A Performance Evaluation of Multihop Relaying vs. Multihop Routing

Luís Carvalho* (lmmc@inescporto.pt), Rute Sofia (rsofia@inescporto.pt)
IAN, UTM, INESC Porto**

***Abstract*—This paper is focused on one aspect of *user-provided networking*, namely, on a simple form of wireless relaying. The paper provides a first analysis of wireless relaying based on a realistic setting of a wireless testbed. The benchmark chosen for this analysis is a specific form of multihop routing, the Dynamic MANET On-demand protocol. Performance results are presented in the form of jitter, end-to-end delay, and average packet loss. Results achieved show that relaying experiences less variability than its routing counterpart, as well as an overall slightly better performance.**

***Index Terms*—wireless networks, relaying, multihop routing, user-provided networks.**

I. INTRODUCTION

Wireless Local Area Networks (WLANs) are now widely deployed at private homes and public spaces, with a tendency to spread further. They present a viral growth pattern which is not in any way controllable nor manageable by access stakeholders and hence, such growth affects network performance due to the intrinsic features of the 802.11 *Media Access Control* (MAC).

Moreover, these private WLAN environments are normally not used at their full capacity. For instance, it is common nowadays to complement *Digital Subscriber Line (DSL)* access with one wireless *Access Point* (AP) per household, serving in average a few users and a small set of end-user wireless devices (in average, 2 or 3). On the one hand, such configuration implies that radio resources are not being fully used. On the other hand, due to the fact that it is common to have more than one wireless AP within the same range, there is strong spectrum overlap, which undermines the wireless coverage.

The mentioned wireless deployment results in an autonomic spreading of a wireless architecture, which is commonly known as a *user-provided network (UPN)* [15]. Today there are already a few examples of commercial UPNs, e.g. Wifi.com or FON. It should be noticed that a UPN is simply a wireless architecture, and that the commercial entities that manage UPNs behave as *virtual operators. A* virtual operator is simply an entity that assists user-centric connectivity models, i.e., connectivity models where end-users cooperate within specific communities by sharing Internet access on-the-fly. In contrast to service providers or access operators, virtual operators do not own/rent any kind of infrastructure nor do they truly provide a service. Their core business relates to the initial coordination (management) of the community in itself.

Another relevant and yet quite simple example of an UPN is an architecture set on-the-fly, based on an end-user terminal with Internet access (e.g. HSPDA Internet access), being such Internet access relayed to a few devices in the wireless range of the laptop. In this type of scenario, quite common today when users travel (e.g., a small family accessing the Internet on a hotel based upon a 3GPP hotspot), there is no need to set up a wireless router. UPNs are further debated in section III-A, but what is relevant to highlight at this stage is that the UPN living-examples of today rely on wireless infrastructure mode (not ad-hoc mode) and on a 1-hop transmission range.

A potential evolution of UPN models incorporating multi-hop relaying has been debated by Sofia et al. [15]. Relaying is appealing due to its simplicity in terms of configuration, status maintenance, as well as network operation. However, there is a performance trade-off associated to relaying, as there is a performance trade-off associated to multihop routing. For instance, at a first glance and considering the MAC layer of 802.11, relaying requires full synchronization of the nodes involved in the process, which may be hard to achieve in a complex network with a large number of nodes.

Throughout our research related with wireless relaying several questions arose: can relaying be as stable as routing in UPNs? If so, can it scale well with the number of nodes in the topology and with an increase in network load, both in terms of transmission rates and of concurrent sources/sessions? And what is the overhead cost associated with implementing relaying, in contrast to routing?

It is within this context that the work presented in this paper is focused. Such work has as main purpose to understand the potential performance trade-off of relaying on realistic settings and having as benchmark multihop routing. Hence, the results provided in this paper are the outcome of experiments carried out in a local wireless testbed which has been set up to mimic as close as possible a realistic environment.

