FastM: Design and Evaluation of a Fast Mobility Mechanism for Wireless Mesh Networks

Luís Couto, João Paulo Barraca, Susana Sargento, Rui L. Aguiar
Universidade de Aveiro, Instituto de Telecomunicações, Aveiro, Portugal
{lcouto, jpbarraca} @av.it.pt, {susana, ruilaa} @ua.pt

Abstract

Although there is a large volume of work in the literature in terms of mobility approaches for Wireless Mesh Networks, usually these approaches introduce high latency in the handover process and do not support real-time services and applications. Moreover, mobility is de-coupled from routing, which leads to inefficiency to both mobility and routing approaches with respect to mobility.
In this paper we present a new extension to proactive routing protocols using a fast mobility extension, FastM, with the purpose of increasing handover performance in Wireless Mesh Networks. With this new extension, a new concept is created to integrate information between neighbor wireless mesh routers, managing locations of clients associated to wireless mesh routers in a certain neighborhood, and avoiding packet loss during handover. The proposed mobility approach is able to optimize the handover process without imposing any modifications to the current IEE 802.11 MAC protocol and use unmodified clients. Results show the improved efficiency of the proposed scheme: metrics such as disconnection time, throughput, packet loss and control overhead are largely improved when compared to previous approaches. Moreover, these conclusions apply to mobility scenarios, although mobility decreases the performance of the handover approach, as expected.

Keywords: Fast Mobility, mesh networks, MeshDV, neighboring tables, handover signaling.

1. Introduction

Wireless Mesh Networks (WMNs) are dynamically self-organized and self-configured networks, where terminals are connected through routers in a mesh topology. WMNs increase the capabilities of ad-hoc networks, such as robustness, power management, reliable service coverage and optimized node mobility (Figure 1). Coverage increases automatically, allowing a continuous addition of terminals and a self-adapting topology. In the mesh infrastructure there are two types of devices: the Wireless Mesh Routers (WMRs) and the Wireless Mesh Terminals (WMTs). WMRs are devices able to provide multi-hop transport mechanisms enabling communication between the terminals in the same or in different WMNs. A terminal can be any type of device with a wireless interface (typically 802.11a/b/g), whether mobile or stationary. In the particular case of the WIP project [2], where this work was performed, terminals will mostly be comprised of laptops, desktop computers or PDAs, all supporting 802.11a/b/g.

In WMNs, terminal mobility occurs whenever a client associated to an access point (or WMR directly) wants to change its point of attachment. To maintain communication with other terminals, it needs to constantly inform active correspondent nodes about its current location. Any mobility solution designed to these networks must be able to quickly update terminals location information with low overhead yet effectively, creating a reliable, non-interrupted communication between nodes. Cellular technologies, such as the ones used in current GSM and UMTS networks, are able to support seamless connectivity between neighbor points of attachment. WMNs, typically using 802.11, are unable to meet the requirements for voice continuity without further solutions.

In this paper, which is an extended version of the work presented in [1], we propose a new mobility mechanism for WMNs denoted as FastM, Fast Mobility support extension for WMNs, an evolution of MeshDV [3] and Enhanced Mobility Management (EMM) [4], inheriting the basic functional aspects, but using neighboring tables and improved handover signaling to avoid packet loss during the handover process, reduce control multicast packets in the network, save bandwidth and optimize the association and disassociation processes of clients to WMRs. The result is a much optimized and effective solution, able to provide voice continuity over WMNs. The obtained results through simulation show a large increase in performance, for both UDP and TCP communications, in terms of disconnection time, throughput, delays and packet losses, not compromising network overhead and network efficiency.

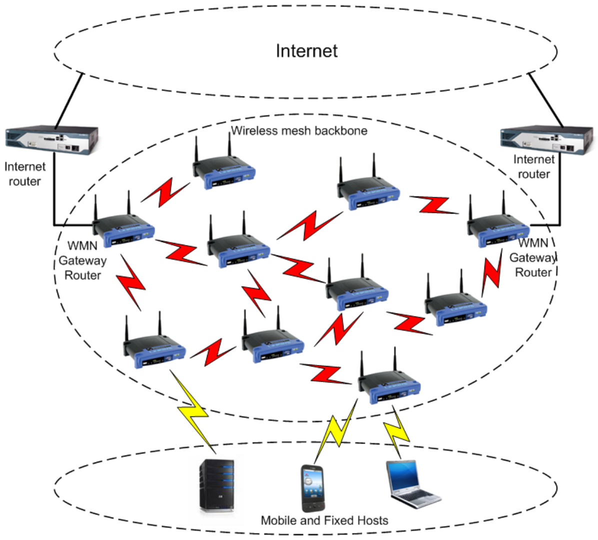


Figure 1. Diagram of a Wireless Mesh Network

The paper is organized as follows. Section 2 presents some of the most relevant mobility mechanisms in WMNs in the literature and their main problems in meeting the requirements of real-time communications. Section 3 introduces the basic routing and mobility mechanism that will be the basis for the protocol enhancements. Section 4 describes the enhanced mobility mechanism, FastM, our proposal for improved mobility in WMNs. Then, Section 5 depicts the simulation scenario and the obtained results, through different scenarios, without and with mobility of nodes. Finally, Section 6 concludes this paper and Section 7 discusses how FastM can be applied to large scale scenarios and describes proposals for future work.

2. Related work

There are already many mobility mechanisms for WMNs in the literature. In this section we describe some of the most relevant mechanisms, stating their benefits and drawbacks.

Ant [5] is a network-based local mobility management scheme for WMNs. Ant introduces some techniques to optimize handovers, such as: a) using the MAC-layer association event as signalling messages, b) maintain IP address of terminals unchanged while moving inside the WMN, and c) pre-establishing tunnels between neighboring WMRs, supporting a list of WMRs neighbors created in each WMR. However, Ant presents some problems: a) the IP address of terminals does not reflect the topology, b) pre-tunnels must be available between every WMR neighbors, which introduces a scaling problem, and c) there is a centralized location server, managing all the location information of the network. In a small scenario with only 4 nodes, handover timing results in the order of 44.5 milliseconds are obtained [5].

MAMP (Mobility-Aware Multi-Path) [6] is a new scheme that uses the interconnection between Serving Access Points (SAPs) and is supported on the existence of a Gateway. It is a multi-path mechanism for packet forwarding, creating a large number of connections between every node, with multiple alternative routes. In this scheme, when a mobile node registers in the network, a message will be forwarded from the correspondent SAP to the gateway, creating routing paths in every SAP that receives the message to the mobile node. Meanwhile, each SAP broadcasts to its neighbors the appearance of a new mobile terminal, and, recursively multi-path routes are created. This solution presents good performance, reducing the handover delay comparing to other techniques, but needs SAPs to have large capacity to deal in a large number of routes. Being this a proactive mobility protocol, it also gives the mobile host the responsibility to trigger the mobility process in the network.

MobiMESH [7] is a WMN mechanism where the network is organized in two sections (backhaul and access), each with a separate IP addressing space. MobiMESH uses a cross-layer mechanism associating MAC and IP layers, making possible to correctly announce associated clients on the backbone routing in a lightweight and fast manner. Results show that, in average, handover using MobiMESH takes 100 milliseconds. However, the association of MAC and IP layers may cause address conflict, and a complete conflict-free strategy may require a central location server or complex interaction between mesh routers.

