An urban mobility model and predictive handover scheme for mobile IP: OPNET modeling and simulation

Reference:
van den Wijngaert Nik, Blondia Christian.- *An urban mobility model and predictive handover scheme for mobile IP: OPNET modeling and simulation*
Proceedings of OPNET Conference 2004 - s.l., 2004
Handle: http://hdl.handle.net/10067/505730151162165141

An Urban Mobility Model and Predictive Handover Scheme for Mobile IP: OPNET Modeling and Simulation

N. Van den Wijngaert, C. Blondia
Department of Mathematic and Computer Science
University of Antwerp
Middelheimlaan 1, BE-2020 Antwerp, Belgium
E-mail: nik.vandenwijngaert@ua.ac.be

Abstract

This paper proposes both an urban mobility model for OPNET and a low latency handover mechanism for Mobile IPv4. The mobility model uses street data to navigate the nodes in a city: (1) semi-randomly according to street precedence level and a configurable bias towards direction, and (2) different modes of waypoint navigation. When roaming across WiFi access points, the mobile nodes need to conduct handovers to different networks. We propose a scheme where the network learns movement patterns of nodes through standard Mobile IP signaling and can, when sufficiently informed, provide much improved handover conditions. The OPNET implementation of both the mobility model and the handover mechanism is presented, along with simulation results.

Introduction

As we are moving towards ubiquitous wireless network coverage, the need for wireless handover protocols that deliver a certain degree of Quality of Service (QoS) grows. Mobile IP (MIP) [15] has become the standard network-level handover protocol for a mobile node (MN) to roam between different IP networks. While MIP is well suited for macro-mobility (mobility between e.g. different administrative domains), it is not very good for dealing with fast consecutive changes in network point-of-attachment (NPA). The IETF has proposed several solutions to this so-called micro-mobility: Cellular IP (CIP) [6], HAWAII [18] and Hierarchical MIP (HMIP) [9]. HMIP adheres closest to the standard MIP mechanism and will be used throughout the rest of the paper.

The key to handling micro-mobility is to deal with all handoff related messaging locally. One of the principal delay components in MIP is the signaling induced latency. When the MN attaches itself to a new NPA, it first needs to discover a Foreign Agent (FA) active on the new link. This FA will provide the MN with its temporary address on the foreign network, the Care-of-Address (CoA). The MN will then send a Registration Request to the FA, which will forward this to the Home Agent (HA), the mobility agent on the MN's home network. The HA can be at a very large routing distance from the FA, causing all traffic to the MN intercepted by the HA to be routed to the incorrect CoA, as the Registration Request will not have arrived there yet. HMIP attempts to overcome this delay by introducing a tree-like structure of FAs, where each branch is governed by a Gateway Foreign Agent (GFA). When the MN first enters a domain, it conducts a Home Registration, where it registers the GFA's address as its CoA, not the address of the lowest level FA to which it is connected. When it subsequently moves to another FA in the same branch, the MN initializes a Regional Registration, which only goes up to the Crossover FA (the FA common to the path from the old FA to the GFA and the new FA to the GFA). As a result, each local movement does not change the CoA from the HA's perspective. An intelligent structure of FA's in the wireless access network can minimize the routing overhead of the registration messaging.

The second major delay component in MIP is called the handoff induced latency, caused by the gap in network layer communication due to the lower layer handover. When using 802.11b (WiFi) [19] as the wireless access technology, this gap is due to the fact that it is a break-before-make protocol. Whenever the MAC layer decides that the current connection has degraded below a certain level, it breaks all connection with its current access point (AP) and initiates a scanning phase, followed by a re-association and re-authentication sequence to a new AP. This sequence of events can take up to several hundreds of milliseconds [12], during which no network layer traffic is possible at all. The IETF has also proposed to overcome this delay, in the form of Low Latency Handover schemes [7]. This draft consists essentially of two different protocols: Pre-Registration and Post-Registration. The first tries to conclude the network layer handoff before the actual Layer2 switch, while the second tries to postpone the MIP registration until after the Layer2 change while the MN is still able to receive traffic through its old FA (oFA) through the use of tunnels.

It is clear that every MIP extension that tries to bridge Layer2 dynamics cannot rely on Layer3 information alone. A cross-layered approach is essential and this is what Pre- and Post-Registration attempt to accomplish. They rely on the presence of several Layer2 triggers to be available in the system:

1. Anticipation Trigger (*AT*): signals a future handover event and to which AP;
2. *oFA L2LD*: connection is severed with the old FA's (oFA) access point;
3. *nFA L2LU*: connection is established with the new FA's (nFA) AP;
4. *MN L2LU*: all communication is established and the MN is ready to initiate traffic.

Note that chronologically, *AT* < *oFA L2LD* < *nFA L2LU* < *MN L2LU* at all times. While these protocols are very promising, they do not work on WiFi networks: the *AT* and *oFA L2LD* triggers are simply not present. An AP usually only knows when an MN has left its coverage area when it is unable to deliver packets to that MN, which can be some time after the actual handover. *AT* is even more of a problem, as neither the MN nor the oFA receive indications when the MN is going to be conducting a handover and to which AP. Even if *AT* can be generated through other means, the timing between *AT* and *oFA*

L2LD remains a big issue as this is crucial to packet loss and buffering requirements [2].

When the MN knows its current location and trajectory (e.g. it is moving on a train or on a highway) then we can exploit this location information and deploy a Low Latency scheme on WiFi. This scheme is called Location Augmented Bulk Pre-Registration (LABPR) and is presented in [21][22]. It is very well suited to the afore-mentioned mobility scenarios, but less so when the MN moves randomly. This paper presents a handover scheme based on Post-Registration in which the network learns node mobility patterns and is able to provide better handover conditions without the MN's intervention.

A lot of wireless network simulation studies are done in scenarios where node mobility is realized through random waypoint principles in an open environment or other algorithmically generated paths (e.g. Voronoi diagram edges between buildings [11]). Most of the time the wireless propagation model used is freespace, which assumes that all sender and receiver pairs are within line-of-sight (LOS) of each other and no other radio effects other than distance attenuation are considered. We have developed a mobility model based on street data and combined this with advanced radio wave propagation modeling.

The remainder of this paper is organized as follows: first the urban mobility model is presented, along with its OPNET [14] implementation. The second part will be about the new predictive MIP handover scheme, where we will discuss protocol dynamics, the OPNET implementation and several results. We will finish with some conclusions and future work.

Urban Mobility Model: Movement Protocols

While the random waypoint/freespace model can be used to study protocol behavior to some extent, it is not very good at capturing node movement and radio effects in more realistic environments. We propose a mobility system in which streets are modeled based upon following criteria:

- Weight: determines the cost and the probability this road will be selected if it crosses another one;
- Width: road width;
- Directions: one- or two-way street;
- Speed: street traversal speed;
- Pause time: waiting time at the end of the street;
- Street coordinates.

