An Investigation into Multicasting

Nikolay Manolov†, Adel Gamil†, and Stephan Wong‡

†Department of Computer Systems,
Faculty of Computer Systems and Control,
Technical University of Sofia,
Sofia, Bulgaria
{monolov,adi4ever}@Dutepp0.ET.TUDelft.NL
http://www.tu-sofia.bg

‡Computer Engineering Laboratory,
Electrical Engineering Department,
Delft University of Technology,
Delft, The Netherlands
J.S.S.M.Wong@ewi.tudelft.nl
http://ce.et.tudelft.nl

Abstract— **Nowadays, the Internet is being utilized increasingly more for communicating real-time (multimedia) data to multiple recipients, i.e., multicasting. Our goal is to investigate the network processing operations that are involved in supporting multicasting and to determine which operations are time-critical. Traditionally, investigations on such routers have been focussing on packet forwarding and functionalities of other protocols found in lower layers of the TCP/IP protocol stack. In this paper, we present our investigation of the Internet Group Management Protocol that belongs to a higher layer of the TCP/IP protocol stack. The methodology encompasses the creation of benchmarks reflecting the IGMP functionalities and subsequently the profiling of the created benchmarks in order to determine the most time-critical operations. Next to determining the time-critical operations, the mentioned simulation environment also provided results on instruction distribution, cache behavior, and branch prediction accuracy. Preliminary results suggest that the operations involved in multicasting are memory-intensive as about 50% of the operations are memory-related operations. Such results can be utilized in the design of future network processors.**

Keywords— **Benchmarking, profiling, multicast, IGMP, videoconferencing.**

I. Introduction

In today's world, increasingly more people are connected to each other via computer networks and are demanding more types of services in order to communicate with each other. Depending on the type of service that is required, different requirements are placed on the utilized networks. The foremost requirement is the speed in which data is transmitted through the network. Additional requirements include reliability, security, etc. Currently, we are witnessing a new type of service that needs support, namely "one-to-many" transmission[1] in which multicasting plays an important role. Example applications include video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news broadcasts. A requirement posed by the mentioned applications is to transmit data from a single source to multiple receivers. In this scenario, it is impractical for the sender to transmit the same data multiple times (i.e., as many times as there are members), because this would place a huge burden on the network utilized to relay the data. If there are thousands of receivers, even low-bandwidth applications benefit from using multicast. In these applications, the only way to send to more than one receiver simultaneously is by using IP Multicast. It is a bandwidth-conserving methodology that reduces traffic by simultaneously delivering a single stream of information to thousands of recipients.

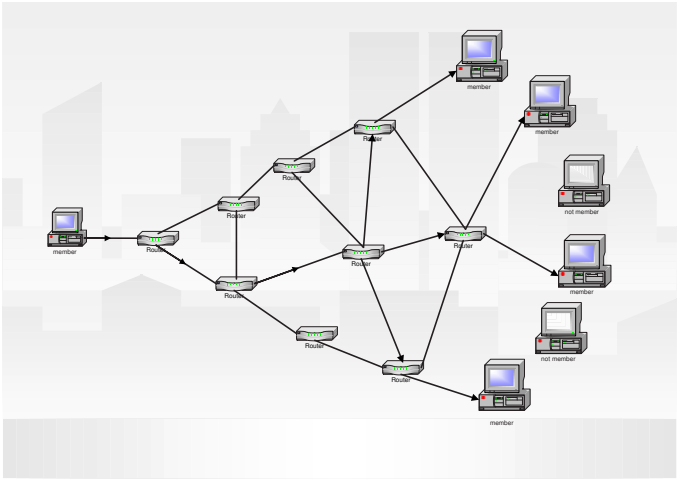

Fig. 1. IP Multicast saves bandwidth by sending packets only to the group members

IP Multicast delivers source traffic to multiple receivers without adding additional burden on the source or the receivers while using the least network bandwidth of any competing technology. Multicast is based on the concept of a group. An arbitrary group of receivers expresses an interest in receiving a particular data stream (see Figure

[1]The term one-to-many is used to the transmission of data from *one* source to *many* destinations.

