PGM Protocol– Pragmatic General Multicast
Its application in file transfer service

Luis Marrone
Laboratorio de Investigación en Nuevas Tecnologías Informáticas,
Facultad de Informática
Universidad Nacional de La Plata
La Plata (1900), Pcia de Buenos Aires, República Argentina
lmarrone@info.unlp.edu.ar

María Claudia Abeledo
Universidad CAECE, Universidad Nacional de La Plata
La Plata (1900), Pcia de Buenos Aires, República Argentina
cabeledo@mail.linti.unlp.edu.ar

Abstract

The aim of this article is to explain new projects taking place about secure file transfer techniques by multicast. In this matter we will analyze Pragmatic General Multicast as a way to obtain better results in terms of transfer rate and a development in the way of using net resources. To conclude, there will be a comparison among the results of this specific file transfer and the FTP over TCP conventional technology.

1. Introduction

Nowadays the most known transport protocol is TCP. Implementing multicast reliable scenery should sent messages using IP multicast and simulate how TCP reach when receives the ACK ok this.

A reliable multicast protocol based in ACK end to end would be impossible to achieve because it will not allow scalability. Each reception will produce a confirmation, so the source will receive a huge quantity of messages producing and ACK's implosion. Through the sending of more messages back to the source (by the increase of receivers) the source and the links in the middle will collapse because of their saturation.

Apart from that, the memory should be proportional to the numbers of receivers in order to keep stable the storage state of them.

2. ¿What is PGM?

Pragmatic General Multicast is a multicast transport protocol for applications which require data provided by a source to many receivers.

PGM ensures that a receiver will be able to receive every packet of transmission, to repair or detect an unrecoverable lost packets from the group in which it participated. PGM scalability can be obtained through a hierarchical way: FEC Forward Error Correction, NAK elimination and NAK suppression (Negative ACK policy to elimination and suppression).

In the normal progress of data transfer, the source sends sequenced packets of data via multicast (ODATA) and the receivers send selective ACK negatives for each packet that has been detected as lost in a sequence via unicast. These NAKs are sent hop by hop by the net elements to the source and it sends via multicast a confirmation of reception (NCF). Net elements send NAK´s up to the source that originated the package up to his potential of reception. The restored data (RDATA) can be provided by the source and the designated local repairer (DLR) in response to a NAK.

Later we will analyze transport reliability

NAKs corresponding to similar non received packets will not be repaired by NAKs suppression policy. The receivers will act as if this NAK was received, even if it was not. It doesn't matter to the source if one or more NAKs about the same packets have arrived. The receivers are able to know other NAKs sent through the confirmation of NAKs multicast reception. It is made by a NCF packet which is sent via multicast by the source. There is a configurable delay to send NAK´s. It prevents from implosion (as well as NAK suppression.

DOI 10.1109/CERMA.2007.131

Nevertheless, the implosion must not happen, so the delays should increase according to the receivers but, too much longer delays may produce inconveniences like an enormous window transmission and the impossibility of a receiver to send a NAK before the end of the session.

Another way to improve scalability is the hierarchic way. A tree is built for a multicast reliable session, formed by as many receivers as special intermediary nodes. NAK or ACK messages are sent from a node to the superior in the hierarchic tree which adds or eliminates the information before sending it to the superior twigs. If many inferior nodes lost a packet of information, only one NAK will be sent to the superior part of the tree. This is a NAK suppression for the same packet.

The hierarchic way can be used for something else. It can restrict the sending of reparation in order to send only to those sub trees that contain a receiver in need of reparation. As the structure grows, the lost packages will be proportional to the number of receivers, this is a real trouble. For example: with a 1.000.000 of receivers and a lost packet percentage of 0.01%, the probabilities that every node receives the packet sent is less than 10^{-43}.

This means that any node of the net will loose the packet. Each packet must be sent at least twice, so as to every receiver receive every packet. This situation reduces to a half the use of bandwidth.

FEC policy, in order to prevent this situation, should allow the correction of loses through the different receivers. For example: if a receiver loses his packet 1, another loses his packet 2, only one packet of reparation containing packets 1 to 7 will allow both receivers to repair the mistake.

