Investigating the Optimal Performance of Multicast Communication by Simulation

Eleftherios Stergiou, Dimitrios Liarokapis and Gerasimos Meletiou
Department of Informatics & Telecommunication Technology, TEI of Epirus, Arta, Greece

Keywords: Simulation, Performance Evaluation, Multicast, Protocols, Multi-criteria, Quality Specification.

Abstract: The multicast mechanism is one method of data communication in data networks which aims to transfer data to a group of receivers on a network in an efficient manner. In this work the performance of four well known and widely used multicast protocols are investigated using OMNeT++ open software, which was chosen for this purpose. Individual performance metrics are determined by executing simulation experiments and in addition a unique overall performance indicator is defined to solve the multi-criteria decision problem that is revealed as the network configuration and the service conditions vary. This performance evaluation approach can be used by network protocol designers for building and exploiting optimal protocols when setting up networks so as to achieve the best performance under the multicast traffic load and quality specifications.

1 INTRODUCTION

Today, as multimedia applications are increasing, they need efficient distribution to a large number of end users over large distances. This trend demands a multicast means of distribution rather than a unicast one. Multicast transmission decreases the usage of network resources compared to unicast and broadcast transmission, which target one recipient and all recipients in a segment, respectively. Figure 1 depicts the differences between unicast and multicast transmission. In our effort to investigate the performance of multicast communication in various types of data networks, our interest is focused on the performance of the protocols which are used in this communication technique. By now a number of multicast protocols have been implemented and tested separately that work perfectly on the internet and intranets. A well-balanced and effective protocol must be able to cut out the quantity of states that is stored in the routers.

In this paper, four well known and widely used multicast protocols are selected and their performance is investigated. Two of them belong to sparse type protocol and that are the Core base Trees (CBT) protocol and the Protocol Independent Multicast-Sparse Mode (PIM-SM), while the other two, the Protocol Independent Multicast-Dense Mode (PIM DM) and the Distance Vector Multicast Routing Protocol (DVMRP), belong to dense-type protocols. The first pair of protocols is suitable for groups where a very low percentage of the nodes are subscribed to the multicast session. These sparse-mode protocols assume relatively small numbers of multicast clients. In contrast, the dense-type protocols (PIM DM and DVMRP) are ideal for groups where many of the end-users subscribe to multicast packets. Dense-mode multicast routing protocols flood packets across the network and then prune off branches where there are no recipients, while sparse protocols have the ability to explicitly construct a tree from each sender to the receivers in the multicast group.

The abovementioned protocols differ significantly in terms of the approach used to implement the many-to-many communication model. They differ in terms of scalability and in that some of them store state-related information in all routers while in the remainder only routers currently participating in the transmission keep these data. Both types of protocols employ different types of trees which have advantages and disadvantages that need to be considered by the network designer. Thus the comparison of results was considered as an interesting topic.

To study the performance of protocols other than by taking direct measurements, which is expensive, it is possible to use a special purpose simulator or an

Stergiou E., Liarokapis D. and Meletiou G..
Investigating the Optimal Performance of Multicast Communication by Simulation.
DOI: 10.5220/0004479401080116
In *Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (SIMULTECH-2013), pages 108-116
ISBN: 978-989-8565-69-3

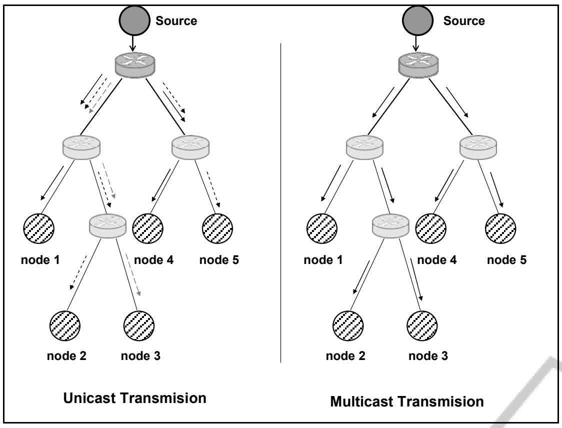

Figure 1: Unicast and multicast modes of data forwarding.