1). This group does not have any physical or geographical boundaries. The hosts can be located anywhere on the Internet. Multicast packets are replicated in the network by routers that employ multicast-aware routing protocols, e.g., MOSPF, PIM, and DVRMP. In this approach, routers know the exact addresses of group members allowing them to more efficiently route the multicast packets. Therefore, hosts interested in receiving data from a particular group must join the group. This is achieved by utilizing Internet Group Management Protocol (IGMP). More specifically, hosts inform the closest router they are connected to that they are interested in receiving multicast messages sent to certain multicast groups. The IGMP also provide support for the routers to periodically check whether the members of a certain group are still active. Consequently, IGMP must be available in last hop routers, i.e., the router directly connected to the group member's network, and host operating system network stacks, and it must be used by the applications running on those hosts. The latest version of the protocol is IGMPv3. IGMPv3 supports applications that explicitly signal sources from which they want to receive traffic. The benefits from this capability are better bandwidth utilization and security.

Our goal is to investigate the processing requirements for routers in order to support IGMPv3. This will be achieved by creating benchmarks based on existing source codes. Profiling of the benchmarks will help us to identify where the protocol bottlenecks are. Furthermore the simulation results from will also provide an insight to determine eventual hardware implementations of time-critical functions.

This paper is organized as follows. Section II discusses in more detail what multicasting is and the associated protocol IGMPv3. Subsequently, we take a brief look at the benchmarking. In Section III, we explain the details of our benchmarking, e.g., the simulator that we use, the datasets, the exact data structures in the benchmark, etc. Section IV presents the simulation results. Section V presents the conclusion of this paper.

II. Background

In this section, we provide that background on multicasting and discuss it in more detail. More specifically, we present the Internet Group Management Protocol (IGMP), multicast addressing, and the benefits of the latest version of IGMP, namely IGMPv3. Finally, we provide some background on benchmarking.

The Internet Group Management Protocol (IGMP)
Multicast is a point-to-multipoint routing technique that allows IP traffic to be sent from one source to multiple recipients. There are many reasons which make the multicasting capability desirable. The first reason and advantage is that multicasting decreases the network load. Assume that an application, e.g., a stock ticker, wants to transmit packets to hundreds of hosts. It is impractical for the sender to generated hundreds of the same packets and then route these packets through the network. Additionally, this approach will also generate a huge network load. In this case, multicasting reduces network load by replicating the packet(s) at the forks of the multicast delivery tree only when it is necessary. The second reason pertains resource discovery in which a router would query other routers to determine their services. Again, it is impractical for such a router to send the same query to each and every other router. Similarly, using multicast only a single query needs to be sent to the multicast group comprising the router to which a query must be sent.

In a small local area network, multicasting can be implemented by introducing a central multicasting-aware router that keeps track of the number of multicast groups and which hosts belong to which group. In a wide area network, e.g., the Internet, the approach of using a single central router is impractical and not efficient. The solution is to turn some routers into multicast routers. Such routers must be aware that certain hosts that they are directly connected to are members of certain groups and they must maintain this information. The manner in which to establish multicasting on the Internet has been specified in the Internet Group Management Protocol (IGMP). Additionally, specific IP addresses were defined to specify multicast groups. Thus, by sending a packet to such a multicast address, all members belonging to that group will receive the packet. The routing is performed through standard routers and the replication of packets is performed by the multicast routers. Therefore, in joining or leaving a multicast group, the immediate multicast router must be notified.

The IGMP is utilized by (multicast) routers to periodically check whether the known group members are still active. It provides the information required in the last stage of forwarding a multicast message to its destinations. This way multicast routers within networks know about the members of multicast groups on their directly attached networks and can decide whether to forward a multicast message on their network[2]. In case there is more than one multicast router on a given subnetwork (LAN) , one of the routers is elected as the "querier" and assumes the responsibility of keeping track of the membership state of the multicast groups which have active members on its subnetwork. Based on the information obtained from the IGMP

[2]The forwarding technique is much more complicated in IGMPv3

the router can decide whether to forward multicast messages it receives to its subnetwork(s) or not. After receiving a multicast packet sent to a certain multicast group, the router will check and determine whether there is at least one member of that particular group on its subnetwork. If that is the case the router will forward the message to that subnetwork. Otherwise, it will discard the multicast packet.

In the development of the IGMP, three versions have been standardized over time. We present a short summary of them and focus more on version 3 (IGMPv3) since it is provides the most extensive set of operations to enable efficient multicasting. This is also the reason why we have chosen to focus on IGMPv3 in our investigation on multicasting. The three version are discussed in the following:

• *Version 1:* specified in RFC-1112 [4], was the first widely-deployed version and the first version to become an Internet Standard.

• *Version 2:* specified in RFC-2236 [5], added support for "low leave latency", that is, a reduction in the time it takes for a multicast router to learn that there are no longer any members of a particular group present on an attached network.

