause a large number of receivers to create a higher NAK rate or even attempt to jeopardize the bidding mechanism by providing false bids about its rate limitations.

While such behavior can be anticipated from an adversary, we do not expect it to

- Make high bids and evade paying its cost as calculated by source,
- Compromise a network element in order to change the value of bids or fiddle with the policing action, or
- Take advantage of other protocol specific vulnerabilities, such as those pointed out for PGM in [2].

These security issues are analyzed in Chapter 8.

Based on the concepts of the auction-based mechanism and policing discussed in Chapter 4, we have described in this chapter, how the mechanism can be implemented in PGM. Our implementation does not use local repairers. The future work section in Chapter 8 suggests how the mechanism can be extended to that scenario. In the next chapter we present the results from our experiments on the auction-based mechanism used with PGM. As mentioned earlier, we did not implement the auction mechanism, but have provided a means for an exchange of the necessary information between the group members and the service provider. We assume that such a mechanism exists and does the necessary computation.

6. EXPERIMENTAL VALIDATION OF THE SOLUTION

Chapters 4 and 5 describe the concepts of how we can use an auction-based mechanism such as GVA in a reliable multicast scenario and it's implementation in PGM, to overcome the 'slowest receiver problem'. To complete the picture, in this chapter, we present the results from our experiments conducted using a simulator. Note that as discussed in earlier chapters, for experimental purposes, we assume that an auction-based mechanism such as GVA already exists with the service provider (the source). We extend the functionality of PGM to carry the information required by the source to calculate the optimum rate.

6.1. Simulation scenario

Section 3.3 describes the various parameters and topology that we used for conducting the experiments. The results presented in this chapter are for an implementation of PGM, that uses an auction based mechanism such as GVA at the source, to compute the optimum rate and costs, along with network-layer policing at each NE. As mentioned earlier, we have not implemented the actual algorithm at the source to compute the optimum rate, but assume that such a mechanism already exists. We do however implement the necessary messages that are required to carry the related information from the receiver to the source and back to the receiver. The details of this implementation are described in Chapter 5. For the purpose of our experiment, a pre-decided value is used for optimum rate and the threshold level for NAK rate is calculated from it (the pre-decided value for optimum rate is assumed to be the output of the auction-based mechanism, such as GVA). Both these values are then propagated downstream to inform the receivers of the rate, and the NE of the threshold level to use for policing. The

reliable multicast protocol used was centralized-PGM (no support for DLRs) with ‘advance with data’ flow control mechanism, modified to provide rate control with network-layer policing. Fig. 11 shows the receivers that were generating NAKs intentionally during the experiments conducted with AWD before the modifications.

6.2. Simulation results and analysis

On running the simulations we obtain the plots below. Fig. 15 shows the advancing transmit window vs. time with 15 malicious receivers. We observe that the transmission rate of the source remains more or less the same for the various NAK rates.

