Developing Tools for Parsing ns-2 Results: IEEE802.21 Case Scenario

Hugo Marques[1], José Ribeiro, Paulo Marques, and Jonathan Rodriguez[2]

[1] Instituto Politécnico de Castelo Branco
Escola Superior de Tecnologia
Av. do Empresário, Castelo Branco
hugo@ipcb.pt
[2] Instituto de Telecomunicações,
Campus de Santiago, Aveiro
jcarlosvgr@gmail.com, {pmarques,jonathan}@av.it.pt

Abstract. Recent work by the United States National Institute of Standards and Technology produced new add-on modules for the popular network simulator tool, ns-2, that allow users to simulate vertical (heterogeneous) handover scenarios in a network topology consisted by different radio access technologies such as WiMAX, Wi-Fi and UMTS. The developed Media Independent Handover functionality was based on draft 3 of the recent IEEE 802.21 standard. By using this modified version of ns-2, this paper presents two new tools to parse the resulting ns-2 tracing files and provide a graphical view of simulated scenarios and common metrics for evaluating the handover performance.

Keywords: 802.21, handover, heterogeneous networks, MIH, ns-2, simulation.

1 Introduction

Considering the coverage provided by all IEEE 802 networks (wireless and wired) available today, it seemed rather intuitive that somehow we should be able to switch between these technologies while maintaining a seamless connection. This is the main scope of the recent IEEE 802.21 standard [1].

It all started in November 2002 in a Handover Tutorial [2] organized by Roger Marks and Brian Kiernan where there was a discussion about a common handoff approach for IEEE 802 networks. An IETF call for interest took place in the beginning of 2003 and in 2004 a Working Group (802.21) was formed. In the following years the IEEE 802.21 WG had to solve the problem on how to seamlessly interconnect a wide variety of non-interoperable heterogeneous networks. Key obstacles to this task were [3]: (i) the lack of scalability due to limited support by operators; (ii) lack of standard handover interfaces; (iii) limited quality of service (QOS) guarantees during handover and; (iv) level of supported security when roaming across different access networks. Security was considered to be a hard task so it was not included in the IEEE 802.21 standard; it is being currently dealt by the IEEE 802.21 Security Task Group (802.21a) [3].

J. Rodriguez, R. Tafazolli, C. Verikoukis (Eds.): MOBIMEDIA 2010, LNICST 77, pp. 347–358, 2012.

To cope with these issues the Advanced Network Technologies Division (ANTD) of the National Institute of Standards and Technology (NIST)[1], developed a Media Independent Handover (MIH) implementation for the network simulator, ns-2 [4], in the scope of the Seamless Mobility project [5]. A thorough description of their implementation work can be found on the project website and also on [6-11].

Since the release of the mobility package by NIST, as part of their joint work with IEEE 802.21 and the IETF, numerous research studies have used these modules. [11] evaluates the performance of an adaptive channel scanning algorithm for IEEE 802.16 using a previous version of the WiMAX module that is used in this article. In [12] the handover latency for the cases where UDP and TCP carry MIH signaling messages is compared and some of the design tradeoffs are presented. The evaluation of the performance of a vertical handoff scheme between 802.11 and 802.16 wireless access networks with respect to signaling cost, handoff delay and QoS support can be found in [13]. [14] evaluates a proposed cross-layer mechanism for making intelligent handover decisions in FMIPv6 in terms of handover latency, packet loss and handover signaling overhead and [15] evaluates a new enhanced Media Independent Handover Framework (eMIHF) that allows for efficient provisioning and activation of QoS resources in the target radio access technology during the handover preparation phase.

The use of ns-2 with NIST modules has also been used to propose new implementation guidelines to the new security extension for IEEE 802.21. In [16] different authentication techniques are compared, namely re-authentication and pre-authentication, that may be used in order to reduce the time and resources required to perform a vertical handover. [17] measures the performance of the authentication process in media independent handovers and considers the impact of using IEEE 802.21 link triggers to achieve seamless mobility and [18] proposes an extension to current network selection algorithms that takes into account security parameters and policies, and compares the handover performance with and without the proposed extension.