open simulator. Developing a special purpose simulator (Weijia et. al., 2005) is a custom solution which is usually completed in one protocol. That drawback forces us to use open simulation software. For building up the multicast model in computer engineering, there are various discrete simulators such as, for example, OMNeT++ (Veselý et al, 2012), NS2 (Bartczak and Zwierzykowski, 2007); (Tarik et al, 2001), and OPNET software (Wang et al, 1999). In all cases it is possible to define static or dynamic network models and to use graphical interfaces. Finally it was decided that OMNeT++ open software would be used as well as it provides an opportunity to verify the network design in a safe and cheap environment. This open simulator provides the ability to test a network with regard to a failure hypothesis or to trace a flow of multicast streams among sources and receivers. Discrete events simulation like OMNeT++ provides a means to check the behaviour of a network across a wide range of test scenarios, which is not always feasible in laboratory conditions. Contrary to this, obtaining measures from a real multicast network architecture is very expensive and complex as various collaborating components are involved.

1.1 Previous Work

Stergiou et al. (Stergiou et al., 2007) present an interesting multicasting architecture that operates using a multicasting firewall over the Multicast Control Protocol (MCP). They evaluate the transition times of IGMPv2 reports on a multicast control server. In 2008 the same group (Stergiou et al, 2008) also analysed a broadband multicasting system in an IPv4 environment using the IGMP and MCOP protocols, where a Gigabit Ethernet was used as the delivery network in the client's segment. This evaluation study provides measurements for the two most significant performance metrics, the required bandwidth and the round trip time of packets versus the number of multicasting clients.

Chuang and Sirbu (Chuang and Sibru, 1998) studied the cost advantages of a multicast protocol. Their study took into consideration the link cost but ignored the cost of routing table memory and the CPU usage. In Chuang and Sirbu's paper, the *normalized multicast tree cost* is defined as the ratio of the total length of the multicast distribution tree to the average length of the relevant unicast path. According to (Chuang and Sibru, 1998) this *normalized multicast tree cost* is proportional to the power, N^k, where N denotes the number of routing nodes that have subscribed to the multicast group while k is a factor ranging between 0 and 1. The previous sentence is called *Chuang–Sirbu's Law*. The weakness of this study is that it ignores the cost of nodes as it is the cost of routing the table and the bandwidth utilization. Moreover it is worth noting that in (Chuang and Sibru, 1998) the pricing model for the multicast mechanism was also discussed. Chuang and Sirbu's approach was the first step in the study of the performance problems of such protocols. Phillips et al. (Phillips et al., 1999) tried to correct the abovementioned weakness of *Chuang–Sirbu's Law*, which is presented in (Chuang and Sibru, 1998). In sequence to that, the Phillips's group analysed the above formula, *Chuang–Sirbu's Law*, for k-ary trees and for general networks that were not k-ary. Another effort that continued the research of Phillips et al. is the study by Mieghem et al. (Mieghem et al, 2001), who investigated the function of reachability, which depicts the number of distinct sites that are equal to a constant number. The number of nodes is a specific constant number of hops starting from a source that has been evaluated. The study of Mieghem et al. shows exponential behaviour, as the number of routings in the internet that can be reached from a root grows exponentially with the number of hops. Also the same group states that multicasting provides efficient transmission of data when the receivers are widely spread. Moreover, Zaballos et al. (Zaballos and Segui, 2006) examined a few routing properties of the IGP routing protocol using an OPNET simulator. Finally, amazing and enviable work has been presented in the field of multicast protocols which are applied to wireless networks (Gupta et al, 2010); (Bhasin et al, 2012). All the aforementioned works were a considerable asset which we held when we started this project.

1.2 This Work

Here, a specific network configuration is obtained as a testbed for our experiments. Multicast-type streams running from load sources to multicast clients via intermediate decreasing carrier links are considered (see Section IV). Based on this network scenario, a model was subsequently implemented using OMNeT++ software. The specific model was run for four multicast protocols which were selected to complete the multicasting communication and which were the main subjects for testing. Two of them were dense-mode protocols (DVMRP, PIM DM) while the other two were classified as sparse-mode protocols (CBT, PIM SM). The major motivation for this approach was to find a method of distinguishing some counterparts of multicast protocols in relation to receivers' population segments.