Both speed and pause times are drawn from any supported statistical distribution. The majority of other road data can be extracted from GIS (Geographical Information System) databases and everything is combined in a GDF file, easily accessible to OPNET processes. Along with this street definition file is a "curb" definition file which lists the obstacles in the environment. As the name suggests, these will usually be the edges of buildings along the streets.

Street data is parsed into a weighted directed graph, where the cost of each street segment is inversely related to its weight. Whenever the shortest path is required from one vertex to another, we can use e.g. the Dijkstra algorithm on this graph.

Figure 1 shows part of the map of Antwerp, as used in Modeler. Mobile nodes can be placed randomly on the map, as they will clip to the nearest vertex when the simulation starts. These nodes have three modes of movement: (i) Target Flags, (ii) Random Waypoint and (iii) Random Next Hop.

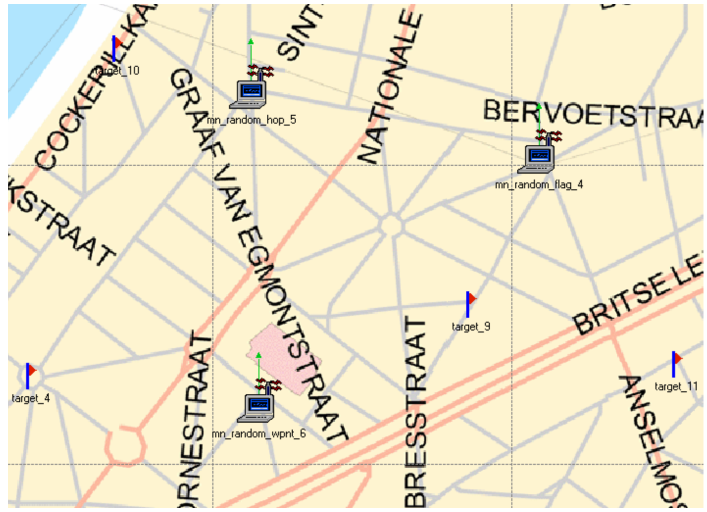

Figure 1: Downtown Antwerp

(i) Target flags are of a new node model type, which doesn't have any process models contained within. There is only one node level attribute which defines this node as a *Target Flag*. These flags can then be randomly placed in the simulation scenario. MNs can be configured to move to these flags in a random fashion, or they can be given an order in which to visit them (and do this once, or keep cycling the sequence).

(ii) In Random Waypoint mode, the MN moves to a randomly selected vertex in the graph using the shortest path (minimal cost) algorithm.

(iii) The Random Next Hop movement selects the next one-hop target vertex to move to whenever the MN arrives at a junction.

   a. If the different outgoing streets at the junction have different weights, selection is based on these weights.

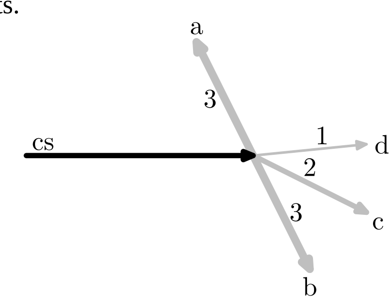

Figure 2: Weight-based road selection

In the example shown in Figure 2, let cs be the current street the MN is on when it reaches the junction. Road a or b will be selected with a probability of 1/3, c with a probability of 2/9 and d with a probability of 1/9.

   b. If the outgoing road segments have the same weight, selection is based on the cosine of the angle between the current street vector and the outgoing street vector.

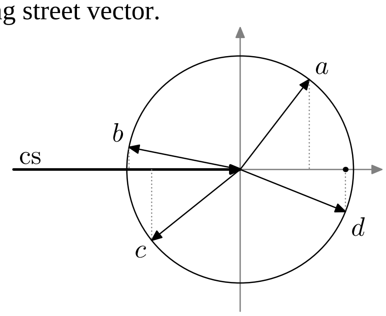

Figure 3: Direction-based road selection

Figure 3 shows four outgoing streets at a junction with all the same weights. Looking at the unity circle around the intersection, we see that road vector d has the smallest angle and hence the largest cosine, followed by a, c and then b. The mobility model has a configurable bias value ranging from 0 to 10, where 0 means that there is no bias towards direction and that all roads have equal chance of being selected. A bias value of 10 makes the MN always choose the "most forward" direction. The optimal value depends on the street scenario used, but a bias ranging from 3 to 6 will yield good results in a lot of cases: it prevents the MN from going around in circles, while still providing a good degree of randomness.

Urban Mobility Model: Propagation Modeling

The basic freespace model for radio wave propagation modeling is:

$$PL_{dBm} = 32.4 + 20\log(dist_{km}) + 20\log(freq_{MHz})\ .$$

As this only account for signal attenuation vs. distance, it is insufficient for capturing radio effects like reflection, diffraction and scattering in urban environments. The most accurate way of modeling these effects is through a ray-tracing scheme, but is computationally too expensive. We chose to implement the semi-deterministic empirical models of COST231-Walfish-Ikegami [4] and Har-Xia-Bertoni [10].

Model Name	Attributes
COST231-Walfish-Ikegami	Average Building Height
	Building Separation
	Environment Type
Har-Xia-Bertoni	Average Building Height
	Environment Type

Table 1: Propagation Model Parameters

The COST231-Walfish-Ikegami model is composed of the freespace loss term corrected by terms for multiple screen diffraction loss and rooftop-to-street diffraction and scatter loss. It is best suited for AP placement above building rooftops and larger coverage areas. The Har-Xia-Bertoni model attempts to provide accurate models for AP placement near or below rooftops in low-rise environments and at streetlamp height in high-rise scenarios. It differentiates between these two environment types to determine whether to take into account radio propagation over rooftops (low-rise) or to focus on diffraction around buildings (high-rise).

The extra parameters that both models rely on are listed in Table 1. Other input parameters like transceiver height and frequency are extracted from the packet's *Transmission Data* attribute set.

Urban Mobility Model: OPNET Implementation

The models discussed above have been implemented in OPNET Modeler in a centralized approach. Mobile nodes only need to have certain node-level attributes configured for the mobility model to work, which are listed in Table 2.

Attribute Name	Possible Values
Street Mobility Mode	Node Random Next Hop Random Waypoint Target Flags
Target Flag Mode	Random Single Sequence Cycle Sequence
Target Flag Sequence	Compound attribute listing target flag names

Table 2: MN Mobility Attributes

The centralized mobility process model is depicted in Figure 4 and is the only process in the mobility utility node. At the *begsim* interrupt, it iterates over all the mobile nodes in the network (who have their *Trajectory* attribute set to VECTOR) in the *set_in_motion* state. This process keeps track of all the nodes configured for street mobility. Whenever they reach the end of a street segment, their *distance* and *bearing* attributes are adjusted to put them on the next selected street. These movement update events are triggered by function interrupts.