Considering the above mentioned research studies, and others not referred here, the authors feel the need to develop new tools to parse the resulting tracing files of a ns-2 simulated scenario in order to provide: (i) a more clear view of the simulated scenarios; (ii) metrics to evaluate the vertical handover performance and (iii) a better explanation of IEEE 802.21 standard.

The remainder of this paper is organized as follows: section 2 makes a quick overview of *ns-2* output tracing data; Section 3 presents the new implemented tools and finally, Section 4 presents conclusions and future work.

2 Quick Overview on ns-2 Tracing Data

After running a simulation, ns-2 outputs the network triggered events to an ASCII trace file. In order to have network tracing events we need a link, and that is *where*

---

[1] NIST is a non-regulatory federal agency within the U.S. Department of Commerce. NIST's mission is to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology.

ns-2 interposes the trace elements, see Fig. 1. The following are trace elements used by *ns-2* when monitoring a link:

- enqT_ - stores information of each packet that arrives and is queued at the link;
- deqT_ - stores information for every packet dequed (this packet will not be dropped);
- dropT_ - stores information for every packet that cannot be queued due to overflow. This packet is dropped;
- rcvT_ - stores information for every packet received at the end of the link.

The other elements define the link, see [19] for a better understanding of these variables.

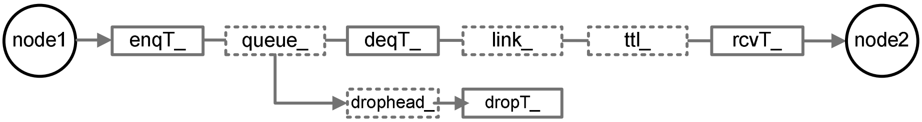

Fig. 1. Tracing objects in a simplex link [19]

When using the command *trace-all*, all traceable simulation events are written in the output file, which can generate huge files. A workaround for this is to activate only a subset or a specific trace element. Nevertheless, sometimes the *trace*-all is the only option to troubleshoot a problem, correlate different events in time or simply because the data is required for result analysis and statistics. Additionally, besides the normal tracing data, users can also write debug information in a file if extra information, such as events not related to packets, is needed. To handle with this amount of ASCII information, a parsing operation, using scripting language such as *bash* or PERL, is needed. With parsing, users can reorganize and filter data for post processing, such as execute mathematical operations.

A more powerful way to collect network information is through queue monitoring, it allows tracking packet arrival/departure/drop statistics, and may optionally compute averages of these values. Detailed information on how to use and configure queue monitoring in ns-2 can be found in [19].

The current structure of *ns-2* trace files is different from earlier versions of the simulator (i.e., ns v1), and for wireless links this structure is still being updated. As an example, a sample output for an *ns-2* trace file is shown in Fig. 3.

Trace event	Time	From node	To node	Packet type	Packet size	Flags (IP &TCP)	Flow identifier	Source address	Dest. address	Sequence number	Packet ID
Trace element that triggered this trace: + (enqT_) - (deqT_) d (dropT_) r (rcv_T)	Time in seconds, at which the trace was triggered	ns-2 node ID for the node who sents this packet	ns-2 node ID for the next hop for this packet	Type of packet: ack cbr tcp	Size of packet in bytes, as encoded in its IP header	Flags for IP and TCP flow control	Identifies the specific flow to which the packet belongs	The packet's source address (using ns-2 addressing scheme)	The packet's destination address (using ns-2 addressing scheme)	The IP sequence number for this packet	The unique identification for this packet in simulation