PGM uses a hybrid scheme including:

◊ Suppression

◊ NAK elimination

◊ Constraint Forwarding: compelling the ascending flow of information in the tree.

◊ FEC (Forward Error Correction) to reach scalability.

The hierarchy is built based on net elements which are able to transmit through PGM. These elements are typical designed routers in order to support PGM over IP multicast, so PGM is less efficient over some elements that were not built to support this protocol.

The senders send periodically an SPM (Source Path Message) to set up PGM hierarchy (it is also useful for other purposes that we will analyze later. SPM packet has the address of the node which is just over it. Net elements, when they send an SPM, replace this address with is own so as the sub nodes can know the node just over it.

Net elements that do not support PGM can operate transparently among PGM nodes with this technique. If multicast routing changes of PGM elements make operations outside PGM, SPM is going to help with the actualization of PGM tree.

PGM distribution tree grows when net elements designed to support PGM are just a few, so in terms of scalability they will have a more critical role by suppression and FEC.

PGM only requires multicast transmission from the sender to the receiver. PGM makes an efficient use of backchannel's bandwidth, producing better results in asymmetric nets. It has a high channel capacity from the sender channel to the receiver's but they have a restricted back channel from the receiver's to sender's.

PGM does not work with applications dependant on ACK´s sending policy to groups of receptors already known. Neither uses different sources.

On the other hand, PGM allows the receivers to join or leave the group whenever they want, leaving reliability only in the present transmission window. It is more efficient for applications that can support some levels of recuperation in the end, than those that recover lose. This does not mean that the unrecoverable lost packages are going to be recovered by PGM. Reliable transmissions are expected only if the sender did not enter in the transmission window in an offensively way. The applications in PGM multicast can choose:

- If the previous information was sent successfully, it can continue sending new information, instead of enumerating
- Set up the importance of sending information to satisfy new receivers (that have joined the group).

PGM tree is limited to the utilization of net elements that support the protocol of that tree. A multicast PGM source determines a sequence of data packets for the receivers (ODATA). The receivers send a NAK by unicast to the immediate superior node of the tree when they realize that the packets were lost in the sequence. That node confirms NAKs' reception to its receivers and to its associated nodes by a multicast NAK confirmation. The repairs are generated by a source or a DLR in response to a NAK. This happens because PGM net elements do not start ODATA packets or provide reparation. A repair consists on a resending packet or a FEC packet, depending of the session parameters. Before sending a NAK, receivers set a random back off and suppress the NAK if the NCF has already received the repaired data.

3. Performance

3.1 Memory requirements of network elements.

PGM requires net elements to store information about its status. For example: the status of the path to the source, through the direction of the path to that from a SPM and the multicast session that applies for it.

PGM could run out of memory for every NAK of every session if many PGM sessions are using net elements.

In this case it could end to operate transparently, reducing memory requirements up to a level they can adapt themselves.

This scenery is more likely to take place in routers located in Internet backbones when the numbers of sessions can become streams.

3.2 Use of back channel

PGM includes a number of options to make use of the back channel more efficiently in NAK transmission.

It supports the option OPT_NAK_LIST in order to include enough individual NAKs in the same NAK packet.

A NAK packet, for a sole loose has a length of 56 bytes, while the packet with the option OPT_NAK_LIST is of 64 bytes. However backchannel traffic can be reduced significantly.

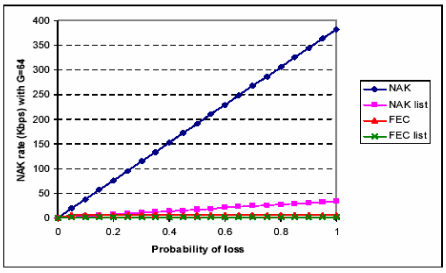


Figure 1

Figure 1: NAKs' rate from a sole receiver vs. loose probability. 1500 bytes were sent 874 times per second for a transfer data rate of 10 Mbps. A group size of 64 has been used for FEC.

