Dynamic Metric Choice Routing for Mesh Networks

Rafael Lopes Gomes, Waldir Moreira, Vagner Nascimento, Antonio Abelém
Universidade Federal do Para
Belem, Para, Brazil
Email: rafaellgom@gmail.com, {mraranha, vagner, abelem}@ufpa.br

***Abstract*—In this paper we propose an extention to the proactive Optimized Link State Routing (OLSR) protocol in order to provide quality of service and support to applications using the TCP protocol. The proposal performance, called OLSR Dynamic Choice (OLSR-DC), is evaluated using the Network Simulator. These simulations show the extension's performance based on the particularities of TCP and UDP applications.**

I. Introduction

During the last years, there was a great advancement in wireless technologies, resulting in the appearance of several types of network to satisfy several needs of the market [23].

One type of network resulting from such advancement is the Wireless Mesh Network (WMN) that is a multihop, self-configurable network with dynamic topology and routing, and that have nodes with communication, in the physical level, through a variation of the IEEE 802.11 standard (Wireless Local Area Network - WLAN).

WMNs are a special case of ad hoc networks, however the main difference between them is that WMN nodes have a fixed location, despite the fact that these locations are not previously determined. Thus, their routing algorithms have many features in common. Furthermore, client nodes in a mesh network do not necessarily need to act as routers, which represents a less complexity in them.

According to [5] there are many uses for WMNs such as: extending coverage range of companies/universities; offering services in places where cabling is not feasible due to cost and/or physical barriers; establishing communication in emergency situations such as earthquakes; provide opportunity for digital inclusion; and giving support to military and rescue operations.

WMN routing protocols must have self-management, self-configuration, and self-healing features. Thus, different protocols were developed [5].

However, these protocols aim to answer the needs of the applications for which they were developed, and each application has its peculiarities [14]. Thus, these protocols, in general, can not answer all the WMNs needs.

This paper's main goal is to propose an extension for the proactive OLSR protocol based on the dynamic choice of metrics, and test its feasibility for implementation in a mesh network scenario recently deployed at the Federal University of Pará (UFPA) [1], in addition to testing its viability in more generic and abstract scenarios. This is done through simulations in Network Simulator (NS) [10] where packet loss, dropping percentage, throughput, blocking probability, delay and jitter are observed.

The metric for routing is chosen based on the type of packet processed at the moment, at the transport layer, thus aiming to answer the applications' needs according to their peculiarities without influencing one another. The proposal is entitled OLSR - Dynamic Choice (OLSR-DC).

This paper is organized as follows. Section 2 show the related works. Section 3 explains the OLSR protocol and some extensions. Section 4 presents the OLSR-DC extension. Section 5 presents the simulation results. And, finally, Section 6 presents the conclusions and future work.

II. Related Works

In this section we presented the previously published related works, providing a relevant information about the WMNs. Problems and challenges are shown in [23], detailing issues and techniques to improve the WMNs performance.

[5] provide a overview about the mesh technology, challenges and concepts concepts on which WMNs must be based.

The reason for the existence of many WMN routing protocols is presented in [14]. And information on OLSR protocol is found in [7].

In [3], to achieve QoS routing, is proposed a solution based on multiple metrics, as in [16].

[2] use composite metric, in OLSR protocol, to improve the routing.

III. OLSR Protocol and Extensions

The OLSR protocol [7] is an adaptation of the link state algorithm, acting as a proactive protocol which uses routing tables obtained through message exchange on the network topology.

According to [15], an OLSR protocol advantage, from the Quality of Service (QoS) perspective, is its proactive nature which allows routes to be avalible before the source needs to start transmitting.

However to [2], the hop count metric defined in the original OLSR is not capable to provide QoS support because a selected path based on this metric may not have the QoS requirements demanded by applications.

In this context were developed some extensions to the OLSR protocol, some of these extensions are listed below.

A. OLSR-ETX

This extension aims to find routes with the least expected number of transmissions that are necessary to a packet be delivered to and confirmed by the destination. Thus, all routing decisions are based on the ETX metric [9].

B. OLSR-ML

In this proposal, the ETX is interpreted as the probability of a successful round-trip, unlike OLSR-ETX where the ETX value reflects the expected number of transmissions. Therefore, the route will be selected according to the greatest success probability [18].

C. OLSR-MD

The main idea of this extension is measuring the link delay, calculating it through the AdHoc Probe technique. With this, the MPR node set and the routing table can be calculated based on the link delay to each neighbor [8].

IV. OLSR - Dynamic Choice

The extensions development to the OLSR protocol, aims to enhance its features and, consequently, its performance. These extensions have followed several approaches such as changes in MPR set selection [3], new metrics for packet routing [8] [9], among others.

Most of these existing protocols aim to provide QoS for different multimedia applications, in other words, to ensure that the network can provide the resources demanded by the applications regarding delay, bandwidth, jitter, blocking probability, throughput, packet loss, bit error rate, and others.