SMesh [8] uses unmodified WiFi interfaces on terminals. Connectivity and transport is provided by a group of access points, creating the WiFi backbone. Results achieved with SMesh present good performance, with a handover latency time approximately equal to zero (ignoring hardware latency). The main feature contributing to these results in SMesh is the fact that during a handover, traffic to the mobile node is sent by the access points using multicast. However, multicast will consume additional bandwidth. Moreover, in 802.11, multicast data rates are lower than unicast.

Ad-hoc On demand Distance Vector (AODV) Pre-handoff Route Discovery (AODV-PRD) [9] focuses on using concepts of ad-hoc networks, such as the routing protocol (AODV [10]), and optimizes them in a Wireless Mesh Network scenario. The main idea behind AODV-PRD is that a mobile node has always knowledge of the location of an alternative correspondent node, like a backup. The aim is to have a solution with low latency network-layer handovers and with low overhead, reducing routing discovery process when movement occurs. When a mobile node detects that the SNR value of the current correspondent node falls below a given threshold, it initiates a link-layer scan to detect neighbor wireless mesh routers. From this list, the mobile node selects the wireless mesh router which was detected with the highest SNR value, as its new correspondent node in the handover process. At this time, AODV-PRD is integrated with the signaling scheme of Fast Mobile IPv6 (FMIP) [11] by extending the *Fast Binding Update* and *Handover Initiate* signalling messages with a pre-handoff route discovery request option. This mechanism follows the intention of having mobility mechanism free of changes in the hardware of wireless mesh nodes; however, it also suffers the problem of the adaptation to a wireless mesh network scenario.

QMesh [12] uses a different concept from the one that gave birth to wireless mesh networks. This solution assumes the existence of several gateways, and QMesh uses a common practice to always assign a user to the nearest gateway. When a user moves and associates with a new correspondent node that is closer to a different gateway than its current one, it automatically performs a handover between gateways. This solution is purely location-based, application-transparent, and does not incur a high performance impact, promoting an efficient gateway assignment. One of the main points that QMesh secures is the Quality of Service (QoS) in the mobility management solution. Parameters that incur QoS degradation and additional costs, e.g., network distances and congestion, server (gateway) loads, an optimized gateway assignment algorithm must be taken into account. QMesh has two types of different decisions to manage, one regarding QoS and other regarding mobility management in the wireless mesh network. Mobile nodes can then choose between: migrate between two gateways, and choose a gateway upon a transition. QMesh performs a trade-off between assigning the user to the gateway that provides the best QoS at any given time, and reducing the number of costly gateway handovers. However, it has to be evaluated the costs involved in the monitoring of the QoS parameters, congestion and loads, and if they compensate not to assign a user at its nearest gateway. To manage this solution, some tables need to be created in every router. QMesh maintains two main data structures in each mesh router: a LUC (Local User Cache), which holds the MAC addresses of the mobile nodes whose 802.11 beacons are received by the mesh router, and a GUC (Global User Cache), which holds the mapping of the MAC addresses of the mobile nodes that this node is aware, through the APs that it can be reached at. There is a balanced decision between what is best for the handover process and what is best in terms of QoS.

Geo-mobility [13][14] is an approach that adopts a geographical solution for mobility and location management in spontaneous wireless mesh networks. Like most of the solutions based on locations points, this one needs that nodes know exact geographical positions, by means of GPS or similar, while other nodes can estimate their relative positions. With all this data, it is possible to create a global topologically consistent addressing space. The resulting addressing space is pseudo-geographical, in the sense that the coordinate space is virtual and relative, but anchored in the real world through the exact geographical positions of some routers. Then, an End-Point-Identifier (EID) is used in every mobile node. The EID is a value adopted by the mobile node that remains the same while the mobile node moves around in the wireless mesh network. At a given location, the mobile node uses the address of a nearby router to become reachable from any other location in the WMN. As other solutions, Geo-mobility needs to have some type of location service, where the correspondence of each mobile node between its EID and its current address is made. The location service, being distributed and scalable, is composed by several hash functions giving the robustness needed. Virtual Home Region [15] is a concept adapted to the location service, with the necessary modifications in order to fully adapt to a wireless mesh scenario. The main point that turns this solution adapted to several situations, is the flexibility that is offered based on the movement of a mobile node. If a node moves in short distances, the addresses of nearby routers are topologically close to each other (geographical addressing), and updating addresses can be done in a lazy way (low signaling overhead). The handover performance depends on the update rate of the location service (which intrinsically is involved with the addressing update), and it is assumed that a mobile node moves between closer mesh routers. Re-routing mechanism is performed during the handover through the shortest path. The drawback of this solution is that the non-standard-IP addressing mechanism used can introduce more overhead in the global-state-routing update.

Some other solutions, like Wireless mesh Mobility Management (WMM) [16] try to combine optimization for mobility management and for routing mechanism. Using IEEE 802.11 as the link-layer handover procedures, WMM creates in every wireless mesh router a sort of location service while routing data packets. The location cache brings efficiency to routing packets with mobility of nodes. Every mobile node has a correspondent router; in WMM it is called Serving Mesh Access-Point (SMAP). It is the SMAP that manages all the location information of the mobile nodes assigned and in its radius. When mobility occurs, the SMAP is updated by the location management, combined with a re-forwarding technique of data packets, using the old and new SMAP in the data flow. Two different cache tables are used to support these procedures, one for routing and other for location. In WMM they are called the routing and proxy table. The first manages the routing paths in the wireless mesh network; the second manages the location information of mobile nodes. Using both tables makes WMM more robust and functional in a dynamically network. Every router has detailed information about the mobile nodes present in the network. One disadvantage of this solution is the overhead and signaling that exists inside the WMN. In large networks, the mesh routes are very solicited, needing to have high performance standards to respond to all solicitations and route correctly in the network.

Finally, although IEEE 802.11r [17] is able to provide fast handover between stations, it is not able to handle the L3 recovery of the network, in terms of routing paths and neighbor information, in order to reduce the latency on the update of the new path in the overall network. Our proposed mechanism will be able to work both with unmodified IEEE 802.11 and IEEE 802.11r, since our approach is performed at the IP layer.

3. MeshDV and EMM

In this section we present two mobility mechanisms in larger detail, MeshDV [3] and Enhanced Mobility Management (EMM) [4], as they will be the basis of our proposal, FastM.

A. MeshDV

MeshDV is a solution proposed for WMNs based on equipments composed by two wireless interfaces, each dedicated to a different sub-network: one offering connectivity to end-user terminals; the other forming a self-organized wireless backbone. The *client* interface is configured as an access point, while the interface used to maintain the wireless backbone, the *mesh* interface, is configured in ad-hoc mode. These two sub-networks will have different routing and addressing mechanisms operating on them. Highly adaptable routing solutions are required in the transport sub-network enabling WMR to route traffic from and to terminals. For this task, it was proposed a routing solution based on the Destination-Sequenced Distance Vector (DSDV) routing protocol [18] running in IPv6. Clients only need to maintain information about their current point of attachment to the network. Traffic is sent towards each correspondent WMR and no modification to the routing protocol is required at the terminals.