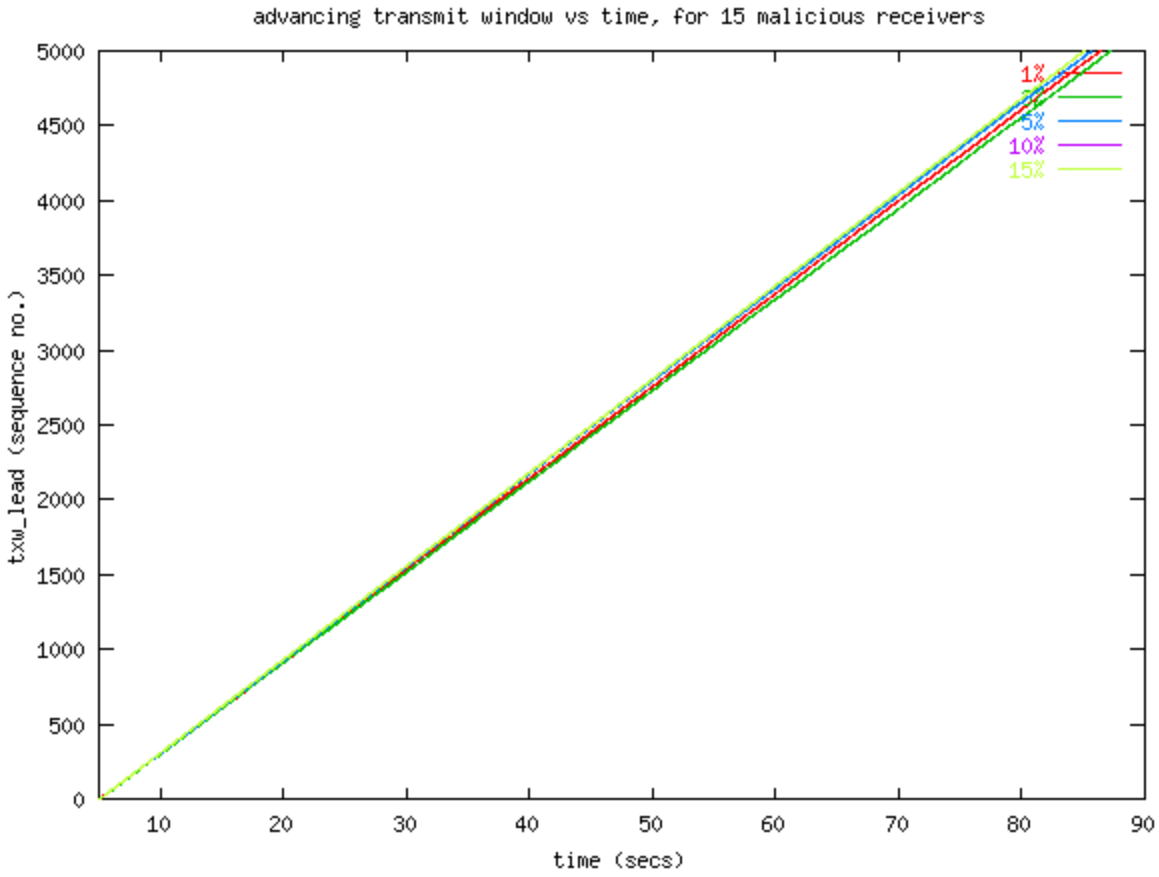

Fig 15: Rate optimization with network policing: Advancing transmission window sequence no. vs. time, for various NAK rates with 15 malicious receivers generating NAKs.

The slight variation in the transmission rates is due to slight leniency given to the receivers for requesting some repairs. Also, some time is allowed to lapse before the data transmission begins to allow the source to complete the optimum rate computation phase, which explains why it took a little longer to complete the data transmission as compared to AWT and AWD.

Fig. 16 shows the cumulative retransmissions sent by the source vs. time.

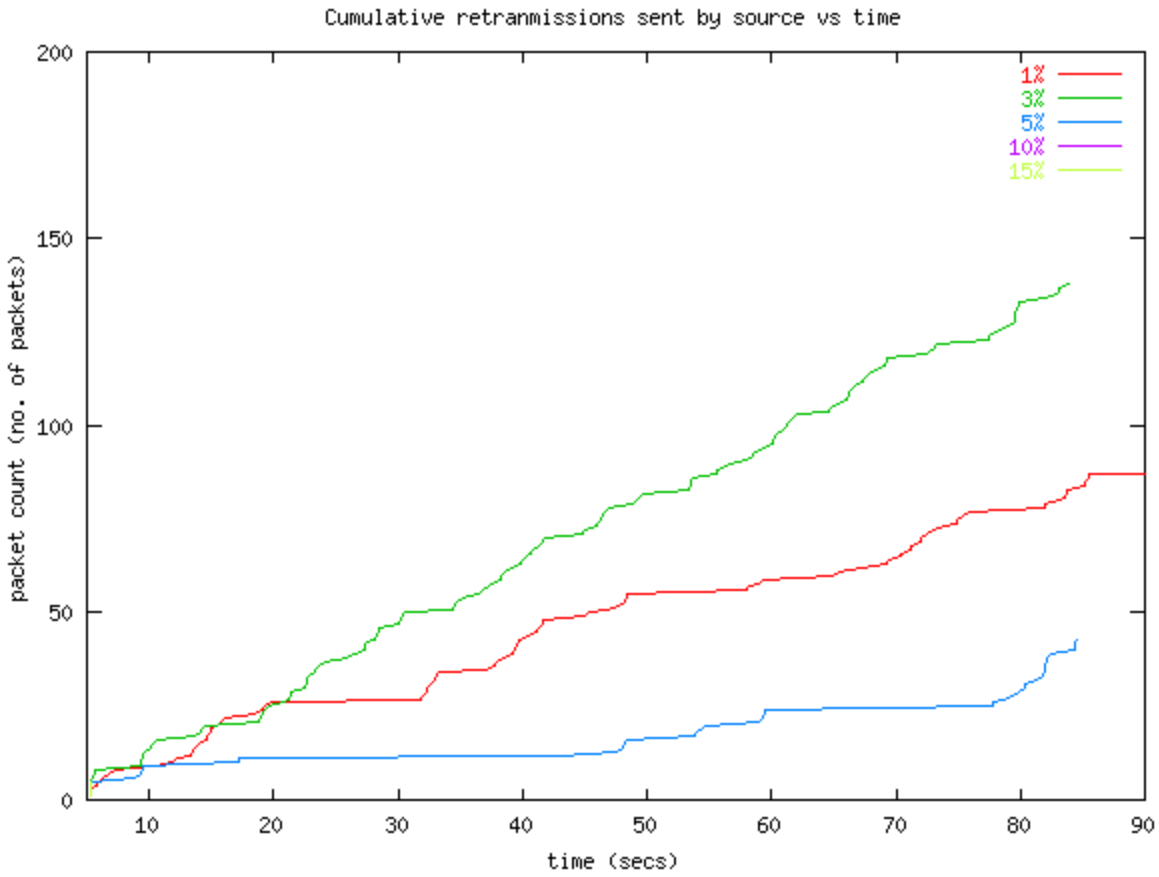

Fig 16: Rate optimization with network policing: Cumulative retransmissions sent by the source vs. time, for various NAK rates, with 15 malicious receivers generating NAKs.

We observe that for higher NAK rates from the 15 receivers, the retransmissions sent by the source reduce. The retransmissions, for NAK rates corresponding to more than

3% of total ODATA packets, keep reducing as fewer and fewer repair requests get through the network elements to the source. RDATA sent for NAK rates corresponding to 10% and 15% of total ODATA packets are too few to be visible on the plot. At higher NAK rates, the NAKs are sent very frequently. Thus, when the NAK rate is calculated based on the last 5 NAKs on that interface, the NAK rate is always found to be higher than the threshold level. And hence no NAKs are able to get past the policing NE. If there were any NAKs sent over a longer time interval, long enough for the NAK rate to be less than or equal to the threshold level, they would have been forwarded by the policing NE. But this was not the case observed during the experiments. Thus, we see almost no retransmissions sent by the source at the higher NAK rates.

