A Decentralized Traffic Management Approach for Ambient Networks Environments

María Ángeles Callejo-Rodríguez, Jorge Andrés-Colás, Gerardo García-de-Blas,
Francisco Javier Ramón-Salguero, and José Enríquez-Gabeiras

Telefónica I+D,
Advanced Networks Planning Department,
Emilio Vargas 6, 28043 Madrid, Spain
{macr327, jorgeac, ggdb, fjrs, jeg}@tid.es

Abstract. This paper presents a decentralized traffic management solution suitable for Ambient Networks environments, where heterogeneous networks will have to cooperate with a high degree of dynamicity, both in traffic patterns and network topologies. Considering IP as the base inter-network technology in these environments, the proposed mechanism autonomously interacts with existing intra-domain routing protocols to improve traffic performance. The proposal has been evaluated by simulation and has been shown how it significantly improves the traffic performance with respect to the solutions currently deployed in networks. For the two simulated scenarios, the proposed solution is able to manage 38% and 15% more traffic than current solutions when the network starts to be congested. Anyway, the behavior of the proposed solution is currently being analyzed in more dynamic scenarios in order to check its goodness for different Ambient Networks environments.

1 Introduction

The Ambient Networks concept [1] aims to provide open and scalable solutions for the near-future networking world where heterogeneous networks, from personal and vehicular networks to access and core transport networks, will have to cooperate to offer ubiquitous communication services to the end-users.

In addition, these scenarios include a wide range of traffic patterns to be carried, with different mobility degrees and performance requirements. Considering IP as the base inter-network technology for Ambient Networks, this paper proposes a decentralized traffic management solution based on the extension of existing static IP intra-domain routing protocols to automatically adapt their routing tables to current traffic dynamics. In this way, traffic management mechanisms autonomously interact with the control plane of the network.

It has to be noted that the work described in this paper focuses on an intra-domain scope. Inter-domain solutions implying routing information exchange among different operators are left for further study.

Regarding intra-domain IP routing algorithms, traditional proposals are based on the dissemination of the network topology in order to allow each router

J. Schönwälder and J. Serrat (Eds.): DSOM 2005, LNCS 3775, pp. 145–156, 2005.

in the network to infer the path with the minimum associated cost. Thus, new routing algorithms are required in order to achieve traffic flows to be forwarded through the available network resources in such a way that no link becomes overloaded and congestion is avoided. For this objective, the usage of routing algorithms based on multipath schemes is required, although their use entails the usage of sub-optimal paths, that is, those paths with costs higher than the optimal ones. This situation can generate routing loops decreasing the efficiency of the routing mechanism. Moreover, routing loops make worse the traffic performance in those scenarios where multipath routing proposals are most interesting: networks with high traffic load, near to or already in a congestion state. Therefore, a thorough study is needed to avoid these loops in the most suitable way. In this paper, a new mechanism for the avoidance of routing loops, called LAP (Loop Avoidance Protocol), is presented. LAP can be used as an extension of any intra-domain IP multipath routing mechanism.

It has to be noticed that solutions based on packet tunneling, such as MPLS-TE (MultiProtocol Label Switching Traffic Engineering) [2] have been discarded beforehand, since they are considered as not flexible enough for the high-dynamic environments envisaged for Ambient Networks. Besides, an IP native solution for traffic management benefits from the scalability and simplicity of IP, which are strongly reduced with the usage of tunneling solutions.

The rest of the paper is structured as follows. In Section 2, a state of the art of routing alternatives for Ambient Networks are presented and the reasoning behind the selection of the MRDV (Multipath Routing with Dynamic Variance) mechanism [3] as the most suitable approach is introduced. Then, a detailed description of LAP is shown in Section 3. Next, Section 4 presents simulation results showing the performance of LAP jointly with MRDV. Finally, Section 5 includes conclusions and further steps.

The research work presented in this paper has been developed within the framework of the Ambient Networks project, partially funded by the European Commission under the Information Society Technology (IST) priority within the Sixth Framework Programme.

2 Routing Alternatives for Ambient Networks

This section surveys existing intra-domain IP multipath routing solutions that can be used in order to optimize network resources in an Ambient Networks environment.