The remainder sections are organized as follows. After this introduction, section II goes over related work, highlighting our contribution. Section III describes notions associated with relaying, multihop routing, as well as the environment that we relied upon to perform an adequate evaluation. Section IV is dedicated to the performance evaluation and to the results analysis, while section V presents conclusions and next steps.

II. RELATED WORK

The environment where wireless relaying seems to be more interesting are today's autonomic wireless architectures, which

---

* Corresponding author.

**Internet Architectures and Networking (IAN), UTM, INESC Porto. Rua Dr. Roberto Frias, 378, 4200-465 Porto.

can be seen as residing on the last hop (local-loop or just the final segment of it towards the end-user) of the Internet. These spontaneous and privately owned wireless deployments spread based upon cooperation incentives between end-users and between end-user and operators. In related literature, spontaneous wireless architectures are addressed by different names, being UPN a specific form of such networks.

Sofia et al. [15] describe notions and challenges related with UPNs. Our work is focused on UPN environments, namely, on wireless architectures that rely on a user-centric relaying model, where connectivity sharing is placed in the end-user device. This gives the means for trusted nodes to take advantage of such connectivity, and hence the inherent infrastructure results from what is today being applied in privately owned WLANs, which are in their majority infrastructure-mode based.

Albeit being an intrinsic feature of the wireless MAC Layer, multihop relaying has been considered in a number of related work. For instance, R. Gitlin et al. address two-hop relaying aspects in [16] with the purpose of providing capacity enhancements to cellular networks. This scheme works by having nodes with dual-mode WWAN/WLAN connections periodically advertise their 3G channel conditions through the WLAN interface. Neighbouring nodes, which may or may not have a 3G interface, can then select the node with the best link quality to act as a relay. Their two-hop relay scheme is simple and effective in avoiding the overhead associated with multihop routing, but still improves the performance of the network. The authors show that an adequate relay selection can improve the overall network throughput in about 200-400%.

Following the same line of thought, Pan War et al. introduce CoopMAC [11], a new MAC protocol where high data rate stations assist low data rate stations in relaying. CoopMAC is therefore a 2-hop cooperative relaying scheme. The authors show that by minimizing the transmission times (and although increasing the number of transmissions), throughput is increased and delay and interference are reduced. One last unexpected outcome from their analysis is that the energy consumption in the relay nodes is even reduced, as they stay longer in idle periods, which is another argument in favor of cooperation.

The two mentioned proposals recur to 2-hop relaying to improve the capacity of wireless networks. Our work addresses multihop relaying in its simplest form, not involving optimal relay selection and provides a performance evaluation based on realistic network conditions.

S. Gormus et al. [7] provide a first comparison of relaying to a specific form of on-demand routing with the purpose to assess network performance in terms of throughput increase. This is done by analysing how to optimize the placement of relays (cooperative relaying). They show that there is a gain in terms of network throughput. Their work is the one that is closer to ours. However, while the authors address relay selection and placement having in mind to increase network throughput, our work relates to a more realistic setting, where the purpose is to assess up to which point could relaying be interesting if using a real set-up (without cooperative relaying), when compared to on-demand routing.

In what concerns multihop routing related work, the most popular routing protocols are: *Ad-hoc On-demand Distance Vector* (AODV) [13], and its updated version, *Dynamic MANET On-demand* (DYMO) [14]; *Optimized Link State Routing* (OLSR) [9]; *Dynamic Source Routing* (DSR) [10]; *Temporally Ordered Routing Algorithm* (TORA) [12]. Comparative studies, be it by means of simulations or by means of real testbed based experiments, show that in face of node mobility, reactive protocols have better performance than their proactive counterparts [3]. The analysis related to multihop routing performance in different settings shows that the best multihop routing approach to apply in UPN environments is a reactive on-demand approach. Hence, we considered one implementation of DYMO.

The next section goes over the setup of the evaluation framework we relied upon.