Fig. 17 shows the cumulative lost packets for a specific dropping receiver. The nature of the plot is almost the same for the rest of the dropping receivers. We observe from the plot that as the NAK rate increases, the number of packets lost permanently also increases since more and more repair requests are filtered by the network elements. From the simulation we also observed that the other receivers that did not send NAKs did not undergo any loss and received the complete transmission with no missed packets.

Comparing Fig. 15 with Fig. 6 and Fig. 11, we observed that source data transmission rate remains almost stable for the pricing mechanism and for ‘advance with time’ mechanism, while transmission rate reduces drastically as NAK rates increase for ‘advance with data’ mechanism.

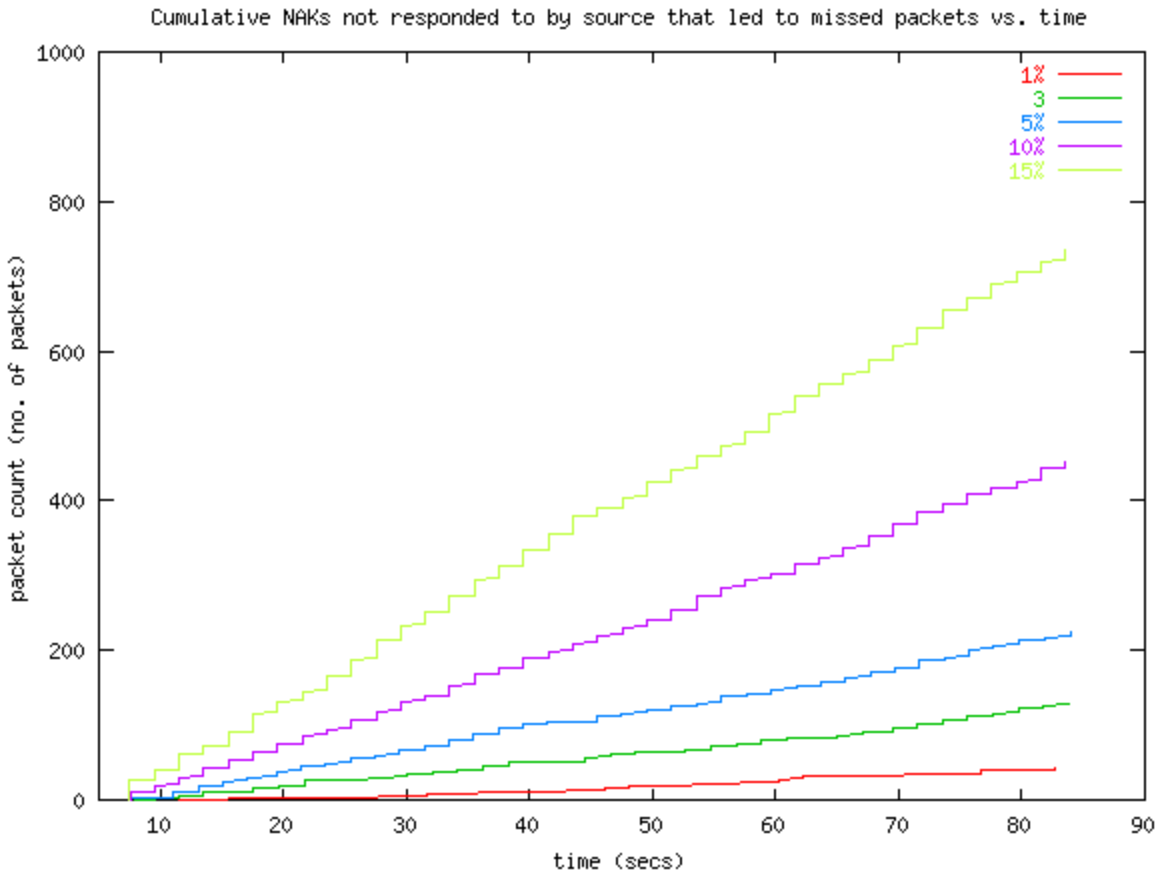

Fig 17: Rate optimization with network policing: Cumulative NAKs not responded by source that led to missed packets for a malicious receiver vs. time, for various NAK rates with 15 total malicious receivers.

Comparing Fig. 9 with results of AWD and pricing mechanism simulation, we observe that reliability for non-malign receivers is very high for the pricing mechanism and AWD mechanism, while they are poorer for AWT.

For AWT, the source throughput is maintained irrespective of the malicious receivers, while reliability was poor for both malicious and non-malicious receivers. For AWD, reliability was guaranteed for the non-malign receivers, while the non-malign receivers also had high reliability, but the source throughput was badly affected. By using the pricing mechanism for rate control with network-layer policing, we are able to achieve best of both worlds. The

source throughput is controlled and stable; also reliability is guaranteed for the non-malign receivers. The malicious receivers on the other hand have poor reliability based on their NAK rates and threshold value.

In the next chapter we look at the performance of the system using source policing. As mentioned in Chapter 4, the advantages of using source policing is simpler and easier implementation in the network, since the only changes that will have to be done will be with the source.