• *Version 3:* specified in RFC-3376 [1] supports applications that explicitly signal sources from which they want to receive traffic. With IGMPv3, receivers signal membership to a multicast host group in the following two modes:

– INCLUDE mode. In this mode, the receiver announces membership to a host group and provides a list of IP addresses (the INCLUDE list) from which it wants to receive traffic.

– EXCLUDE mode. In this mode, the receiver announces membership to a host group and provides a list of IP addresses (the EXCLUDE list) from which it does not want to receive traffic. This indicates that the host wants to receive traffic only from other sources whose IP addresses are not listed in the EXCLUDE list. To receive traffic from all sources, like in the case of the Internet Standard Multicast (ISM) service model, a host expresses EXCLUDE mode membership with an empty EXCLUDE list.

Version 3 is designed to be interoperable with versions 1 and 2. According to the protocol, in networks where present hosts running different versions of IGMP, the routers must operate in version 1 and version 2 compatible modes. The same rule holds for the hosts, when there is a router running lower version of the protocol.

For delivering a multicast packet from the source to the destination nodes on other networks, multicast routers need to exchange the information they have gathered from the group membership of the hosts directly connected to them. There are many different algorithms such as "flooding", "spanning tree", "reverse path multicasting" for exchanging the routing information among the routers. Some of these algorithms have been used in dynamic multicast routing protocols such as Distance Vector Multicast Routing Protocol (DVMRP), Multicast extension to Open Shortest Path First (MOSPF), and Protocol Independent Multicast (PIM), that exists in two variations – Sparse Mode(PIM–SM) and Dense Mode(PIM–DM).

Multicast Addressing

A Class D IP address is assigned to a group of nodes defining a multicast group. The most significant four bits of Class D addresses are set to "1110". The 28-bit number following these four bits is called "multicast group ID". Some of the Class D addresses are registered with the Internet Assigned Numbers Authority (IANA) for special purposes. The block of multicast addresses ranging from 224.0.0.1 to 224.0.0.255 is reserved for the use of routing protocols and some other low-level topology discovery or maintenance protocols. Addresses ranging from 239.0.0.0 to 239.255.255.255 are reserved to be used for site-local "administratively scoped" applications, and not Internet-wide applications. There are some other Class D addresses already reserved for well-known groups such as "all routers on this subnet", "all DVMRP router" and "all OSPF routers".

IGMPv3 Benefits

In this section, we discuss several capabilities that the Source Specific Multicast (SSM) provides resulting in two main advantages. The first capability is that a host can specify to receive packets *only* from a specific source address. The second capability is that a source specify to receive packets from *all but* specific source addresses. These capabilities result in two main advantages of the IGMPv3, namely:

• Optimized bandwidth utilization – The receiver may request to receive traffic only from explicitly known sources. This decreases the network workload, because unwanted traffic is not replicated by the routers anymore.

• Improved security – No denial of service(DoS) attacks from unknown sources. In SSM, multicast traffic from each individual source will be transported across the network only if it was requested. In contrast, the older versions of IGMP allow forwarding traffic from any active source sending to a multicast group to all receivers requesting that multicast group. In Internet broadcast applications, this behavior is highly undesirable because it allows unwanted sources to easily disturb the actual Internet broadcast source by simply sending traffic to the same multicast group. This situation depletes bandwidth at the

receiver side with unwanted traffic and thus disrupts the undisturbed reception of the Internet broadcast. In SSM, this type of denial of service attack cannot be made by simply sending traffic to a multicast group.

All these advantages make the protocol very useful in networks with heavy IP traffic and frequent DoS attacks.

III. Implementation

In the previous section, we discussed the multicast environment and the Internet Group Management Protocol (IGMP), in particular, version 3 – IGMPv3. In this section, we discuss the methodology we utilized to perform profiling. We have taken application C code reflecting the functionality of IGMPv3 and turned it into a benchmark. Subsequently, we compiled the benchmark to run on a simulator. Utilizing the simulator, we were able to gather data on execution time of specific functions and instruction distribution of the benchmark. The profiling results are discussed in Section IV.

The *sim-outorder* simulator

In order to simulate the performance of our benchmark we have performed a simulation using SimpleScalar's *sim-outorder*[2]. We simulated a 4-way superscalar processor with 128 KB of direct mapped level 1 (L1) data cache; 512 KB of direct mapped level 1 instruction cache; 1 MB unified level 2 (L2) data and instructions cache. The L1 and L2 cache latencies are set to default values of 1 and 6 cycles, respectively. The simulated processor uses a bimodal branch predictor with 2048 table entries.