Fig. 2. ns-2 trace file format for wired links

The first 5 lines represent the traces for wired links, and the last two are for wireless links. As we can see, wired links have a different output structure than wireless links. For wired links the output contains 12 fields distributed according to Fig. 2.

```
- 10.000004 0 1 cbr 1240 ------- 0 0.0.0.3 2.0.4.0 0 23
r 10.000004 0 1 cbr 1240 ------- 0 0.0.0.2 2.0.3.0 0 22
+ 10.000004 1 17 cbr 1240 ------- 0 0.0.0.2 2.0.3.0 0 22
- 10.000004 1 17 cbr 1240 ------- 0 0.0.0.2 2.0.3.0 0 22
r 10.000004 1 17 cbr 1240 ------- 0 0.0.0.1 2.0.2.0 0 21
(...)
r -t 10.000004170 -Hs 17 -Hd -2 -Ni 17 -Nx 550.00 -Ny 550.00 -Nz 0.00 -Ne -
1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.1 -Id 8388610.0 -It cbr
-Il 1240 -If 0 -Ii 21 -Iv 30 -Pn cbr -Pi 0 -Pf 0 -Po 0
f -t 10.000004170 -Hs 17 -Hd 8388610 -Ni 17 -Nx 550.00 -Ny 550.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.1 -Id 8388610.0 -It
cbr -Il 1240 -If 0 -Ii 21 -Iv 29 -Pn cbr -Pi 0 -Pf 0 -Po 0
```

Fig. 3. Sample of a *ns-2* output trace file

For wireless links, a new improved trace format has been introduced in the latest versions of *ns-2*. Refer to section 16.1.6 in “The NS Manual” [19], for details on this trace format.

3 Developed Tools

This section describes two tools developed in the scope of the ICT-HURRICANE [20, 21], a Specific Targeted Research Project (STREP) supported by the European 7th Framework Programme. Both tools were developed in MATLAB®, a powerful tool for mathematics with a large choice of graphics and also capable of providing a friendly graphic user interface (GUI). The tools can ran in any operating system with support for MATLAB®.

3.1 Intersectus

Intersectus is a tool that allows users to see the coverage area of the radio access technologies (RAT) and the MNs’ trajectories in a simulated scenario. A screenshot of the tool is depicted in Fig. 4. With this tool, users can:

- Randomly distribute MNs in the topology;
- Plot the trajectory for every MN in the simulated scenario;
- Determine the theoretical number of handovers in the simulation;
- Determine the number of failed handovers;
- Measure the percentage of successful handovers;
- Measure the overall system packet loss;
- Measure HO delay for each MN and for each RAT