7. THROUGHPUT-RELIABILITY TRADEOFF

With the ‘rate optimization using pricing’ technique, we are able to steady the source transmission rate once it has been fixed. With the existing ‘advance with data’ mechanism we are able to obtain high reliability at the cost of throughput. However, without the complexity of the network-layer policing, it is also possible to have an intermediate mechanism that works on source policing, to provide a tradeoff between reliability and throughput, as shown in this chapter.

The source policing mechanism will also need to employ an auction-based mechanism, such as GVA (described in Chapter 4) to compute an optimum rate at which to transmit. This information can then be used by the source to perform policing on NAKs.

7.1. The tradeoff mechanism

The only feedback received by the source is the NAKs from the receivers. As explained earlier in source policing, if the source were to control the received NAKs and decide on which ones to reply, it is possible for the source to control the throughput and reliability.

The selection of threshold level by the source is based on how in responding to a certain NAK will the throughput be affected. On receiving a certain NAK and taking into consideration the current rate of transmission, it is possible for the source to predict how the throughput would be affected if it replied to the NAK. Thus, based on the leniency level required, depending on the tradeoff with reliability, source fixes a threshold level for responding to the NAKs. If in responding to a certain NAK, the

throughput would fall below the threshold level, the source then chooses not to send a repair in response to NAK. Whether the source responds or ignores a NAK, it has to send an NCF immediately on receiving a NAK as specified by the PGM protocol.

We conducted the experiments under the same conditions and parameters as used for the earlier experiments. These parameters and the topology used are described in section 3.3. The threshold level used for these experiments permitted a NAK rate corresponding to 3% of total ODATA packets from 15 malicious receivers. This level of threshold was selected as a matter of choice and any other level could be chosen and throughput and reliability would change accordingly. Higher throughput and lesser reliability would be obtained for a low threshold level, and vice versa for higher threshold level.

Fig. 18 shows the advancing transmission window vs. time. We observe from the plot that the ODATA transmission rate remains the same for NAK rates corresponding to 3% of total ODATA packets and higher (in Fig 18 below, these lines are overlapping). This is because, the transmission rate at 3% is used as a threshold level, and hence at NAK rates of 3% and higher, the transmission rate is the same (overlapping). The nature of this plot is very similar to that of AWD (see Fig. 11) for 1% and 3% NAK rates. The difference lies at higher NAK rates. With AWD, the throughput reduces further, but with source policing they remain same as that of 3% NAK rate.

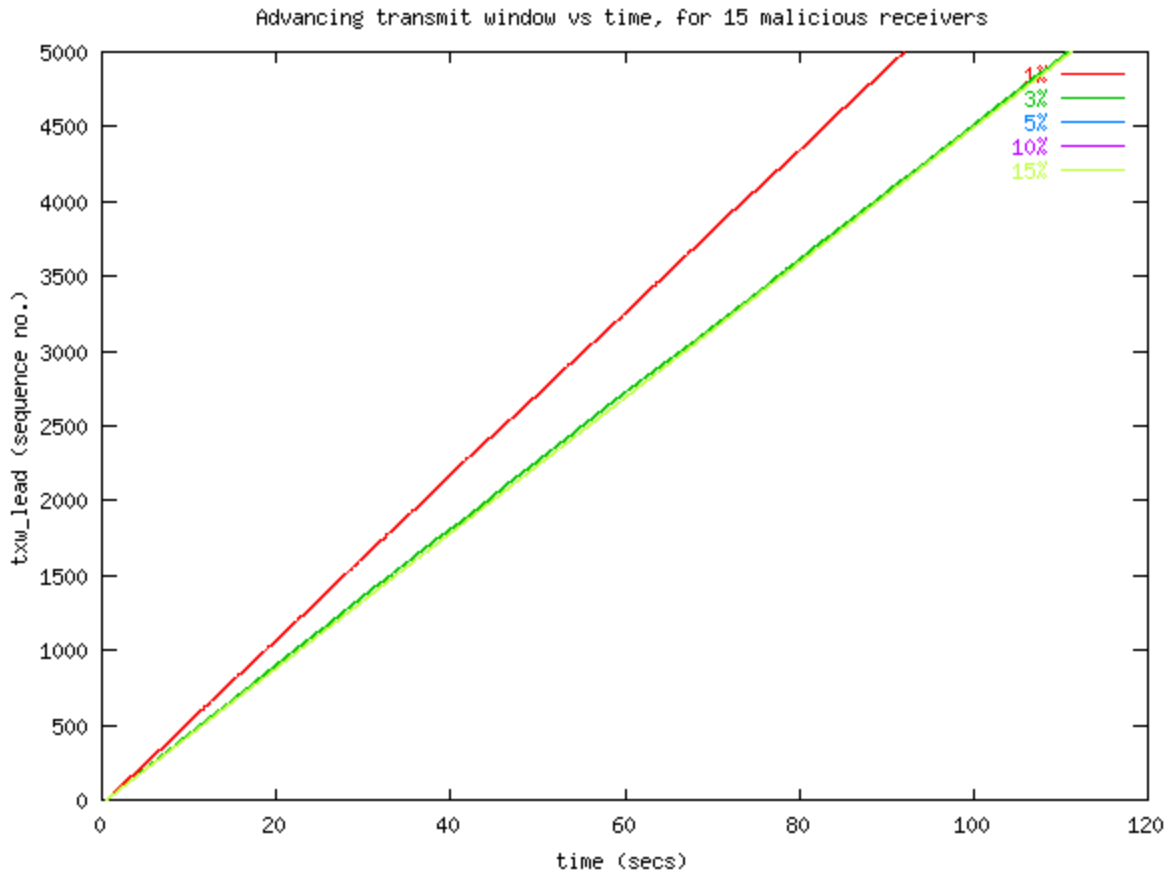

Fig 18: Source policing: Advancing transmission window sequence number vs. time, for various NAK rates, with 15 malicious receivers generating NAKs.