QoS prioritization emerged from the growth of wireless Internet which led to the development of several small applications and devices that use the resources of these networks. So, wireless network users expect to obtain the same types of services that are offered in wired networks [6].

In this context, most of the existing proposals end up not focusing on more traditional computer network applications such as those that use the Transmission Control Protocol (TCP) as their transport layer protocol.

According to [21], about 80% of the existing Internet traffic is based on TCP.

From our bibliographic research, we found out that there is not a routing protocol that aims to answer the needs and characteristics of each packet type. Therefore, this paper proposes OLSR-DC to provide not only QoS for multimedia applications, but that can ensure the performance of TCP-based applications.

Our work began with a performance study on the two transport layer protocols, UDP and TCP, in wireless networks, as well as, the main problems faced by them in this network type. In addition, we also observed the performance of each metric regarding these transport layer protocols. The metrics considered for this paper were the Hop Count [7], Expected Transmission Count (ETX) [9], Minimum Loss (ML) [18], and Minimum Delay (MD) [8].

A. Study on the TCP protocol

The TCP is classically known as a connection-oriented unicast protocol which offers reliable data transmission through its flow and congestion control mechanisms [19].

It was originally designed for wired networks which have low packet loss rates because of the low bit error rate. So, when a packet is lost, the protocol assumes that the loss occurred due to network congestion [4].

TCP becomes more efficient when it experiences a small packet loss which results in lower network saturation, and consequently lower retransmissions number.

Unlike wired connections, wireless ones have high error bit rates with great variation in the signal quality and other factors that are consequently inherited by mesh networks. Thus, TCP is not fully appropriate for mesh networks.

However, TCP may have a performance improvement if used with one of the OLSR metrics. Thus, we carried out simulations on NS [10] using different generation seeds with a confidence interval of 95% according to [13]. The scenario was the mesh network deployed at the UFPA campus (Figure 1). This campus is located in the city of Belem by the Guama river and is composed by many buildings surrounded by parking areas and a large number of trees. It is also divided by a small river, the Tucunduba.

Figure 1. UFPA's Mesh Network

Table I shows the simulations parameters. These parameters represent the network and region characteristics, as well as the used equipment. Path Loss Exponent and Shadowing Deviation parameters were based on the measurements showed in [16].

Table I
SIMULATION PARAMETERS

Parameters	**Values**
IEEE Standard	802.11b
Propagation Model	Shadowing
Antennas	Omnidirectional, 18dB gain
Router's Carrier Sense Threshold	-76dBm
Router's Transmit Power	-80dBm
Frequency	2.422GHz (Channel 3)
Path Loss Exponent	1.59
Shadowing Deviation	5.4dB

We evaluated the performance of each OLSR version varying the number of TCP flows competing for the medium. The flows involved the following points: Capacit →Grad(Básico), Grad(Profissional) → Música, Grad(Básico) → Incubadora e Laboratorios → Secom. Each simulation was run for 50 seconds and performed 50 times.

Amongst the several proposed TCP versions in the literature, TCP-Reno is the most widely used on the Internet and is the standard protocol for the majority of the operation systems. So, we considered it for our simulations.

The protocols analyzed in the simulations were OLSR, OLSR-ETX, OLSR-MD and OLSR-ML. For each protocol, we evaluated the blocking probability, dropping percentage and throughput.

Figure 2 shows the dropping percentage for the OLSR versions considered in this proposal. As it can be seen, OLSR-ETX had the lowest loss, this happens because the protocol succeeded in having a greater number of transmitted packets over the lost ones.

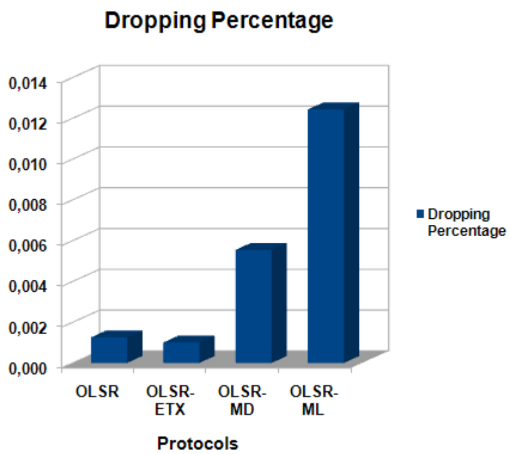


Figure 2. Dropping Percentage

Figure 3 shows the throughput of each flow for each OLSR version. OLSR and OLSR-ETX versions had the best throughput values, but in the overall, ETX had the best results.

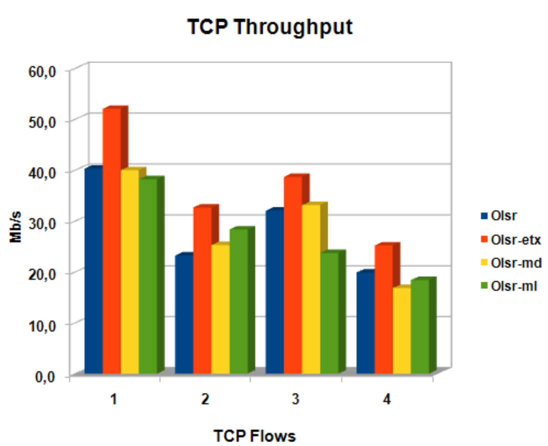


Figure 3. Throughput

The results regarding blocking probability are shown in Figure 4, where it can be observed that OLSR-ETX had the best performance amongst the others.