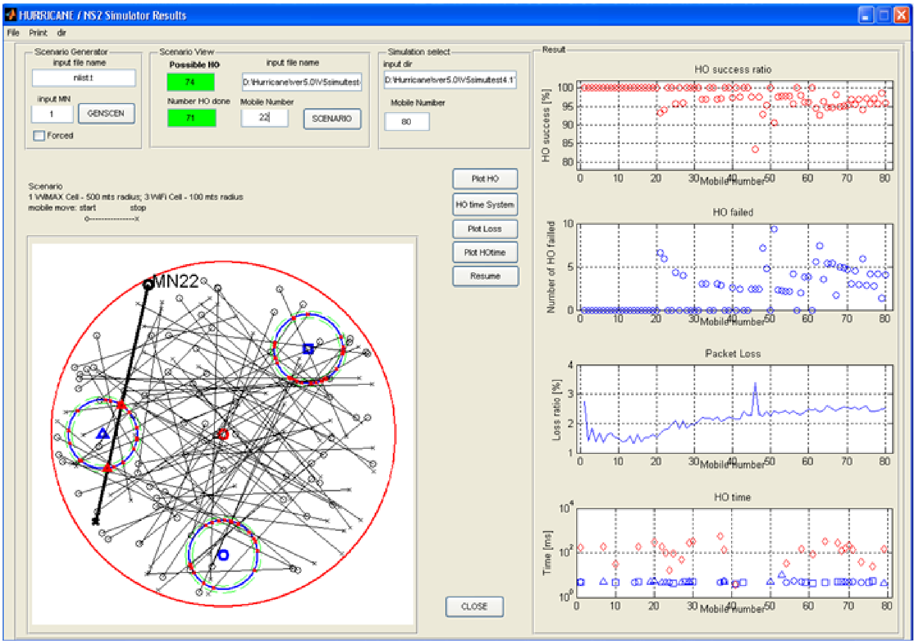

Fig. 4. The *Intersectus* tool

When simulating IEEE 802.21 scenarios in *ns-2* with variable number of nodes and random start/end positions, it is important to determine the number of handovers that should be triggered by MNs. Comparison between this value and the effective number of handovers that were produced in *ns-2* simulation is important because it is an indication of the handover success rate as the number of mobile nodes (MNs) increase in the simulation. Since *ns-2* does not provide this information, a new innovative way of determining such a value is proposed. A handover happens when a MN crosses the coverage boundary of a RAT; this is independent of start and end positions of the MN. The code for the intersection function, that determines if an intersection between a line and a circle exists, is as follows:

```
function [Ixa,Iya,Ixb,Iyb,real]=intersect(x1,y1,x2,y2,a,b,r)
    m=(y2-y1)/(x2-x1);
    K=a^2 + (y1-b)^2-2*m*(y1-b)*x1+m^2*x1^2-r^2;
    A=1+m^2;
    B=2*m*(y1-b)-2*a-2*x1*m^2;
    C=K;
    Ixa=(-B+sqrt(B^2-4*A*C))/(2*A);
    Ixb=(-B-sqrt(B^2-4*A*C))/(2*A);
    Iya=y1+m*(Ixa-x1);
    Iyb=y1+m*(Ixb-x1);
    real=0;
    if(((B^2-4*A*C)<=0))
        real=1;
    end
```

Using the scenario generator (GENSCEN button in Fig. 4) the user can randomly distribute MNs in the topology, effectively creating new simulation scenarios and then obtain the trajectories of each simulated MN. The total number of intersections between the MNs trajectories and the RATs boundaries gives the theoretic number of handovers in the simulation. The user can also load a previous scenario by inputting a file name in the scenario generator.

The trajectories for 80 MNs triggering a total of 74 handovers can be seen in Fig. 5; the big circle corresponds to a WiMAX coverage area, the three smaller circles correspond to three Wi-Fi cells coverage and the lines correspond to MN trajectories. Each line starts with a small circle (MN start position) and ends with a cross (MN end position). The intersections between lines and the three Wi-Fi circles are indicated by small triangles. Each Wi-Fi cell has three doted circles representing the power boundaries described in [9].

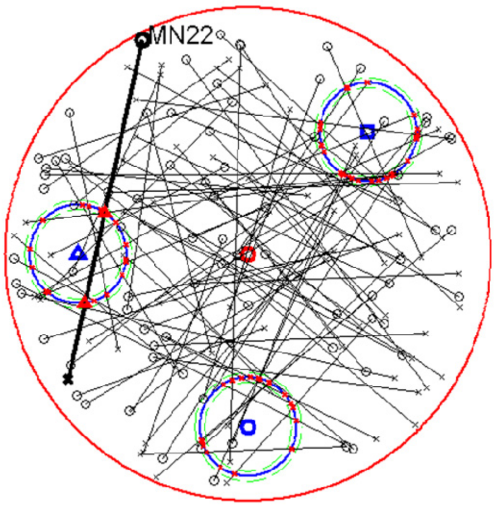

Fig. 5. Determining handovers in a simulated scenario by using an intersection function

In the ‘Scenario View’ we can see the number of handovers triggered by a specific scenario, the number of failed handovers, see [6], and also highlight a specific MN in order to easily indentify it in the topology.

In ‘Simulation Selection’ the user inputs the location where the *ns-2* tracing files were stored. Upon loading the correspondent files, *Intersectus* plots 4 graphics:

- Handover success ratio – shows the percentage of successful handovers for each increase in the number of MNs.
- Number of failed handovers – shows the number of failed handovers for each increase in the MNs.
- System packet loss – shows the packet loss (in packets) for the entire topology, includes packet loss both in WiMax and Wi-Fi.
- Handover delay – shows the handover delay for each MN that triggered an handover. It also identifies to which RAT cell the handover (or handovers) were executed.

The files needed to run the tool as well as their interdependencies are depicted in Fig. 6. The *bash* file is a script that passes initial parameters to the OTcl simulation file, and filters trace outputs in order to do handover and packet loss calculations. The results for this operation are saved in multiple files. The *nlistMN.t* and *nHOtmMN.tm* files are generated for each MN, for example if a simulation starts with one MN and ends with ten, there are ten *nlistMN.t* and *nHOtmMN.tm* files. A sample for *nlistMN.t* and *nHOtmMN.tm* can be seen in Fig. 7 and Fig. 8.