III. EVALUATION FRAMEWORK

In order to assess relaying performance based upon the questions presented in section I, a local wireless testbed has been set up and configured with a number of parameters. The global evaluation framework included two main building-blocks: i) implementing and evaluating a simple form of relaying; ii) relying on a specific form of multihop routing as benchmark for the validation aspects.

A. UPN Basics

Fig. 1 illustrates a generic scenario for a simple UPN scenario which will assist the explanation of the two building blocks of the evaluation framework.

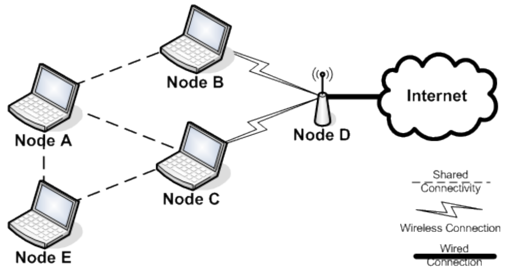

Figure 1. Generic UPN scenario. Node D is an AP, nodes B and C represent relays and are on the direct range of node D. Nodes A and E represent regular end-user devices.

In such scenario, node D corresponds to an AP owned by a specific user, who is willing to share his/her Internet connection with other users within his/her community. Nodes B and C are in the direct connectivity range of node D. However, nodes A and E are not in the direct range of D but they would profit to have access to the Internet connection shared by D. A and E can profit from such Internet access either by relying on multihop routing or on multihop relaying. There are three main differences between relaying and routing: i) there is no path computation involved on relaying; ii) status information concerning paths are not kept on relaying; iii) both approaches can support multipath, but while routing embodies an end-to-end perspective, relaying integrates a local perspective only (selection of the optimal successor set is a local optimization problem, not a global one).

B. Relaying Block

The ability to do relaying is an intrinsic feature of any end-user device as of today, given that relaying can be performed on OSI Layer 2 or 3 with any available operating system.

Relaying is performed on Layer 2 by recurring to bridging. Again relying on Fig. 1, in order for node A to profit from the Internet access of node D, then node B and/or node C have to become a *relay*, i.e., a software bridge has to be established between his two interfaces, allowing both networks to be seen as a single segment for the upper Layers of his stack and therefore, allows traffic from D to be seen by A. In this case, nodes A, B, and D are not necessarily connected to form an ad-hoc network.

As mentioned, relaying can also be performed by relying on features of the OSI Layer 3, i.e., IP masquerading. Masquerade works by having each node change the source IP of each packet it forwards, to its own IP address, like NAT. When a packet comes in the reverse path, nodes do a similar procedure, changing the destination address instead. This is actually different from routing, and in user-provided networking has implications in terms of traceability [15].

OSI Layer 2 relaying has the advantage of forwarding the packets in an earlier stage of the TCP/IP stack, meaning that in a network card which has the MAC layer logic implemented directly in its firmware, forwarding is done with no additional processing overhead. However, when using a network card with a softmac layer, the processing of the MAC information is already done by software. As such, there is no additional overhead in relaying on Layer 3.

Relaying in our evaluation framework is performed on OSI layer 3, recurring to IP masquerade, given that it was felt that it would be better to compare relaying vs. routing at the same OSI Layer.

Given that no path computation mechanism is involved, relaying requires a specific selection of one or more successors. For the sake of simplicity, we configured the preferential neighbour by default. In a real-world implementation a specific mechanism related to trust management and cooperation incentives would have to be the basis for such a selection.

C. Multihop Routing Benchmark

There are several implementations of both AODV and DYMO publicly available, mainly for Linux systems. Such implementations either fall into user-space or kernel-space, and such categorization has a significant impact in terms of performance and ease of use. Keeping everything in kernel-space can cause compatibility issues, as well as causing potential system instability due to buggy implementations. User-space implementations can be easier to implement and to use, but normally achieve a lower performance. Tests have shown that a user-space daemon can take up to 1 order of magnitude more in packet forwarding than its kernel-space equivalent [5].