As for throughput, it is observed that the OLSR protocol based on the ETX metric achieves the best overall performance by efficiently using its routes for the flows, thus avoiding larger packet loss.

This best performance is due to the fact that OLSR-ETX can efficiently distribute traffic flows in way to keep enough resources in the network so that the traffics had the best performance possible.

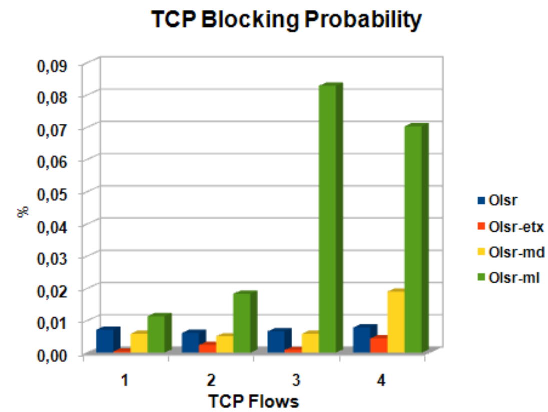


Figure 4. Blocking Probability

Thus, we observed that the ETX metric has the best performance regarding the TCP characteristics. Therefore, we decided that packets which have TCP as their transport layer protocol should be routed based on the ETX metric.

B. Study on the UDP protocol

The UDP protocol is known to be the transport layer protocol based on the "best effort" paradigm, in other words, it does not have reliability mechanisms such as in the TCP. This approach is used to provide a quicker packet delivery.

By not using such mechanisms, UDP makes a tradeoff between efficiency and reliability. That is why it becomes suitable for applications that require packets to arrive as quick as possible and in the right time.

The usage of UDP protocol has increased in the last years with the emergence and popularization of multimedia applications on the Internet and local networks.

So, another concern emerged regarding the ability of networks to provide QoS to such applications [22].

Within this context, applications using the UDP protocol become more efficient from the moment they have the packets being delivered with low delay, and a small variation of it which results in smaller packet loss because these packets will not have to be retransmitted.

With that in mind, we carried out simulations considering the scenario, parameters, and methodology used for the TCP evaluation.

Differently from the TCP study, we now evaluate 6 UDP traffic using Constant Bit Rate (CBR) which characterizes 3 VoIP calls. The packet size was of 40 bytes and the bit rate was of 8Kb/s, this was made to represent the G.729 codec. It is worth remembering that each VoIP call is represented by two UDP flows since it is a bi-directional application.

The VoIP calls involved the following points Capacit ↔ Incubadora, Grad(Basico) ↔ Musica and Grad(Profissional) ↔ Secom. As in the previous simulations, we also considered OLSR, OLSR-ETX, OLSR-ML and OLSR-MD where we observed the blocking probability, dropping percentage, throughput, delay and jitter for each flow.

Figure 5 shows the dropping percentage observed for each OLSR version. It is observed that OLSR-MD has the lowest values amongst the versions due to its great transmission efficiency with respect to the packet loss.

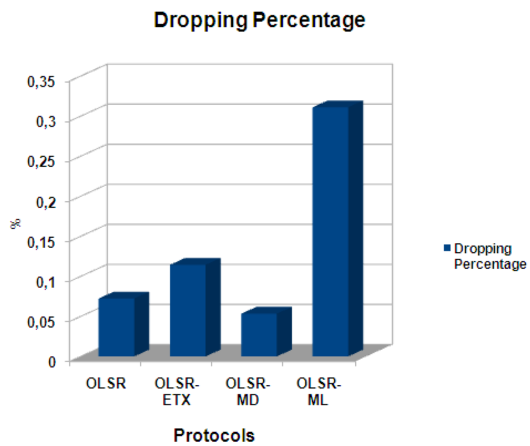


Figure 5. Dropping Percentage

Figure 6 presents the throughput of each flow for the versions. It is observed that all OLSR versions, but OLSR-ML, had a considerably good throughput. OLSR-MD had the best throughput over the others.

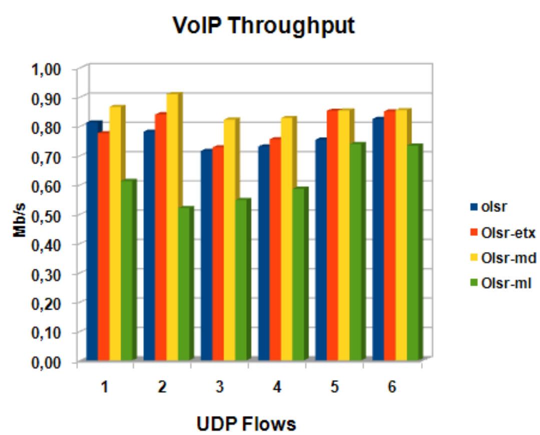


Figure 6. Throughput

The blocking probability is shown in Figure 7 where we can see that all protocols, but OLSR-ML, had almost the same values. It is also observed that OLSR-MD had the best performance.