In figure 1 it's shown the way the traffic is minimized with opposite NAK, with a group size of G=64 using loose options and FEC.

Net utilization

PGM and IP header overhead limit net utilization to the 97.1%.

Other factor is SPM messages, considering that they are sent by a transfer rate 1 per second. Even in 10 Mbps the impact is insignificant.

Even in dial up the utilization of the net is over 90%, in spite of the increasing of SPM in 5 per second.

High speed transmission resources

Now we are going to analyze high speed transmission using PGM. A PGM implementation can not be sent by high transfer rates if the whole system is unable to send IP packets in the same transfer rate. This requires adequate NICs' net buffers establishments and kernel adjustments.

Additionally, emission buffers for PGM transmission windows are not worrying in low transfer rates. A transmission window of 30 seconds with a transmission rate of 10 Mbps requires more than 375 Mb of space in the buffer.

If the transmission is not entirely restricted to the physic memory, page mistakes can degrade performance severely. In order to ensure high speed operation it is needed to have enough RAM memory in order to support the whole transmission window.

The receivers may have similar memory requirements when the FEC is used. The case in which a packet is lost by every group in the transmission, every k-packet will be needed to be detoxified. The consequence of this situation is that the packet must be kept till the fulfilling of groups reception. So, the receivers must keep (k -1)/ k window transmission. Once again if they are not in the memory, the performance will be degraded

A PGM receiver should warn the loose in the buffer, it is possible, because PGM uses route IP and does not have any construction for flow control.

If any sender is sending at high speed, receiver buffer will overflow or a net congestion will take place.

The reaction to this congestion must be handled in a different way than the loose produced by the buffer overflow, so the sender transfer rate will be adjusted. In order to reduce buffers' overflow in the receivers, the applications will need to know the acquisition process for the reading of PGM packets from the net. They must also allow that PGM does this process with a high priority compared to other task levels in the corresponding applications.

Also, the operative system connected to the buffers require modifying the sizes, all this in order to handle the capacities that PGM traffic can reach.

The first approach

We get the source code with granted stability and the documentation to the implementation of the PGM

host to some sendings of Free BSD V in www.iet.unipi.it/~luigi/pgm.html.

It also contains:

- Complete implementation of the basic host mechanism of PGM (FEC and DLR are not implemented yet)
- Experimental option to FIN options.

It allows demonstration with a diskette under FreeBSD version.

These characteristics allowed using PGM and TCP sockets. A simple application of the test, pgmcat is also available allowing transference of simple available file in multicast modality, with complete congestion control. We count on TCPdump as a tool for traffic capture on the net. It's important to consider the fact that it is available for UNIX platforms and Windows' with similar features.

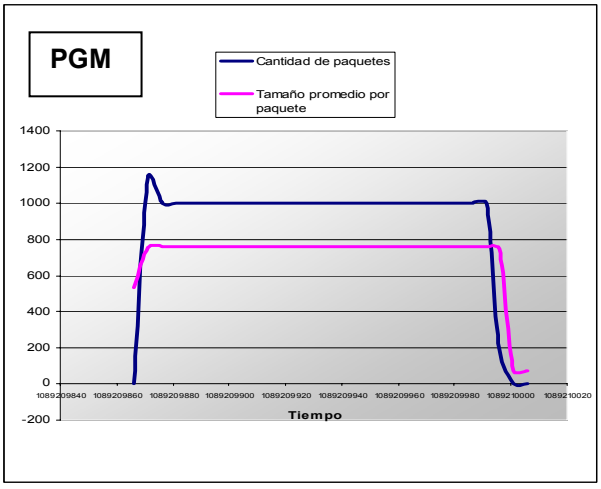
4. Measure over file transfers

4.1 Description of the work

FTP (File transfer protocol) is a commonly used application. We took into consideration the client/Server scheme through standard parts. We used FTP with a server that allowed anonymous FTP. Ports: 20/21.

We did many file transfers inside the CAECE University labs' LANs using an already installed FreeBSD 4.0 version with some modifications to support PGM.

