Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Redundancy Elimination in GPRS network

QIU Qin-long[1], ZHANG Dong-mei[2], MA Jian[2], ZHANG Da-jiang[2]
([1]National Key Lab of Switching Technology and Telecommunication Network,
Beijing University of Posts & Telecommunications, Beijing 100876, China)
([2]Nokia Research Centre, Beijing 100013, China)
E-mail: qiuqinlong@sohu.com; dongmei.zhang@nokia.com; jian.j.ma@nokia.com; dajiang.zhang@nokia.com
Received Mar. 1, 2005; revision accepted Oct. 17, 2005

Abstract: The mechanisms of TCP's retransmission and reset will result in redundant packets. These redundant packets are often sent unnecessarily to the user over a slow last-hop link delaying useful traffic. This is a problem for wide-area wireless links, such as General Packet Radio Service (GPRS), because unnecessary transmissions waste already limited radio bandwidth, battery power at the mobile terminal and incurs monetary cost due to charging by data volume. The paper first describes a GPRS model, then discusses how to eliminate the redundant packets in GPRS network and presents the simulation results in Network Simulation 2 (NS 2). The more traffic is, the more the network can benefit. In heavy traffic, it can even get more than 30% improvement in throughput. Average delay and loss percent are also lowered.

Key words: GPRS, TCP, Redundancy Elimination, Network Simulation 2 (NS 2)
doi:10.1631/jzus.2006.A0477 **Document code:** A **CLC number:** TU375

INTRODUCTION

GPRS is a packet-switched wireless wide area network deployed worldwide (Brasche and Walke, 1997). Performance evaluation of GPRS is an active research area, especially TCP performance. The TCP mechanism will produce redundant packets in wired-and-wireless network because of the bad wireless environment; obviously, these redundant packets will harm the GPRS network performance. The paper clarifies how redundant packets are produced and eliminated. We will evaluate the network performance according to its throughput, average delay and loss percent mainly.

The paper first introduces GPRS network, TCP mechanism and NS 2, then introduces GPRS network model and traffic models, and then detailedly describes the mechanism of Redundancy Elimination in the last Section 3, where some test results, main conclusions, including the benefits and drawback of Redundancy Elimination, are given.

GPRS network

GPRS is a new bearer service for the Global System Mobile Communication (GSM) that greatly improves and simplifies wireless access to packet data networks, e.g., to the Internet (GSM 03.02, 1998). GPRS adds two new nodes to GSM: SGSN and gateway GPRS support node (GGSN) (Fig.1) and applies a packet radio principle to efficiently transfer user data packets between mobile stations and external packet data networks (GSM 03.02, 1998).

GPRS improves utilization of radio resources, offers volume-based billing, higher transfer rates, shorter access times, and simplifies the access to packet data networks (GSM 03.05, 1999). The European Telecommunications Standards Institute (ETSI)'s standardisation of GPRS in recent years is of great interest to many GSM network providers.

TCP

TCP is intended for use as a highly reliable host-to-host protocol between hosts in packet-switched computer communication networks, and in inter-

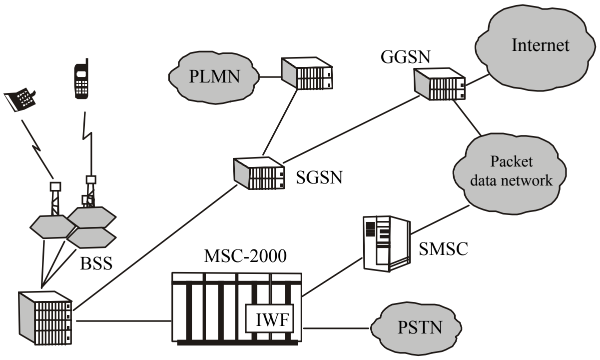

Fig.1 GSM and GPRS network

connected systems of such networks (IETF RFC 793, 1981).