Each WMR has a Local and a Foreign Client tables (LC and FC) that keep track of clients present in the network: the LC table contains the list of clients directly assigned to the WMR; the FC table contains the information about clients and their correspondent WMR, which is required in order to allow communication between these nodes and the local ones. MeshDV uses a tunnel-based approach creating a communication channel between end terminals. Terminals only need to know the IP address of the destination client and query the current WMR (using Address Resolution Protocol - ARP - or IPv6 Neighbor Discovery mechanisms). When the client queries the WMR for the location of a given node, the WMR will search its LC table. If the node is not local, it then queries other WMRs in the network and adds this information to the FC table. The client only needs to send packets to its correspondent WMR. The WMR will then create a tunnel for the communication with the correspondent WMR of the destination client.

The module making all this process transparent makes use of the Neighbor Discovery Protocol (NDP) [19], which is ubiquitous in all systems. This way, clients do not need any additional mechanism to communicate, making possible the integration of off-the-shelf equipment without modifications. Traditionally, nodes use NDP to maintain track of the local neighbors and check their local reachability. NDP uses a set of packets and caches to share and maintain information related to nodes in a network. Using MeshDV, the protocol will alter the operation of NDP (at the WMR), allowing impersonation of the remote terminals.

MeshDV introduces several additional messages in order to manage communication, association, and disassociation events. These messages only exist in the backhaul part of the network and are mostly related to the discovery and advertisement of clients:

- MCREQ – Multicast Client REQuest – This type of message is sent by a WMR when the location of a client in the wireless mesh network is unknown.
- UCREQ – Unicast Client REQuest – This is a periodic message that is generated by the WMR to check if the information present in the FC table regarding a particular client is still valid. The purpose is to confirm if reachability still exists.
- CRREP – Client Request REPly – When a WMR receives a MCREQ or a UCREQ and if the client is connected (present and active in the LC table), the WMR answers with this type of message with the requested information.
- CWIT – Client WIThdraw – When a client disassociates from a WMR, this message is sent to all the WMRs that requested information about this node, effectively

notifying them of the withdraw event. Upon receiving this message, a WMR will update his FC table.

- CWREP – Client Withdraw REPly – When a WMR receives a CWIT message and has an entry with the client address in the FC table, it sends a CWREP packet to the WMR that issued the CWIT announcing that the entry was deleted from the table.

In MeshDV, mobility management is based on feedback from the wireless card (MAC layer) and periodic messages (IP Layer). The problem with this approach is that it is affected by the beacon timeout configuration of the wireless driver. When timeouts are considerably long, it is possible that (incorrect) information regarding some node is kept in a WMR for a long time, resulting in connectivity problems. WMRs, as defined by MeshDV, are responsible for all tasks of the handover process, communicating with the other WMRs in order to update caches and maintain information coherent. Standard versions of MeshDV use an approach of self-detection (a predictive approach) where a mobility manager module is responsible for managing the handover process. While being a valid approach, it has poor performance in the real world. This is more noticeable with active communications because, while the association of a client with a new WMR is a fast process, packets will still be delivered to the old location for some time. The result is high packet loss during the handover period until caches expire (a few seconds). This process must be performed in a completely transparent manner to the terminals and consuming the minimum bandwidth. Also, handover must be a fast process with minimal packet loss, giving terminals the possibility of maintaining active communications across different attachment points.

B. EMM

Enhanced Mobility Management (EMM) [4] is a new optimization to MeshDV, designed to improve mobility management, and reduce handover delay. With this solution, a new reactive approach for mobility management is proposed, with the detection of the clients during their movement performed by the new WMR. This solves the refresh delay problem created by the NDP cache [19]. Results show that EMM [4] reduces the handover latency time in MeshDV Network, in some cases from 3 minutes to only a few seconds or less than one second. The main change that EMM adds to MeshDV is a new type of message that is sent when a client changes its WMR association. EMM also proposes modifications to some of the original messages and mechanisms first proposed in MeshDV. Figure 2 depicts de EMM message sequence diagram of a handover process.

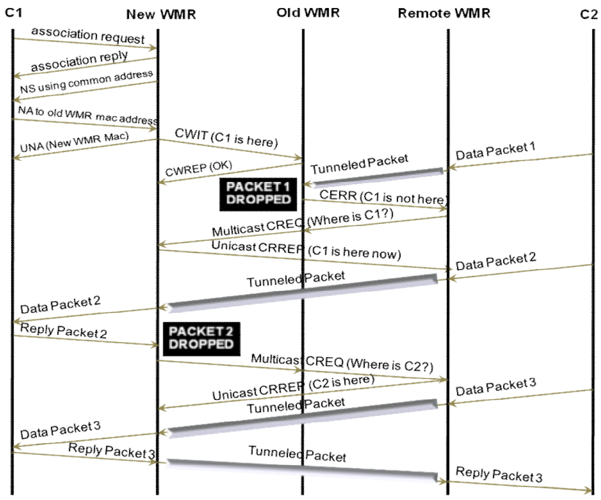


Figure 2. Communication diagram of EMM

One of the messages added is the Unsolicited Neighbor Advertisement (UNA), which is sent by a node to inform its neighbors that its link-layer address has changed, correctly updating the NDP cache of its neighbors with the MAC address of the new WMR. EMM adds an important feature to WMRs: when movement occurs, the old and the new WMRs exchange information related with the old association of the client. This way, the new WMR will receive from the clients the address of the WMR where they were associated (via NDP) and update with its own address (UNA message). Meanwhile, the new WMR informs the old WMR of the association context related to a client. Also, the old WMR will be instantly notified about the new location of its former client and clean local caches.

When a WMR receives traffic from one of its clients with a destination MAC address different than its MAC address, the packet is not discarded. Instead, the packet is tunneled and forwarded to the proper WMR serving the destination address. Then, the WMR sends an UNA message to the client in order to update its NDP cache with the value corresponding to the MAC address of the WMR.

Another message that is added to the MeshDV system is the Client Error message (CERR), which is sent by the destination WMR directly to the source WMR, informing that the client is not associated with it anymore. Receiving a data packet to a node that was removed from the LC table triggers this message.

EMM corrects most of the issues affecting the original MeshDV proposal aiming to be adaptable to wireless mesh networks in general. Results in [4] show reduced disconnection times by a large factor. However, EMM still achieves results in the order of seconds, thus it is unable to support voice continuity in wireless mesh networks. The problem is that data packets in transit during the mobility process are dropped by the WMR previously used by the client. Another issue is the existence of a large control overhead sent to the network when mobility occurs. The existence of multicast packets during the mobility process for the WMR to be able to locate the sender and receiver client after a handover, creates a large number of unnecessary packets, floods the network and increases the time of the process to locate nodes in the network. Solutions to provide mobility in WMNs must be able to offer better performance with respect to adapt to the actual requirements of nodes, and at the same time, be easy to deploy in real networks. Even being a valid and important improvement, EMM is still not able to support real-time communication during the handover process.

4. FastM: Fast Mobility Support extension

FastM was developed to further optimize mobility in WMNs. It is an evolution of EMM and MeshDV, inheriting the basic functional aspects, but modifying key functionalities, and adding a significant set of effective features. As MeshDV, FastM uses both local and foreign tables, and proposes a new table named Neighbor Client table (NC) used to keep information about neighbor nodes. With the addition of this table, which effectively exploits locality, a new set of control messages is used in order to maintain the coherence of the mesh network. These changes aim to avoid packet loss for the duration of the handover process, reduce control multicast packets in the network, save bandwidth and optimize the association and disassociation processes of clients to WMRs. The result is a much optimized and effective solution providing services like voice continuity over WMNs.