In [4] we analysed the most popular implementations of AODV and DYMO available, and concluded that NIST-DYMO was the only one matching our needs.

The NIST-DYMO implementation was coded for Linux kernel 2.6.8, but our testbed required at least version 2.6.27 due to, among other issues, virtualization support. So the first task performed was to port NIST-DYMO, replacing the deprecated function calls with their recent equivalents.

Besides porting the code to the newer kernel, NIST-DYMO had some implementation faults, like the way routes are acquired and maintained and some race conditions in route handling. One other issue about this implementation is the lack of interoperability between different hardware architectures. We improved these features, turning NIST-DYMO into a more robust, cross-platform and updated version that can easily be deployed and used in real-world scenarios. We tested this implementation in the testbed, with 6 nodes of different architectures and the results were good. The code is available at [1].

IV. PERFORMANCE EVALUATION

This section is dedicated to the performance evaluation which attempts to answer the questions that led to this work. The experiments here described were carried on a realistic testbed composed of six nodes. Four nodes had an x86 architecture, while two were embedded (mips architecture) APs. All nodes were part of the same ad-hoc network, configured on channel 8 in order to minimize overlap with devices around the testbed, and the maximum rate of 802.11g (54Mbps) was considered. Due to space constraints, all the nodes in the network have been kept physically close and iptables filtering was used to drop overheard packets.

For each experiment, MGEN was used to model VoIP traffic, according to G.711 [2]. A specific set of bursty UDP flows were started following a Poisson distribution, each contributing to the average network load. Each individual flow modeled a VoIP session, with an ON period with an exponential average of 4s and a duty cycle of approximately 70% [6].

Nodes have been synchronized before each experiment with a local NTP server. Each experiment lasted 5 minutes, with an additional warm-up period of 40 seconds. Experiments I and II were run 10 times with different random seeds and results have been obtained within a 95% confidence interval. However, due to time constraints, it was only possible to run Experiment III five times and, as a result, the confidence intervals are not shown. Furthermore, for each experiment set, the number of flows on the system was varied to emulate different network loads.

Results were obtained in the form of *packet loss*, *end-to-end delay*, and *jitter*.

Packet loss is expressed in percentage and defined as the ratio between the number of lost packets and the number of packets that were sent.

End-to-end delay for a packet x is defined as the time between when the packet was sent and when it was received.

Jitter is here used to denote packet jitter per flow, i.e., the inter-packet delay variation. Packet jitter is therefore computed relying on the inter-packet delay between packet *x* and its predecessor *x-1*. Then, the total jitter for a specific flow is defined as the average of every packet jitter.

Three main experiments have been run, as described next.

A. Experiment I

The first experiment run relies on the topology illustrated in Fig. 2. It is a linear topology with an average six hop path, being F the only source, and Dest the destination. The network load is then varied by having F sending 5, 10, 20, 30, and 40 bursty UDP flows as described. This corresponds roughly to rates of 40KBps, 80KBps, 160KBps, 240KBps and 320KBps, respectively. The experiment models a scenario where DYMO is expected to show a worst-case scenario behavior [8]. The purpose is to understand if in such conditions, relaying experiences the same scalability problems.

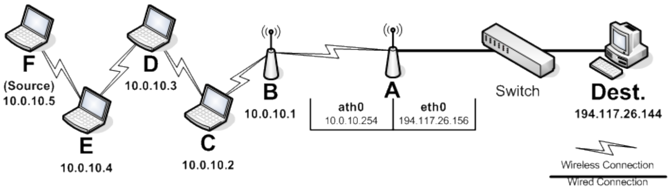

Figure 2. Topology I.