With a specific service scenario in a particular network construction, the performance trends of four natively multicast protocols were determined and compared in a quantitative manner. OMNeT++ open software was preferred as the platform for simulation development because it is an open simulator and does not have limitations in supporting the tracing of data flows. Moreover, the concept of the ANSA (Automated Network Simulation and Analysis) extension module, which expands the INET framework in the OMNeT++ simulation environment, was used and sufficiently exploited here. This simulation comprises the called INET framework in the OMNeT++ environment with some external tools that are created for implementing those specific multicast protocol configurations. In addition to the above, the OMNeT++ platform was chosen because it has the capability to support hierarchically nested modules, and has the ability to handle flexible module parameters, although it is commonly accepted that this product is a rather complicated software to manage. Thus, one obvious contribution of this paper is the simulation approach and the improved router model incorporated in the OMNeT++ model.

Contrary to the majority of publications, which present separate values of some performance metrics without correlating the various results among them, in this paper a general performance indicator is introduced that can be extended to include a more positive or negative performance factors, evaluating any protocol's strengths and weaknesses respectively. So, the introduction and the usage of a unique overall factor is another significant contribution in the field of evaluation of protocols. Finally, the rationale behind this performance methodology is to provide a tool for estimating the overall performance of protocols in an inexpensive, quick, accurate, and transparent way for a given number of receivers.

The remainder of this paper is organized as follows: in Section 2 we introduce the performance metrics as performance criteria that are related to multicast protocols and that are taken into consideration in our study, and in Section 3 the simulation testbed used is presented. Section 4 presents part of the results of our performance approach, which has been conducted through simulation experiments, and finally Section 5 concludes with basic remarks.

2 METRICS DEFINITIONS

In order to consider the quality of service (QoS) at the receiver side, we need to define some clear parameters which will show the QoS in a quantitative manner. Several parameters can indicate the QoS at the end-user view such as the *latency* of packets, the *jitter*, the *packet loss*, the availability of connecting a node in a multicast tree, the time period of joining, and so on.

Bearing in mind the need to have a reliable approach, we chose to define the following parameters which can show the QoS at the end-user side sufficiently succinctly.

2.1 Relative Packet Latency from Source to end User (L_r)

The *relative packet latency* from *source to end user* (L_r) can be defined as the ratio of *latency* which packets presents when they travel from sender to receiver in unicast mode (L_u) to the corresponding *packet latency* when they travel in multicast mode (L_m). Thus, the *relative latency* (L_r) from source to end user is given by the following expression:

$$L_r = L_u / L_m \quad (1)$$

The *packet latency* transmitted in multicast mode is basically dependent on the number of sources. If the number of sources in a system proliferates then the time consumed increases on each Randezvous Point (RP) router considerably, which raises the total *packet latency* from sources to end users.

2.2 Relative Jitter Parameter (J_r)

The *relative jitter* parameter (J_r) is defined as the ratio of the average *jitter* of packets which are transmitted by multicast mode transmissions from source to receiver ($\overline{J_m}$) to the corresponding average *jitter* of the packets transmitted in unicast mode ($\overline{J_u}$). Hence,

$$J_r = \overline{J_m} / \overline{J_u} \tag{2}$$

Jitter variation is a parameter that basically determines whether a user who handles multimedia will have a pleasant experience or not.

It is clear that the above two defined metrics have a relationship and that they are directly involved with the QoS at the listeners' side. Contrary to this, network utilization is an alternative aspect of the performance problem. The efficiency or the gain of multicasting in terms of network resource consumption can be compared to the corresponding unicast one. Similarly to the above (the viewpoint of the performance of the end-user), various parameters need to be defined in order to obtain indications of the effectiveness of the network when it manages and handles multicast traffic. Subsequently, we chose to define two major performance parameters through which the capacity of the network is evaluated when it deals with multicast traffic.