Internet users often abort their transfers in progress, for example by clicking on the “Reload” button or another link in a Web browser. Analysis of backbone Internet traces shows that 15%~30% of all TCP connections are aborted via a reset (MAWI, 2003). In GPRS network, most of the traffic will be based on the TCP, so it will experience similar behaviour. Packets from aborted transport connections are often sent unnecessarily to the users over a slow last-hop link, just like the Um reference point of the GPRS network.

The redundant packets are generated not only from the TCP’s reset, but also from the normal case in TCP. For example, mobile station (MS) receives the packet x from the Internet (Fig.1), because of the Um interface is a slow link, the retransmission timer times out before receiving the acknowledgement packet of packet x. Thus the Internet will resend packet x according to TCP’s mechanism (Comer, 1995). The packet x is an obviously redundant packet that should be dropped instead of being sent to MS.

In this section, we discuss two causative reasons for the generation of redundant packets and will continue to discuss how to eliminate these redundant packets in GPRS network in the following sections.

NS 2

NS 2 is a discrete event simulator targeted at networking research and provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. The current highest version of NS 2 is NS 2.28 (released Feb. 3, 2005). NS 2 used in this paper is NS 2.26.

MODEL OF THE NETWORK

In this section, we describe the GPRS network model in NS 2. As the traffic will also affect the simulation results, the GPRS traffic model is also described here.

In NS 2 simulation tools (NS 2 notes and documents, VINT project, http://www.isi.edu/nsnam/ns), the network elements are denoted and implemented as agent, and the interface as queue. Fig.2 lists the functionalities of network elements and the interfaces respectively, which are described in detail in (Qiu *et al.*, 2004).

The agents, i.e. the network elements, deal with the packet processing, e.g. segmentation and reassembly. The queues, i.e. the interfaces, are responsible for queuing, scheduling, and resource management (Table 1).

Table 1 GPRS key elements in NS 2

Node	Agent	Queue
MS,	MSAgent,	Um_DL_Queue, Um_UL_Queue,
BSS,	BSSAgent,	Gb_DL_Queue, Gb_UL_Queue,
SGSN,	SGSNAgent,	Gn_DL_Queue, Gn_UL_Queue,
GGSN,	GGSNAgent	Gi_DL_Queue, Gi_UL_Queue
Server		

The QoS supporting in GPRS network is simulated in our simulation model. There are five different QoS classes, which will be treated differently in Gb and Um (Heiskari, 2003).

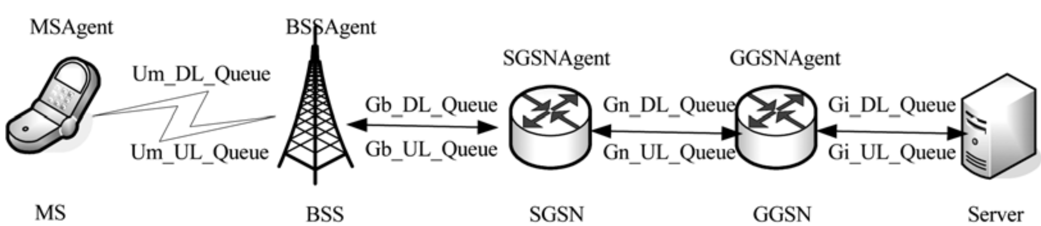

Fig.2 GPRS network model in NS 2

The following network elements are included in our simulated GPRS network: (1) 1 server (to provide HTTP, FTP and signalling services); (2) 1 GGSN; (3) 1 SGSN; (4) 1 BSC; (5) 30 BTSs.

There are three sectors in each BTS and three time slots (TSLs) in each sector, altogether 9 GPRS TSLs in down link (DL) direction per BTS; while in up link (UL) direction, there are only 3 GPRS TSLs in each BTS as GPRS traffic is asymmetric normally.