The IGMP benchmark

For simulation purposes, we decided to use *igmprt* [3] routing daemon and compiled it with SimpleScalar's gcc v2.7.2.3. The daemon implements the router side processing of IGMPv3, namely maintaining the IGMP table. Each entry in the table has the following structure: **(Interface, (Group records), Filter mode for each group, (Source records))**. The fields in this structure are discussed in the following:

- **Interface** is the upstream interface where the daemon is running.
- **Group records** represent the multicast groups that are reported on that interface.
- **Filter mode for each group** is either INCLUDE or EXCLUDE.
- **Source records** contains the reported IP addresses, used for source based filtering for each group.

The records in the IGMP table are updated by means of periodic reports sent by hosts in response to General Queries, Group Specific Queries or Source and Group Specific Queries. We modified this behavior in such a manner that the program reads these reports from the file *input_stream*, instead of receiving them from the network. In this file, every line contains one report.

The functionalities that we look at are those of receiving IGMPv3 membership reports, assembling them in the *igmp_report_t* structure and passing them to a function called *igmp_interface_membership_report_v3*. That function deals with the maintaining of IGMP table and sending queries to the query queue. The processing is done in the following steps:

1. It calls the function *igmp_interface_group_add* to determine whether the multicast group in the report is preset on the interface where the report is received from. The function creates a group entry if one is not found.
2. The function *igmp_group_rep_add* is invoked, which checks whether the sender's IP address is in the group's structure for that interface, and if it doesn't it is added.
3. The record type (*is_in*, *is_ex* etc.) is determined and the corresponding function *igmp_group_handle_isin* or *igmp_group_handle_isex* are called, respectively. These two functions handle the updates of group records – add or remove source IP addresses, reset group and source timers, change the state of the router(from INCLUDE to EXCLUDE) and send "Group Specific Query" and "Group and Source Specific Query".

Following these steps, the IGMP table is updated, the group and source timers are updated also and the necessary query messages are sent to the output message queue. The actual sending of these messages is not implemented, because in this investigation we are only interested in the IGMP table updates.

Data types and datasets

The IGMPv3 Membership Report format [1] differs from that of IGMPv1 and IGMPv2 Membership Reports due to the SSM capabilities of IGMPv3 (discussed in Section II). In our benchmark, this report is represented in the structure *igmp_report_t*:

```
typedef struct _igmp_report_t { u_char
igmpr_type;//version and type of IGMP message
u_char igmpr_code;//subtype for routing msgs
u_short igmpr_cksum;//IP-style checksum
u_short igmpr_rsv;//reserved
u_short igmpr_numgrps;//number of groups
igmp_grouprec_t* igmpr_group;//group records
} igmp_report_t;
```

In the original code of *igmprt*, the amount of group records per report was 1 (*igmp_grouprec_t igmpr_group[1]*), there-

fore we decided to allocate memory dynamically in runtime (*igmp_grouprec_t* igmpr_group*) depending of the number of groups in every single report.
Every group record (*igmp_grouprec_t*) has the following structure:

typedef struct _igmp_grouprec_t{
u_char igmpg_type;//record type
u_char igmpg_datalen;//amount of aux data
u_short igmpg_numsrc;//number of sources
struct in_addr igmpg_group;//the group being reported
struct in_addr igmpg_sources;//source addresses*
} igmp_grouprec_t;

In the original code of *igmprt*, the amount of source addresses per group was 1(*igmpg_sources[1]*), therefore we decided to allocate memory dynamically in runtime(*in_addr* igmpg_sources*)depending of the number of sources in every single group.

The actual data that we process comes from file *input_stream* containing randomly generated reports. We performed 3 simulations with 3 different datasets containing 1000, 2000 and 5000 reports, respectively. Each report contains between 1 and 5 multicast groups with different record types (*is_in, is_ex, to_in* etc.), and each group has from 1 to 5 source addresses.

IV. Experimental Results

In the previous section, we presented the simulator, the benchmark, and the data types and datasets used in our investigation. In this section, we present the results on instruction distribution, cache behavior, and branch prediction accuracy.

Instruction Distribution

The simulated functionality includes the operations performed when receiving IGMPv3 membership report. The function *igmp_interface_group_lookup* is called by *igmp_interface_membership_report_v3* and it is responsible for determining whether the reported group persists on the IGMP table. This processing takes a major amount of processor cycles. It consumes more than 90 percents of the *igmp_interface_membership_report_v3* cycles (see Figure 2).