With the neighbor table, all updates made to the local and foreign tables in any WMR are broadcasted to all its neighbors (TTL=1). From real experiments, we notice that handovers are typically performed to neighbor WMRs. In this case, when a data packet reaches an old WMR, the address will be found in the neighbor table and the WMR automatically re-tunnels the packet towards the new location of the client, avoiding packet loss. This produces extremely fewer Client Request packets and speeds the handover process. Our solution does not try to maintain tables consistent, and we assume some incoherence may occur. Nevertheless, if a node has inconsistent information, the algorithm will resort to standard node location mechanisms, guaranteeing proper operation.

A. Signalling Approach

FastM uses a new set of messages to improve the mobility process:

- CLIENT_IS_THERE – When a WMR receives data packets to a client that is in its neighbor table, the WMR checks its foreign table, and a message is sent to the WMRs that had communication with that client. This message contains the new location of the client, enabling FastM to also support mobility that involves other WMRs besides the direct neighbors.
- CLIENT_IS_THERE_CONF – By receiving a CLIENT_IS_THERE message, the WMR is notified that a node moved. Then, this WMR sends to the WMR at the new location, a confirmation stating that a client attached to the first one is communicating with a client attached to the second one. This message acts as a confirmation to the handover process.
- TABLE_UPDATE – This message is broadcasted to neighbors (TTL=1) when the Client table changes. Updates like associations, disassociations, and handovers are then known by every WMR in the neighborhood.
- TABLE_UPDATE_HELLO – In order to periodically refresh node status, and every 30 seconds, all WMRs send a TABLE_UPDATE_HELLO announcing that they are still alive, and that no changes had occurred to the Client table.

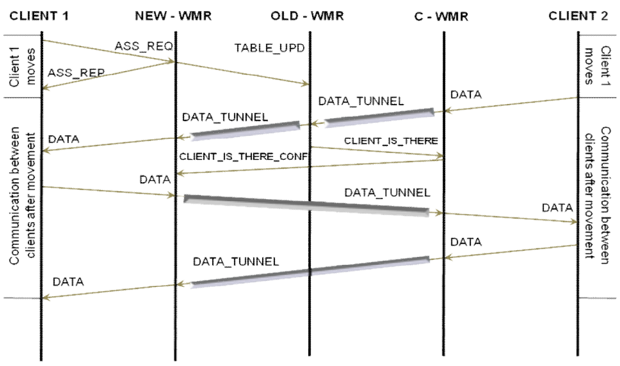


Figure 3. Communication diagram of FastM

Figure 3 depicts the communication diagram of a handover. The changes that characterize this procedure are immediately noticed: there is a tunnel between old and new WMRs, which enables packets to reach their destination until the protocol finally converges; also, no multicast packets are sent to locate clients in the network, in the case of location information is present in the neighbor table. The process occurs as follows:

1) Client 1 issues an ASSOCIATION_REQUEST message to a new WMR. This is a standard 802.11 message. Scanning delays with the new association in

IEEE 802.11 are out of scope of our work. The problem of loosing performance with scanning delays can be resolved using solutions like the one described in [20].

2) The new WMR accepts the association and sends an ASSOCIATION_REPLY message (also standard 802.11) to Client 1. Because the Client table is updated, it broadcasts a TABLE_UPDATE message to all neighbors WMR notifying others about the topology change.

3) When the correspondent WMR forwards packets from Client 2 to Client1, their destination is the old WMR because the correspondent WMR is not informed of the handover process. Data packets arriving to the old WMR are re-tunneled to the new WMR, thus no loss occurs only delay is added. The old WMR will then send a CLIENT_IS_THERE message to the correspondent WMR notifying it that Client 1 has left to the new WMR.

4) Correspondent WMR answers to the new WMR with a CLIENT_IS_THERE_CONF message. This confirms the new location and informs the new WMR about the location of clients that were communicating with the Client 1.

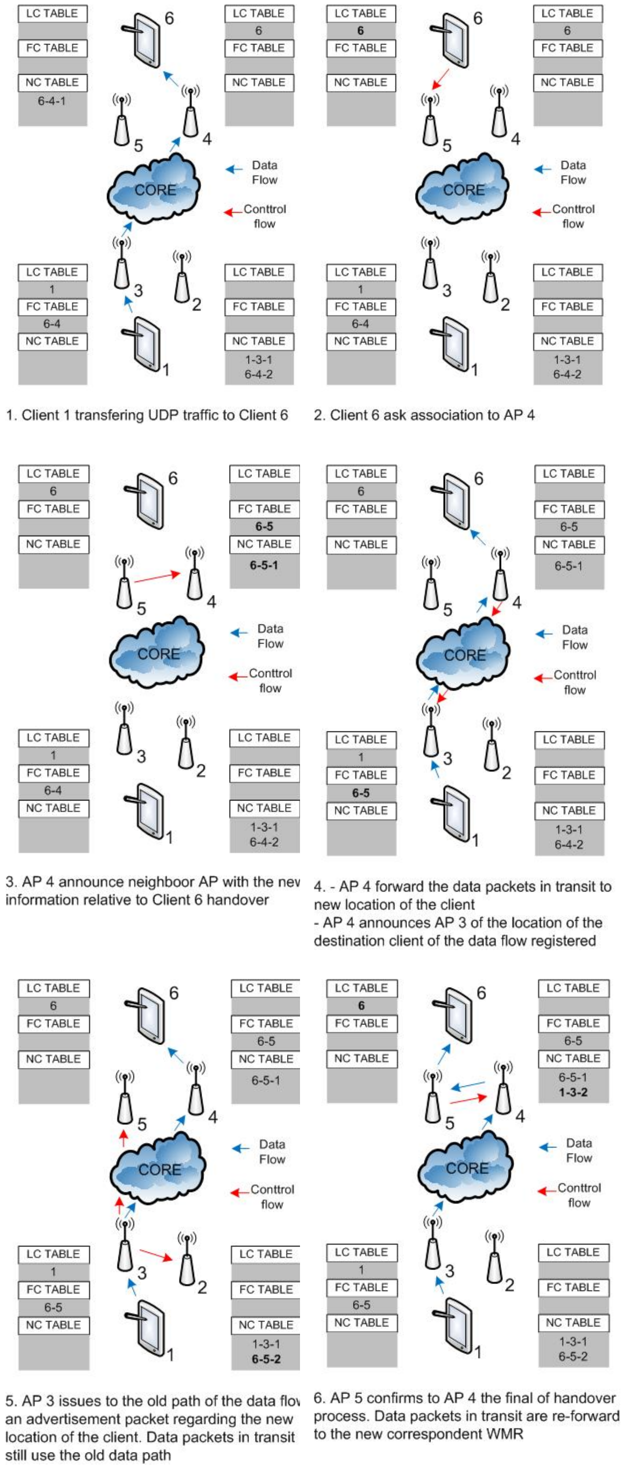
Basically, these are the steps that FastM makes in order to complete a handover process. We can see that in this process the signaling required to perform the handover is mainly between neighbor nodes, which decreases the handover latency and control overhead. There are also changes in the Client Table of each WMR in order to store location information thus predicting future handovers and facilitating them.