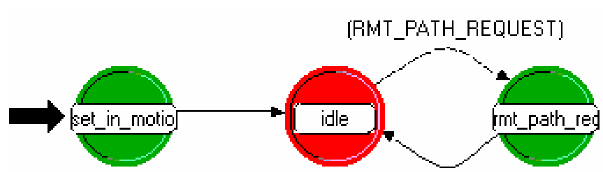

Figure 4: Street Mobility Process

The street mobility process rests in an idle state and the only interrupt able to change this state is a *RMT_PATH_REQUEST* event, which allows mobile nodes to request the remainder of their path. This can be used to e.g. simulate GPS systems and is only valid when the Random Waypoint or Target Flag mobility modes are used.

The COST-231 and Har-Xia-Bertoni propagation models are implemented in the WLAN pipeline stage *wlan_power.ps.c*, which discovers the mobility process through the process wide registry. Both propagation models make the distinction between LOS and non-LOS conditions between sender and receiver. Here the curb file is used to determine if any of the defined obstacles intersect with the straight line drawn from sender to receiver. If the curb file is very large, computation times can become very large because this check has to be done for each transmission. An optimization mode has been implemented, which calculates a set of streets for each fixed wireless node (to APs). It works as follows: if an AP can see both street endpoints, it is placed in the LOS list for that AP. If it can see only one endpoint, then it is placed in the "maybe LOS" list. Now whenever a packet is transmitted, the street the MN in question is on is checked for occurrence in either list. If a match is found in the "maybe LOS" streets list, full LOS computations are made. If the street is found in neither list, non-LOS conditions are assumed. This optimization may yield false negatives in some cases, of which Figure 5 is one. Neither endpoint of the MN's current street is visible from APs point of view, but there is a clear transmission path between the two transceivers. These cases however, are quite infrequent and it is up to the user to decide if this is an issue in the scenario under study.

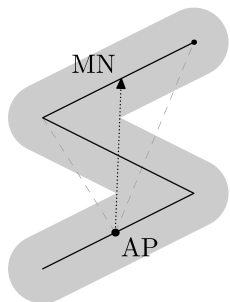

Figure 5: LOS Optimization False Negative

Table 3 summarizes the mobility manager's attributes.

Attribute Name	Comments
Path Loss Modeling	
COST231-Walfish-Ikegami Parameters	See Table 1
Har-Xia-Bertoni Parameters	See Table 1
Optimized LOS Determination	Enables LOS efficiency mode
Path Loss Model	Any of above models (including freespace)
Street Mobility	
Curb File	Defines scenario obstacles
Direction Bias	Controls relative angle weight
Position Update Interval	For animation refresh purposes
Start Time	Start mobility from…
Stop Time	Stop mobility from…
Street File	Defines street attributes
Coordinate Correction	If displayed scenario map and Street File coordinates do not line up

Table 3: Mobility Manager Attributes

A Predictive Handover Scheme

As our predictive handover protocol uses techniques from Route Optimization [16] and Post-Registration, they will be briefly explained first.

The main goal of Route Optimization is to remove the overhead caused by triangular routing: all traffic sent by a CN to the MN is first sent to the Home Network, where it is intercepted by the HA and tunneled to the CoA. Route Optimization allows for CNs to tunnel traffic directly to the MN's CoA. Because CNs have no idea of the MN's movement, the MN attaches a Previous Foreign Agent extension to its Registration Request (RegReq). The nFA removes this extension and sends a Binding Update message to the previous FA which will delete its binding entry for the MN and forward all incoming traffic to the nFA. Any CN tunneling traffic directly to this out-of-date CoA will receive a Binding Warning message to update their forwarding tables. This technique is known as Smooth Handover and is illustrated in Figure 6.

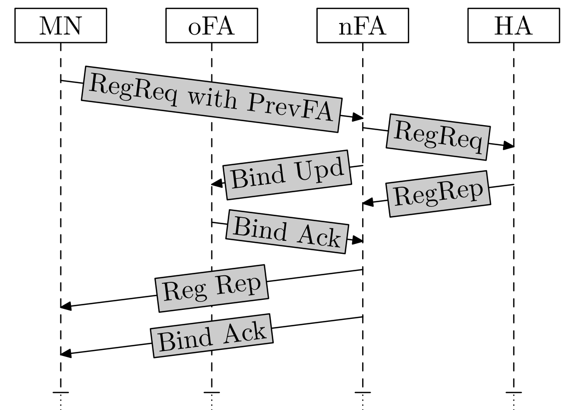

Figure 6: Smooth Handover

The operation of the Post-Registration handover scheme is illustrated in Figure 7. When a node is connected to a foreign network served by oFA, all traffic is forwarded there from a correspondent node (CN), either directly if Route Optimization is used or via the HA. The received packets are then de-tunneled at the CoA and put on the local link for delivery to the MN.

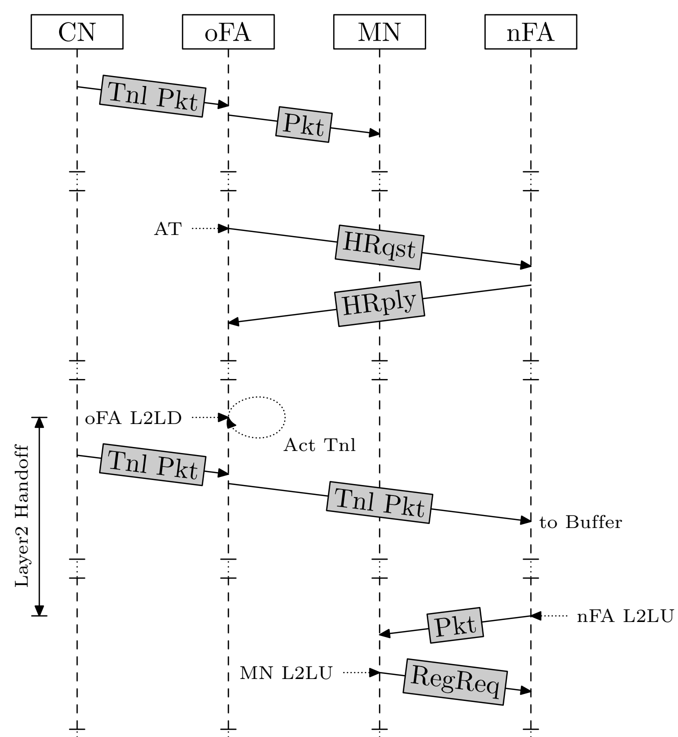

Figure 7: Post-Registration

When a Layer2 handover event is going to occur in the near future, an *AT* is received at the oFA who sends a Handoff Request (HRqst) to the nFA specified by the *AT*. When this message is replied to with a Handoff Reply (HRply), a tunnel has been set-up (but not activated) between the two FAs. When the *oFA L2LD* trigger is received at the oFA, the tunnel is activated (*Act Tnl* event) and all received traffic is re-tunneled to the nFA where it is placed in a buffer if the *nFA L2LU* trigger has not been received yet. When the MN associates itself with the nFA's AP, the buffered packets are forwarded. At any later point in time (*MN L2LU*), the MN can choose to register itself with the nFA and have all incoming traffic tunneled there directly.