2.3 Relative Usage of Links (U_r)

Considering one-to-many communication, in which a source distributes messages (packets) to m different, uniformly distributed destinations along the shortest path, in the unicast mode of transition, these messages are sent m times from the source to each destination. One of the main properties of multicasting is that it can economize on the number of links traversed. Mieghem and Janic present a study (Mieghem and Janic, 2002) which shows how the number of links in a multicast tree changes as the number of multicast users in the group changes. According to (Mieghem and Janic, 2002), the stability of a tree tends to a Poisson distribution for a large number of receivers. It is obvious that the greater the number of links that are used in a multicast communication, the greater the utilization of links and usage of resources of the network. Hence a performance parameter is defined in order to show the usage of the network. This metric is called the *relative usage of links* and is defined as the ratio of the number of multicast hops to the number of unicast hops of a packet stream.

$$U_r = (\textit{number of multicast hops}) / (\textit{number of unicast hops}) \tag{3}$$

where $0 \le U_r \le 1$.

It is apparent here that when the number of members of a multicast group increases, the number of hops also pullulates but this increment has a much lower rate than that in the corresponding unicast-mode communication.

2.4 General Performance Factor (*GP*)

From our general experience of multi-criteria system performance it became apparent that the performance metrics can be divided into two major categories: those that cause the system (here the performance of a protocol) to behave better when they are maximized and those that cause the performance of the system to improve when their values are minimized.

We depict the first group of performance metrics as $a_{\max} = \{a_{1,\max}, a_{2,\max}, \ldots, a_{\mu,\max}\}$ and the other group as $b_{\min} = \{b_{1,\min}, b_{2,\min}, \ldots, b_{\nu,\min}\}$, where μ and ν are the number of performance factors to be maximized and minimized respectively.

The optimal solution is to have high values of the first group of performance metrics and low values of the other group. Thus, it is interesting to carry out a general evaluation using exclusively one factor. We wish to have only one factor which will reveal the overall performance. So, this required overall performance factor is defined by relying on the correlation of the two individual groups of metrics. Nevertheless due to the fact that each metric has different units and ranges, it is necessary to normalize them to obtain a common reference value domain. We call this factor the *general performance* (GP) factor, which is formally defined as:

$$GP = \sqrt{\sum_{i=1}^{\mu} (a_{i,\min})^2 + \sum_{j=1}^{\nu} \left(\frac{1}{b_{j,\max}}\right)^2} \tag{4}$$

Supposing we now have three or more protocols and their *general performance* factors, GP_1, GP_2, GP_3, and so on, respectively. If we have the following inequality regarding the general performance, $GP_1 < GP_2 < GP_3 < \ldots$, then we can say that System 1 is better in comparison to System 2, System 2 is more powerful than System 3, and so on. It is clear that the evaluation and comparison between protocols is transformed into the evaluation of a unique factor and the ideal performance of the system occurs when the value of the GP indicator tends towards zero. It is noteworthy that the GP factor represents the length of the corresponding vector in two-dimensional space and a smaller value indicates a better performance of the examined system. Following the above, another interesting point when we study multi-criteria decision-making problems is that the importance of each criterion is a design problem and depends on the general environment and special considerations. When we want to set a special weight on each individual metric, we have the ability to assign a weight to each performance coefficient and the above formula can be replaced by:

$$GP(w_i, w_j) = \sqrt{\frac{\sum_{i=1}^{\mu}(w_i \cdot a_{i,\min})^2 + \sum_{j=1}^{\nu}(w_j \frac{1}{b_{j,\max}})^2}{\sum_{i=1}^{\mu} w_i + \sum_{j=1}^{\nu} w_j}} \qquad (5)$$

where w_i and w_j are the corresponding weights of the minimized and maximized performance metrics. In all cases the reference value domain of the *general performance factor* ranges from 0 to 1. The assumption that $b_{j,\max} \neq 0$ must be satisfied in any case.

In this work we limit our study to the use of three performance metrics, knowing that it is feasible to include additional performance factors that could be chosen to evaluate the performance of the multicast protocol with better accuracy. Additional performance parameters that can be considered include the multicast *packet loss*, the *number of states* that are utilized by each intermediate router in order to achieve the multicasting delivery, and so on.