B. Example of FastM Operation

Figure 4 uses an example to show the several steps to reach a successful handover with the FastM mechanism, and to better introduce the changes to the several tables in the handover process. These tables are used to manage neighbor handovers with low latency and without errors.

In Figure 4, in step 1, it is shown the communication between node 1 and 6, and the 3 client tables, local, foreign and neighbor, of the mesh nodes 2 to 5. For example, in step 1, where client 1 is transferring UDP traffic with client 6, node 3 contains: node 1 in its local client table, which is the node directly connected to it; node 6 in its foreign client table (6→4, means that node 6 is connected to node 4), which is the node in the mesh network but not connected to it; and no information in its neighbor client table, since it has no knowledge about neighbors with communication. Node 2 contains no information in local and foreign tables, since this node does not have active communications. However, node 2 has information in its neighbor table of node 1, which is connected to node 3. Since the information in 3 comes from the local table, the third number is 1 (1→3→1). Notice that this third number does not address a specific node, but informs if the information came from a local (1) or a foreign (2) table. Node 2 also contains information in the neighbor table about node 6 that is connected to 4 and whose information came from the foreign table of node 3, and therefore, the third number is 2 (6→4→2). Node 4 is the same case as node 3 in terms of the neighbor table, and contains node 6 in its local table (the node it is communicating with). Finally, node 5 has no information on local and foreign tables, similarly to node 2, and contains node 6 in its neighbor table, which is connected to node 4 and whose information came from the local table of node 4 (and therefore the third number is 1: 6→4→1).

When node 6 moves, it sends an association message to node 5, which will contain now node 6 in its local table (step 2). Then, in step 3, node 5 announces node 4 (neighbor node) that node 6 is performing handover; now node 6 is in the foreign table of node 4 (obtained through node 5), and the neighbor table contains node 6, connected to node 5 and whose information came from the local table of node 4. Then, in step 4, the node 4 forwards the data packets in transit to the new location; it also announces to node 3 the location of node 6. Node 3 updates its foreign table with node 6 connected to node 5. Node 3 then sends information of the new location of the node to the old path of the communication flow, which triggers an update of the neighbor table of node 2, which now contains information from node 5 (step 5). At this stage, all nodes in both new and old communication paths have information on the handover of node 6, and then node 5 confirms to node 4, the previous node associated to node 6, that the handover process is terminated. Node 4 updates its neighbor table with information of node 1 connected to 3, and it will finish the process of forwarding data packets (step 6). Finally, the handover process is finished and the packets are forwarded through the new path (step 7).



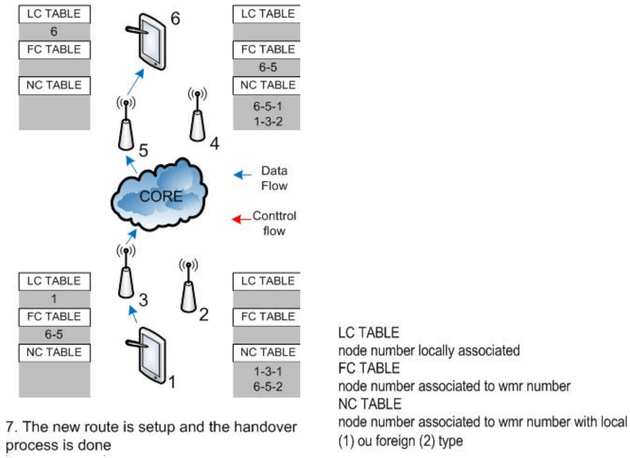


Figure 4. Steps in the handover mechanism of FastM

5. Simulation results

The solution described above was implemented and tested using the ns-2 simulator, version 2.31 [20] on an Ubuntu Linux environment.

Two different scenarios were evaluated: one to compare FastM enhanced MeshDV (MeshDV FMS) with standard MeshDV, and one to provide a more detailed study of the MeshDV with FastM solution in mobile environments with increased mobility. Evaluation of FastM MeshDV in comparison to EMM MeshDV will be addressed in future work.

In order to compare the FastM solution with standard MeshDV, a cloud-type scenario has been chosen with 37 nodes, in a square flat area of 6400 m^2 (800m x 800m). The number of nodes and the topology was chosen to enable the change of several paths during the handover. The MAC layer protocol is the IEEE 802.11 with 11 Mbps of maximum bandwidth, and 150 m of transmission range for every node. This scenario is depicted in Figure 5.

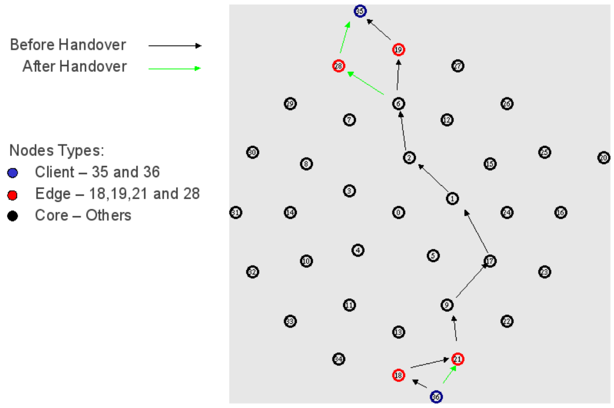


Figure 5. Simulations scenario: comparison between MeshDV and FastM

UDP and TCP flows are generated between two clients nodes as shown in Table 1. In all simulations the receiver node performs handover at t=270 seconds, while the sender node switches attachment point at t=370 seconds. Total simulation time is 450 sec.

Table1. Characteristics of the scenarios

	Configuration 1	Configuration 2
Traffic type	UDP CBR	TCP
Packets size	84 bytes	1060 bytes
Sending rate	100 pkt/sec	N/A
Number of flows	1	1

In order to better mimic the real world, artificial delays have been introduced in the WMN. These delays are used to emulate the delay required for the network interface to change channel and the network stack to configure a new address (50 milliseconds). The value is derived from previous work performed [22]. Other relevant aspect is the artificial control delays implemented in ns-2 to approximate even more the simulations to real situations, in what refers to the implementation of MeshDV, both in EMM and FastM. All other values are set to their defaults. Control packets are retransmitted if an answer (or action) is expected and was not detected, with a backoff starting at 1s.

Using this simulation environment, we evaluate MeshDV without and with FastM, according to the following metrics: throughput, packet loss and control overhead.

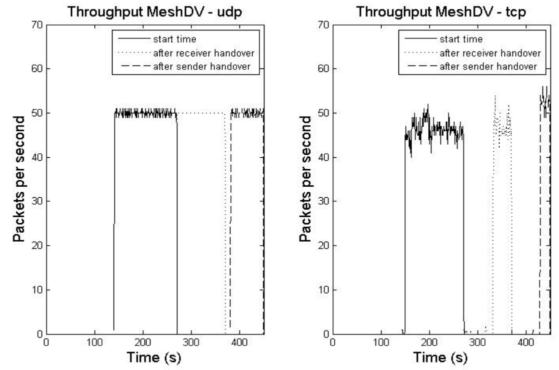


Figure 6. Throughput of MeshDV in the static case and in case of handover (receiver at t=270s, sender at t=370s).

