

# A Multi-homing Extension of Wireless Node Implementation in NS-2

Qinghua Wang, Tingting Zhang

Department of Information Technology and Media

Mid Sweden University, 85170 Sundsvall, Sweden

Email: {qinghua.wang, tingting.zhang}@miun.se

**Abstract**—Due to the rapid growth of wireless communication, it is now believed that wireless devices with multiple interface support will highly increase in the near future. In the research community of Network Simulator, there has recently been quite a large interest in extending the current wireless node implementation so as to be able to include more than one radio interface to work simultaneously.

We focus our efforts on the extension of multi-homing ability of ns-2, which has been the most used network simulator in wireless network research. Multi-homing here is defined as the ability of a single end-point to support multiple IP addresses by the use of multiple interfaces. The ns-2 simulation environment is a flexible tool for network engineers to investigate how various protocols perform with different configurations and topologies. However, the support for wireless and mobile networking in ns-2 is monolithic and not entirely consistent with basic design principles so that it makes it quite difficult, if not impossible, to extend a multi-homing support based on current mobilenode design. This paper describes how we extend the ns-2 framework to include support for an imitated wireless multi-homed node.

**Index Terms**—Multi-homing, multiple interfaces, ns-2.

## I. INTRODUCTION

In recent years the tremendous success and demand met by wireless technologies, have fostered a considerable amount of research work ranging from propagation models, radio planning, power consumption, medium access, routing protocols to QoS and security issues. Topics such as mobile ad-hoc networks (MANETs) and wireless sensor networks for instance have been the focus of a lot of studies and papers during the last few years. In the MANETs and sensor network research community, a lot of studies have been based on the simulation. And among the network simulators, ns-2 [1] has been used in nearly half of them and become a de facto standard, if we refer to [6].

The object-oriented design of ns-2 allows for straightforward creation and use of new protocols. The combination of easy in protocol development and popularity has ensured that a high number of different protocols are publicly available, despite not all of which are included as part of the simulator's release. Its status as the most used MANETs simulator has also encouraged further popularity, as developers would prefer to compare their work to results from the same simulator.

However, ns-2 is currently not perfect in supporting MANETs simulation research. It does not support well for some important features. An example of such a feature needed

today is the multi-homing ability for a wireless node. A multi-homed node is defined as a node that enjoys the ability of supporting multiple IP addresses simultaneously, and this kind of ability is usually implemented by adding multiple interfaces, including wired and wireless interfaces, in a single node. The utilization of a multi-homed node can be versatile. For example, the feature of redundancy in interface numbers can be used to provide reliable transmission and increase the survivability of the session in the presence of network failures, such as what The Stream Control Transmission Protocol (SCTP) [3] is doing. Another important utilization is to use the multi-homed node as a communication switcher in a heterogenous network. A network can become heterogenous because of different reasons, such as the need of the cohabitation between a network with updated protocol and an old network, or network with nodes coming from different providers, or simply different type of networks needing to communicate. In this case, the multi-homed node must possess the ability in forwarding and translating packets among different types of connected networks. A sketch graph of one heterogenous network interconnected by multi-homed nodes is shown in Fig. 1.

In this paper, we extend the current ns-2 wireless node design to support the ability of multi-homing, so it can be used to perform the simulation of the heterogenous wireless network interconnection.

In the following of this paper, Section II introduces the ns-2 wireless node design and the existing efforts in extending its multi-homing ability. Section III further details the wireless node design in current ns-2 release, which is the basis of our multi-homing extension. Section IV then gives the design details of our multi-homing extension. An initial ns-2 implementation of the presented multi-homing extension is given in section V. Finally, the conclusion is given in section VI.