This scheme tends to produce better handover quality than Pre-Registration at the cost of slightly higher packet delay due to the extra tunneling hop. Post-Registration is not very dependent on the timing between *AT* and *oFA L2LD*; they just have to be spaced far enough apart to allow full tunnel set-up. Unfortunately this protocol is unusable on a WiFi network, as the *AT* and *oFA L2LD* triggers are not available there. As stated before, the MAC layer has no idea to which AP it is going to go next, because this is dynamically discovered during the scanning phase.

We propose a low latency predictive handover scheme based on the idea of Post-Registration, and call it simply Predictive Post. It is a completely network-initiated protocol, where the MN only needs to implement the basic Route Optimization extension. Using the Smooth Handover mechanism, each FA can learn the movement patterns of the visiting nodes. Consider an MN moving from FA1 to FA2 and then to FA3:

1. MN associates with FA2's AP and sends a Registration Request with PrevFA extension to FA2 (which hereby learns the address of FA1).
   → FA2 sets up a temporary association
   FA1 ➔ (MN).
   → MN starts timing visit.
2. MN moves from FA2 to FA3 and again sends a Registration Request with PrevFA extension, this time with an extra field that holds the time that it spent under FA2's control.
   → FA3 sends Binding Update (with visit time) to FA2
   → FA2 completes association
   FA1 ➔ *self* ➔ FA3, visit time: x sec.

As nodes move about, each FA will construct tables like the one shown in Table 4.

From	To	Count	Visit Time
FA1	FA3	12	Duration[12]
	FA7	3	Duration[3]
FA4	FA9	6	Duration[6]

Table 4: Mobility Table Example

Based on these observations, whenever an MN registers with an FA, that FA can predict which network(s) the MN will move to next. It can then choose to set-up Post-Registration tunnels with any number of most likely next-hop FAs. When looking at urban traffic, the number of possible next-hops is usually very small and results will show that a high predictive accuracy can be obtained. When all nodes move at the same speed, then all handovers will be seamless. This is generally not the case however and estimating when to activate the appropriate tunnels can be problematic and will be discussed in more detail in the results section.

Predictive Post: OPNET Implementation

The Predictive Post protocol is integrated into our MIP stack discussed in [20] and [21]. This stack is completely integrated into OPNET's standard model library and supports following protocols:

- Standard MIP
- Hinted Cell Switching (HCS) [8]
- Hierarchical MIP
- Route Optimization
- Optimized Route Optimization [17]
- Low Latency Handover Schemes
  - Pre-Registration
  - Post-Registration
- LABPR
- Predictive Post.

A number of above implementations have been used in a number of performance analysis studies [1][2][3][5]. While most of the new node and process models are detailed in [20] and [21], I will highlight existing features and introduce the new ones in the following two sections.

Predictive Post: Node Models

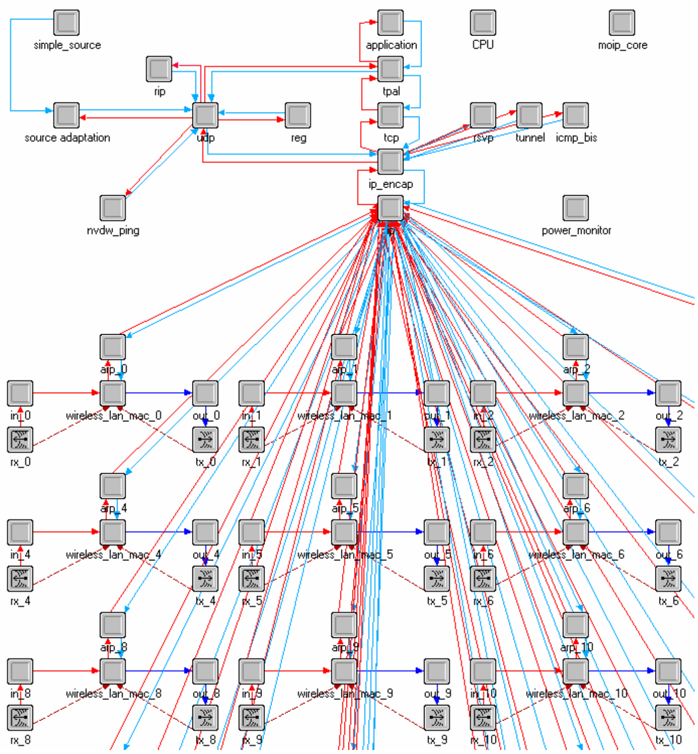

Figure 8: MN Model

Since Layer2 mobility was not supported by the OPNET WLAN models when the stack was first developed, a multi-homed approach was taken to model MNs. Figure 8 shows the structure of this node model. It was originally derived from *wlan_wkstn_adv* and that basic structure is still visible: the entire stack from *application* down to *ip* has been preserved. The MN is equipped with as many wireless interfaces as there are BSS IDs to which it can roam. The modifications to the IP routing logic make sure that only one interface is used at all times and that packets received on the other ones are discarded. There is a gate module between each *wlan_mac* and its transceivers which can be configured to block all traffic if the interface they're serving is not flagged as the active one. This multi-homed modeling has several distinct advantages. First of all there is the added flexibility and control: we can either use very strict scripts in which the every handover event is defined or we can opt for e.g. more realistic modeling of the Layer2 scanning phase. Secondly, these interfaces also needn't be of the same type. We use primarily WLAN interfaces, but any one of these can be replaced by e.g. a GPRS or Bluetooth interface.

Interfacing to *ip_encap* are two new modules: *tunnel* and *icmp_bis*. These provide functionality that could also be modeled as child processes of the *ip* module, but are better maintainable as interface modules to *ip_encap*. The *tunnel* process takes care of IP-in-IP tunneling and has several buffers in place to allow FAs to hold on to traffic for nodes that are not yet under their control. Transmission and processing of agent solicitations and advertisements is done by *icmp_bis*. All MIP-related signaling messages like Registration Requests/Replies, Binding Updates/Warnings, tunnel set-up messages,… are handled by the *reg* module which communicates through UDP.

The *moip_core* and *power_monitor* modules are unique to mobile node models. The *moip_core* process is a distribution point for all handover related events. It can also send "deus-ex-machina" trigger events to FAs in the network to simulate triggers that are not available through Layer2 means This allows evaluation of e.g. a protocol specified by an IETF-draft even though all required information is not given by the other layers.