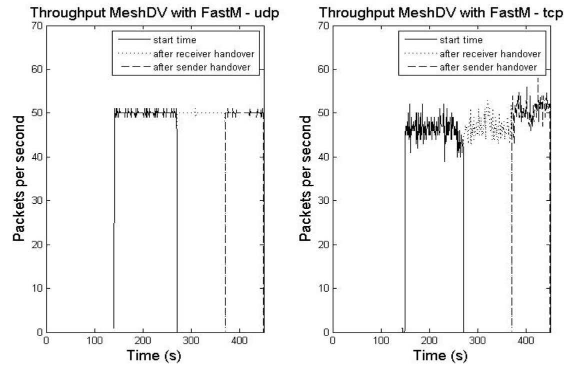


Figure 7. Throughput of MeshDV with FastM in the static case and in case of handover (receiver at t=270s, sender at t=370s).

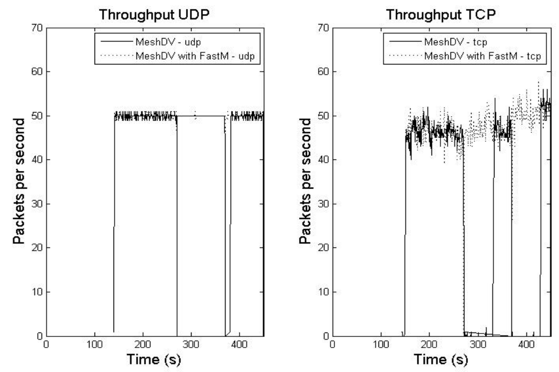


Figure 8. Throughput comparison of MeshDV with and without FastM. The later shows little interruption.

A. Throughput

As depicted in Figures 6, 7 and 8, the results obtained during the handover process, without and with FastM extension, for both configurations, are quite different.

The first curves describe throughput when no handover has occurred. In this situation, and for all cases, throughput is stable. When the receiver changes its point of attachment (second curves), throughput may take some time to reach its previous values if MeshDV is used.

FastM shows a rapid recovery. If the sender moves (last curve), disruption occurs again and can last for several tens of seconds (using MeshDV). Table 2 compares these disconnection delays with physical testbed results, obtained with MeshDV and EMM and previously presented by its authors in [4].

The testbed experimental results of both MeshDV and EMM (Table 2) show handover times in the orders of seconds; even in EMM (which is the best case) the handovers when using UDP sessions require 2 to 4 sec, while the ones of TCP sessions require 35 to 40 sec. In simulations, the values are considerably lower. However, TCP sessions in MeshDV induce disconnection delays in the order of 60s, which is unacceptable for normal communications (even if not real-time). This happens due to inappropriate scanning process, sub-optimal NDP cache handling, and 802.11 scanning. TCP performs worst than UDP due to the rate adaptation algorithm, which takes a long time to recover when packets are being lost.

Table 2. Disconnection time comparison

	Receiver Handover (s)		**Sender Handover (s)**	
	UDP	**TCP**	**UDP**	**TCP**
MeshDV[1)]	15	240	190	185
EMM[1)]	2	40	4	35
ns-2 MeshDV[2)]	0.279	61.287	0.340	59.537
ns-2 MeshDV[2)3)4)]	0.279+ [0...180]	61.287+ [0...180]+ [0...30]	0.340+ [0...180] +[0...30]	59.537+ [0...180]+ [0...30]
ns-2 FastM[2)4)]	0.118	0.344	0.089	0.141

1) Results obtained on a physical testbed (see [4])
2) Results obtained through simulation in ns-2
3) The wireless driver used by the AP's in the physical testbed, has a scanning delay, which varies between 0 and 180 seconds, and increases handover time by ~90s.
4) Due to the use of NDP in clients, there's a cache update delay, which ranges between 0 and 30 seconds. This will increase handover time by ~15s.
(Inclusion of notes 3) and 4) aims at providing results closer to the ones expected in real world scenarios)

Comparing the disconnection times in Table 2 between MeshDV and FastM (through simulation), the values obtained in FastM are significantly lower. With UDP, even when both sender and receiver clients move, traffic values are reduced and the timeout imposed by the wireless driver and NDP are suppressed; in this case, the handover time is reduced from 300 milliseconds to 100 milliseconds, a 3 times improvement. Using TCP traffic, the differences are even more evident: the techniques implemented in FastM (which minimize packet loss) are able to lower the disconnection time to milliseconds (between 100 and 300 milliseconds) compared to the 60 seconds of MeshDV (an improvement of more than 200 times). This large disconnection delay happens due to TCP congestion avoidance mechanism. When disconnection occurs, and both delay and loss figures increase, TCP will reduce the packet rate in order to minimize loss. Because disconnection time spans for several seconds, the exponential backoff will increase to high values, further increasing disconnection time for TCP applications. Ultimately, sessions may be terminated and then restarted. When using FastM, because disconnection time takes only a few hundreds of milliseconds, backoff never reaches high values, and TCP recovers more rapidly. Moreover, TCP creates two flows requiring routing, which greatly increases disconnection time.

B. Dropped Packets

Figure 9 shows the amount of dropped data packets in every scenario. The results using FastM in MeshDV show a large decrease of dropped packets during handovers, both in UDP and TCP traffic.

Using UDP traffic this difference is more noticeable due to a 10 seconds gap in which the communication between the clients is non existing due to NDP session timeout (as the handover occurs at t=370s, it only needs 10 seconds for the NDP timeout, since it is issued every 30 s). In this period, there is a large number of dropped packets, as the session in the client is not updated to the address of the new WMR. When the receiver handover takes place (t=270s), there is a small number of dropped packets, during the period the receiver changes attachment point, and the new WMR searches the location of the correspondent client (sender node).

With respect to FastM, there are no dropped packets when the receiver changes attachment to a new WMR. This is due to the re-tunnel of data packets in transit. When the server handovers to a new WMR, some packets are lost during the time it takes to disassociate and associate to a new WMR. An improvement of 97.3% (from 546 to 15 packets) is obtained with the use of FastM in MeshDV. Using TCP, FastM also shows great improvements, in this case of 35.1% (from 57 to 31 packets). As it can be shown in Figure 10, there are fewer packets generated when FastM is not used in MeshDV, which will also result in less packets being dropped. This happens because the location of clients is unknown after handovers. With FastM, packet generation is constant and some losses exist due to the TCP characteristics, such as drop links and full queues, during the simulation period. During handover, only 7 packets are dropped (4 in the first, 3 in the second handover) between the disassociation and association times of clients to a new WMR.