The delay of each flow is shown in Figure 8 where OLSR, OLSR-ETX and OLSR-MD had close results. Original OLSR had a subtle advantage over the others.

As occurred for the delay, the jitter values, shown in Figure 9, present a relatively similar performance among the verions, especially between OLSR-ETX and OLSR-MD. However, the best performance was achieved by original OLSR, with a minor delay variation.

From the simulations, we observed that the OLSR version which had the best overall performance regarding UDP packets was OLSR-MD, while for delay and jitter OLSR had the best performance.

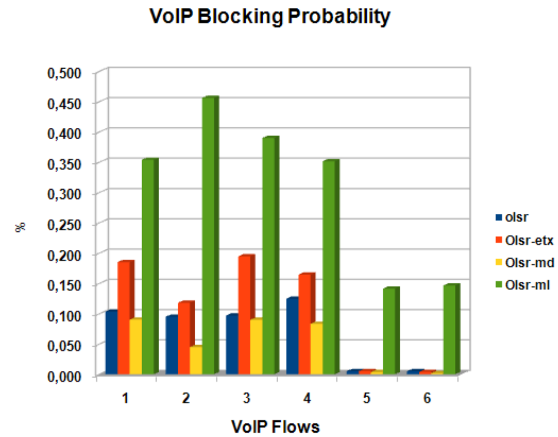


Figure 7. Blocking Probability

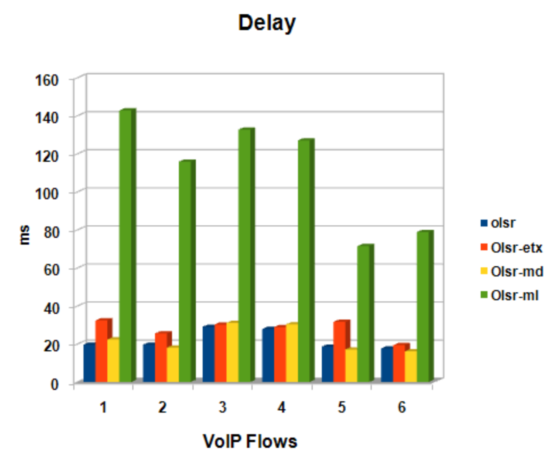


Figure 8. Delay

So, we observed that, in generak, the MD metric has the best performance. Therefore, we decided that packets which have UDP as their transport layer protocol should be routed based on the MD metric.

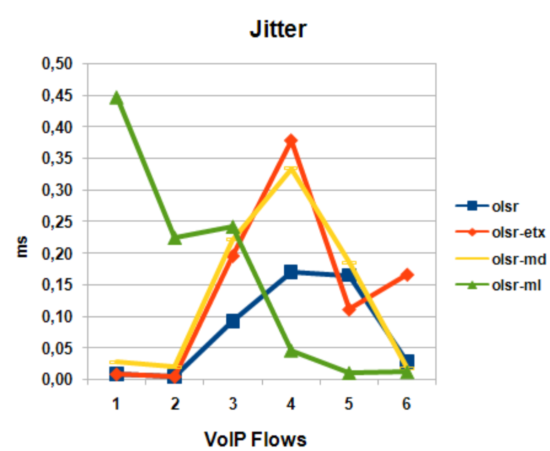


Figure 9. Jitter

C. Changes in the OLSR Protocol

The study conclusion about the performance of each protocol, TCP and UDP, according to the metric used, showed us that the best option for routing TCP packets is to use ETX, as well as the best option for routing UDP packets is to use MD.

From this analysis, we made the following changes in the OLSR protocol:

1) ETX metric calculation changes: The strategy used by the OLSR-ETX protocol to analyze incoming messages is the following: initially the protocol expects for a packet with a given identification number (IDN), when the first message is received, the protocol access its header and compares the packet's IDN with the expected number.

If these numbers match, the protocol updates the expected IDN and processes the packet. Otherwise, it concludes that the packet with expected IDN was lost, then it updates this expected IDN up to its maximum value (default 65535) until it is able to find the received packet's IDN. If this IDN is not found, the expected IDN is restarted, the lost packets number is updated, and then the link quality value can be calculated.

However, MD uses the AdHoc Probe technique for measuring link delay. In this technique, pairs of fixed-size packets are sent to calculate the delay considering clock desynchronization. However, MD uses an extra IDN besides the one OLSR uses. Thus, both packets from a same pair have the same OLSR packet number.

With this, at the time the second packet of the pair is received, the ETX metric calculation is expecting a new IDN. But, instead, it receives a packet with the same IDN from the previous iteration.