4.2 Obtained Results

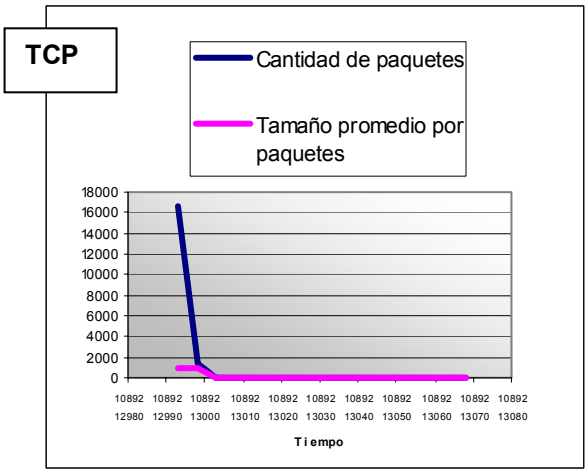


PGM	Time	Amount of packets	Average size per packet
	1089209866	3	533,33
	1089209871	1146	757,08
	1089209876	1001	761,31
	1089209881	1002	760,62
	1089209886	1001	761,31
	1089209891	1002	760,62
	1089209896	1001	761,31
	1089209901	1002	760,62
	1089209906	1002	760,62
	1089209911	1001	761,31
	1089209916	1002	760,62
	1089209921	1001	761,31
	1089209926	1002	760,62
	1089209931	1001	761,31
	1089209936	1002	760,62
	1089209941	1001	761,31
	1089209946	1002	760,62
	1089209951	1002	760,62
	1089209956	1001	761,31
	1089209961	1002	760,62
	1089209966	1001	761,31
	1089209971	1002	760,62
	1089209976	1001	761,31
	1089209981	1002	760,62
	1089209986	1002	760,62
	1089209991	1001	761,31
	1089209996	195	747,51
	1089210001	1	72
	1089210006	2	72

Results obtained were totally different for the case of transmission using TCP:

From Server (listening clients)

Example for one client:



TCP Time	Amount of packets	Average size per packet
089212993	16549	1035,86
1089212998	1294	1032,72
1089213003	0	0
1089213008	0	0
1089213013	0	0
1089213018	0	0
1089213023	0	0
1089213028	0	0
1089213033	0	0
1089213038	0	0
1089213043	0	0
1089213048	0	0
1089213053	0	0
1089213058	0	0
1089213063	0	0
1089213068	8	44,75

Effective time transfers are lower to TCP than to PGM . Another job about this job is added.

The next job shows that through TCP under Windows Platform, the transference is faster but in a total inefficient use of bandwidth.

The disadvantages will be analyzed in the conclusions.

The job was made over data took from the FTP server (SERVER.DUMP) and from the client (CLIENTE.DUMP).

Server direction: IP: 192.168.4.160 (windows 2000 server)

Client direction: IP: 192.168.4.161 (windows 2000 professional)

Used command : Microsoft Windows

C:/WINDUMP -i 2 -w [data file] -p tcp

16622 packets seen, 16622 TCP packets traced

elapsed wall clock time: 0:00:00.079313, 209574 pkts/sec analyzed

trace file elapsed time: 0:00:09.872585

TCP connection info:

2 TCP connections traced:

TCP connection:

First packet: Fri Nov 12 19:13:26.174958 2004

Last packet: Fri Nov 12 19:13:36.047543 2004

Elapsed time: 0:00:09.872585

Total packets: 7

TCP connection 2:

First packet: Fri Nov 12 19:13:26.182005 2004

Last packet: Fri Nov 12 19:13:35.859631 2004

Elapsed time: 0:00:09.677626

Total packets: 16615

The graph is similar to the exposed under FreeBSD, but with a maximum use of the available bandwidth.

5. Conclusions

We can conclude after analyzing lab tests and according to the theoretical environment included before that:

1. PGM uses rationally the effective bandwidth in file transfers and keep it constant along the entire transfer process. As it can be seen from the table, each client received a constant file transfer rate.