## II. RELATED WORK

Ns-2 is the most popular simulation tool for MANETs. It began as ns (Network Simulator) in 1989 with the purpose of general network simulation. Its wireless and mobile networking support extension is originally developed by CMU Monarch group [2] in the late 1990's and later integrated in the mainline code. The primary purpose of this extension was to enable simulation of wireless networks, in particular multihop ad hoc networks. The support of sensor network simulation

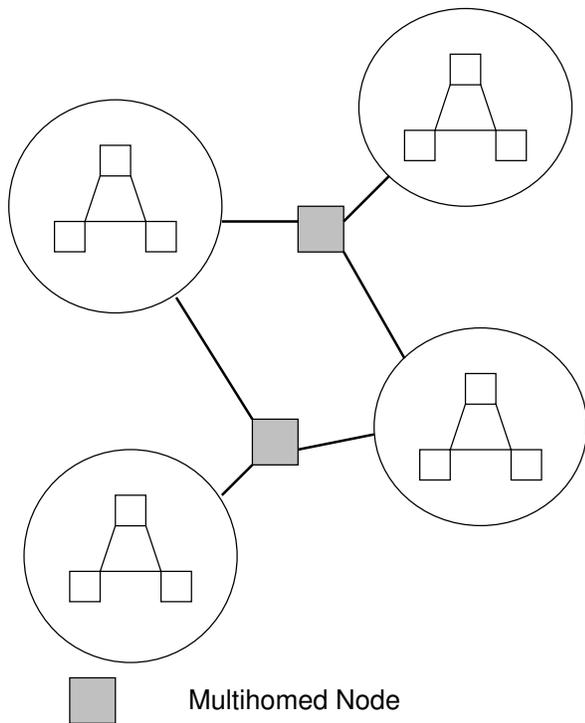


Fig. 1. An interconnected heterogeneous MANET

in ns-2 is also based on CMU Monarch extension, with a small number of add-ons. The CMU Monarch extension will be further detailed in Section III and our wireless multi-homed node extension will be based on the CMU Monarch wireless node design.

There is actually already one multi-homing implementation included in current ns-2 release. It is the implementation of The Stream Control Transmission Protocol (SCTP) provided by University of Delaware. However, it's not a satisfying solution for the multi-homing requirement of MANETs. The first argument, against the use of SCTP for the realization of multi-homing in MANETs, is that the multi-homing in current SCTP implementation is only used for the transport redundancy between exactly two endpoints, and it does not support the interconnection in a heterogeneous network, which however is our focus. The second argument is that the multi-homing feature in current SCTP implementation is not an independent feature and always need to be used together with SCTP, which however suppresses the use of other transport protocol, such as TCP and UDP. The last argument is that the support of SCTP in the existing Internet infrastructure and MANETs is minimal. An extreme example is the redundantly deployed sensor network, where transmission reliability is not so important and a connectionless transport protocol similar to UDP is preferred for the saving of limited energy.

In Ref. [5], the authors try to (in the format of a howto) address all the changes that are needed on ns-2 source code (both TCL and C++) so as to be able to extend current wireless node implementation to support multiple interfaces. However,

in the current wireless node structure, only a single routing agent can be attached. The manual of ns-2 [7] said that a wireless node may have more than one interfaces, it is assumed that a unique routing protocol agent runs over all the interfaces. What Ref. [5] can help is to support such kind of multiple interfaces. Not to mention massive of technical work needs to be done to support such a multiple interface implementation, such kind of support is not sufficient for the interconnection requirement of heterogeneous MANETs.

Researchers in the university NTNU realized the insufficiency in current ns-2 wireless node design, and thought the current wireless node design has made it difficult to extend new features [8]. Thus they provided a modified design of ns-2 wireless node, and based on this modified wireless node design, a multiple interface wireless node with each interface attached with an independent routing agent can be easily built. The drawback is that this modified wireless node design is not included in the ns-2 release currently and there are few publicly available mobile protocol implementations that can support this modified design, although it is considered to be more reasonable.

Another drawback of Ref. [5] proposal and NTNU proposal [8] is that although multiple interfaces can be supported by their proposals, all interfaces must share the same Source/Sink agent. That means, all interfaces must share the same IP address, since a Source/Sink agent must and can only be bound with a static IP address in the current implementation. According to the definition of multi-homing, both these two proposals can not provide the real sense multi-homing ability for a wireless node.