Four different coding schemes, CS-1 to CS-4, are defined for the GPRS Radio Blocks carrying RLC data blocks. For the Radio Blocks carrying RLC/MAC Control blocks code CS-1 is always used. The exceptions are messages that use the existing Access Burst (see 3GPP TS 45.003, e.g. Packet Channel Request) (MAWI, 2003). We only use one coding scheme CS-2 that is not true in the real world. It should be changed in further study.

In Gb interface, each type of class is scheduled by WRR (Weighted Round Robin) algorithm based on different priority weight, and stored in different size buffer (WRR, 1999). In air interface, each type of class is scheduled by WFQ (Weighted Fair Queue) algorithm. The QoS class definition and related QoS parameters are listed in Table 2. The parameters' value of quantum and weight is defined according to the requirements of different applications. In the Gb interface, the bigger the quantum is, the more chance the class gets to be scheduled. In the air interface, the smaller the weight is, the more chance the class gets to be scheduled (Montes *et al*., 2003).

Several agents such as MSAgent, BSSAgent, SGSNAgent and GGSNAgent are used to handle data packets transmission on the MS, BSS, SGSN and GGSN respectively. Accordingly, several classes, class MSAgent, class BSSAgent, class SGSNAgent, class GGSNAgent are defined to identify each agent. Since these agents have some common features, a base class GPRSAgent is defined, and the other agents are derived from GPRSAgent which is derived from NS Agent.

METHODOLOGY

In this section, the mechanism of Redundancy Elimination is introduced in detail. The mechanism of Redundancy Elimination is divided into two parts according to different causative reasons of generating the redundant packets, i.e. Fast Reset and Normal Redundancy.

Fast Reset

The standard TCP receiver generates reset (RST) packets after receiving (and discarding) packets on an aborted connection. We propose an algorithm for preventing delivery of aborted data over the last-hop link Um_DL_Queue, which is named Fast Reset. When SGSNAgent receives a reset packet in up link (UL), drop all packets that belong to the same flow to that of the reset packet in SGSNAgent and BSSAgent. Thus, these unnecessary packets will not be transmitted to MS saving limited radio bandwidth and battery power of MS.

Normal Redundancy Elimination (Fig.3)

When server sends data ($seq=x$) to MS, cell reselection causes long buffer time. When RTO expires, server will have to resend the data ($seq=x$). After a while, MS receives the data ($seq=x$) and responds with an ACK packet. This ACK packet arrives at Gb interface and finding that there is duplicate packet here ($seq=x$), drops this duplicate packet. If by the time when the duplicate data ($seq=x$) arrives at SGSN, SGSN has already received the ACK ($ack=x+1$), the duplicate data ($seq=x$) will be dropped here. The red scenario is implemented in DL-SGSN-Queue; the green scenario is implemented in SGSNAgent (*seq* is the TCP's sequence number; *ack* is the TCP's acknow-

Table 2 GPRS traffic model in NS 2

ClassID	Traffic type	Simulated traffic	Gb interface (quantum)	Buffer size	Air interface (weight)
0	Signalling traffic/Streaming	CBR (constant bit rate)	100	Hard configure	1
1	Interactive traffic 1	HTTP	5	10%	3
2	Interactive traffic 2	HTTP	3	20%	3
3	Interactive traffic 3	HTTP	2	30%	3
4	Background traffic	FTP	1	40%	9

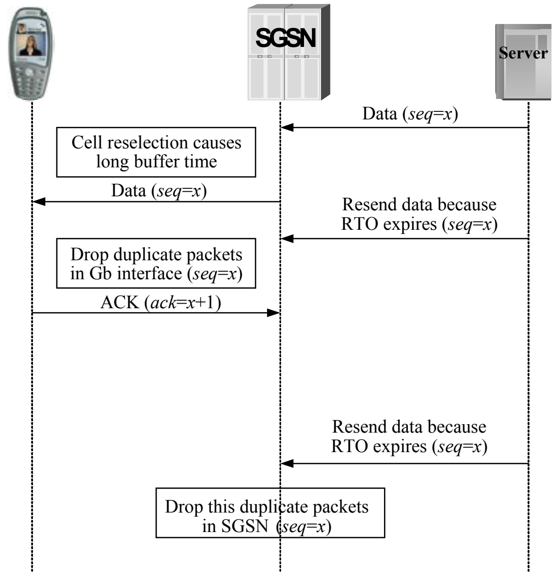

Fig.3 Normal Redundancy Elimination

ledge number).