Fig. 3 shows that when the network is underused, both DYMO and relaying attain low delay values. Around 30 flows, both DYMO and relaying see a significant increase in delay, which is simply a consequence of the network becoming congested. Our hypothetical explanation for this behavior relates to high network congestion, and to the fact that in such cases some of the required path setup signaling is lost, thus resulting in more variability for the case of DYMO. Overall, relaying shows a slightly better performance than DYMO. Our explanation for this behavior relates to the control overhead that DYMO attains.

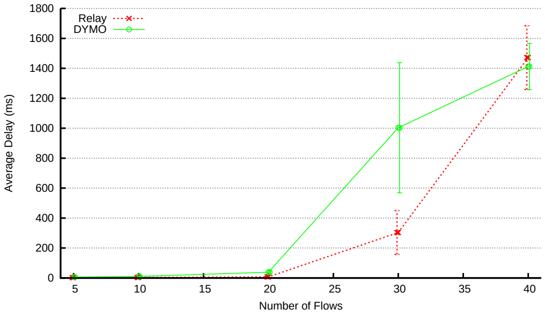

Figure 3. Average delay in experiment 1. Average delay achieved in milliseconds (y-axis) for the different loads represented by the number of average concurrent flows on the system (x-axis).

Let us now look into the results related with packet loss. In Fig. 4, we can denote that the increase in packet loss closely follows the increase in delay. In this case, both parameters grow exponentially with more than 20 flows. This happens because, as stated before, the network becomes saturated.

In regards to the behaviour of relaying when compared to DYMO, relaying again shows a slightly better behaviour as with packet loss. This is also justified by control overhead.

Jitter results (cf. Fig. 5) are the most relevant in this scenario, given that they corroborate the conclusions drawn from the previous plots, namely, that DYMO attains more variability than relaying, due to the need to build routes in a network that shows high variability due to bursty traffic.

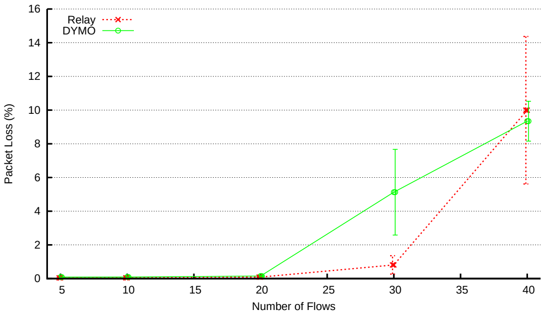

Figure 4. Packet loss in experiment 1. Packet loss in percentage (y-axis) against the average number of flows in the network (x-axis).

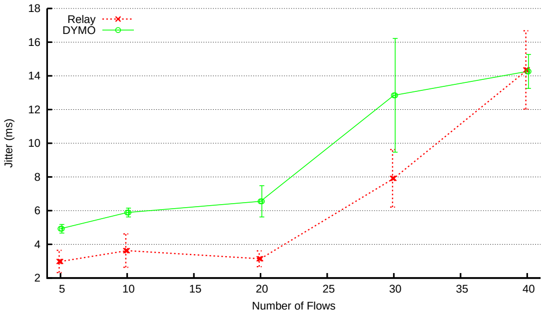

Figure 5. Jitter in experiment 1. Jitter is provided by the y-axis, in milliseconds vs. the number of concurrent flows (x-axis).

Some additional remarks relate to the variability observed in this scenario and corroborated by the large 95% confidence intervals. This was due to the high number of hops in the scenario, which in turn causes a lot of unpredictable MAC contention, in particular due to the fact that every node is within reach of every other. Furthermore, as there is no centralized control over the network, there is no warranty that every node will get a fair share of transmission time slots.

B. Experiment II

In order to assess if the results obtained in the previous scenario were due to the settings, a new topology was set, as illustrated in Fig. 6.