Fig. 19 shows the cumulative retransmissions sent by the source. From the plot we see that for NAK rates greater than 3%, i.e. higher than the threshold level, the number of RDATA sent by the source is controlled and hence is the same for all these rates. Once the ODATA transmission is complete, pending RDATA are sent without control, which explains the increase in RDATA rate towards the end of the session. Again, this plot is very similar in nature to that obtained with AWD (see Fig. 12). The number of retransmissions is similar for both methods for NAK rates up to 3%.

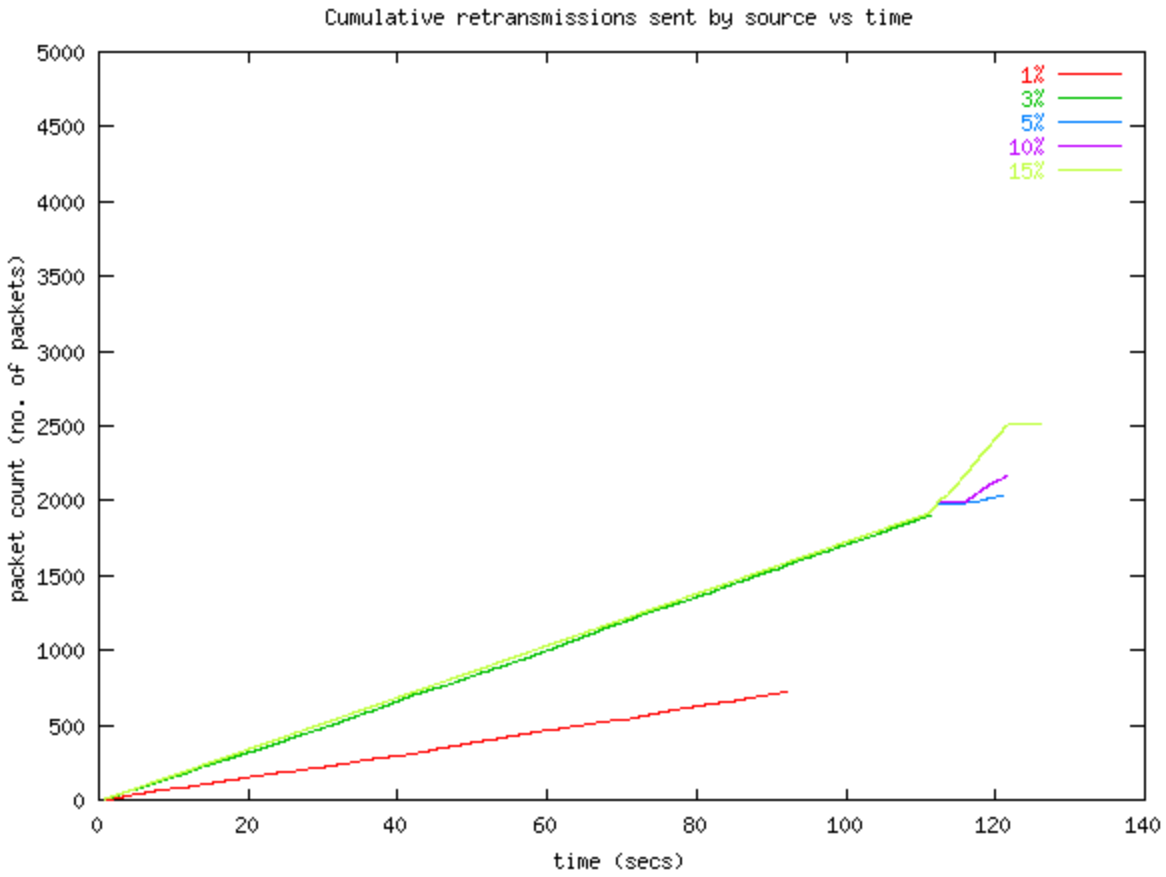

Fig 19: Source policing: Cumulative RDATA sent by the source vs. time, for various drop rates, with 15 receivers dropping packets.

The malicious receivers may lose some packets permanently, if it was generating NAKs by dropping packets, since all repair requests received by the source are not responded to. Fig. 20 shows the cumulative packets lost by a dropping receiver. The nature of this plot is similar to that while using the network policing as seen in previous chapter (see Fig. 17). Also similar to AWD mechanism and network policing, non-malign receivers do not undergo any loss and obtain complete transmission.