The most deployed multipath routing algorithm in current IP networks is ECMP (Equal-Cost MultiPath) [4], which is inherently supported by common intra-domain routing protocols, such as OSPF (Open Short Path First) [5] and ISIS (Intermediate System to Intermediate System) [6]. In ECMP, all paths with minimal cost are equally used to route traffic. Nevertheless, its scheme does not split the traffic according to a balanced load criterion, as all paths are required to have the minimal cost.

As a more dynamic approach, OMP (Optimized Multi-Path) [7] allows routers to shift load from heavily loaded paths to less loaded ones by means of the use of the global state-network information: new paths can be inferred by other routers in the network since updated and accurate information about the link loads of all the nodes in the network must be exchanged; and thus make OMP not scalable enough in those scenarios where traffic demands are highly variable. Another algorithm, AMP (Adaptive MultiPath) [8] is based on local network-state information for path selection. Thus, each router only distributes information about the load on each link to only its immediate neighbors.

With a similar approach, MRDV [3] does not require the exchange of any load information: each router running MRDV algorithm allows non-optimal paths to be used according to a variance factor reflecting the load on the next hop. Consequently, a MRDV router only has to monitor the load on its own links and can coexist with non-MRDV routers in the network. This approach is interesting for Ambient Networks environments due to both its decentralized scheme and the ability of its gradual introduction in networks allowing a smooth migration towards a full MRDV-enabled network. Next subsection briefly describes the MRDV basis.

2.1 Overview of Multipath Routing with Dynamic Variance (MRDV)

MRDV combines multipath routing with variance and distributed dynamic routing protocols. The core concept of the MRDV algorithm is that the number of alternative paths towards a destination depends on how occupied the links are. Multipath with variance routing algorithms allow traffic to each destination to be carried by other paths in addition to the paths with the minimum cost if the comparison between its metric and a threshold meets the following rule:

$$M \leq M_{min} \cdot V \tag{1}$$

where M is the metric of the path, M_{min} is the metric of the optimal path and V is the variance parameter of the output interface towards the next hop in the optimal path.

MRDV adjusts the variance parameter dynamically, according to the average load that the router detects in the next hop of the optimal path towards the destination. A different variance is defined for each output interface: every router monitors load in its adjacent links and modifies the variance of those interfaces according to their load.

According to the variance, new paths will be considered as suitable: load is distributed among these suitable paths, but the traffic offered to every path is inversely proportional to the path cost, so that the less cost a path has, the more traffic it receives. MRDV distributes traffic properly even when not all the interfaces are overloaded. In this case, only these overloaded links overflow traffic to other interfaces. Therefore, this algorithm is decentralized and IP compatible, and also adds the ability to adapt the variance to the traffic demand automatically.

With this approach, every router reacts to its own view of the network state: the average load of its adjacent links. The forwarding decisions are only based on local information and not on global information, as happens with other routing solutions that modify link costs according to the network status. However, two issues must be considered to prevent instability problems in MRDV. First, the variance must describe a hysteresis cycle, where relative increments in variance are proportional to relative increments in average load. Considering that the minimum variance is 1 (ECMP situation), the expression will be the following:

$$\left.\begin{array}{l} \frac{\partial V}{V} = K\frac{\partial \rho}{\rho} \\ V(\rho = 0) = 1 \\ V(\rho = 1) = V_{max} \end{array}\right\} \Rightarrow V = 1 + (V_{max} - 1) \cdot \rho^{K} \tag{2}$$

where K is any real positive number and a design parameter, and V_{max} is the maximum possible variance.

Therefore, the hysteresis cycle is defined by the values of K for each of the two sections (from now on, K_{up} for the ascending curve V_{up}, and K_{dn} for the descending curve V_{dn}) and a common parameter V_{max} for the maximum variance. These parameters define the behavior of the algorithm. For simplicity, $K_{up} = 1/K_{dn}$ is proposed.

The other key issue regarding MRDV stability is the choice of the frequency to refresh the variance parameter as a trade-off between response time and accuracy in measures. Based on our experience with MRDV simulations, the update interval should never be less than about ten seconds, since a shorter update interval could lead to a too unstable behavior in the presence of bursty traffic.