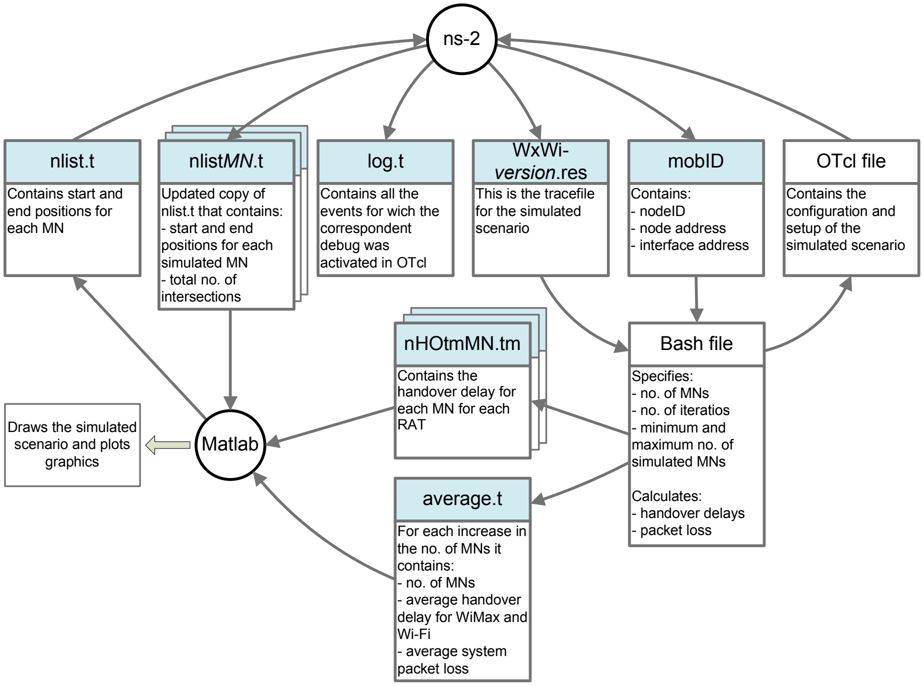

Fig. 6. Files (shade) needed to operate *Intersectus* and their interdependencies

0	941	300	429	676
1	229	536	224	92
2	700.0	90.0	300	90
3	2	2	0	0

Fig. 7. Sample content for *nlistMN.t*

1	0.0049	0	0.00472	0.0238
2	0	0	0	0
3	0	0	0	0.0231
4	0	0	0	0
5	0	0	0	0
1	0.0049	0	0.00472	0.0238
2	0	0	0	0
3	0	0	0	0.0231
4	0	0	0	0
5	0	0	0	0
5	2	0	0	0

Fig. 8. Sample content for *nHOtmMN.tm*

In *nlistMN.t* there are 3 MNs (0 to 2). The first three lines identify the MNs and their start and end positions in the simulation. The last line is included by *ns-2* and contains the number of simulated MNs, the number of intersections, the number of successful handovers and the number of failed handovers, the last column isn't used.

The file *nHOtmMN.tm* contains information regarding 5 MNs and 2 simulation iterations. With the exception of the last line, the first column identifies the MN and the following columns indicate the handover delay for the three Wi-Fi cells and WiMAX cell, respectively. The last line includes the number of simulated MNs and the number of iterations, columns 3 to 5 are not used.

The *average.t* file, see Fig. 9, contains the results for average handover delay and average packet loss when the number of MNs in the simulation increases by increments of 10. The first column identifies the MN, second and third columns contain the average handover delay to Wi-Fi and WiMAX respectively and columns 4 to 6 contain the average packet loss for the entire system, for Wi-Fi and WiMAX respectively.

```
...
71 143.681 459.675 2.65 1.05 1.6
81 156.048 544.68 3.69 1.03 2.66
91 189.154 605.222 4.52 1.52 3
101 214.714 645.895 4.5 1.5 3
```

Fig. 9. Sample content for *average.t*

3.2 Civitas

Civitas is an animation tool that allows users to see the movement of multiple MNs in a city as well as monitor relevant characteristics of the handover signaling and communication link. A screenshot of the tool is depicted in Fig. 10.