2. PGM assures that the files can be transmitted simultaneously. We mean those files whose clients belong to the same multicast group. Each of them received the file transfer in the **same time interval**. Considering point 1, all the clients received the file transfer **simultaneously with constant rate. This is very much important indeed regarding file transfer via multicast related to file transfer via TCP in a unicast way.**

3. It is not possible to get these transfer rates features with another transport protocols, as in the case of TCP.

4. Anyway it is important to remember that PGM is in an experimental stage, even for many mas-

sive applications and it can't be used in all platforms as is the situation with another protocols.

5. The file transfer carried on through TCP shows that in a first attempt there is a lower transfer time, but because of TCP congestion control that resumes in a decrease of transfer rate spoils the initial best performance.

6. Besides, under TCP we have no simultaneous transmission. So that each connection requires the whole net resources.

7. PGM, assuring a simultaneous transmission, achieves a file transfer service incomparable to TCP.

8. The Labs where these tests were carried on, are considering PGM adoption. This could be the initial step towards a more ambitious PGM implementation, including voice and video transmission.

6. Bibliography and Webliography

Distributed Systems – Concepts and Design – G. Colouris, J. Dollmore, T. Kindberg.- Second Edition - Addison Wesley

Developing IP Multicast Networks – Cisco Systems – Beau Williamson -Cisco press.

Manual de FreeBSD – FreeBSD Documentation Project.

RFC 3208 – PGM Reliable Transport Protocol Specification – T.Spakman, J. Crowcroft y otros.

RFC 959 - File Transfer Protocol -J. Postel, J. Reynolds

RFC 793 - TRANSMISSION CONTROL PROTOCOL - DARPA INTERNET PROGRAM

RFC 2236 - Internet Group Management Protocol - W. Fenner - Xerox PARC

L. Rizzo, "PGMCC: A TCP-friendly Single-Rate Multicast Congestion Control Scheme", Proc. of ACM SIGCOMM -2000.

L. Rizzo, "A PGM Host Implementation for FreeBSD", http://www.iet.unipi.it/~luigi/pgm.html

M. Psaltaki, R. Araujo, G. Aldabbagh, P. Kouniakis, and A. Giannopoulos, "Pragmatic General Multicast (PGM) host implementation for FreeBSD.", http://www.cs.ucl.ac.uk/research/darpa/pgm/PGM_FINAL.html

[3] The PGM Reliable Multicast Protocol – J. Gemmell, T.Speakman, J. Crowcroft y otros.

Implementation and Evaluation of Pragmatic general Multicast – Derek Chen-Becker, Manj Singla.- 2002

CISCO IOS SOFTWARE RELEASES 12.0 T- PGM Router Assist - http://www.cisco.com/en/US/products/sw/iosswrel/ps1830/products_feature_guide09186a00800879a2.html#wp5038

http://www.rediris.es/rediris/boletin/54-55/ponencia12.html - Uso de IGMPv3 para evitar ataques DoS en redes multicast- . F. Gómez Skarmeta,A. L. Mateo, P. M. Ruíz

N. Ishikawa, N. Yamanouchi, O. Takahashi. "IGMP Extension for Authentication of IP Multicast Senders and Receivers". Internet-Draft. Agosto 1998.

W. Fenner. "Internet Group Management Protocol. Version 2". RFC 2236. Noviembre 1997.

A. Gómez Skarmeta, A. L. Mateo Martínez, P. M. Ruiz Martínez. "Access Control in Multicast Environments: an Approach to Senders Authentication". Proceedings of the IEEE LANOMS'99, pp 1-13. 1999

B. Cain, S. Deering, A. Thyagarajan. "Internet Group Management Protocol. Version 3". Internet-Draft. Noviembre 1999.

B. Quinn. "SDP Source-Filters". Internet-Draft. Mayo 2000.

M. Handley, V. Jacobson. "SDP: Session Description Protocol". RFC 2327. Abril 1998.

S. Kent, R. Atkinson. "Security Architecture for the Internet Protocol". RFC 2401. Noviembre 1998.