MRDV has been implemented in Network Simulator 2 (ns-2) [9] and evaluated in different scenarios. Detailed results can be seen in [3], where MRDV is compared with OSPF without and with ECMP. In a realistic scenario with a typical backbone topology composed of 12 nodes and traffic with different burstiness degrees, the network is able to carry around 35% more traffic with MRDV than OSPF without ECMP, and around 15% more than OSPF with ECMP. In spite of these promising results, routing loops were affecting negatively to the traffic performance in these simulations. Thus, a mechanism to avoid them was considered as a key requirement for a satisfactory traffic management solution.

3 Description of Loop Avoidance Protocol (LAP)

In order to develop an algorithm for avoiding loops, a distinction between primary and secondary loops has been made. Primary loops, as Fig.1.a shows, appear when a node A tries to introduce a new sub-optimal path to reach D through B, which has A as the next hop of the optimal path to D.

Secondary loops are shown in Fig.1.b and Fig.1.c. In the first one (primary path sees a secondary one), there are both an optimal path (from B to A, however B has not A as its next hop in the optimal path to D) and also a secondary one (from A to B) to reach the same destination. In Fig.1.c (secondary path sees a secondary one), a loop is caused by two secondary paths, each one with its own percentage of routed traffic, α and β.

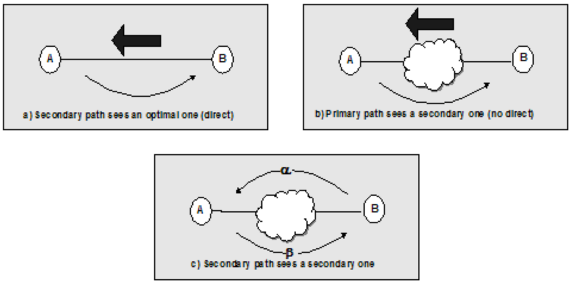
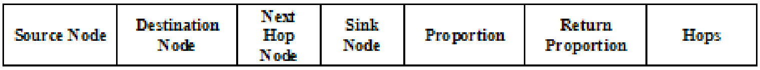


Fig. 1. Types of loops



Source Node	Destination Node	Next Hop Node	Sink Node	Proportion	Return Proportion	Hops

Fig. 2. Structure of the LAPM. LAPM is composed of the next fields: *SourceNode* (id. of the node wanting to establish the secondary path), *DestinationNode* (id. of the destination node), *NextHopNode* (id. of the next hop of the secondary path *SourceNode* wants to establish to reach *Destination*), *SinkNode* (id. of the node starting the return phase), *Proportion* (direct proportion of the traffic sent by *SourceNode* to *Destination* through *NextHop* that reaches *SinkNode*), *ReturnProportion* (proportion of the traffic sent by *SinkNode* to *Destination* that reaches *SourceNode*) and *Hops* (number of hops that can be still leaped)

Taking into account this classification, two different mechanisms are proposed when a node is going to install a new secondary path: avoidance of primary loops and avoidance of secondary loops, described in Sections 3.1 and 3.2, respectively.

3.1 Avoidance of Primary Loops

Avoiding primary loops only requires a simple process to be computed at each router: when a router X is going to install a new sub-optimal path, if the candidate to new Next Hop (NH) to reach a destination has the router X as the next hop of its optimal path to reach the same destination, this new secondary path is discarded. Since X knows both the topology and the link-state information of the network, it is able to infer the optimal paths of NH by means of applying a Dijkstra algorithm [10] and no additional information exchange is required.