It may however be interesting to take a look at the ns2-MIRACLE framework [9], where multiple modules are allowed to coexist within each layer of the protocol stack. Thus multi-homing may be supported using their framework. In this paper, we are not going to present a subversive redesign of the ns2 structure, rather than provide a short cut of implementing multi-homed node with a few add-ons.

### III. CURRENT WIRELESS NODE DESIGN IN NS-2

This section gives a brief overview of the wireless networking support in ns-2, originally developed as extensions to ns-2 by the CMU Monarch group [2]. Each wireless node is an independent entity that is responsible for computing its own position and velocity as a function of time, and each wireless node consists of a series of components. Basically, these wireless node components follow the ISO network stack. The physical layer includes radio propagation models, radio interfaces with adjustable parameters such as transmission power and receiver sensitivity and antennas models. The link layer includes medium access control (MAC) protocols needed in such environment, and in addition address resolution protocol (ARP) for the MAC address translation in any outgoing IP packet. The network layer are implemented as a routing agent. All these entities are linked up together inside the wireless node. Fig. 2 shows the basic schematic layout of a typical wireless node.

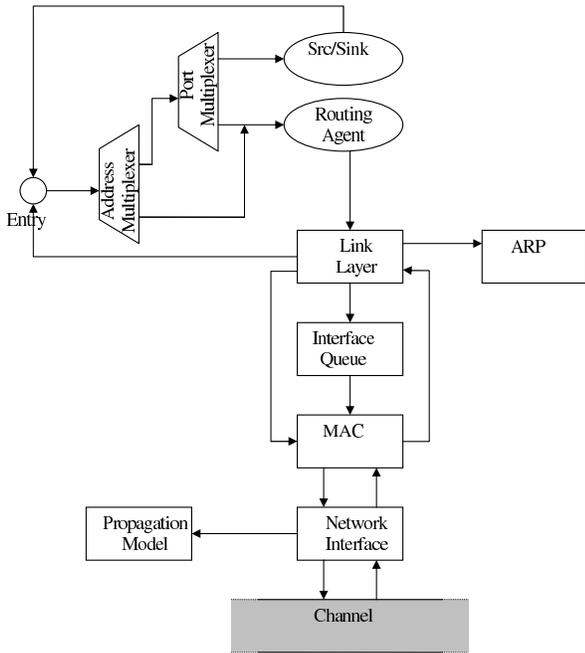


Fig. 2. Schematic representation of a wireless node (CMU Monarch implementation)

#### A. Outgoing packets

Packets generated by a source agent attached to a wireless node are sent to the node's entry point, except routing packets (such as routing request) generated by the routing agent which are directly passed down to the link layer. The address classifier compares the destination address with the local address and, if it does not match, forward the packet to the routing agent by default. The routing agent is then responsible for setting the next hop address before handling the packet down to the link layer. The link layer object queries the ARP object to map the IP address to a MAC address. If ARP does not have a valid mapping, it sends an ARP request to resolve the IP address to the MAC address. Once the MAC address of the next hop is known, the packet is queued in the interface queue (IFq). The MAC object takes packets from the queue and sends them when appropriate. At the physical layer the node position and transmission power are recorded in a metadata packet header<sup>1</sup> before the packet is transmitted onto the wireless channel. A copy of the packet is distributed to all the other interfaces on the same channel after a propagation delay depending on the distance between the sending and receiving nodes.

#### B. Incoming packets

Upon reception of a packet, a propagation model is used to determine the power at which the interface is receiving the packet. It is assumed that nodes do not move significantly while transmitting or receiving a packet so that a single signal

<sup>1</sup>This is called the common header in ns-2 terminology. This common header is not designed according to any wireless protocol and is only used for the ease of simulation.

strength value is computed. The physical layer object determines then whether the packet has actually been successfully received by this node or not. Every incoming packet received at the MAC layer without error or collision on the medium is forwarded to the node's entry point. If the packet is at its final destination, the address classifier passes it directly to the port classifier which delivers it to the proper sink agent. Otherwise, the packet is to be forwarded by the routing agent.