3 EXPERIMENTAL TESTBED AND SIMULATIONS

A specific networking schema is selected for use as a basic testbed for checking multicast communication (See Figure 2). This network consists of three distinct zones related to the values of speed and bandwidth. The carrier links that connect these distinct areas is considered to have a corresponding scalar structure, as they serve the load from the sources to the multicast clients. According to our scenario, three multicast senders create multicast traffic at a rate of 256 kbps, obtained as a root of traffic generation, and those servers send load to the core network. The services that are employed on the multicast servers are Web and FTP services which are working over the TCP transport layer, and also the deployed TFTP service which operates over the UDP transport layer. The multicast data streams traverse the network's tree. specifically the streams pass from the network backbone (WAN) to an intermediate MAN area, then arrive at the local areas and finally terminate at the end traffic consumers. In the WAN area, 5 basic routers are obtained which are connected together via E3 type connections. Those WAN area routers are connected via E2 carrier links with intermediate second level routers. Each second level routing node (belonging to the MAN area) is able to support a maximum of 10 different MAN type networks. Also, continuing the tree hierarchy regarding the bandwidth and data speed, the second level routing devices are connected via E1 type links with the third level routers. Each third level routing device directly supports two Ethernet type LANs, which are connected via 10 Mbps links. Finally, three different nodes are connected on each LAN, which are the final recipient of the multicast load. The end data consumers of multicast traffic are selected randomly in order to be spread throughout the network architecture under study. Special care was taken to ensure that the ultimate receivers do not belong to the same group.

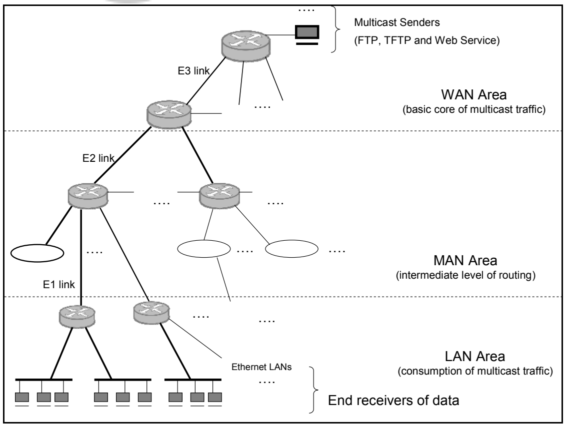

Figure 2: The under study networking schema.

The rationale behind our approach was to firstly create a simulation model of multicast routing and data delivery architecture including the visualisation of the distribution tree and end-user behaviour. At a secondary level we wanted to investigate optional scenarios, accompanied by thorough analysis. To accomplish this task, OMNeT++ version 4.2 and an INET 20111118 framework is used to model the network running the multicast traffic. We use the INET framework modules as the state of art of simulation modelling. A module in the OMNeT++ environment is a list of commands enforcing a required behaviour, which represents a counterpart device in the network. Specifically, the model that is created use the `inet.networklayer.ipv4`: INET's module. At the start of modelling the IGMPv2 protocol (RFC 2236) is used (`igmp`: INET's module), since IGMP is a standard protocol that has the ability to manage membership in multicast groups. According to the IGMPv2 specification, this protocol uses three distinct types of message: *Membership Query*, *Membership Report* and *Leave Group*. As a consequence, the INET modules `networkLayer`, `routingTable` and `interfaceTable` are used to build up and manage a third level of communication in the system.

Regarding the transport layer, corresponding INET modules for TCP and UTP protocols were used, as well as subsequently FTP, HTTP and TFTP services for setting up corresponding sessions. Unfortunately the implementation of the TCP protocol in the OMNeT++ environment has not been state of the art. Some important features of modern TCP implementation, particularly Selective Acknowledgements (SACK) and complete Flow Control, were missing (Reschka et al, 2010). However this simulation used the existing INET module for the TCP sessions. Thus the modules `TCPConnection`, `addSockPair`, etc. are correctly exploited by this model. Furthermore, among the other modules included in the INET framework, the `OSPFRouter` module is used for typical router implementation in our system.

In the first phase of development we used all the above described modules, which already exist in the INET framework of OMNeT++. However, the freely available OMNeT++ simulator is not fully-compliant with the specific protocols under investigation. That forces us to implement modules for the following multicast routing protocols: CBT, PIM SM, PIM DM and DVMRP. In the second phase, special purpose modules are created which include the required multicast features and which are ultimately bound to the ANSA type router. The additional modules are formed in XML language. Simulation model for OMNeT++ was revealed immediately after formed the XML configuration of real devices and links. Our simulation tool was able to setup all devices and a variety of links and initial information, finally running on the network scenario under study.