When this occurs, ETX assumes that the expected packet was lost. Then, it initiates the process previously explained.

This really makes the ETX metric calculation extremely inefficient, because, for each packet pair, when the second packet is received, ETX performs the search that, in this calculation context, becomes useless.

With that in mind, we changed the algorithm so that, when the second packet from a pair is received, ETX will only process it if its IDN is different from the previous packet.

This solution was adopted since it could solve the repeated IDN problem, still keeping the same ETX metric calculation strategy described in [9], and not influencing in the MD metric calculation.

2) Changes in the Routing Algorithm: After the adjustments made in the OLSR protocol to efficiently calculate both metrics, ETX and MD, according to [9] and [8], changes were made in the routing algorithm, so that it could search for the best available paths based on both metrics.

The algorithm used was the well-known Dijkstra algorithm [17].

It was developed to calculate the lowest cost route in graphs. This fact makes the algorithm not suitable for oaur proposal.

Therefore, we studied some adjustment possibilities to apply the Dijkstra algorithm to OLSR-DC.

To make the Dijkstra algorithm calculate the routes without being run twice, lists and control variables were used which helped in the data organization so that it could perform efficiently.

In addition to the changes mentioned above, we made another change in the algorithm paradigm. Originally, the Dijkstra algorithm, when faced with routes that have the same cost based on the used metric, it chooses the first route found which means that will choose the route with the lowest number of hops.

After observations regarding routes that have the same cost, we concluded that the route chosen should have the largest number of hops.

This approach was adopted because a route with a large number of hops has each of its segments with a low packet loss probability which becomes much more interesting for a efficient packet delivery.

As for delay, such metric is not influenced by number of hops a route has. Thus, we decided if two routes had the same cost, the algorithm would choose the route with the lowest packet loss probability.

3) Changes in the Routing Table: Originally, the OLSR protocol performed the discovered routes insertion in the routing table according to the number of hops to reach the destination node [7].

This insertion strategy continued to be used in OLSR-DC, because it was found that it does not have a negative influence in the proposal.

In the proposed extension, we modified three existing fields in the OLSR and added three more. For the OLSR-DC, the routing table would keep only the R_dest_addr field of the original approach, and would have the following new fields:

- R_next_addr_quality: node which we need to send the packet to reach the destination, according to the established route that has the best link quality;
- R_next_addr_delay: node which we need to send the packet to reach the destination, according to the established route that has the lowest link delay;
- R_dist_quality: hops needed to reach the destination, according to the chosen route which has the best link quality;
- R_dist_delay: hops needed to reach the destination, according to the chosen route which has the lowest link delay;
- R_iface_addr_quality: interface output address corresponding to the route with best link quality to the destination node;
- R_iface_addr_delay: interface output address corresponding to the route with lowest link delay to the destination node.

In OLSR-DC, we kept the OLSR strategy of recalculating the routing table every time that changes occur in the link local information, as well as the information on topology since the table is made from the data obtained from this information.

4) Changes in the Packet Forwarding: Below we show the steps performed by OLSR protocol to the data packet forwarding according to [7]:

- Access the packet header, to extract the information for the following steps;
- From the extracted information, it is checked if the packet is addressed to the current node or whether it is broadcast packet. If the current node is the destination, then the packet is sent to the upper layers. Otherwise, the process continues;
- After checking destination, check in the routing table if the packet's destination is available. If the destination is not found, the packet is discarded. Otherwise, processing continues;
- At this point, it is extracted the need information from the table (e.g. the next hop), and they are added to the packet;
- Finally, the packet is encapsulated by the lower layers and then sent to its next hop.

Basically two changes were made in the steps mentioned above. First, in the information extraction, in addition to the originally used information, the process must also access the field that indicates the packet's transport layer protocol being used, so this information could be used in the next step.

Subsequently, in the next hop checking step, we implemented in a way that the table field used to extract the needed information is accessed according to the packet's transport layer protocol found earlier.

In other words, UDP packets will use the table field regarding the path with the lowest link delay, the R_next_addr_delay field, while TCP packets will use the table field regarding the path with the best link quality, the R_next_addr_quality field.

With this, the packet is routed according to the metric that provides it with a better performance in the network.

5) Metric selection to be used for determining the MPR node set: The control messages forwarded through the nodes that are part of the MPR set which are addressed to all network nodes, are incorporated into UDP packets for transmission in the network.

When a message is lost, it will not be retransmitted. With this, the information about the topology will, for a long period, not be update.

So, we decided to adopt the ETX metric to be part of the selection criterion for the MPR node set, because this metric perform a better message delivery.