3.2 Avoidance of Secondary Loops

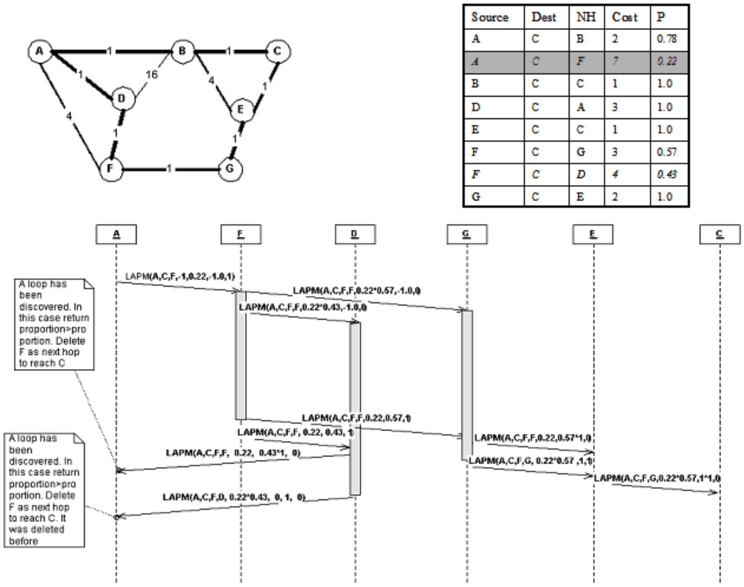
An information exchange is required in order to know whether a secondary loop will exist if the secondary path is installed and, if so, avoid it. We define the LAPM (Loop Avoidance Protocol Message) as the normalized message required for this information exchange, whose structure is shown in Fig.2.

This mechanism defines three main phases: a *forward phase* (calculation of the percentage of traffic routed by the forward path), a *return phase* (calculation of the percentage of traffic routed by the reverse path) and a *discovery* *phase* (triggered if a loop is discovered, where the secondary path is deleted if *Proportion* is lower than *ReturnProportion*).

The *forward phase* is triggered by a node (N), trying to establish a new secondary path to a destination (D) by routing a traffic percentage (p) through the next hop (NH). This node sends a new LAPM to NH, initialized with the set of values (N, D, NH, -1, p, -1.0, $MaxHops$) according to the LAPM format defined in Fig.2. $MaxHops$ is a configurable parameter that defines the depth of the algorithm and represents a trade-off between loops avoidance and extra load in the network.

Once N sends the LAPM to NH, it is processed according to a defined set of actions to be triggered when a LAPM arrives.

When a node receives a LAPM, it firstly checks if the received message is a forwarding LAPM (*ReturnProportion* is equal to -1). In this case, if *Hops* is greater than zero, the node must resend the message to its next hops to reach



Source	Dest	NH	Cost	P
A	C	B	2	0.78
A	*C*	*F*	*7*	*0.22*
B	C	C	1	1.0
D	C	A	3	1.0
E	C	C	1	1.0
F	C	G	3	0.57
F	*C*	*D*	*4*	*0.43*
G	C	E	2	1.0

A loop has been discovered. In this case return proportion>proportion. Delete F as next hop to reach C

A loop has been discovered. In this case return proportion>proportion. Delete F as next hop to reach C. It was deleted before

Fig. 3. Example of the avoidance of secondary loops. The figure shows a topology (top-left), the paths to reach C from all the nodes in the network (top-right) and a sequence diagram with all the messages exchanged by the nodes in the network when A wants to establish a new secondary path to reach C (bottom). In this example, we can distinguish the forward phase (A sends a new LAPM to F, F resends this message to its next hops with updated values of *Proportion* and so on), the return phase (e.g. when the timer expires, F sends to all its next hops to reach C a new LAPM with the initialized value of *ReturnProportion*) and final the discovery phase (A receives a LAPM with *Source* equals to A, and compares the values of the *Proportion* fields and deletes from its routing table F as a possible next hop to reach C).

D with updated values of $Proportion$, taking into account the percentage of traffic that the node routes through each one, p_i; therefore, for each next hop to reach D, the node has to resend the received LAPM with updated values for $Proportion$ ($Proportion * p_i$) and $Hops$ ($Hops - 1$).