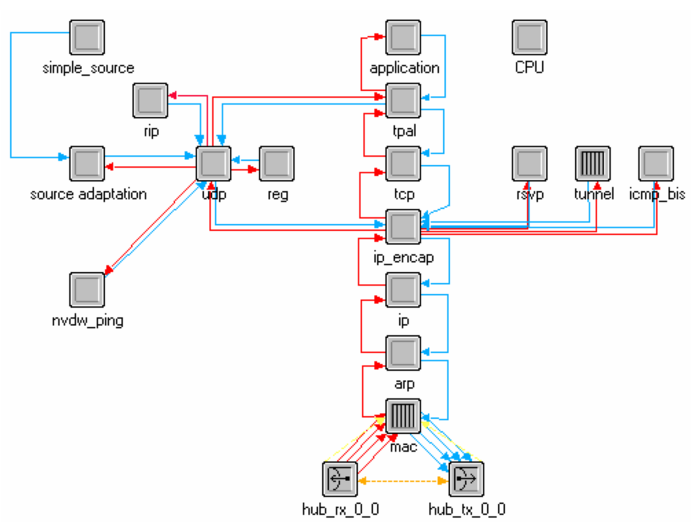

Figure 9: FA Model

The FA node model is shown in Figure 9 and is basically the *eth_wkstn_adv* node model enhanced with the required processes for mobility support. The *source_adaptation* module seen in both nodes acts as a UDP packet wrapper for *simple_source*-generated traffic to allow easy traffic generation without the need for configuring applications profiles.

Predictive Post: Process Models

Network layer mobility can be initiated through very controlled scripts or dynamic mechanisms, where criteria are based on MN location or received signal strength. The most important process in the MN is *moip_core* depicted in Figure 10. This is the central dispatching point for sending handover events to other modules both in the surrounding node and in FAs. It is also here that the mobility method is chosen from the scripted or one of the dynamic methods. The most important attributes are shown in Table 5.

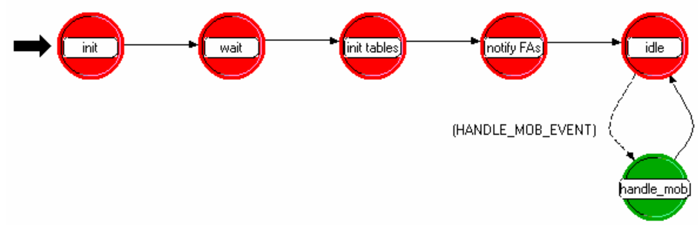

Figure 10: *moip_core* Process

Attribute Name	Comments
Mobility Method	Possible values: • Scripted • Dynamic: Nearest AP • Dynamic: Signal Level • Dynamic: Realistic Scanning
Mobility Sequence	Defines the node's mobility script
Interface Configuration	Specifies which interface is tied to the Home Network
Home Agent	HA's IP address
Sequence Repetition Count	Number of times to repeat the mobility script
Dwell Time	Pause time after each mobility script sequence
Discard Duplicate Packets	Instructs the MN to discard duplicate packets at the IP level
Dynamic CoA Discovery	Specifies the handover protocol to use when not using a mobility script: • Normal • Normal with HCS • MI PreReg • TT PostReg • LABPR • Predictive Post

Table 5: *moip_core* Attributes

If the scripted mobility mode is selected, then the script is specified as an attribute of this process, along with some other parameters like script repetition count and pause time in between repetitions. When scripts are parsed here, all the future Layer2 handover events for this node are scheduled. If one of the dynamic methods is selected (explained later in this section), this process receives triggers as they occur, usually from the *power_monitor* process. Handling of triggers scheduled by a script or received dynamically is identical. Its most important attributes are listed in Table 1.

Closely related to *moip_core* is *power_monitor*, shown in Figure 11 and is only used when the node is not using a mobility script (i.e. configured for Dynamic Mobility). It makes handover decisions depending on whether the distance or power level mode for mobility is used.

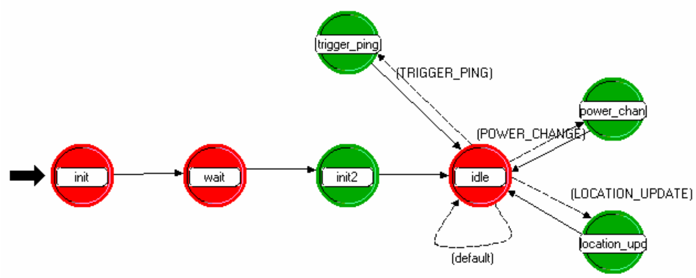

Figure 11: *power_monitor* Process

When the dynamic mobility mode is set to distance-based, this process registers itself for position updates through a call to *op_ima_obj_pos_notification_register*. Whenever an AP becomes closer than the current one, all four handover events to the new AP are scheduled at *moip_core*: *AT*, *oFA L2LD*, *nFA L2LU* and *MN L2LU*. There is also support for GPS functionality required for correct operation of the LABPR protocol. It is able to extract a list of APs the surrounding MN is going to be visiting based upon either a Trajectory defined in the Network Editor or, when using street mobility, the remaining path to the random waypoint or target flag. The *trigger_ping* state is only used in the LABPR scheme and is used to initiate an RTT timing sequence between the MN and the GFA.

A second mode for the mobile to conduct dynamic handovers is through power level monitoring. This process was originally connected to each wireless transceiver through statwires to monitor the received power level. Since OPNET Release 10.0 however, the concept of WLAN Beacons has been introduced and is used to evaluate APs on their reliability. There are basically two reasons why an AP's reliability figure can be lowered: received power level is too low for the receiver to achieve a lock on the signal or interference from neighboring senders is too high (yielding high bit error rates). Either way, after a few consecutive bad Beacon receptions, the reliability figure drops below a certain threshold and the scanning phase is initiated. When the *wlan_mac* process discovers through the process-wide registry that it is running in an MN, all AP reliability calculation logic is redirected to the *power_monitor* process. So each time the function *wlan_ap_reliability_eval* (defined in *wlan_support.ex.c*) would have been called, it is replaced by another function call which schedules remote interrupts that are filtered so that each interface only forwards events for Beacons received in the BSS they are configured for.

AP reliability evaluation in *power_monitor* is conducted largely similar to how it is done in the standard WLAN process. After a certain number of negative Beacon receptions, a handover is initiated and a new AP is discovered depending on the state of the *Realistic Scanning* attribute. If this is disabled, then the propagation modeling routines are called to determine which station has the strongest signal strength (taking into account sender power levels, frequency and location), after which the four handover events mentioned before are scheduled to *moip_core*. If *Realistic Scanning* is enabled however, only the *oFA L2LD* handover event is scheduled (*AT* does not exist as we have only now realize we're doing a handover and the new AP has yet to be discovered) and a scanning phase is entered. A user definable number of correct consecutive Beacons need to be received for an AP to become eligible for a handover. Because we have one interface tied to every BSS a node can roam to, the scanning phase doesn't need to spend a certain amount of time in each channel waiting for Beacon receptions. Therefore, the attribute *Min Time for 802.11 Re-Association* is used (which can be set to any available distribution) to model the Layer2 gap. After the required number of correct Beacons is received by any interface and the minimum Layer2 handoff time has expired, a new AP is elected and the remaining two handover events (*nFA L2LU* and *MN L2LU*) are scheduled at *moip_core*. Do note that when using this handover mode, deployment of Low Latency schemes or LABPR is not possible (no *AT* event).