The `StandardHost` module is used for making simple host (or receiver) representations. Moreover each end user is assigned a queue which is capable of collecting and keeping 50 packets at most. That value of the buffer size is considered to be very common in such experiments. Originally 5 receivers were set up in our network system, and step by step this reached a value of 30 receivers. Those receivers were spread randomly throughout the multicast tree.

We ran the simulation many times, again setting all the parameters to separately obtain results in the form of mean values for each multicast protocol.

Metrics such as *packet's latency* of packets, *minimum variation of delay*, and *number of multicast hops* employed for each protocol, were collected. However, the analytical structure of the developed model, as well as the results presented here, is limited by the boundaries of a typical document. The obtained statistics from the simulator were raw numbers that we subsequently handle in order to produce normalised values which are presented in the next section. Extensive simulations to validate our results have also been undertaken.

4 PERFORMANCE RESULTS

4.1 Relative Packet Latency (L_r)

Figure 3 illustrates the *relative packet latency* (L_r) that appears when the packets travel from the sources to the end consumers versus the number of end multicast traffic consumers for each studied protocol. It is obvious that the sharing mode transmission needs more time cycles in comparison to the unicast mode because the transmitted packets have to be first multiplied and then transmitted.

According to the definition of relative *packet latency* (Formula (2)) and the results depicted in Figure 2, the *relative packet latency* of multicast-mode traffic in the CBT, DVMRP, and PIM DM protocols is $L_r \le 0.6$. Thus the *packet latency* of multicast traffic is:

$$L_m \le L_u / L_r \cong L_u / 0.6 \cong L_u \cdot 1.666 .$$

That means that the maximum possible value of multicast *packet latency* may be equal to 160% of the corresponding unicast mode packet latency.

On the other hand, the relevant *packet latency* of the PIM SM protocol (the second adjacent bar of each four-tuple set of bars) reaches at most $L_r \cong 0.9$. Hence, the *packet latency* of multicast traffic is $L_m \le L_u / L_r \cong L_u / 0.9 \cong L_u \cdot 1.111.$

That means that the maximum packet latency of the data when the PIM SM protocol is applied never exceeds 111% of that of the corresponding unicast-mode traffic.

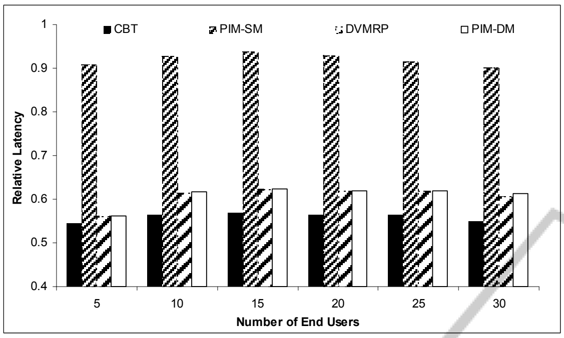

Figure 3: Relative sources-to-end packet latency versus number of end users.

Hence, from all of the above it is concluded that the packet latency of the CBT protocol (the first adjacent bar of each four-tuple set of bars) presents the maximum deviation of the corresponding unicast-type packet latency (and thus the worst case) while the packet latency of the PIM SM protocol shows the minimum deviation of the corresponding unicast mode (the best case in terms of *packet latency*), and Fig. 3 confirms and illustrates this result quantitatively. In all cases of the multicast protocols it can be noticed that when the number of receivers become greater than 10, the traffic delay from source to client tends to present stability. Moreover it is worth noting that the gain in packet latency shown by the PIM SM protocol over the dense-type of protocols studied here can be considered as marginal, especially when the number of listeners becomes large and the differences are negligible.

4.2 Relative Usage of Links (U_r)

Figure 4 plots the relative usage of links (in relation to unicast mode) when multicast traffic is employed for various numbers of receivers and for each examined multicast protocol for the given network and service selection. From this graph it is obvious that the two dense-type multicast protocols (PIM DM and DVMRP; the first two successive bars in each four-tuple bar group in Fig. 5) employ a considerable number of links especially when the number of receivers is small (less than 25). For example, for an extremely small multicast client population equal to 5 the quantity of links that are engaged by dense-type protocols is approximately 25% over of the corresponding links' quantity that are employed by the sparse-type protocols.