In addition to the sending the updated LAPM to its next hops, the node starts the *return phase*. In order to aggregate the forwarding proportions belonging to the same routing tree (whose key is defined by $SourceNode$, $Destination$ and $NextHop$), each node must maintain a list with the sum of the $Proportion$ fields received in different LAPMs for the same routing tree. Consequently, when the return phase starts, if there is another registry in the list for that routing tree, the value of its proportion is updated (the received $Proportion$ is added to the stored value). If not, a new registry is added to the list with the values included in the received LAPM, and a timer is triggered for that registry. When this timer expires, the node sends a new LAPM to each next hop of its routing table to reach D. The node initializes a new LAPM with the values stored in the list for $SourceNode$, $Destination$, $NextHop$ and $Proportion$ and for $ReturnProportion$, $SinkNode$ and $Hops$ it uses p_i (traffic proportion routed to reach D from the specific next hop), the identifier of the node and the configurable parameter $MaxHops$, respectively.

On the other hand, when a returning LAPM is received ($ReturnProportion$ different from -1), the node checks if the value of $SourceNode$ is equal to its own node identifier. If this condition is met, a secondary loop has been discovered and the discovery phase starts. Otherwise, if $Hops$ is greater than zero, the return phase continues and the LAPM is resent to all the next hops to reach D by updating the values of $ReturnProportion$ ($ReturnProportion * p_i$, where p_i is the proportion of traffic sent by this hop) and $Hops$ ($Hops - 1$) fields.

Similarly to the return phase policy, in the *discovery phase* each node must also maintain a list to manage the received return LAPMs containing information about $Destination$, initial $NextHopNode$, $SinkNode$ (that one that initialized the return phase), $Proportion$, $ReturnProportion$ and a timer to check if the path must be deleted. Therefore, when a loop is discovered, the node firstly checks if $ReturnProportion$ of the received LAPM is greater than the $Proportion$ contained in the same message. In this case, the node deletes from its routing table the $NextHop$ to reach $Destination$. If this is not the case, and there is another registry in the list with the same values of $Destination$, $NextHop$, $Proportion$ and $SinkNode$, the value of $ReturnProportion$ is updated by means of adding the just-received $ReturnProportion$. If not, the node introduces a new registry in the list with the values received in the LAPM and the initial value of the timer (also proportional to the $MaxHops$ configuration parameter). When the timer expires the node checks if fixed $Proportion$ is equal or lower than the store, and maybe updated, $ReturnProportion$. In this case, the secondary path to reach $Destination$ through $NextHop$ is deleted.

Fig.3 shows how the phases defined above converge and allow a router wanting to establish a new secondary path to avoid loops.

4 Evaluation of the Proposal by Simulation

The proposed solution for intra-domain traffic management in Ambient Networks, (MRDV+LAP), has been implemented in Network Simulator 2 (ns-2) [9] in order to evaluate by simulation the efficiency of the proposal. Two different topologies have been used in these simulations: a basic low-meshed topology with seven nodes and a more realistic and meshed topology with twelve nodes.

The traffic pattern used to feed these topologies is composed of both TCP and UDP traffic. For each pair of nodes in the network, each node contains at least one FTP application (and sends/receives TCP traffic to/from the other nodes in the network) and both constant and exponential bit rate applications are established. All the nodes send traffic rate to all the other nodes in the network, which is multiplied by a scale factor to increase the traffic load level, as it can be seen in further graphs.

In order to evaluate the performance perceived by this traffic, the evolution of the loss ratio and the mean delay for the UDP traffic and the average throughput obtained for the FTP applications have been analyzed. Moreover, the evolution of the loop probability has been monitored in order to analyze the efficiency of LAP.

For each simulated traffic level, ten simulations with different seeds have been performed, in order to estimate the error associated to a given statistical confidence interval. The further graphs show the mean value, while the highest values of the errors for a confidence interval of 90% are given in the figure captions for reference.

The following subsections present these scenarios and analyze the obtained results.

4.1 Basic Scenario

Firstly, we have evaluated the performance of the protocol in a basic scenario, whose topology is shown in Fig.4. The traffic matrix of this scenario is defined according to the principles explained above; however, in order to simulate a typical interconnection point to external networks, traffic rates of UDP flows sent or received by node 1 are increased a 30%. Regarding FTP applications, the file sizes sent by the sources depend on the traffic level of each mechanism.