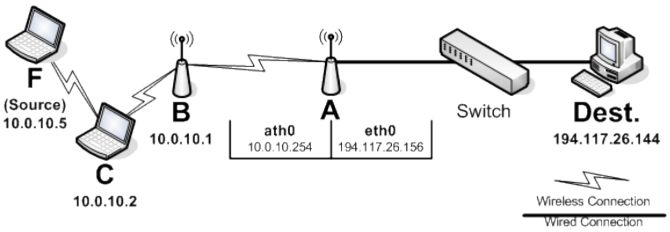

Figure 6. Topology II.

This is still a linear topology where the average path length was reduced from 6 to 4 hops. This now corresponds to a scenario that is beneficial to DYMO. Traffic settings were kept similar to the ones described in Experiment I (cf. section 4.1) but the number of flows on the system was varied from 10 (80KBps) to 60 (480KBps) to understand if results persisted for higher loads. Let us look first to results in the form of end-to-end delay, plotted in Fig. 7.

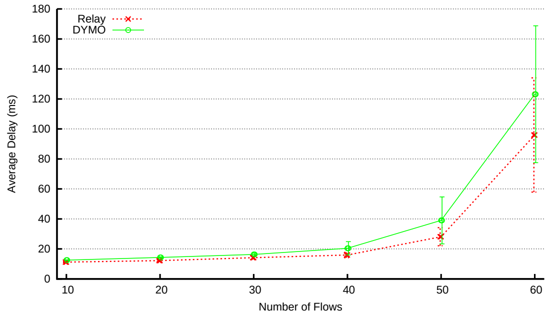

Figure 7. Experiment 2 delay.

Relaying still attains slightly lower values than DYMO, which is coherent with DYMO's control overhead. In regards to Experiment I, results now obtained attain significantly lower values in average. Saturation of the network is now reached around 50 flows and hence the more abrupt raise in the delay.

Packet loss results show a similar behavior (cf. Fig. 8). Even with as many as 50 concurrent flows, the results are good enough for real time traffic constraints [8], with a maximum packet loss of around 0.4% and 40ms of maximum delay. This is simply a consequence of the lower number of hops in the topology used. Only with 60 concurrent flows do we get a significant increase in delay and in the number of lost packets.

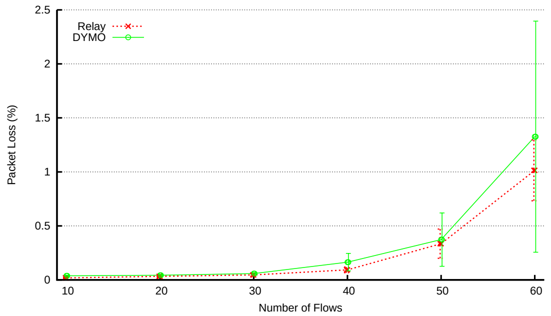

Figure 8. Experiment 2 packet loss.

Looking at the jitter behavior (cf. Fig. 9), DYMO again shows more variability than relaying. Overall, jitter is also significantly lower than in the previous set of results. The value rises with the increasing number of concurrent flows, but does not vary more than 3 milliseconds.

The results obtained with this experiment show that relaying is indeed more stable that DYMO, and that such stability is not a consequence of a worst-case scenario for multihop routing, given that the results are similar both with 6 and 4 hops. The stability observed seems to be independent of the average path length, but this is something that can only be further corroborated by running additional experiments in more complex settings, and varying the average path length. We leave this to explore as future work.

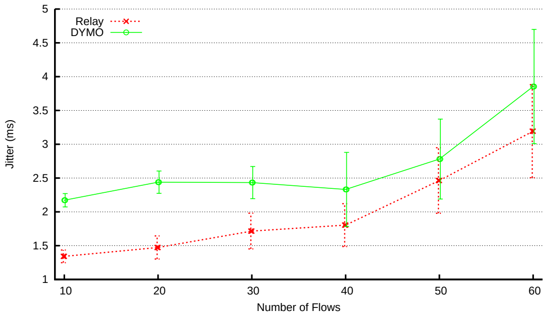

Figure 9. Experiment 2 jitter.