V. Results

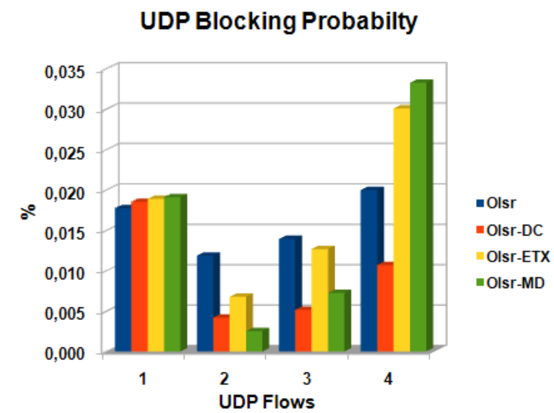
In this section, we show the performance of OLSR-DC protocol along with the other protocols considered in this paper. As in Section IV, we considered the same scenario, parameters, and methodology for the simulations discussed here. However, these simulations were run for 150 seconds.

Table II shows the information for each flow including ID, source, destination, begin/end of transmissiona, and the traffic type.

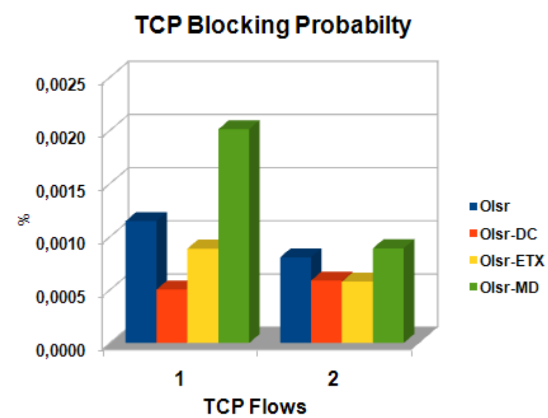
The figures with the simulations results are shown below, and the throughput and blocking probability figures show the TCP flows and the UDP flows separately.

Table II
Flows Configuration

Flow	Source	Destiny	Begin	End	Traffic Type
1	Laboratrios	DI	50	130	TCP - Reno
2	SECOM	Msica	60	140	TCP - Reno
3	CAPACIT	Grad(Profissional)	30	100	UDP - CBR
4	Grad(Profissional)	CAPACIT	30	100	UDP - CBR
5	CT	Grad(Básico)	40	110	UDP - CBR
6	Grad(Básico)	CT	40	110	UDP - CBR



(a) Blocking Probability of UDP Flows



(b) Blocking Probability of TCP Flows

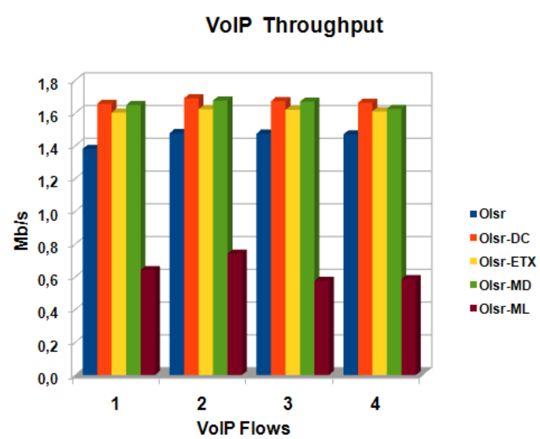
Figure 10. Blocking Probability

Figure 10a shows the blocking probability of each VoIP flows (four UDP traffics) where flows 1 and 2 represent the first VoIP call, and flows 3 and 4 represent the second VoIP call. The OLSR-ML results were omitted from the figures due to its high values of blocking probability over the others.

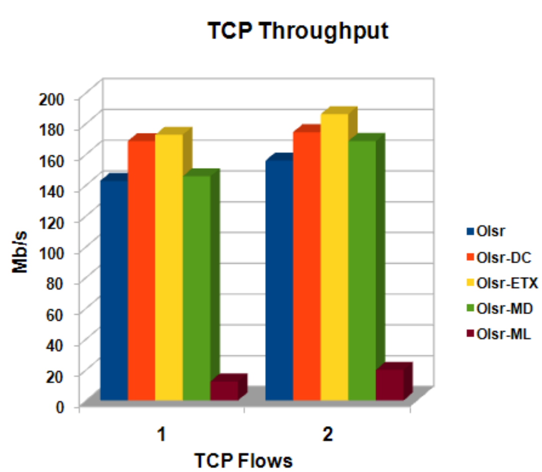
As it can be seen, our proposal had the best overall over the others for most of the UDP flows. OLSR-MD performed better than OLSR-DC only for the second flow. And for the fourth flow, OLSR-DC had the best performance. At the time of this transmission, such flow was suffering with interference from the other flows. This shows that the proposal is able to perform better even with interference and high distance between the communicating nodes.

Figure 10b shows the performance of the TCP flows also regarding the blocking probability. For the first flow, OLSR-DC had best overal performance over the other versions. We believe that the subtle difference between OLSR-DC and OLSR-ETX for the second flow is due ato the interference experienced by this flow. Also these low values of blocking probability for the TCP flows are explained by the distance between the communicating nodes which explains the good performance of original OLSR since the number of hops is very small.

Despite of OLSR-DC and OLSR-ETX have the same metric for packet routing, the best performance of OLSR-DC is due to the changes made in the routing algorithm described in Section IV.