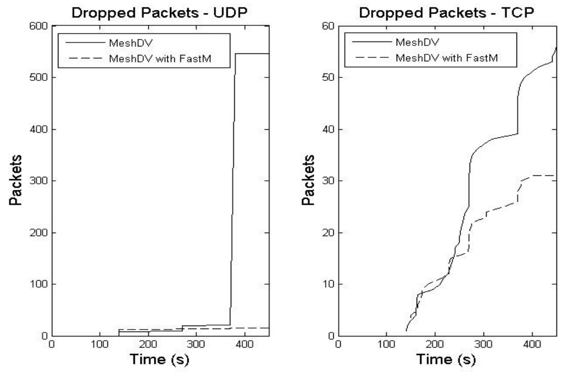


Figure 9. Sum of dropped packets after stabilization

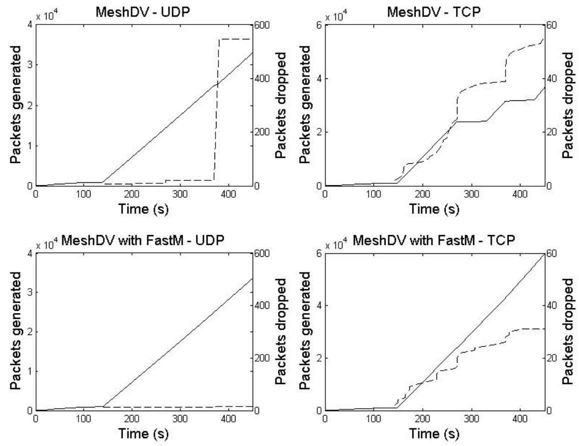


Figure 10. Comparison between generated and drops packets

C. Control Messages

In what refers to control messages (Figure 11), there are significant changes using MeshDV with and without FastM. While analyzing the performance of MeshDV without FastM, we see a typical and coherent ladder shaped process during the simulation. This is present either using UDP or TCP traffic. The initial control packets (t=10s) are due to the initial associations of the clients to the WMRs. At t=140s, when data transfer starts, some control packets are generated in order to locate the correspondent clients in the WMN. With TCP traffic, there are two client location processes (the second takes place at approximately t=143s), due to the packet flows of TCP (data and ack). Then, in each handover, control packets are generated in order to locate the clients. In TCP, due to the loss of links, some exchanges of packets are performed after the handover takes place, causing a disconnection time during this period.

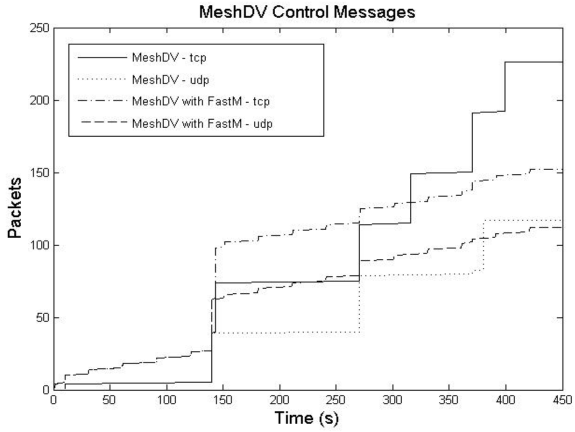


Figure 11. Sum of control packets sent to the network.

Using FastM extension in MeshDV, when mobility takes place, the number of control packets is reduced. Due to the existence of the neighbor client table, there are TABLE_UPDATE_HELLO messages every 30 seconds. During the handover, FastM reduces the control packets from 117 packets to 112 packets in UDP traffic, and from 226 packets to 152 packets in TCP traffic. With this traffic, an improvement of 32.7% in control packets is obtained. This will be even more significant in a scenario with larger number of mobile nodes and more frequent handovers. Please consider that the improvement we achieve is expressed in terms of number of packets sent to the network, which is more appropriate considering that the medium is shared. A single message sent by a node will lead to the generation of multiple packets (one for each forwarding node, plus one sent by the origin). By better exploring locality, from one side communications are reduced, and from the other they involve nodes in closer proximity, thus reducing the impact of the routing protocol.

D. Mobility Speed

In this section we analyse the performance of the FastM solution with respect to the speed of the mobile client. The scenario is similar to the previous one, but nodes are placed on a square grid instead of a circle, as shown in Figure 12. This square scenario was used in order to simplify the mobility pattern and the estimation of the position (and point of attachment) of the mobile node. All WMR nodes are fixed, while one mobile node moves around the others at speeds ranging from 4m/s to 16m/s. This node (node 26), starts its movement at the lower left corner, and moves first to the right, then up, left and down, stopping near to node 2. In this scenario, the mobile node is communicating with one of the fixed nodes either using TCP or UDP. When using TCP, it sends packets with 1000 bytes of data (1040 bytes including TCP and IP headers); bandwidth is limited only by TCP internal contention mechanisms. When using UDP, the node sends a CBR flow with packets of 1000 bytes and a constant periodicity of 10 milliseconds. As in the previous case, total simulation time is 500s and the flows start at t=100s.

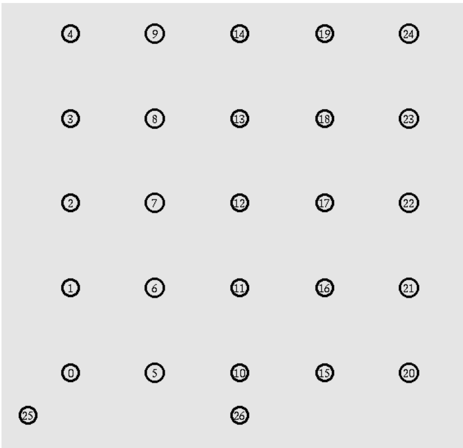


Figure 12. Simulations scenario: FastM with increased mobility

Concerning delivery rate in Figure 13, we observe that the total number of packets received decreases with the increase of velocity. These results are inline with our expectations and reflect the proper operation of the routing protocol and handover approach in the mesh network: mobility decreases the time between handovers, and with more frequent handovers, the losses are increased, which consequently decreases the throughput. One interesting aspect is that delivery rate seems to decrease more rapidly at lower speeds, reaching stabilization at higher speeds. Above 10 m/sec, results show that the protocol is able to maintain some minimal delivery rates. This shows the effectiveness of the neighboring tables: with higher handover frequency, there is less time for tables to be disseminated and normal MeshDV handover process occurs (the effect and enhancement of FastM is minimized). More aggressive table maintenance strategies could help increasing the performance; still, the resulting higher overhead would probably nullify the overall performance improvement. As expected, UDP achieves higher delivery rate than TCP due to the inexistent flow control mechanisms in UDP; however, the trend is similar to both types of communications.

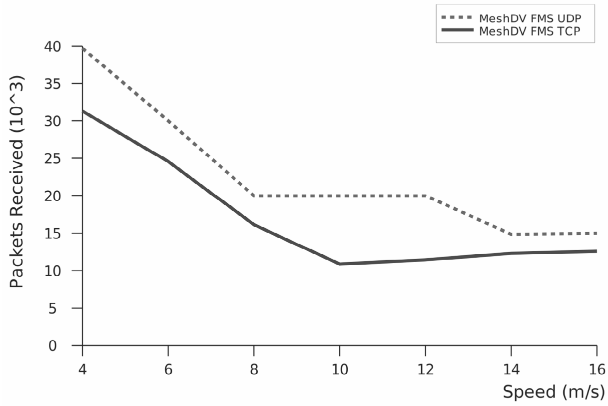


Figure 13. Comparison between throughput in FastM in TCP and UDP as a function of mobile node velocity

According to the results presented in Figure 13, we would expect that overhead would be higher when neighbor tables are being maintained (more maintenance in small mobility scenarios), and this is shown in Figure 14. This graphic depicts the number of control packets sent to the network as a function of speed of the mobile node. For lower mobility, as nodes slowly hop between the WMRs, neighbor tables are filled and propagated to their neighbors, thus producing a higher overhead. As the nodes start to move with a higher speed (and we realize from the graphic that the critical point is at 10 m/s), the neighbor client tables construction becomes more inefficient, since there is less time to propagate the tables to the neighbors, and this reduces total overhead.

Related to the difference between UDP and TCP curves, TCP has, in most of the cases, increased overhead, as in the first scenario depicted in Figure 11. For increased speed, the difference in overhead is not so noticeable: this is due to the mobility effects on TCP sessions.