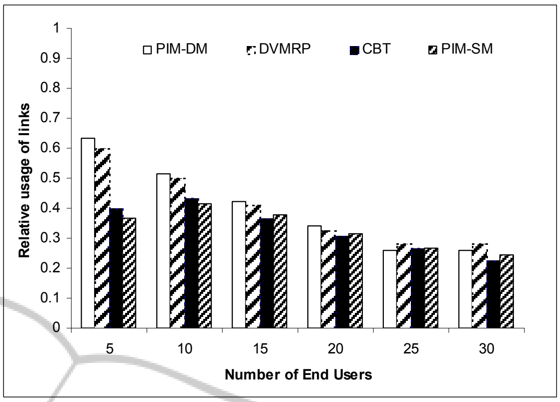

Figure 4: Relative usages of links of protocols versus number of end users.

On the other hand, when the number of listeners exceeds this threshold (25 listeners) all of the multicast protocols have similar behaviours regarding the metric for the *usage of links* as there is only a slight discrepancy between their employed links. Also, as the number of receivers increases and the service becomes more fully multicast, the exploitation of the network is improved considerable.

Especially, observing Fig. 4, it can be observed that when the number of listeners is more than 25 the gain in network sources exploitation tends to exceed 80% of the corresponding unicast service. That clearly reveals the main advantage of the multicast way of transferring data.

4.3 General Performance (*GP*) Indicator

For calculation of the g*eneral performance* indicator three metrics were taken into consideration: *the packet's latency* and the *jitter* variation which were considered with equivalent weights, equal to 40% (w=0.4) and the *usage of network's links* with its weight equal to 20% (w=0.2).

The first pair of metrics is related to the performance of the *end user* aspect, while the third correlates to capacity of the network. Hence this chosen pattern of weights is useful, especially when there is more interest on the QoS on the receivers' site and less interest on achieving a high network capacity. Moreover, is noteworthy that the metric of *packet's latency* is a maximised type factor due to its definition (see formula (2)) while the other two metrics are minimised type factors in our performance scenario. Fig. 5 presents the *general normalised performance* factors as the number of receivers is increased (values domain: 5–30). In the same plot, the general behaviour of multicast streams according to the listeners' view appears to have a small variation (stability) for each protocol.

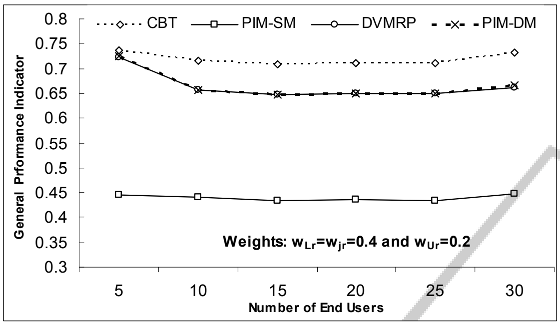

Figure 5: The general normalized performance indicator versus number of end users.

Furthermore, the graph in Fig. 5 clearly shows that the best overall performance is presented when the PIM SM is applied; resulting in a lower value of the GP indicator of approx. 0.45. In contrast to this, the other sparse type protocol (CBT) presents the worst overall performance which is about 28–30% worse in comparison with the PIM SM case protocol for the given network configuration, range of listeners, obtained metrics and specific pattern of weights between them.

It is worthwhile noting that the dense type protocols endow an overall yield that is between the performance of the two sparse protocols (approx. 22% worse in comparison to the PIM SM performance and 10% better than the CBT protocol). Also, it must not be ignored that the tension that have all the protocols of slighting deterioration (slight rise in the value of the GP indicator) when the number of listeners exceeds 25 and is increasing.

Finally it is worth mentioning, and to be reminded again, of the variety of network schemas for the chosen metrics and obtained weights that can be studied by applying this methodology for detecting where the performance edge of each multicast performance protocol exists.