(a) Throughput of UDP Flows



(b) Throughput of TCP Flows

Figure 11. Throughput

Figure 11a shows the performance of the UDP flows regarding the throughput. OLSR-DC had the best overall performance which is a result from the adaptations made. Besides being based on the same metric, OLSR-MD had the second best results.

The performance of the TCP flows regarding the throughput are shown in Figure 11b. OLSR-ETX had the best overall values. Besides being the second best, OLSR-DC was able to keep a good throughput level for the TCP flows.

The delay for both TCP and UDP flows are shown in Figure 12 where flows 1 and 2 represent the TCP flows and flows 3 to 6 represent the UDP flows. Regarding the TCP flows, OLSR-DC performed better than the OLSR-ETX. Although they suffered with high delay, TCP applications are more delay tolerant.

The very small delay values of OLSR-MD is explained by number of packets it managed to send which was very low too, and since this delay in function of the number of received packets, this explains this fake best performance. As for the UDP flows, OLSR-DC protocol had the best overall values, being only beaten by OLSR-MD in the first UDP flow (flow 3 in Figure 12), in a few moments being overcome by the OLSR-MD protocol.

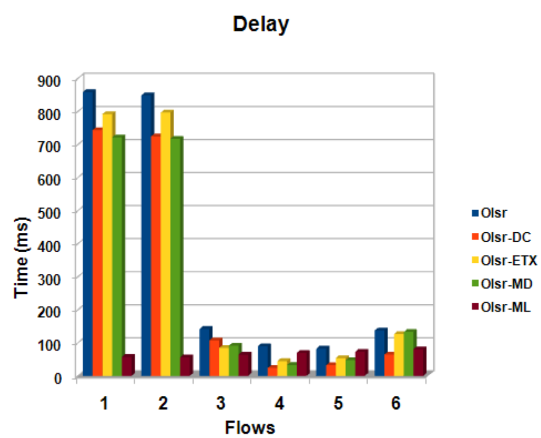


Figure 12. Delay

Jitter is shown in Figure 13 where only the jitter values for each of the UDP flows are considered since such performance measure is not important for TCP flows. OLSR-ML and OLSR had the best overall jitter values, however OLSR-DC's performance remained below the acceptable threshold when it comes to VoIP applications which shall not degrade the quality of such calls.

The fact that OLSR-MD and OLSR-DC use the same metric for packet routing, makes these two protocols achieve similar results for the VoIP flows. The difference between the protocols is due to the changes that were made in the routing algorithm of OLSR-DC protocol.

VI. Conclusion and Future Works

This paper presented an extension to the OLSR routing protocol based on dynamic choice of metrics, which is based on the transport-layer protocol used by the packet processed at the moment of routing. The main goal with this is to try to answer the needs of the applications according to their particularities without influencing one another.

We showed that OLSR-DC can have a better, and sometimes a similar, performance if compared to protocols that use only

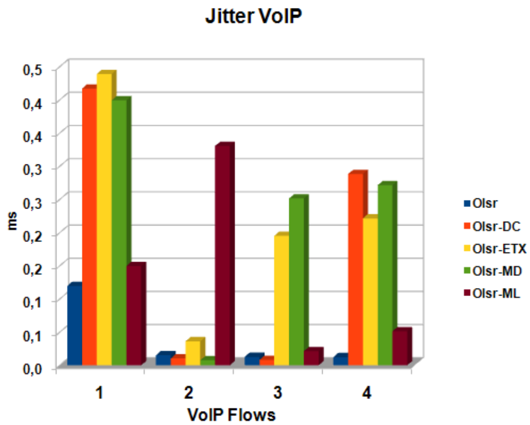


Figure 13. Jitter

one metric, as OLSR-ETX and OLSR-MD. The ETX metric was proved to perform better for TCP-based applications, and the MD metric does quite well for UDP-based applications.

In this context, the protocol OLSR-DC aimed to extract the capacity of each of the metrics above, in others words, attempts were made to have a performance similar to the protocol OLSR-ETX regarding to applications that use the TCP, and similar performance to OLSR-MD with respect to applications based on UDP protocol, generally multimedia applications.

So, we showed that the protocol OLSR-DC has a similar performance to the protocol OLSR-ETX and higher than the protocol OLSR-MD, when it comes to applications that use the TCP protocol. Similarly, the OLSR-DC had a performance, whether dealing applications that use UDP, similar to the protocol OLSR-MD, and sometimes better than this and the protocol OLSR-ETX.

Therefore, this paper proposed an extension of the OLSR protocol that, despite the fact that it generates a subtle increase in processing and memory consumption, can better serve the applications independently of the transport layer protocol being used.

As future works, we intend to implement the proposed extension for the wireless mesh network deployed at UFPA, and thus evaluate this proposal's performance in a real testbed.

Another intention is to study the possibility of adding other metrics, or replacing the used ones, to provide the routing considering other characteristics of a packet.

Besides that, we plan a refinement of the routing algorithm proposed, aiming to reduce the extra processing that was generated with the current algorithm, and some ideas to help with this improvement can be based on the following works [20], [17], [12] and [11].