The attributes that control most of the Layer2 timing events are listed in Table 6.

Attribute Name	**Comments**
Min Time for L2 Re-Assoc	Minimal L2 scan time.
Nr Beacons for L2 Re-Assoc	Number of correct beacon receptions before AP is considered reliable.
Offset MN LU from nFA LU	Time after *nFA L2LU* when MN should initiate MIP registration.
Offset AT to oFA LD	Time between *AT* and *oFA L2LD*. Only usable when not using realistic scanning.

Table 6: *power_monitor* Layer2 Attributes

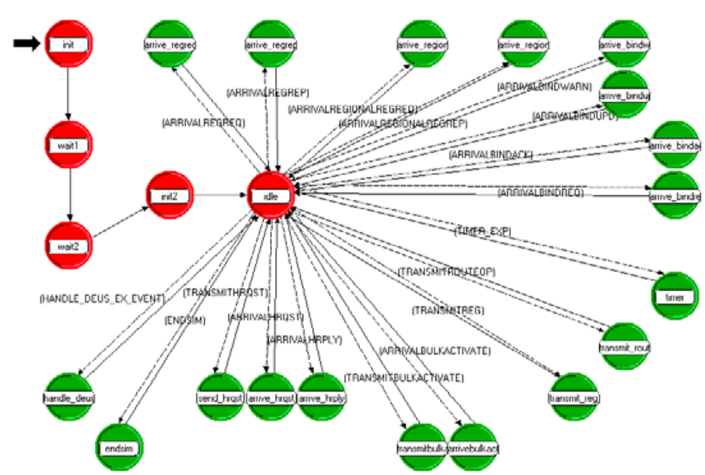

Figure 12: *moip_reg* Process

The *moip_reg* process shown in Figure 12 contains the Predictive Post handover logic. It is a reactive process where control returns to an idle state after each packet reception or remote trigger. When a Registration Request is received here by an FA, the PrevFA extension is removed and a Binding Update is sent to the oFA. At this time, the first part of the mobility table entry can be formed. The oFA which received the Binding Update can complete its mobility table entry and update the Count and Visit Time values. The tunnel set-up messages are also handled here and transmission of a HRqst message (in *send_hrqst* state) is triggered by arrival of a Registration Request and subsequent next-hop prediction. The parameters of importance in *moip_reg* for Predictive Post are listed in Table 7. Note that only FAs need to have these parameters defined. The MN only has to have the Smooth Handover extension enabled.

Attribute Name	**Comments**
Correction Term for Visit Time Prediction	To encourage optimistic predictions
Max # of Likely Next Hops	To increase prediction accuracy
Number of Visit Times for Prediction	Defaults to 10
Visit Time Prediction Mode	Avg, Min or Imm

Table 7: *moip_reg* FA Configuration Attributes

Predictive Post: Simulation Results

We chose a Manhattan style scenario to conduct our experiments in, where all nodes will have the same movement behavior (i.e. due to scenario symmetry, we can focus on one node and assume all other nodes will show similar results).

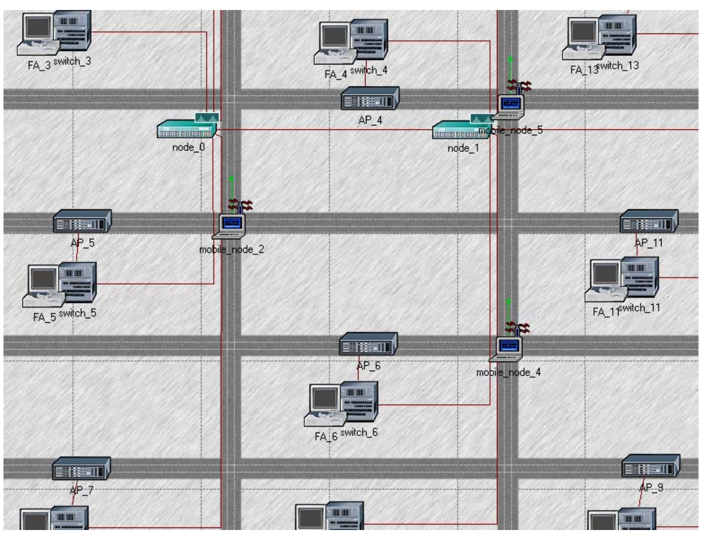

Figure 13: Reference Network

Part of this network is displayed in Figure 13: city blocks are 250m by 100m with 20m-wide one-way streets in between (the total scenario is 5 x 7 blocks). APs are placed in the middle of street blocks at a height of 7m. As FAs are modeled as regular hosts, every FA is connected with an AP through an Ethernet switch which is in turn connected to a router. MNs are placed randomly in the network, at a height of 1.5m. The Har-Xia-Bertoni propagation model is used, with environment type set to *High-Rise* and average building height set to 40m. The movement mode of each node is set to *Random Next Hop* with a direction bias of 1 or 2: they will have a slight tendency to continue in the direction they are on, but still provide enough randomness by making turns. Traffic flows from a CN to one MN in the form of UDP packets sent every 50ms.

Every time an MN associates itself with a new AP, an *nFA L2LU* trigger is received at the nFA, which checks if it has a buffer enabled for the registering node (i.e. if a HRqst has been received from the oFA). If so, a value of 1 is written to the prediction accuracy statistic and a 0 otherwise. The graph of Figure 14 is generated by taking a moving average filter with a size of 250 over the results. When setting up a tunnel only to the most popular one, we get an accuracy of about 60%. Increasing the number of predictions to 3 attains almost perfect prediction, with the possible cost of added overhead of course. If the tunnel to the FA the MN is effectively moving to is activated first, then there is practically no overhead as the remaining pending tunnels are destroyed when the oFA receives the Binding Update message. Generally this will not be the case however, but it should be noted that any overhead caused by tunneling to incorrect next-hop FAs is local and will be limited to the access network only.

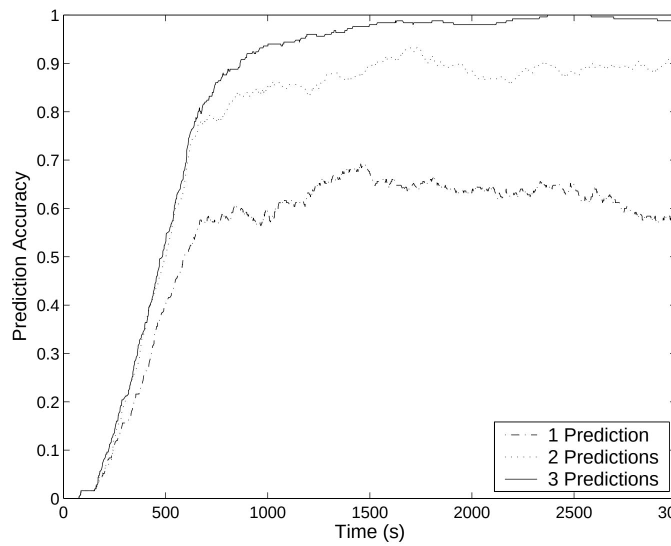

Figure 14: Prediction Accuracy

The time spent under an FA's control, as reported by a MN, is basically a result of 2 factors:

1. MN speed: street traversal and possible pause time;
2. MN trajectory: taking a turn can cause a MN to switch sooner than it would have if it went straight on.