This scenario has been simulated with different routing options: OSPF with ECMP (ECMP), ECMP with MRDV (MRDV), MRDV avoiding primary loops,

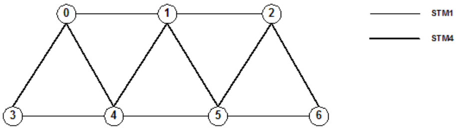


Fig. 4. Topology used in the basic scenario. Link delays set to 5 ms, link capacities to STM1 (155 Mbps) and STM4 (622 Mbps).

and MRDV with full LAP up to 1-hop loops. It has been monitored the UDP loss ratio, the TCP throughput and the loop probability and looped traffic obtained in each option. Fig.5 shows the mean values of these parameters.

Due to the usage of basic MRDV with respect to ECMP, it can be seen that UDP losses are postponed with MRDV and the congestion point for TCP traffic also appears later. For example, if we compare the traffic level when the loss ratio in both options overpasses a threshold of 1%, basic MRDV can manage around 38% more traffic than ECMP. Moreover, for a traffic level resulting in a loss ratio for UDP traffic of 1% in the case of MRDV, ECMP obtains 2.5% of losses. For this same traffic level, the average number of bytes received by a FTP application is 19.5% higher in the case of MRDV with respect to ECMP. As these results show, the use of MRDV can significantly improve the performance in the network for both UDP and TCP traffics.

The simulation results also show how LAP clearly improves the MRDV performance with respect to the basic MRDV option: for the same traffic level resulting in a loss ratio of 1% in MRDV, the avoidance of primary loops reduces to 0.19% the loss ratio and with the use of LAP with one hop, 0.18%. This small difference is due to the low possibility of routing loops with more hops since the topology is low-meshed. Looking at the obtained the loop probability in each case, it is reduced from 4.6% to 0.9%.

If we compare the traffic level when the loss ratio exceeds a 1%, LAP allows to duplicate the traffic level carried by the network. Finally, regarding the performance of TCP traffic, the congestion point, located where the number of received packets decreases for higher traffic levels, appears with a traffic increment of around 33% in the case of MRDV+LAP with respect to the congestion point of ECMP.

As expected, loop probability has been decreased and is even maintained when the traffic level is increased, as Fig.5 shows. This is because, although new routing loops appear, LAP discovers them and removes them from the routing tables. The remaining loops are those with more hops, which LAP is not considering in exchange of introducing less traffic overhead in the network.

4.2 Realistic Scenario

We have also evaluated the performance of MRDV with LAP in a more meshed topology (shown in Fig.6) based on the core network of the reference transport network presented in the IST-LION project [11].

In order to make TCP traffic more realistic, five Edge Nodes (EN) have been attached to a Core Node (CN) with links of 1ms delay and 10 Mbps capacity. These ENs run FTP applications and establish TCP connections with other ENs in such a way that fifteen FTP transactions running during the whole simulation time are established between each pair of CN nodes, that is, three per each EN associated. Regarding the UDP traffic, a rate proportional to the population of both source and sink cities has been set between the CNs.

Fig.7 presents the obtained simulation results, where MRDV without any mechanism for avoiding loops performs worse than ECMP due to the high ap-