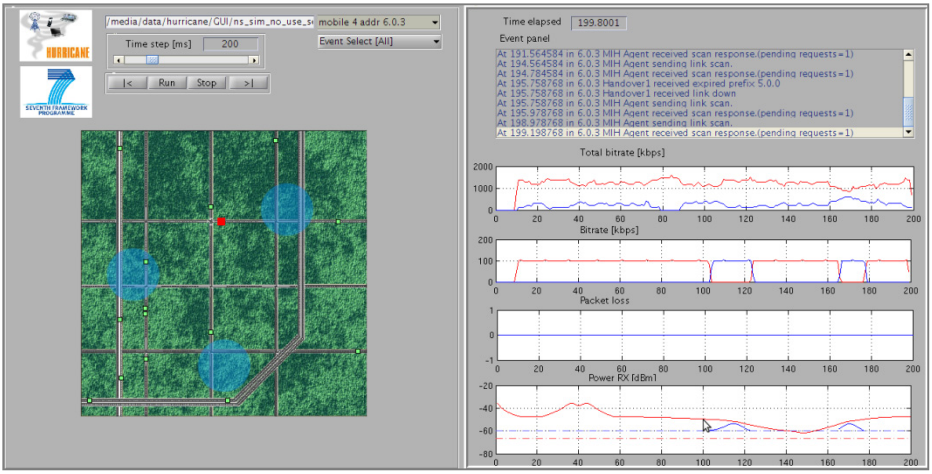

Fig. 10. The *Civitas* tool

With this tool, users can:

- See the movement of multiple MNs in a city scenario deployed with Manhattan and Highway mobility models for ns-2;
- Select a specific MN to monitor;
- Select the speed of the animation;
- Select which types of events to monitor (all, MIH, MIPv6, amongst others)
- Visualize in real time:
  - o the total throughput for each of the available RATs in the system;
  - o the throughput and packet loss for a specific MN in the topology
  - o the received power in each of the MNs' interfaces

The files needed to run the *Civitas* tool as well as their interdependencies are depicted in Fig. 11.