On the other hand, the functions involved in table updating (*igmp_group_handle*_xxxx) take no more than 8 percents from *igmp_interface_membership_report_v3* cycles. When we use larger datasets, the IGMP interface table grows. Figure 2 shows an increase in the group lookups time for larger IGMP tables. The frequent loads from the memory, caused by the table lookups executed by

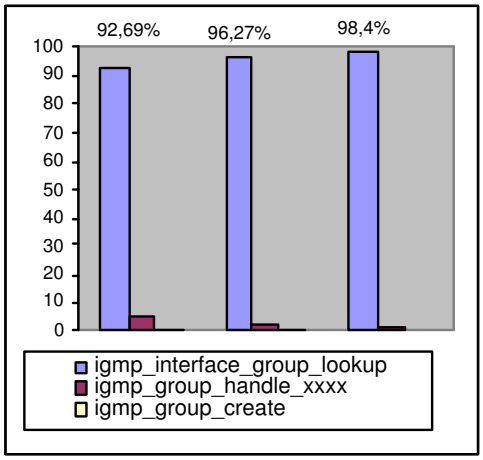

Fig. 2. Functional Statistics

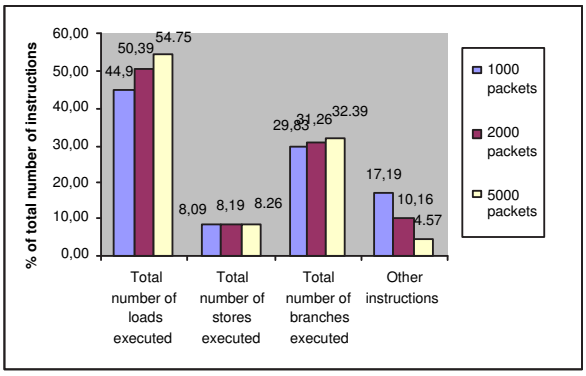

Fig. 3. Instruction Distribution

igmp_interface_group_lookup, explain the data-intensive characteristic of our application (see Figure 3). As depicted in Figure 3 the memory operations take more than 50 percents of all instructions. The condition branch operations have relatively high distribution, which is caused by the high number of different flags and field checks when receiving packet.

Cache Behavior and Branch Prediction

The results for the cache behavior are summarized in Table I.

Number of packets	1000	2000	5000
Branch address predict. rate	0.9881	0.9933	0.9971
Branch direction predict. rate	0.9882	0.9934	0.9971
L1 data cache miss rate	0.1032	0.1152	0.1213
L1 instruction cache miss rate	0.0068	0.0040	0.0018
L2 unified cache miss rate	0.0045	0.0388	0.1897

TABLE I
Cache behavior

The table shows a high branch address and direction prediction values for all datasets. We can also observe the increase in data cache miss rate and decrease in instruction cache miss rate for larger datasets. In addition, the last row in the table shows fast increase in Level 2 cache miss when we process larger datasets. The cache size values are described in Section III.

V. Conclusions

In this paper, we presented our investigation of the Internet Group Management Protocol that belongs to a higher layer of the TCP/IP protocol stack. The methodology encompasses the creation of benchmarks reflecting the IGMP functionalities and subsequently the profiling of the created benchmarks in order to determine the most time-critical operations. Next to determining the time-critical operations, the mentioned simulation environment also provided results on instruction distribution, cache behavior, and branch prediction accuracy. Preliminary results suggest that the operations involved in multicasting are memory-intensive as about 50% of the operations are memory-related operations. Such results can be utilized in the design of future network processors.

References

[1] B. Chain, S. Deering, I. Kouvelas, Cisco Systems B. Fenner, AT&T Labs - Research, A. Thyagarajan, Erricson, *Internet Group Management Protocol, Version 3*, RFC 3376 (www.ietf.org/rfc/rfc3376.txt?number=3376), October 2002.

[2] Burger, D. and Austin, Todd M., *The Simplescalar Tool Set, Version 2*, Tech. report, University of Wisconsin, June 1997.

[3] Lahmadi, *An igmpv3-router implementation*, World Wide Web, http://www.loria.fr/ lahmadi/igmpv3-router.html.

[4] S. Deering, Stanford University, *Host Extensions for IP Multicasting*, RFC 1112 (www.ietf.org/rfc/rfc1112.txt?number=1112), August 1989.

[5] W. Fenner, Xerox PARC, *Internet Group Management Protocol, Version 2*, RFC 2236 (www.ietf.org/rfc/rfc2236.txt?number=2236), November 1997.