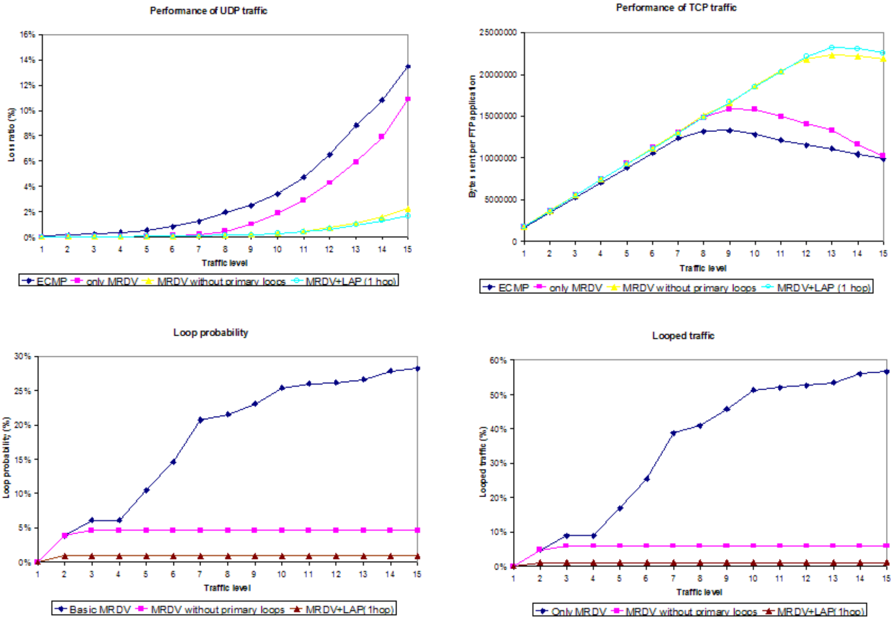


Fig. 5. Results obtained from the basic scenario. Top-left: Loss ratio of UDP traffic (max. error: 2.5%). Top-right: throughput of TCP traffic (max. error: 1.3%). Bottom-left: loop probability. Bottom-right: looped traffic.

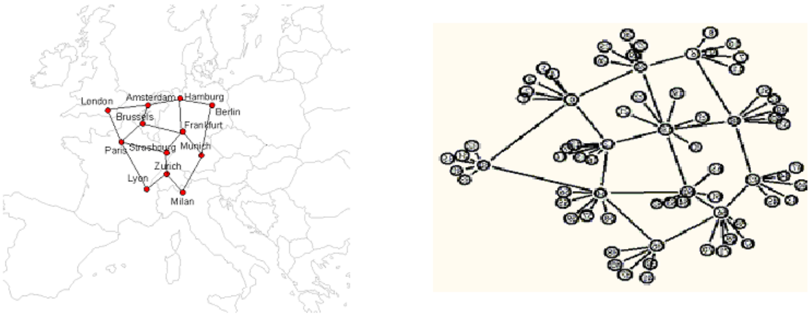


Fig. 6. Topology for the realistic scenario: backbone topology used (left) and topology of the simulated scenario, including TCP traffic sources and sinks (right)

pearance of loops, even with low traffic levels. As it can be seen, loop probability rapidly increases and goes over 50%. Therefore, this scenario justifies the use of mechanisms for avoiding loops.

Nevertheless, the results also show how the use of LAP can significantly improve the performance of MRDV. Specifically, the avoidance of only primary loops, for a traffic level value that causes a loss ratio of 5.8% for UDP traffic, MRDV+LAP obtains a loss ratio of 2.9%, reducing UDP losses around 50%. If