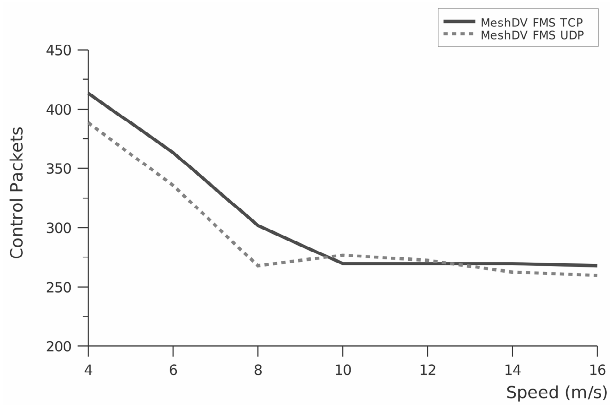


Figure 14. Number of control packets sent to the network as a function of node velocity

6. Conclusions

The support for VoIP applications in current and future WMNs is considered to be vital to its success. However, the wireless medium and multi-hop routing protocols are frequently unable to meet the requirements of seamless mobile nodes handover, while maintaining uninterrupted calls: efficient mobility mechanisms are required to enhance the performance of the communications in these networks in mobile scenarios.

In this paper, we proposed a novel extension to improve the mobility process, denoted as FastM, that integrates mobility and routing mechanisms, and that brings a new way to deal with neighborhoods, using other nodes to maintain information about the organization of the WMN. Results obtained with ns-2 prove the efficiency of the solution and its effectiveness in meeting the requirements of low packet loss, disconnection times, and control overhead, even in scenarios with increased mobility. The disconnection times were reduced to values around 100 milliseconds, being able to meet the real-time communications requirements.

7. Future Work

This mobility mechanism is based in principles that are transversal to other mobility mechanisms that exist nowadays. In this paper this solution was applied to MeshDV; however, its concepts are applicable to most proactive routing protocols. Its support in EMM will be addressed as future work.

New routing processes use concepts related to k-neighborhoods where the algorithm itself organizes the network in several groups, having a number of nodes with a global perspective of the overall network, one for each group. Adapting this concept to the principle behind FastM is intuitive. FastM is supposed to be a mobility management in small perspective, using the relations between neighbors to re-forward packets and adjust the topology when mobility occurs. This way, implementing FastM inside every neighborhood to support micro mobility management and in every group leader to support macro mobility management, it will be created a system with fully support of micro and macro mobility management integration with considerably gains. This integration brings two main advantages: larger routing capacity in large networks and capacity of maintaining k-neighborhoods with high performance. FastM contributes with good results on handovers: no packet loss, low timeout and high performance on route convergence. These two techniques combined add a great robustness to a network when handovers occur. Applying the procedures in very large networks will have no influence in the global performance, as the responsibilities and overloads are distributed between all the k-neighborhoods.

Other future work in this area also concerns the implementation of the mobility approach and the comparison of simulation and experimental results, to assess the behaviour of these mechanisms in real environments. Moreover, a comparison should be made with different mobility solutions.

Acknowledgments

This work was partially supported by the European Commission project IST-WIP under contract 27402.

8. References

[1] L. Couto, J. P. Barraca, S. Sargento, R. L. Aguiar, "FastM in WMN: a Fast Mobility Support extension for Wireless Mesh Networks", The Second International Conference on Advances in Mesh Networks (MESH 2009), Athens (Greece), June 2009.

[2] IST WIP – An All Wireless Mobile Network Architecture, http://www.ist-wip.org.

[3] L. Iannone and S. Fdida, "Meshdv: A distance vector mobility-tolerant routing protocol for wireless mesh networks", *IEEE ICPS Workshop on Multi-hop Ad hoc Networks: from theory to reality (RealMAN'06)*, July 2005.

[4] M. Bezahaf, L. Iannone and S. Fdida, "Enhanced Mobility Management in Wireless Mesh Networks", *Journées Doctorales em Informatique et Réseaux (JDIR08)*, January 2008.

[5] H. Wang, Q. Huang, Y. Xia, Y. WU, Y. Yuan, "A Network-Based Local Mobility Management Scheme for Wireless Mesh Networks", *In Proc. of WCNC 2007*.

[6] Y. Fan, J. Zhang, X. Shen, "Mobility-Aware Multi-Path Forwarding Scheme for Wireless Mesh Networks", *In Proc. of WCNC 2008*.

[7] A. Capone, M. Cesana, S. Napoli and A. Pollastro, "MobiMESH: a Complete Solution for Wireless Mesh Networking", *IEEE*, 2008.

[8] Y. Amir, C. Danilov, et al., "Fast Handoff for Seamless Wireless Mesh Networks", *In Proc. of MobiSys'06*, June 2006.

[9] S. Speicher, C. H. Cap, "Fast Layer 3 Handoffs in AODV-based IEEE 802.11 Wireless Mesh Networks," In Proc. of 3rd International Symposium on Wireless Communications Systems (ISWCS),Sept, 2006.

[10] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, 2003.

[11] G. Koodli, "Fast Handovers for Mobile Ipv6," RFC 4068, Jul., 2005.

[12] E. Bortnikov, T. Kol, A. Vaisman, "QMesh: a QoS Mesh Network with Mobility Support," In Proc. of ACM SIGMOBILE Mobile Computing and Communications**,** vol. 12, no. 1, Jan., 2008.

[13] F. Rosseau, Y. Grunenberger, V. Untz, E. Schiller, P. Starzetz, F. Theoleyre, M. Heusse, O. Alphand and A. Duda, "An Architecture for Seamless Mobility in Spontaneous Wireless Mesh Networks". *in Proc. MobiArch'07, Kyoto, Japan*, August 2007.

[14] F. Rousseau, F. Théoleyre, A. Duda, A. Krendzel, M. Requena-Esteso, J. Mangues-Bafalluy, "Geo-mobility and Location Service in Spontaneous Wireless Mesh Networks," In Proc. of ICT-MobileSummit, Stockholm, Sweden, June 10-12, 2008.

[15] X. Wu, "VPDS: Virtual Home Region based Distributed Position Service in Mobile Ad Hoc Networks," In Proc. of IEEE International Conference on Distributed Computing Systems (ICDCS), June, 2005.

[16] D.Huang, P. Lin, C. Gan, "Design and Performance Study for a Mobility Management Mechanism (WMM) Using Location Cache for Wireless Mesh Networks," IEEE Transactions on Mobile Computing, vol. 7, no. 5, May, 2008.

[17] IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Fast Basic Service Set (BSS), July 2008.

[18] C.E. Perkins and P. Bhagwat, "Highly Dynamic destination-sequenced distance vector routing (DSDV) for mobile computers." *in Proc. ACM SIGCOMM 94, London, UK*, Oct. 1994, pp. 234-244.

[19] T. Narten, E. Nordmark, W. Simpson, "RFC 2461 – Neighbor Discovery for IP Version 6 (IPv6)".

[20] I. Ramani, S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks", *Infocom 05*, March 2005.

[21] The ns-2 Manual (Oct. 2, 2006), http://www.isi.edu/nsnam/ns.

[22] L. Couto, J. Barraca, S. Sargento, R. Aguiar, "Fast Mobility in proactive routing protocols", *ICT-MobileSummit 08*, July 2008.