UL in SGSNAgent

If there is an incoming packet in UL direction in SGSNAgent, check whether its *ptype* is ACK, if it is not, just send this packet (Because the packet is HTTP UL's request data packet). If the packet's type is ACK, it means the packet is ACK packet to down link data (HTTP response). In this case, get the information (flowid, sport, seqno_) of this packet, and then update its session table according to the information. Finally, send the ACK packet to GGSNAgent. In the case of UL data transfer, TCP sender may send duplicate acknowledgments. These acknowledgements should not be dropped, in other words only TCP data packets should be dropped, because these ACK packets may be lost between SGSNAgent and MSAgent due to wireless link error.

DL in SGSNAgent

If there is an incoming packet in DL direction in SGSNAgent, check whether its seqno_ is zero (seqno_ equals zero means it is a SYN in NS 2), if it is, just send this packet (Because the TCP connection has not been built yet). If the packet's type is ACK, it means the packet is ACK packet to UL data (HTTP request). In this case, SGSNAgent also just sends the packet to BSSAgent. In other cases, if its seqno_ is bigger than its session table's ackno_, just send the packet. Or drop the redundant packet and resend a copy of the highest ACK received that helps in increasing the congestion window (CWND) in TCP sender (TCP sender will increase the CWND after receiving the ACK packet (IETF RFC 793, 1981)).

Note: Because we use TCPAgent not FullTCPAgent in NS 2, we can only use the seqno_ instead of ackno_.

For every TCP session the SGSNAgent keeps a list of acknowledged packets in order to support our algorithm. If another ACK packet for the same TCP session is received, it is enough just to update the TCP *ack* number.

If a DL packet's fields are the same as those in the stored data table of its session except that the sequence number is less than the acknowledgement number, then this duplicate packet should be dropped (Tables 3 and 4).

Table 3 UL IP and TCP packet-fields stored by SGSNAgent

Field	Length (bit)
IP source address	32 (Ipv4)/128 (Ipv6)
IP destination address	32 (Ipv4)/128 (Ipv6)
TCP source port	16
TCP destination port	16
TCP acknowledgement number	32

Table 4 DL IP and TCP packet-fields to be compared

Field	Length (bit)
IP source address	32 (Ipv4)/128 (Ipv6)
IP destination address	32 (Ipv4)/128 (Ipv6)
TCP source port	16
TCP destination port	16
TCP sequence number	32

HTTP 1.0 vs HTTP 1.1

In HTTP 1.0, most implementations use a new connection for each request/response exchange. In HTTP 1.1, a connection may be used for one or more request/response exchanges, although connections may be closed for various reasons (Fielding *et al*., 1999).

Prior to the persistent connections, a separate TCP connection was established to fetch each URL, increasing the load on HTTP servers and causing congestion in the Internet. The use of inline images and other associated data often requires a client to make multiple requests of the same server in a short span of time. Analysis of these performance problems and results from a prototype implementation are available (Padmanabhan and Mogul, 1995; Spero, 1994). Implementation experience and measurements of actual HTTP 1.1 (RFC 2068) implementations show good results (Nielsen *et al*., 1997).

Because HTTP 1.0 uses more TCP connections, the redundant packets due to RST packets become less, but the redundant packets due to normal case become more. So it is a little difficult to theoretically evaluate different version HTTP's effect on our algorithm in theory. The test results are discussed in the following section.

TEST RESULTS

Test scenario

The test results are based on the following traffic distribution: (1) 30 signalling traffic; (2) 100×3 HTTP traffic (100 for each class); (3) 100 FTP traffic; (4) Simulation time: 2000 s.