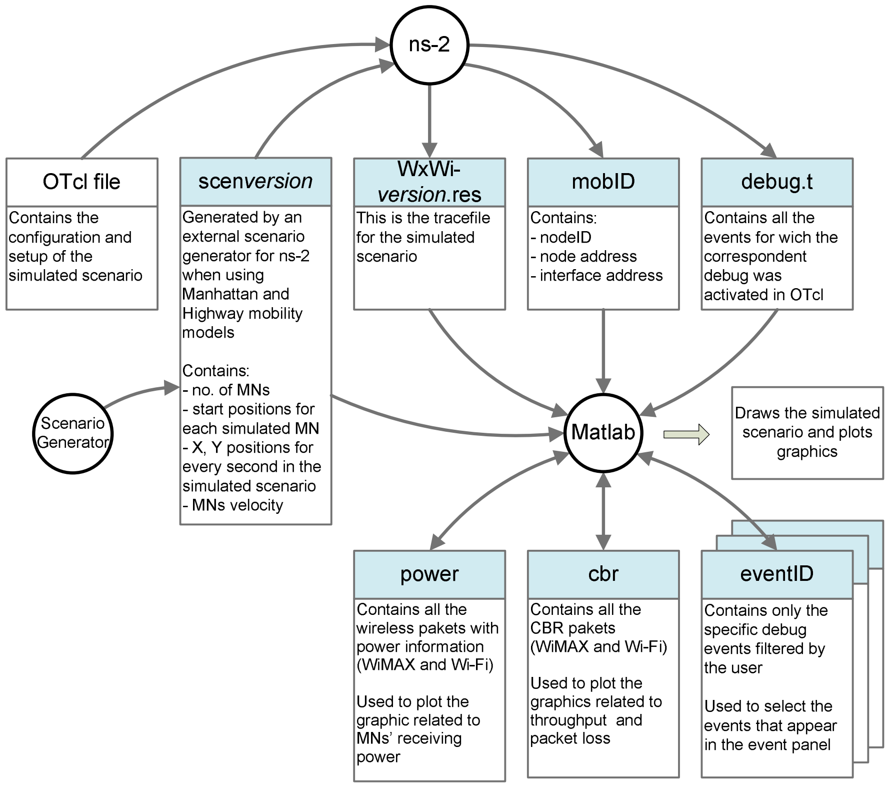

Fig. 11. Files (shade) needed to operate *Civitas* and their interdependencies

The topology used in *Civitas* is the same as the one used in *Intersectus*; three Wi-Fi cells (small circles) inside a bigger WiMax cell. As we can see in Fig. 10, there are 14 MNs (small squares), plus MN number 4 (bigger square) with address 6.0.3 that was chosen to be monitored. Simulation starts at 0s and by analyzing the WiMAX and Wi-Fi received power from the bottom right graphic, it is clear that this MN has coverage from WiMAX and at some point in its path it found a Wi-Fi cell (2 times). At approximately 9s the MN starts receiving 100Kbps constant bit rate (CBR) traffic as depicted in the second graphic (from top). At approx. 105s the MNs executes a handover to a Wi-Fi cell and stays there for approx. 20 seconds, when a handover to WiMAX is triggered. The MN makes another handover to Wi-Fi at approx. 165s, staying on this network only for 10s. The MN ends connected to the WiMAX cell.

From this description it's clear that the handover algorithm is not optimized, since the MN has triggered unnecessary handovers to Wi-Fi. It is also clear that the WiMAX interface was on a 'always on' state, which has power consumption implications. These however, are characteristics of the NIST modules. The *Civitas* tool only interprets the output for the simulation, and does not take any active action in it.

4 Conclusions and Future Work

ns-2 is a powerful tool to simulate wired and wireless networks. The use of C++ in the simulator code has the advantage of reducing packet and event processing time, allowing it to achieve fast execution times. The use of OTcl, besides being a compact and very powerful object programming language, allows a more intuitive interface with the user, when compared to C++.

Bugs in the ns-2 software are still being discovered and corrected; each user is responsible for verifying that his simulation is not invalidated by bugs. Additionally, due to its old code, output files are somewhat difficult to understand and users need to parse them in order to obtain the desired results. In order to better understand the simulation results, it is current practice to put 'debugs' and 'printfs' in ns-2 code. Nevertheless, ns-2 is considered a feasible simulator and as such, it's largely used by academic researchers.

We can say that ns-2 with NIST add-on modules proved to be a valuable tool to simulate IEEE 802.21 handover scenarios and better understand the basic signaling of IEEE 802.21 standard. NIST add-on modules however, only support part of the standard (based on draft 3) and were not conceived to simulate a high number of MNs as some adaptations in ns-2 were needed in order to run the simulations described in this paper. Nevertheless obtained results have proven an acceptable approximation to what could be expected in real case scenarios.

Considering that add-on modules such as WiMAX, Wi-Fi, UMTS, Bluetooth, FMIPv6, amongst others, are constantly being added and updated by the ns-2 community, the importance of ns-2 for 802.21 simulations becomes clear. Not only it will allow simulating complex vertical handover scenarios, prohibitive to do in a real testbed, but also allow modifying such modules in order to model missing primitives and better predict results for a real scenario. As shown by our previous experimental work [6, 22], both WiMAX and Wi-Fi modules need improvements in the scheduling mechanisms and the contention resolution/avoidance for the shared medium in order to better reflect the technology's real behavior.

Both *Intersectus* and *Civitas* tools presented in this paper use information obtained via trace elements of ns-2, however a more efficient and powerful way to collect network information is through queue monitoring. It allows tracking packet arrival/departure/drop statistics, and can be used to compute averages of these values. Users should be aware of this capability when developing new tools for parsing ns-2 output trace files.

Acknowledgments. The work presented in this article was supported by the European project ICT-HURRICANE and the Portuguese Foundation for Science and Technology (FCT) through project AGILE. The authors would like to acknowledge the contributions of their colleagues from the HURRICANE consortium.