5 CONCLUSIONS

Building up the simulations and making the inevitable validation for each protocol was a difficult task and a time-consuming process; also, the open software OMNeT++ does not have a full range of modules ready and does not provide fully compliant tools for the protocols under study. Finally, to sum up the above, in this paper two sparse-type multicast protocols, CBT and PIM SM, and two dense-type multicast protocols, DVMRP and PIM DM, were investigated with regard to their performance. The performance approach using ONNeT++ software was one of the key contributions and the introduction of a unique performance indicator was the other. The simulation experiments focused on only a few metrics out of many that can be explored and studied. Nevertheless, we believe that the rest of the quantitative properties can be extracted without making considerable changes to the current simulation approach. It is obvious that, so far, the research on multicast protocols is far from exhaustive.

REFERENCES

Bartczak T. and Zwierzykowski P., 2007. "Validation of PIM DM implementation in the NS2 simulator", *ISAT, pp. 117-127*

Bhasin Swati, Gupta Ankur, Mehta Puneet, 2012. "Comparison of AODV, OLSR and ZRP Protocols in Mobile Ad-hoc Network on the basis of Jitter", *International Journal of Applied Engineering Research, ISSN 0973-4562 Vol.7 No.11*

Chuang John, Sibru Marvin, July 1998. "Pricing Multicast Communication: A Cost-Based Approach", *Internet Society INET 1998 Conference, Geneva, Switzerland*

Gupta Anuj K., Sadawarti Harsh, Verma Anil K., April 2010. "Performance analysis of AODV, DSR & TORA Routing Protocols", *in IACSIT International Journal of Engineering and Technology, Vol.2, No.2, ISSN: 1793-8236*

Mieghem P. Van, Janic M., 2002. "Stability of a Multicast Tree", *in IEEE INFOCOM 2002, pp. 1099-1108*

Mieghem Piet Van, Hooghiemstra Gerard, Hofstad, Remco van der, 2001. "On the Efficiency of Multicast", *in IEEE/ACM Transactions on Networking, vol. 9, No. 6, pp. 719-732*

Phillips G., Shenker S., Tangmurarunkit H., 1999. "Scaling of Multicast trees: Comments on the Chuang Sirbu Scaling law", *in ACM SIGCOMM 1999, Harvard, MA pp. 41-51*

Reschka Thomas, Dreibholz Thomas, Becke Martin, Pulinthanath Jobin, Rathgeb Erwin P., 2010. "Enhancement of the TCP Module in the OMNeT++/INET Framework", In: *Proceedings of the 3rd ACM/ICST International Workshop on OMNeT++, ISBN: 978-963-9799-87-5*

Stergiou E. , Meletiou G., Vasiliadis D., Rizos G., and Margariti S., November 2008. "Evaluation Study of a Broadband Multicasting Service over a Gigabit Ethernet Delivery Network", *AIP Conf. Proc. Volume 1060, p.p. 392-396*

Stergiou E., Meletiou G., Vasiliadis D., Rizos G., and S. Margariti S., December 2007. "Evaluating the Multicast Control Protocol on a Multicasting Network Architecture", *AIP Conf. Proc. Volume 963, p.p. 943-946*

Tarik Cicic, Stein Gjessing, Oivind Kure, 2001. " Performance Evaluation of PIM SM Recovery". *Proceedings of International Conference on Networking ICN'01, Colmar, France, July 2001*

Veselý Vladimír, Matoušek Petr, Švéda Miroslav, 2012. "Multicast Simulation and Modeling in OMNeT++", In: *Proceedings of the IEEE 5th International ICST Conference on Simulation Tools and Techniques, Desenzano del Garda, IT, ICST, p. 298-301, ISBN 978-1-936968-47-3*

Wang X., Yu C., Schulzrinne H. and Stirpe P., 1999. "IP Multicast Simulation in OPNET", CiteSeerx 10.1.1.18.7365

Weijia Jia, Wanqing Tu, Wei Zhao and Gaochao Xu, 2005. "Multi-shared-trees based multicast routing control protocol using anycast selectio", *The International Journal of Parallel, Emergent and Distributed Systems, Vol. 20, No. 1, pg. 69-84, March 2005*

Zaballos A. and Segui C., 2006. "Analysis and simulation of IGP routing protocols", *Technical report, University of Ramon Llull, Barchelona*