Normal vs Redundancy Elimination (RE)

We evaluate the network performance mainly according to its throughput, average delay and loss percent. Fig.4, Fig.5 and Fig.6 describe the Redundancy Elimination test results respectively based on the scenario above.

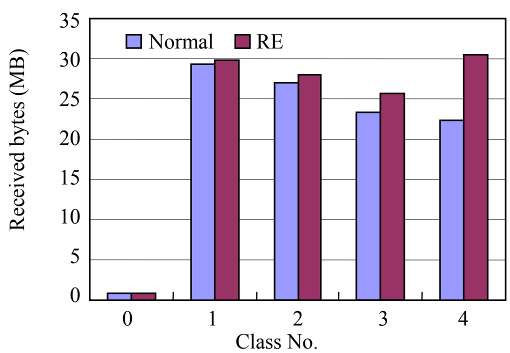

Fig.4 Throughput of normal and RE

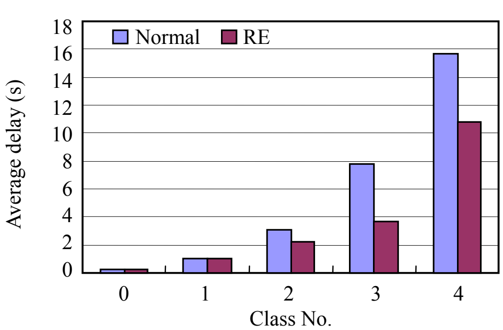

Fig.5 Average delay of normal and RE

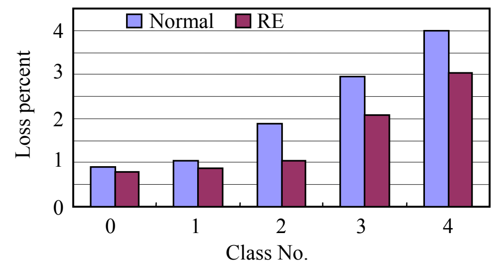

Fig.6 Loss percent of normal and RE

From Fig.4, we can see that the throughput of Redundancy Elimination is improved compared to normal case, especially Class 4, the FTP traffic because Class 4 is the class whose priority is the lowest. If there is not enough network resource, such as wireless resource, it cannot send many packets, while the other four classes with higher priority can still send some packets. Of course, the higher priority it is, the more packets it can send. So the number of packets sent by Class 1 is the most.

Fig.5 shows that the average delay of each class becomes low. Because of the dropping of redundant packets, the packets that are useful get more chance to be transmitted.

Fig.6 also shows that the loss percent of each class becomes lower.

There is one more thing that has to be discussed. In Class 0, the signalling traffic is the class with the highest priority. Many constraints do not affect it, the RED algorithm in SGSN for example. If there is Class 0's traffic, other classes have to wait until it is transmitted completely. So the throughput, average delay and loss percent of normal case are almost the same as that of Redundancy Elimination.

HTTP 1.0 vs HTTP 1.1

HTTP 1.1 has two features compared to HTTP 1.0: persistent connection and pipeline. We set the time of persistent connection to 2 min, which is the default value of Internet Explorer version 5.0 and 6.0. We set the pipeline with two HTTP requests at the value suggested in RFC 2616.

We found that the test result with HTTP 1.1 was almost the same as that with HTTP 1.0, i.e. Redundancy Elimination is of benefit to the throughput, average delay and loss percent of the GPRS network.

ACK vs without ACK

According to the TCP mechanism, after receiver receives a data packet, it has to send an ACK packet back to the sender. In our algorithm, receiver will receive some redundant packets dropped in SGSN, so SGSN should send ACK packets back to the sender. But the appearance of duplicate packets indicates network congestion. So it is hard to say whether or not to send the ACK packet. We test both implementations. The test result in Fig.7 shows the result is almost the same. Though constructing an ACK packet entails some CPU time and wastes some memory (Because we have to remember what ACK we have to construct), we select the implementation that does not send ACK finally.