References

1. IEEE P802.21-2008, Standard for Local and Metropolitan Area Networks: Media Independent Handover Services. IEEE (2009)
2. IEEE 802 Tutorial: Handoff Mechanisms and their Role in IEEE 802 Wireless Standards (2002), `http://ieee802.org/16/tutorial/T80216-02_03.pdf`
3. IEEE 802.21 Task Group a (TGa), `http://www.ieee802.org/21/TGa.html`
4. The Network simulator ns-2, `http://www.isi.edu/nsnam/ns`
5. Seamless Mobility Project, National Institute of Standards and Technology (NIST) (2007), `http://w3.antd.nist.gov/seamlessandsecure/`
6. Marques, H., Ribeiro, J., Marques, P., Rodriguez, J.: Simulation of 802.21 Handovers Using ns-2. Journal of Computer Systems, Networks, and Communications 2010, Article ID 794749, 11 pages (2010), doi:10.1155/2010/794749
7. The Network Simulator NS-2 NIST add-on - IEEE 802.21 model (based on IEEE P802.21/D03.00), National Institute of Standards and Technology (NIST) (2007)
8. The Network Simulator NS-2 - NIST add-on - IEEE 802.16 model (MAC+PHY), National Institute of Standards and Technology (NIST) (2009)
9. The Network Simulator NS-2 - NIST add-on - Mac 802.11, National Institute of Standards and Technology (NIST) (2007)
10. The Network Simulator NS-2 - NIST add-on – Neighbor discovery, National Institute of Standards and Technology (NIST) (2007)
11. Rouil, R., Golmie, N.: Adaptive Channel Scanning for IEEE 802.16e. In: 25th Annual Military Communications Conference (MILCOM 2006), Washington, D.C. (2006)
12. Griffith, D., Rouil, R., Golmie, N.: Performance Metrics for IEEE 802.21 Media Independent Handover (MIH) Signaling. In: Proceedings of the Springer Wireless Personal Communications Special Issue on "Resource and Mobility Management and Cross-Layer Design for the Support of Multimedia Services in Heterogeneous Emerging Wireless Networks" (2008)
13. Zhang, Y., Zhuang, W., Saleh, A.: Vertical handoff between 802.11 and 802.16 wireless access networks. In: IEEE Globecom 2008 (2008)
14. Mussabir, Q., et al.: Optimized FMIPv6 using IEEE 802.21 MIH Services in Vehicular Networks. IEEE Transactions on Vehicular Technology (2007)
15. Neves, P., Fontes, F., Sargento, S., Melo, M., Pentikousis, K.: Enhanced Media Independent Handover Framework. In: IEEE 69th Vehicular Technology Conference (VTC 2009-z Spring), Barcelona, Spain (2009)

16. Izquierdo, A., Golmie, N., Hoeper, K., Chen, L.: Using the EAP Framework for Fast Media Independent Handover Authentication. In: Wireless Internet Conference (WiCON), Maui (2008)
17. Izquierdo, A., Golmie, N., Rouil, R.: Optimizing Authentication in Media Independent Handovers Using IEEE 802.21. In: 18th International Conference on Computer Communications and Networks (2009)
18. Izquierdo, A., Golmie, N.: Improving security information gathering with IEEE 802.21 to optimize handover performance. In: 12th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Spain (2009)
19. Fall, K., Varadhan, K.: The ns Manual (2010), http://www.isi.edu/nsnam/ns/ns-documentation.html
20. ICT HURRICANE project: D4.2: Implementation of handover-enabling modules for network nodes (2009), http://www.ict-hurricane.eu/
21. ICT HURRICANE project: D.5.1a: Intermediate integration of HURRICANE handover framework (2010), http://www.ict-hurricane.eu/
22. Marques, H., Ribeiro, J., Marques, P., Rodriguez, J.: Measuring IEEE 802.21 Handover Performance in ns-2. In: 1st International Workshop on Mobility in Future Internet (MiFI 2010), International Conference on Telecommunications and Multimedia (TEMU 2010), Creete, Greece (2010)