References

[1] Aguiar, E., Bittencourt, P., Moreira, W. e Abelm, A. (2007) Estudo comparativo de protocolos de roteamento para redes Mesh na regio amaznica. 25 Simpsio Brasileiro de Redes de Computadores e Sistemas Distribudos. Maio.

[2] Aslam, N., Phillips, W., and Robertson, W. (2004). Composite metric for quality of service routing in olsr. Canadian Conference on Electrical and Computer Engineering, 2:759762.

[3] Badis, H., Munaretto, A., Agha, K., and Pujolle, G. (2003). Qos for ad hoc networking based on multiple metrics: bandwidth and delay. Fifth IFIP International Conference on Mobile and Wireless Communications Networks.

[4] Bakre, A., Badrinath, B. R. (1995) I-TCP: Indirect TCP for mobile hosts. Distributed Computing Systems, Proceedings of the 15-th International Conference, pages 196 143, June.

[5] Bruno, R., Conti, M., and Gregori, E. (2005). Mesh networks: commodity multihop ad hoc networks. IEEE Communications Magazine, 43(3):123131.

[6] Chinta, M., Helal, S. (2003) ILC-TCP: An Interlayer Collaboration Protocol for TCP Performance Improvement in Mobile and Wireless Environments. Proceedings of the Trird IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, Louisiana, March.

[7] Clausen, T. and Jacquet, P. (2003). Optimized link state routing protocol (olsr). RFC 3626, Experimental.

[8] Cordeiro, W., Aguiar, E., Moreira, W., Abelem, A., and Stanton, M. (2007). Providing quality of service for mesh networks using link delay measurements. 16th International Conference on Computer Communications and Networks, pages 991996.

[9] De Couto, D. S. J., Aguayo, D., Bicket, J., and Morris, R. (2003). A high-throughput path metric for multi-hop wireless routing. 9th Annual International Conference on Mobile Computing and Networking, pages 134146.

[10] Fall, K. and Varadhan, K. (2007) The Network Simulator NS-2. URL: http://www.isi.edu/nsnam/ns/.

[11] Xuhui Hu; Lee, M.J.; Saadawi, T.N.; Progressive Route Calculation Protocol for Wireless Mesh Networks. Communications, 2007. ICC '07. IEEE International Conference on 24-28 June 2007 Page(s):4973 4978.

[12] Ikikardes, T.; Hoftauer, M.; Kaelin, A.; May, M.; A Robust, Responsive, Distributed Tree-Based Routing Algorithm Guaranteeing N Valid Links per Node in Wireless Ad-Hoc Networks. Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on 1-4 July 2007 Page(s):455 462.

[13] Jain, R. (1991). The Art of Computer Systems: Techniques for Experimental Design, Measurement, Simulation, and Modeling. John Wiley & Sons, 1st edition.

[14] Kowalik, K. and Davis, M. (2006). Why are there so many routing protocols for wireless mesh networks? Irish Signal and Systems Conference.

[15] Leguay, J.; Conan, V.; Friedman, T. QoS routing in OLSR with several classes of service. In: 4th Annual IEEE International Conference on Pervasive Computing and Communications Wokshop, 2006, Pisa, Italy, pp. 6.

[16] Moreira, W., Aguiar, E., Abelm, A e Stanton M. (2008) Using Multiple Metrics with the Optimized Link State Routing Protocol for Wireless Mesh Networks .26 Simpsio Brasileiro de Redes de Computadores e Sistemas Distribudos. Maio.

[17] Noto, M.; Sato, H.; A method for the shortest path search by extended Dijkstra algorithm. Systems, Man, and Cybernetics, 2000 IEEE International Conference on Volume 3, 8-11 Oct. 2000 Page(s):2316 - 2320 vol.3.

[18] Passos, D., Teixeira, D., Muchaluat-Saade, D., Magalhes, L., and Albuquerque, C. (2006). Mesh network perfomance measurements. 5th International Information and Telecommunication Technologies Symposium.

[19] Postel, J. (ed.), "Internet Protocol - DARPA Internet Program Protocol Specification", RFC 791, USC/Information Sciences Institute, September 1981.

[20] Sobrinho, J.L.; Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet. Networking, IEEE/ACM Transactions on Volume 10, Issue 4, Aug. 2002 Page(s):541 550 Digital Object Identifier 10.1109/TNET.2002.801397.

[21] Thomson, K., Miller, G. J., Wilder R. (1997) Wide-area Internet Traffic Patterns and Characteristics. IEEE Network, Vol. 11, No. 6, November.

[22] Ying Ge; Kunz, T.; Lamont, L.; Quality of Service Routing in Ad-Hoc Networks Using OLSR. System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on 6-9 Jan 2003 Page(s):9 pp.

[23] Zhang, Y., Luo, J., and Hu, H. (2007). Wireless Mesh Networks: Architectures, Protocols and Standards. Auerbach Publications, 1st edition.