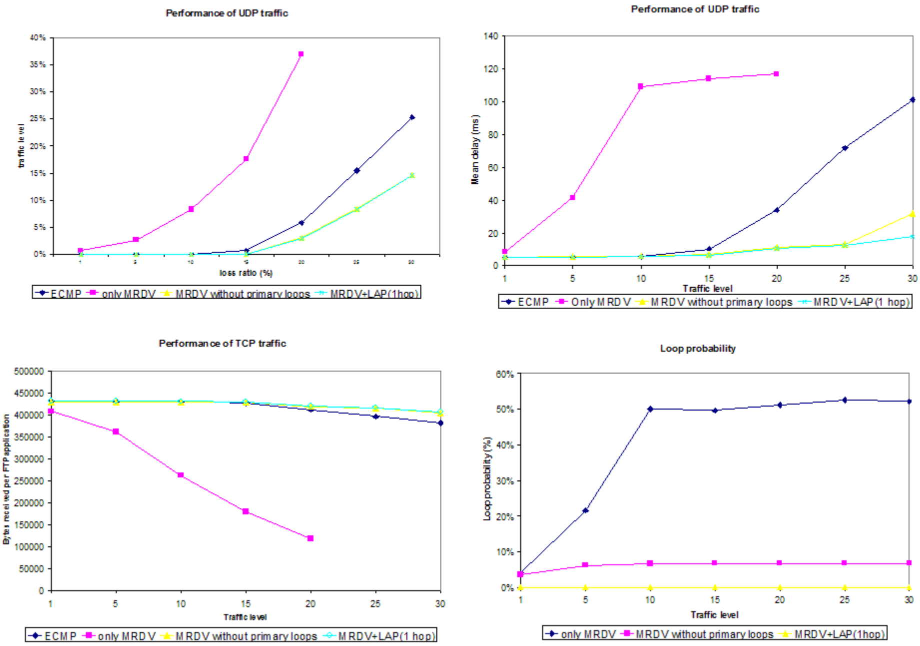


Fig. 7. Results obtained from the realistic scenario. Top-left: Loss ratio of UDP traffic (max. error: 0.4%). Top-right: Mean delay of UDP traffic (max. error: 0.6% and 9% for MRDV and ECMP respectively). Bottom-left: Throughput of TCP traffic (max. error: 0.2%). Bottom-right: loop probability.

we compare the traffic level when the loss ratio in both options overpass a 5%, MRDV with LAP can manage around 15% more traffic than ECMP. However, if we evaluate the TCP performance for the same traffic level (20), the throughput only increases by 2% due to the use of MRDV+LAP instead of ECMP. It is needed a higher traffic level to obtain significant benefits; e.g. for a traffic level of 30, this benefit is around 8%.

The results obtained with one-hop LAP are not significantly better than those ones with just avoiding primary loops because most of the loops are primary ones. In fact, the avoidance of primary loops reduces loop probability to 6%. Anyway, one-hop LAP removes all the existing loops.

5 Conclusions and Further Steps

This paper presents the combination of MRDV and LAP as a satisfactory solution for the traffic management in Ambient Networks. The proposal presents a decentralized friendly-migrable solution to distribute traffic load in the network thanks to the use of MRDV, whose performance is improved by LAP by successfully removing secondary paths that can cause loop appearance. Specifically, the use of LAP is interesting in those scenarios where loop probability is very high, as it has been shown in the realistic scenario.

Moreover, at least in the studied scenarios, only with the avoidance of primary loops, traffic performance is significantly improved. So, the second phase of LAP would not be necessary since the improvement in the traffic performance is negligible. Nevertheless, more simulations with different scenarios are needed to have more confidence in this conclusion.

Another issue that must be analyzed in further steps is the behavior of MRDV+LAP in case of variations during the simulation time in both the traffic matrix (sudden changes in traffic patterns) and the network topology (link and node failures). Also, we have to evaluate how configuration of timers used in the return and discovery phases of LAP can affect the performance of the protocol.

References

[1] Niebert, N., Schieder, A., Abramovicz, H., Malmgren, G., Sachs, J., Horn, U., Prehofer, C., Karl, H.: Ambient Networks: An Architecture for Communication Networks Beyond 3G, IEEE Wireless Communications, Abril 2004.

[2] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., McManus, J.: Requirements for Traffic Engineering Over MPLS, Request for Comments 2702, Internet Engineering Task Force (1999).

[3] Ramón-Salguero, F.J., Enríquez-Gabeiras, J., Andrés-Colás, J., Molíns-Jiménez, A.: Multipath Routing with Dynamic Variance, COST 279 Technical Report TD02043 (2002).

[4] Hopps, C.: Analysis of an equal-cost multi-path algorithm, Request for Comments 2992, Internet Engineering Task Force (2000).

[5] Moy, J.: OSPF Version 2, STD 54, Request for Comments 2328, Internet Engineering Task Force (1998).

[6] Callon, R.: Use of OSI IS-IS for Routing in TCP/IP and Dual Environments, Request for Comments 1195, Internet Engineering Task Force (1990).

[7] Villamizar, C.: OSPF optimized multipath (OSPF-OMP), Internet Draft, draft-ietf-ospf-omp-03 (1999).

[8] Gojmerac, I., Ziegler, T., Ricciato, F., Reichl, P.: Adaptive Multipath Routing for Dynamic Traffic Engineering, IEEE Globecom (2003).

[9] The Network Simulator - ns-2, http://ww.isi.edu/nsnam/ns.

[10] Cormen, T.H., Leiserson C.E., Rivest R.L.: Introduction to Algorithms, MIT Press (1990).

[11] Reference Transport Network Scenarios, IST COST/COST 266, http://www.ibcn. intec.rug.ac.be/projects/IST/NRS.