C. *Experiment III*

The previous two experiments had as purpose to understand if relaying would perform equally to DYMO both in good and bad conditions in terms of average path length. Experiment III has the purpose to understand what could be the impact of having more than one source (more than one station) competing for the wireless media. In order to allow a fair comparison to the previous two scenarios, a new topology has been set as illustrated in Fig. 10. Such topology repeats the settings of the topology used on Experiment II, but now having two sources (E and F) instead of one.

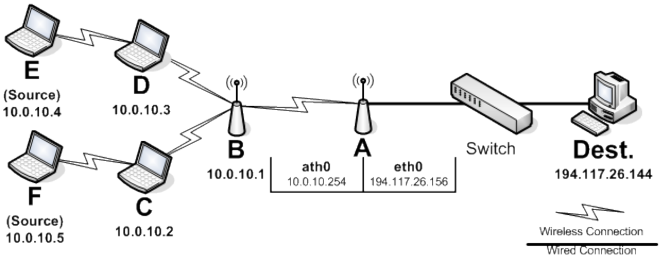

Figure 10. Topology III.

The same load is kept on the network, i.e., concurrent flows vary between 10 and 60. The average delay and packet loss are depicted in Fig. 11 and Fig. 12, respectively.

The first observation to draw from the results obtained is that DYMO and relaying show a closer behaviour with relaying presenting slightly lower values. There is therefore less variability in DYMO in contrast to the previous experiments run. Main reason for this is that the load on the network is being distributed by two sources and hence routes are possibly being kept active for a longer period. In contrast, for the linear topologies relied before, intermediate nodes would see the routes updated more frequently.

Let us now compare these results with experiment II, which holds the same path length but a single source.

The lower average delay observed in Experiment III (cf. Fig. 11) occurs because two sources are contending for the media on the first transmission hop. The same behaviour is observed for packet loss (cf. Fig. 12). In average, experiment III has half of the delay and half of the packet loss of experiment II. This was expected, given that the network load is being generated by two sources instead of one. However, the most significant

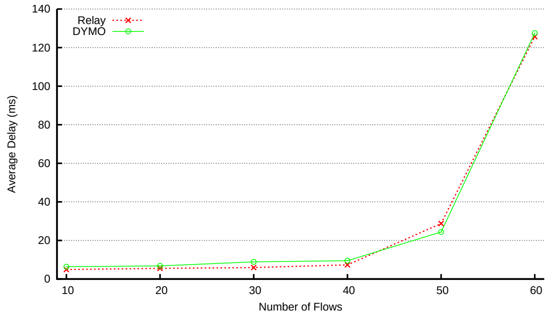

Figure 11. Experiment 3 delay.

remark to be made is that DYMO shows more variability, even for these experiments where congestion is not an issue.

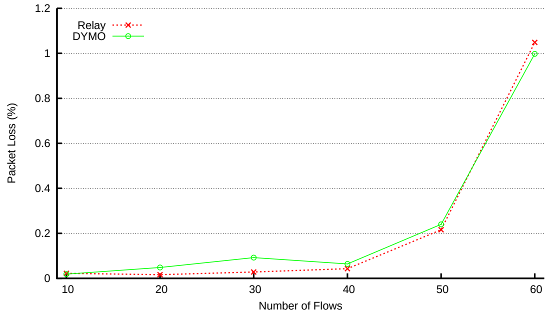

Figure 12. Experiment 3 packet loss.

The values obtained for jitter follow the behavior of that on previous experiments, i.e., it increases almost linearly with the increase in the network load, as shown in Fig. 13.

When compared to the jitter values obtained in Experiment II, there is a reduction of around 50%, which was expected given that the network load is kept the same in average, but split by two sources. Moreover, relaying, which does not have the overhead of control packets, performs better in terms of jitter, being always slightly below its routing counterpart.