If we use for example the last 10 recorded visit times to base our prediction on, we can take a number of approaches in deciding when to activate the tunnel:

- Average visit time (*Avg*);
- Minimum visit time (*Min*);
- Immediate activation (*Imm*).

Figure 15 shows the number of packets lost using each method and compared to packet losses when using basic MIP. MNs move randomly in the network for 2000secs at speeds drawn from *uniform(7,15)* m/s. We do assume high next-hop prediction accuracy to illustrate handover smoothness in optimal conditions. When the MNs don't have any significant pause times, the total number of packets lost using the average estimator is less than half than the losses observed when using plain MIP. This is reduced further when using the minimum estimator and packet loss is eliminated when the tunnel is immediately activated, again at a cost of extra overhead both in network load and buffering at the nFA. These buffering requirements are obviously linearly related with the activation timing: the earlier the prediction, the higher the buffer build-up at the nFA will be.

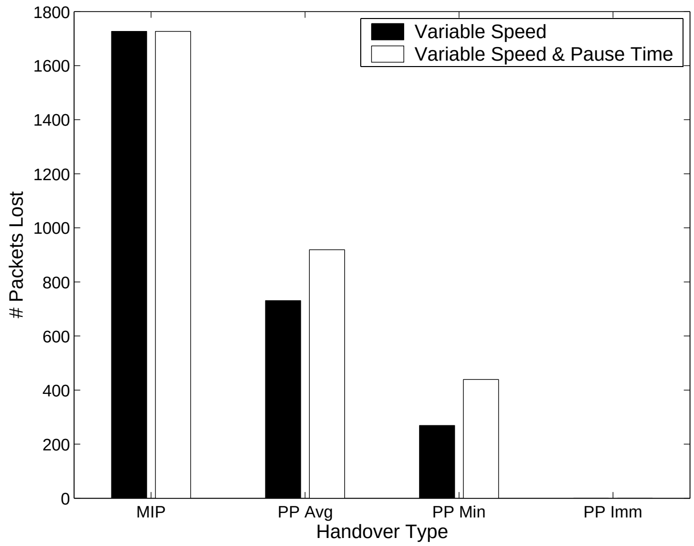

Figure 15: Packet Losses

While the *Avg* and *Min* predictions will react to temporal changes in node behavior (e.g. rush-hour conditions vs. light traffic), as only the last *x* number of visit times are used, they can never provide totally smooth handover conditions. MNs can stop unexpectedly for an unknown amount of time: its tunnel to the next predicted FA will be initiated, but what about traffic arriving at the oFA to which it is still connected? The only answer here is to bi-cast traffic by forwarding it in the tunnel and still sending it on the local link also. This has an associated overhead, but can be limited by placing a timeout on the tunnel so that it will be destroyed if the MN does not make its handover in the allocated timeframe. Unusually long or short visit times will also be taken into account in the visit time prediction, unless these outliers are removed by some means. Figure 15 also shows the packet loss observed over the same amount of handovers as before, but now a short uniformly distributed waiting time is added at each junction (according to *uniform(0,40)*). This extra level of randomness brings down performance somewhat, but it is still noticeably better than standard MIP.

For the following discussion, we assume that all nodes move at roughly the same speed, i.e. at a given FA, every node will report approximately the same visit time. Two different handovers were commonly observed in the reference network: those triggered by sudden drops in power reception or rises of interference level and those triggered by a gradual decline of signal level (area of insufficient coverage). The first kind is illustrated in Figure 16. Even though the timing prediction is now very accurate, a small number of packets are lost because it takes the MN a number of consecutive bad Beacon receptions before the scanning phase is initiated. This type of handover is the most common one and can be improved by adjusting the tunnel activation timer by a small offset. When looking at the second type of handover in Figure 17 however, performance is worse. Even though application traffic cannot be delivered, the scanning phase is not initiated as some Beacons do get through (as they are very short packets) and cause the AP's reliability to be upgraded a few times. The buffered packets that are eventually delivered are those that would have been lost during the scanning phase, which is about 500ms.

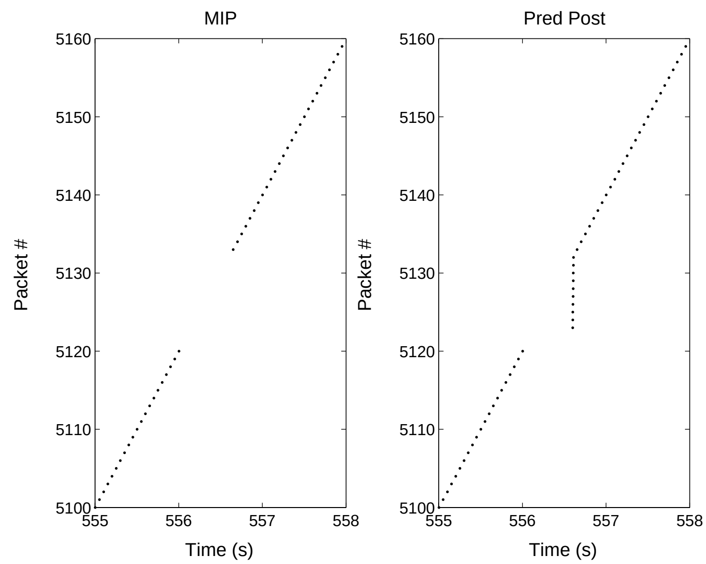

Figure 16: Sudden Handover

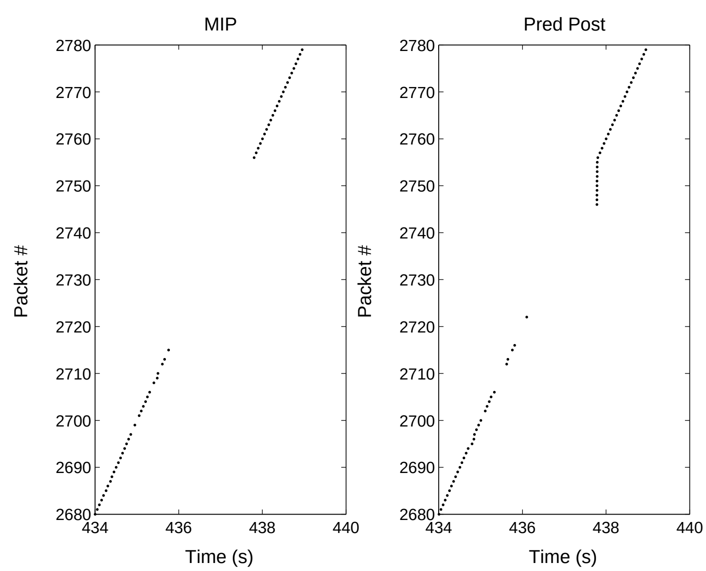

Figure 17: Gradual Handover

Optimal handovers in both cases are illustrated in Figure 18. These were conducted with optimistic prediction times and duplicate packet elimination. Packets can arrive at the MN out-of-sequence and this can have detrimental effects on application behavior. When TCP is used as transport protocol, some more advanced flavor should be used to maintain troughput under these conditions.