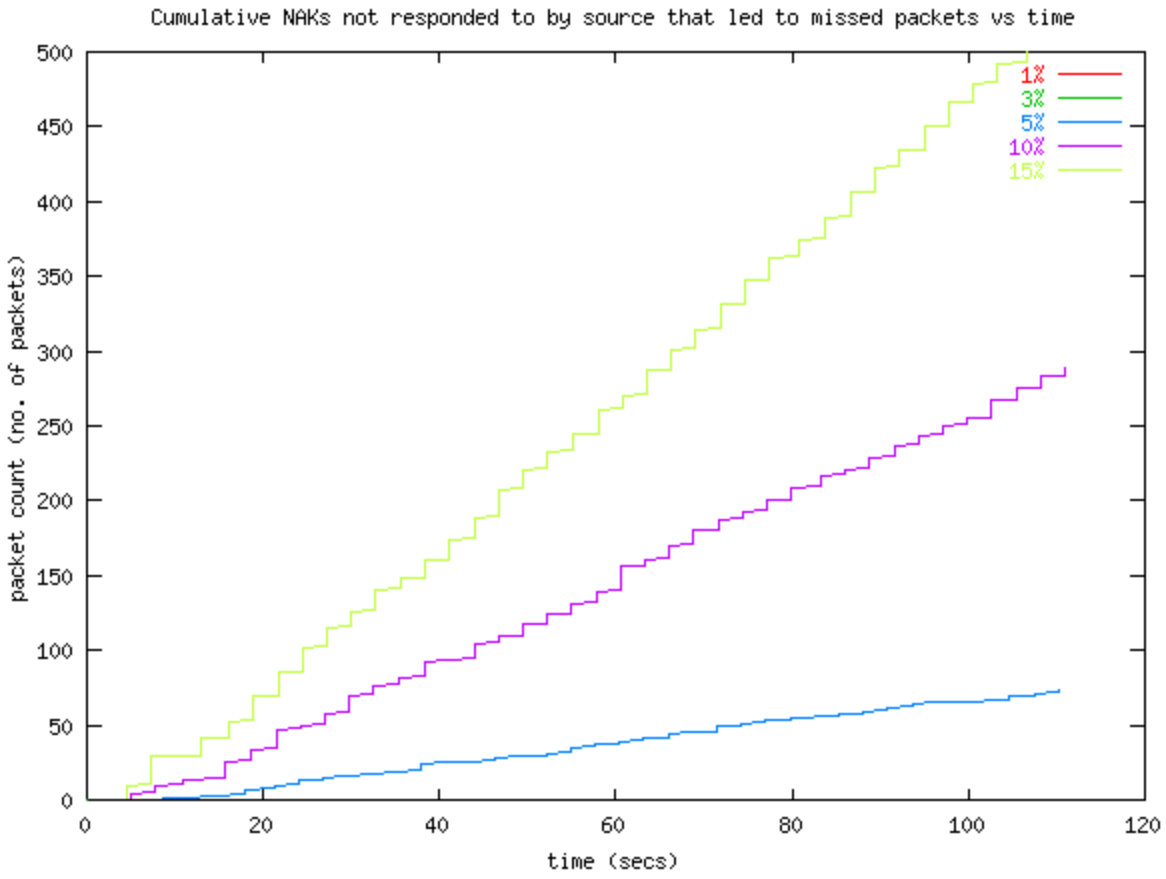

Fig 20: Source policing: Cumulative NAKs not responded by source that led to missed packets for a malicious receiver vs. time, for various NAK rates, with 15 malicious receivers generating NAKs.

Another interesting parameter that can be used for control is the retransmission buffer size. If the threshold level used by the source is large enough to allow the source to respond to many repair requests, a limiting factor will be the size of the retransmission buffer at the source. If the retransmission buffer is small, source will only be able to respond to small bursts of repair requests, while large retransmission buffer size could mean that some repairs reach the receivers late because of spending too much time in retransmission buffer queue. This could lead to repeated requests and waste of network resources. Thus, controlling the retransmission buffer size allows control over reliability of malicious receivers.

The biggest disadvantage of this method as specified earlier in source policing, is that this is a generalized policing mechanism. It is based on the assumption that once the source selects an optimum rate to transmit at, only malicious receivers request repairs. Thus, if non-malign receivers request repairs occasionally (this could be many if the total number of receivers is very large) these repair requests will also be treated similar to the rest and no distinction is made between them. The reason is that it is very hard for the source to identify any malicious receiver or make any distinction based on NAKs. Using network policing, it is possible to generalize the problem to an incoming interface, but by using source policing, this is also not possible. Since NEs use constrained NAK forwarding, only the first of the NAKs for a given sequence number gets through the NE and thus, it becomes very hard to identify the malicious receivers.

Thus, we see from our results from this chapter that by changing the threshold level, a tradeoff can be made between reliability and throughput. A low threshold level permits the source to respond to only a few NAKs, thus keeping threshold high, but very poor reliability for the malicious receivers. On the other hand, a high threshold level allows the source to respond to more NAKs, increasing reliability for the malicious receivers, but at the cost of throughput. The next chapter contains our conclusions from this thesis and some suggestions for extending this work in the future.

8. CONCLUSIONS

In this last chapter of the thesis, we present our conclusions from this thesis. We also point out some areas in which the work shown here can be extended further.

8.1. Conclusions

From our simulation experiments we conclude that using rate optimization with pricing technique along with network-layer policing in a reliable multicast network, it is possible to overcome the problem of the slow receiver and gain control over the source transmission rate. The malicious receivers may undergo large losses depending on their NAK rates, but if the other receivers adhere to the negotiated rate then the malicious receivers do not affect them. The algorithm that is used by the source to compute the optimum rate strongly influences the performance of the session.