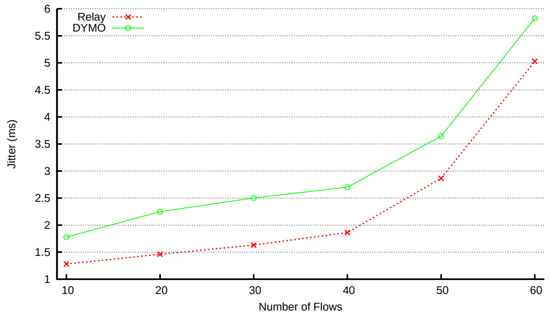

Figure 13. Experiment 3 jitter.

The results of this experiment show that the performances of both DYMO and relaying will equally suffer with the increase in the diversity of source nodes. As we double the number of sources, the performance of the network dropped in all analysed parameters, when compared to Experiment II. In other words, concurrency affects equally relaying and DYMO.

V. CONCLUSIONS AND FUTURE WORK

This paper provides a first analysis of a simple form of relaying vs. a stable form of multihop routing. The motivation for such evaluation was to understand up to which point can relaying be an interesting concept to deal with, in comparison to multihop routing. Several experiments have been run and results have been collected in the form of end-to-end packet delay, average packet loss, as well as jitter.

Albeit being a first step towards the performance of relaying vs. routing, the evaluation performed allow us to conclude that relaying does have interesting features in the tested scenarios, showing less variability than the benchmark used, DYMO.

Being initial work, there is the need to confirm the results achieved by means of scenarios with different parameters, e.g., topologies, traffic matrix, node mobility, energy constraints. These are tasks that will be addressed as future work.

REFERENCES

[1] Ian software server. http://ian.inescporto.pt/software.html.

[2] Traffic analysis (voip) - cisco systems. http://www.cisco.com/en/US/docs/ios/solutions_docs/voip_solutions/TA_ISD.html.

[3] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. pages 85–97, 1998.

[4] L. Carvalho. User-provided Networks: Relaying vs. Ad-hoc Routing, Faculdade de Engenharia da Universidade do Porto, MSc Thesis, July 2009.

[5] I. Chakeres and Belding-Royer E. Aodv implementation design and performance evaluation. *International Journal of Wireless and Mobile Computing*, 2005.

[6] K. Chen, C. Huang, P. Huang, and C. Lei. Quantifying skype user satisfaction. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 399–410, New York, NY, USA, 2006. ACM.

[7] S. Gormus, D. Kaleshi, J. McGeehan, and A. Munro. Relaying and routing in wireless networks: a throughput comparison. In *Vehicular Technology Conference, 2006. IEEE 63rd*, volume 1, pages 440–445.

[8] P. Hofmann, C. An, L. Loyola, and I. Aad. Analysis of udp tcp and voice performance in ieee 802.11b multihop networks. *European Wireless*, 1-4, April 2007.

[9] P. Jacquet, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. pages 62–68, 2001.

[10] D. B. Johnson, D. A. Maltz, and J. Broch. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In *In Ad Hoc Networking, edited by Charles E. Perkins*, pages 139–172, 2001.

[11] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. Panwar. Coopmac: A cooperative mac for wireless lans. *IEEE Journal on Selected Areas in Communications*, 25(2):340 – 354, February 2007.

[12] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. pages 1405–1413, 1997.

[13] C. E. Perkins. Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[14] C. E. Perkins and I. D. Chakeres. Dynamic manet on-demand (dymo) routing, ietf draft (work in progress).

[15] R. Sofia and P. Mendes. User-provided networks: consumer as provider. *Communications Magazine, IEEE*, 46(12):86–91, December 2008.

[16] Hung yu Wei and R.D. Gitlin. Two-hop-relay architecture for next-generation wwan/wlan integration. *Wireless Communications, IEEE*, 11(2):24–30, Apr 2004.