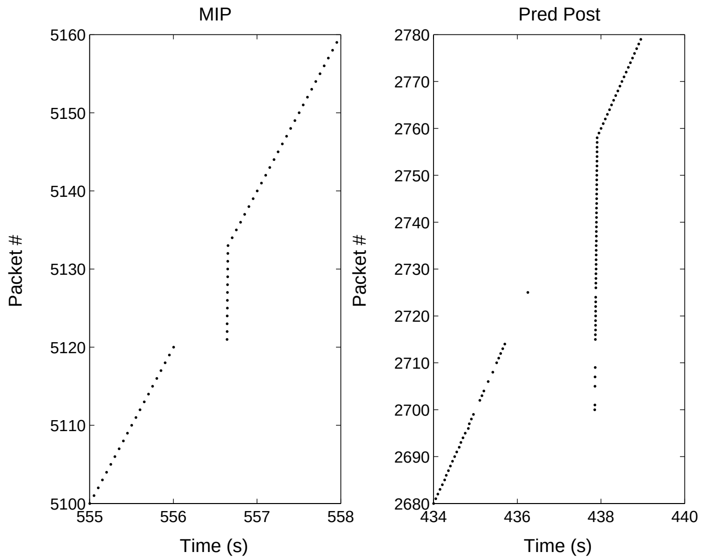

Figure 18: Optimal Handover

Conclusion & Future Work

This paper introduced a new mobility model based on street data. Different modes for MNs to move about are supported to provide realistic movement patterns. To further enhance simulation realism, two new radio wave path loss models were implemented in the wireless pipeline stages.

A new network-initiated MIP handover scheme was also introduced which learns node mobility patterns through standard Route Optimization messaging. Next-hop prediction was shown to be quite accurate in the scenario given, but tunnel timing remains somewhat of an issue. Emergency services or other priority users can make use of immediate tunnel activation, yielding the best results but with higher network load. Application bandwidth could also dictate the method to use in order to keep overhead down. Low bandwidth applications can be given extra performance by increasing the number of next-hop predictions and/or making tunnel activation times more optimistic. Another good application for this protocol is in fixed-rail scenarios where all nodes have very similar speed characteristics and GPS-enabled protocols like LABPR are not possible (e.g. subway systems).

Future work includes deploying this protocol on a larger scale in OPNET and running a wide variety of applications on top. Further research also needs to be done to make the tunnel activation estimator more robust.

References

[1] C. Blondia, O. Casals, N. Van den Wijngaert, G. Willems, “Performance analysis of smooth handoff in Mobile IP”, Proceedings of The Fifth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM2002), Atlanta, USA, September 2002, p.22-29.

[2] C. Blondia, O. Casals, Ll. Cerda, N. Van den Wijngaert, G. Willems, P. De Cleyn, “Performance Comparison of Low Latency Mobile IP schemes”, Proceedings of WiOpt’03, INRIA Sophia Antipolis, France 2003, pp. 115-124.

[3] C. Blondia, O. Casals, Ll. Cerdà, N. Van den Wijngaert, G. Willems, P. De Cleyn, “Low Latency Handoff Mechanisms and their implementation in an IEEE 802.11 Network”. Proceedings of ITC 2003, Berlin, Germany 2003, pp. 971-980.

[4] COST 231 Final Report, “Digital Mobile Radio Towards Future Generation Systems”.

[5] O. Casals, Ll. Cerdà, G. Willems, C. Blondia, N. Van den Wijngaert, “Performance evaluation of the Post-Registration method, a low latency handoff in MIPv4”, Proceedings of ICC 2003.

[6] A.T. Campbell et al, “Cellular IP”, draft-ietf-mobileip-cellular-00-txt, December 1999.

[7] K. El Malki and others, “Low Latency Handoffs in Mobile IPv4”, IETF draft-ietf-monileip-lowlatency-handoffs-v4-04.txt, 2002.

[8] N. A. Fikouras, C. Görg, “Performance Comparison of Hinted and Advertisement Based Movement Detection Methdos for Mobile IP Hand-offs”, Proceedings of European Wireless 2000.

[9] E. Gustafsson, A. Jonsson, C. Perkins, “Mobile IP Regional Registration”, draft-ietf-mobileip-reg-tunnel-06.txt, March 2002.

[10] D. Har, H. Xia and L Bertoni. “Path-Loss Prediction Model for Microcells”, IEEE Transactions on Vehicular Technology, Vol. 48, No. 5, 1999, pp. 1453-1462.

[11] A. Jardosh, E. Royer, et al, “Towards Realistic Mobility Models in Mobile Ad hoc Networks”, Proceedings of The Ninth Annual International Conference on Mobile Computing and Networking (MobiCom 2003), San Diego, USA 2003, pp. 217-229.

[12] A. Mishra, M. Shin and W. Arbaugh, “An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process”. Technical report number CS-TR-4395, University of Maryland, Department of Computer Science, September 2002.

[13] M. Ohta, “Smooth Handover over IEEE 802.11 Wireless LAN”, IETF draft-ohta-smooth-handover-wlan-00.txt, 2002, June 2002.

[14] OPNET Modeler, www.opnet.com.

[15] C.E. Perkins. “IP Mobility Support”, rfc 2002, October 1996.

[16] C. Perkins and D. Johnson, “Route Optimization in Mobile IP”, IETF Internet Draft, draft-ietf-mobileip-optim-11.txt.

[17] C. Perkins and K-Y. Wang. “Optimized smooth handoffs in Mobile IP”, Proceedings of IEEE Symposium on Computers and Communications, Egypt, July ’99.

[18] R. Ramjee, et al, “IP micro-mobility support using HAWAII”, draft-ietf-mobileip-hawaii-00.txt, June 1999.

[19] ANSI/IEEE Std 802.11, 1999 Edition.

[20] N. Van den Wijngaert and C. Blondia, “Integration of IP mobility in OPNET: modeling and simulation”, Proceedings of OPNETWORK 2002, Washington D.C., August 2002.

[21] N. Van den Wijngaert and C. Blondia, “A Location Augmented Low Latency Handoff Scheme for Mobile IP: OPNET Modeling and Simulation”, Proceedings of OPNETWORK 2003, Washington D.C., August 2003.

[22] N. Van den Wijngaert and C. Blondia, “A Location Augmented Low Latency Handoff Scheme for Mobile IP”, Proceedings of ICMU2004, Yokosuka, Japan 2004.