Based on our observations regarding the performance of the four mechanisms described in this thesis, we summarize as follows:

- AWT: all the ODATA is sent over the same time period. There is no feedback from the receivers to the source and hence malicious receivers do not affect source rate, i.e. throughput remains steady. However, reliability is very poor not only for the malicious receivers, but even for the non-malign receivers.
- AWD: Feedback is based on NAKs that lie in the increment window, which provide for flow control. Reliability is maintained in presence of various malicious receivers for all non-malign receivers. Malicious receivers tend to lose a few packets at very

high NAK rates due to retransmission buffer overflow. High reliability is obtained at the cost of throughput, as with increasing NAK rates for multiple receivers, source throughput decreases dramatically.

- Auction-based pricing with network policing: Source rate more or less is independent of the NAK rates and reliability is maintained for non-malign receivers. Non-compliant receivers have little or no reliability.
- Auction-based pricing with source policing: Source rate can be altered by changing the threshold level, thus controlling the reliability level for the malicious receivers.

The biggest drawback of the policing mechanism is that it is extremely hard to identify the malicious receivers. We generalize the problem to be coming from an interface and all NAKs from such an interface are treated equally. Thus, if non-malign receivers lie in a group of receivers connected to an interface or downstream of an interface that is receiving high repair requests due to some malicious receivers, then occasional repair requests from these non-malign receivers may also be filtered out, affecting their reliability.

8.2. Security analysis:

A number of security threats are identified by this thesis. In this section we analyze the threats taken into consideration by our mechanism and also identify other threats that are left out by our discussion.

An adversary, in the form of one or more receivers, may give false bids about its rate limitations to influence a wrong decision by the source. If so, the cost computed by

the source for the malicious receiver will correspond to its high bid. If the source does select a lower rate due to this receiver, then the cost for the receiver will also be comparatively high as compared to that for other receivers.

A malicious receiver may also generate false NAKs though it has received the corresponding ODATA, or may drop packets intentionally to produce genuine NAKs. Though it is not possible to distinguish the NAKs, both have the same implication on the source performance, i.e. reduction in throughput in AWD, or in reliability in AWT, and thus are dealt with identically. Once an optimum rate is computed, all complying receivers are expected to receive the transmission at that rate, and barring a few NAKs, all additional NAKs on the interface are policed. Thus, NAKs sent by a malicious receiver do not reach the source and hence are unable to affect the system performance.

There are also other threats, which are considered to be outside the scope of this thesis. These are:

Evading payment:
A malicious receiver may shirk from paying the cost that has been computed for it by the source. Necessary authentication and cost recovery techniques need to be employed in any bidding mechanism to ensure that bidders keep up their commitments.

Compromised network element:
An adversary may take control of a network element and use to change the bid values of other receivers or to prevent bids from reaching the source, effectively tampering the bidding mechanism. Again, effective authentication mechanisms are assumed to be in use to detect if network elements are malignant.

Protocol specific vulnerabilities

Our experiments are demonstrated using PGM, which has been identified to be having certain vulnerabilities, mentioned in [2] as “Short of full authentication of all neighboring sources, receivers, DLRs, and network elements, the protocol is not impervious to abuse”. So would be the case for any other reliable multicast protocol used with the bidding mechanism. Such protocol specific weaknesses are outside the scope of this thesis.

8.3. Future work:

While our results conclusively show that by using an auction-based pricing mechanism for a reliable multicast session, it is possible to reduce the threat of a denial of service attack on the complying receivers, there are still some avenues left to investigate or pursue further.

A different protocol for carrying cost information:

For the purpose of this thesis, we extend the functionality of the PGM protocol to carry the various pricing information. The extension has been added to existing packet types. It may be more efficient, however, to use a completely different protocol for this purpose. By using SPM messages to carry calculated rate information downstream, we depend on the periodic transmissions of Ambient SPMs to provide weak reliability. Those receivers that detect a missing SPM can use SPMRs to solicit an SPM for the source. If strong reliability is required, this may be achieved well by designing a separate protocol for carrying the rate information from the receivers to the source and back to the receivers.

Support for Local Repairs:

The PGM simulations used for experimenting in this thesis did not have support for DLRs. If DLRs are brought into the picture, the mechanism for calculating optimum transmission rate does not alter. However, policing will need to be different. Since most of the repair is not provided by the source anymore, no bandwidth is wasted by the source for RDATA transmission if all repairs are done locally. Thus, receivers sending out too many repair requests do not slow the source. However, PGM specifies that for all the repairs sent locally, DLRs are required to send NNAKs (Null-NAKs) to provide flow-control feedback. If the source is employing AWD, NNAKs that lie in the increment window reset the transmission window advance timer. These are the NNAKs that can cause source to slow down by delaying transmit window advance. Thus, these NAKs need to undergo policing. Also if DLRs are not able to provide repair, then the NAKs are redirected to the source. Thus, these NAKs also affect source data transmission rate and need to be policed. Policing may be done again at network-layer as in centralized-PGM. It is also now possible to do policing at DLRs. Since most of the repair requests received by the DLR do not affect the data transmission rate of the source, DLRs can implement a generalized policing mechanism similar to the source policing discussed in chapter 4. It can police specific NAKs, viz. those in increment window, or those which the DLR will need to redirect to the source since it does not have its repair available.