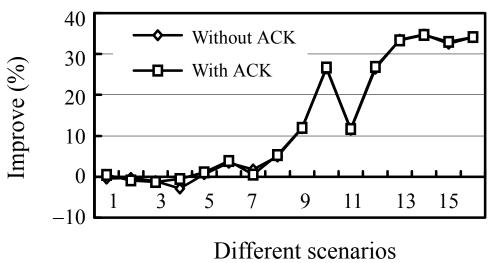

Fig.7 Comparison of different RE implementations

One more thing has to be discussed. In some scenarios, the algorithm harms the performance of the GPRS network. Although it is only a little, we have to tell the reason. In our model, there is cell handover model. Once one user enters the cell handover status, all packets belonging to that user in BTS have to be dropped, that results in the scenario above.

CONCLUSION

From the discussion above, we can see that the Redundancy Elimination algorithm helps improve the network throughput and reduce the average delay and loss percent. Many redundant packets being dropped in BSSAgent and SGSNAgent before being transmitted to MS save limited radio bandwidth, battery power of MS and reduce the monetary cost due to charging by data volume in GPRS network.

Of course, there are some weaknesses in our algorithm. The first thing having to open all DL TCP data packets and all UL ACK packets will consume some CPU resources. When there are redundant packets, the algorithm of dropping packets is a little complicated. It not only needs some CPU resources, but also some new signalling between SGSN and BSS. The session table will charge many memories. So further work will be on how to reduce the size of session table but not affect the performance of our algorithm.

References

Brasche, G., Walke, B., 1994. Concepts, services and protocols of the new GSM phase 2+ general packet radio service. *IEEE Communications Magazine*, (August):94-104.

Comer, D.E., 1995. Internetworking with TCP/IP, Volume 1 (Third Edition). Prentice-Hall International, Inc., USA.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., 1999. Hypertext Transfer Protocol—HTTP/1.1. IETF RFC 2616.

GSM 03.02 (version 7.1.0), 1998. Digital Cellular Telecommunications System (Phase 2+), Network Architecture.

GSM 03.05 (version 8.0.0), 1999. Digital Cellular Telecommunications System (Phase 2+), Technical Performance Objectives.

Heiskari, H., 2003. 3GPP R97 GPRS and 3G R99 QoS. Nokia NRC (unpublished).

IETF RFC 793, 1981. Transmission Control Protocol.

MAWI Working Group Traffic Archive, 2003. Packet Traces from WIDE Backbone.

Montes, H., Lequeux, F., Long, D., Sotta, P., 2003. Analysis of the E-2-E QoS Solution for GPRS. GERAN QoS Project.

Nielsen, H.F., Gettys, J., Baird-Smith, A., Prud'hommeaux, E., Lie, H., Lilley, C., 1997. Network Performance Effects of HTTP/1.1, CSS1, and PNG. Proceedings of ACM SIGCOMM'97, Cannes, France.

Padmanabhan, V.N., Mogul, J.C., 1995. Improving HTTP latency. *Computer Networks and ISDN Systems*, **28**:25-35. Slightly Revised Version of Paper in Proc. 2nd International WWW Conference'94: Mosaic and the Web, Oct. 1994. Http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/DDay/mogul/HTTPLatency.html.

Qiu, Q.L., Zhang, D.M., Ma, J., 2004. GPRS Network Simulation Model in NS-2. The Joint Conference of 10th Asia-Pacific Conference on Communications and 5th International Symposium on Multi-Dimensional Mobile Communications.

Spero, S.E., 1994. Analysis of HTTP Performance Problems. Http://sunsite.unc.edu/mdma-release/http-prob.html.

WRR, 1999. Weighted Round-Robin Schedulers for Advanced QoS in High Speed Networks. Computer Architecture and VLSI Systems (CARV) Lab, Institute of Computer Science (ICS), FORTH Science and Technology Park of Crete, Heraklion, Crete, Greece.