Periodic rate optimization:

The method described in this thesis is for a multicast session where the receivers do not join or leave once the session has started. But if the new receivers want to join or existing receivers want to leave during a session, then in such a scenario the optimum rate at which to transmit

may also vary. Thus, it is required to recalculate the optimum rate and pricing when group membership changes. It is, however, difficult to do so every time a member joins or leaves the session. A periodic pricing algorithm may work in such a scenario. Instead of sending poll request messages before start of session, the sender could periodically send poll request messages with the various rate options. The value specified by a receiver for each rate now corresponds to its value not for the entire multicast session, but for this period of the session till the next poll request arrives. The change in the number of receivers could thus be taken into account with each cycle of polling. This also allows for receivers to change their value for the various rates from cycle to cycle. With GVA, the change in the number of receivers can change the payments computations for the other receivers, which may lead to a new rate being selected. This may not be acceptable to some receivers. As suggested in [6]:

> A utility model that properly considers both the value of admission and the value of continuing service over time is a promising avenue to explore.

Research in synchronized multicast:

Research work in IP multicast has, not given enough importance to applications requiring synchronized and reliable transmissions. There are a number of protocols for providing synchronized receiving, but they lack flow control mechanisms. This leaves a door open for further research at the IP multicast level.

REFERENCES

[1] T. Parks, D. Kassay, and C. Weinstein, *"Vulnerabilities of Reliable Multicast Protocols"*, IEEE Military Communications Conference, Oct 1998.

[2] T. Speakman et al, *"PGM Reliable Transport Protocol Specification"*, RFC 3208, December 2001.

[3] M. Yamamoto, Y. Sawa, S. Fukatsu and H. Ikeda, *"NAK-based flow control scheme for reliable multicast communication"*, Proceedings IEEE Globecom 1998, Vol. 5, pp 2611-2616, November 1998.

[4] D. Woods, *"The wizardry of Multicast"*, Network Computing, Feb 19, 2001.

[5] B. Levine and J. Garcia-Luna-Aceves, *"A Comparison of Reliable Multicast Protocols"*, ACM Multimedia Systems, 6, pp. 334-348, 1998

[6] A. Sureka and P. Wurman, *"Applying the Generalized Vickrey Auction to Pricing Reliable Multicast"*, International Workshops on Internet Charging and QoS Technologies (ICQT) 2002.

[7] *"IP Multicast: Inside Look"*, InformIT.com, article from Cisco Press, October 19, 2001.

[8] S. Banerjee and B. Bhattacharjee, *"A Comparative Study of Application Layer Multicast Protocols"*, Computer Science Dept, University of Maryland College Park.

[9] P. Milgrom, *"Putting Auction Theory to Work: The Simultaneous Ascending Auction"*, Stanford University, Department of Economics, 1998.

[10] Z. Kangxin, L. Jianhua and Z. Hongwen, *"Sender Delay comparison of IP-based Reliable Synchronous Collaboration"*, IFIP/IEEE – International Conference on Communication Technology (ICCT), 2000.

[11] R. Piantoni and C. Stancescu, *"Implementing the Swiss Exchange trading system"*, Digest of Papers, 27th

International Symposium on Fault-Tolerant Computing Systems, pages 309-313, July 1997.
[12] B. Charron-Bost, X. Defago, and A. Schiper, *"Time vs space in fault-tolerant distributed systems"*, Proceedings of the Sixth International Workshop on Object-Oriented Dependable Systems, January 2001.
[13] K. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky, *"Bimodal Multicast"*, ACM Transactions on Computer Systems, 1999.
[14] NS network simulator. Available at http://www.isi.edu/nsnam/ns/
[15] Developed by Yunxi Shi, Washington University at St. Louis. Available at http://www.arl.wustl.edu/~sherlia/rm/pgm.tar.gz
[16] K. Yamamoto, Y. Sawa, M. Yamamoto and H. Ikeda, *"Performance evaluation of ACK-based and NAK-based flow control schemes for reliable multicast"*, in proc. IEEE TENCON2000, pp. I.341-I.345, Sept. 2000
[17] K. Miller, K. Robertson, A. Tweedly and K. White, *"StarBurst Multicast File Transfer Protocol (MFTP) Specification"*, <draft-miller-mftp-spec-03.txt>, IETF draft, April 1998.
[18] B. Whetten, S. Kaplan and T. Montgomery, *"A high performance totally ordered multicast protocol (RMP)"*, in Theory and Practice in Distributed Systems, Springer Verlag, vol. LCNS 938, August 1994.
[19] C. Papadpoulos, G. Parulkar and G. Varghese, *"LMS: A Router-Assisted Scheme for Reliable Multicast"*, Submitted for publication to IEEE/ACM Transactions on Networking.
[20] S. Paul, K. Sabnani, J. Lin, and S. Bhattacharyya, *"Reliable Multicast Transport Protocol (RMTP)"*, IEEE Journal on Selected Areas in Communications, Vol. 15 No. 3.
[21] H. Varian and J. Mackie-Mason, *"Generalized Vickrey Auctions"*, Technical report, University of Michigan, 1995.

[22] S. Pingali, D. Towsley, and J. Kurose, *“A comparison of sender-initated and receiver-initated reliable multicast protocols”*, in IEEE JSAC, Vol. 15, No. 3, April 1997.
[23] Distributed Systems Department Collaboration Technologies Group, *“Introduction to the MBone”*, http://www-itg.lbl.gov/mbone/