#### IV. MULTI-HOMING SUPPORT BASED ON CURRENT WIRELESS NODE DESIGN

We have mentioned in Section II that multiple interfaces can not be easily supported based on current ns-2 wireless node design. And even if multiple interfaces were made supported in a single wireless node, such as Ref. [5] and [8] have done, it still does not satisfy the multi-homing requirement of MANETs. Thus, we consider to use multiple existing ns-2 wireless nodes to imitate a single wireless multi-homed node. That means, for each interface in the imitated multi-homed node, an ordinary non-multihomed node is initialized to simulate it. For convenience, we will use the name "interface node" for all non-multihomed wireless nodes that are used in imitating the multi-homed node. Then all these "interface nodes" must be associated in some way so that they seem to be just different interfaces belonging to the same host node. This idea is not new, since the current ns-2 SCTP implementation realizes multi-homing like this, and this idea is also mentioned as a comparison in [8]. But we will show our design can provide the multi-homed node with the ability to support heterogenous network interconnection, which SCTP implementation does not provide except for its inappropriate use in MANETs. In addition, the worry mentioned by Ref. [8] about the information sharing and packet forwarding among interfaces will be shown not to be a problem in our design.

Basically, to use multiple "interface nodes" to compose a single wireless multi-homed node, there are two problems needed to be solved. The first is all physically related operations, such as node movement, conducted by all "interface nodes" must look like acted by the same physical node. The second is all logically related operations, such as internal information sharing, must seem to be conducted within a single node.

##### A. Physically imitating a wireless multi-homed node

All "interface nodes" hosted at the same wireless multi-homed node should always be physically bound together throughout the simulation run. In the current ns-2 release, users can configure the mobility scenario for each ordinary non-multihomed wireless node in a TCL application file. The node's mobility scenario is specified by assigning the node's initial location together with the moving speed and moving direction change covering the whole simulated time span. To ensure all "interface nodes" are always at the same place, people can configure them one by one with the same mobility scenario. The way in configuring an "interface node" is the same as that in configuring an ordinary node. An exception is

that one can not configure a random mobility scenario for a multi-homed node by configuring all “interface node” one by one with a random mobility scenario. The reason is each time a random scenario is configured, it’s different from the random scenario before. Thus, “interface nodes” actually have different random scenarios. A special function needs to be designed to ensure the mobility of all “interface nodes” are updated with the same random number generation, if a random scenario is necessary.

*B. Logically imitating a wireless multi-homed node*

Bound together physically is not enough, “interface nodes” still need to be bound together logically. This is done through the association among different “interface nodes”. In our design, we use two additional kinds of modules to associate “interface nodes”.

A Core Application module<sup>2</sup> available to all attached “interface nodes” is used to simulate the application of the multi-homed node. Like that in an ordinary node in the current design, there is a Src/Sink agent attached to each “interface node” which is responsible for the application-level packet sending and packet receiving on that concrete interface. Then, the same application, e.g., the Core Application, is bound with all the corresponding Src/Sink agents on all “interface nodes”. The decision about which one or several interfaces are responsible for a special packet delivery is made inside the Core Application. A multi-homed node with such a Core Application and two “interface nodes” is shown in Fig. 3.

Another module called Switcher module is used to support the packet forwarding among “interface nodes”. So the packet received from one interface can be forwarded to another interface (usually connected with a different network), thus provide the multi-homed node with the ability to interconnect heterogenous networks. The position of this Switcher Module and its connection with other modules are also shown in Fig. 3. Different from current design, a packet sent down by the routing agent is firstly received by a Switcher module, and then the Switcher module decides whether deliver it down to the local link layer, or forward it to the entry point of another interface. The Switcher module here is actually responsible for the routing among interfaces. Since different interfaces in a multi-homed node can be connected with different heterogenous networks, the routing ability provided by the Switcher module together with that provided by the routing agent make the packet exchanging among heterogenous networks possible.

The inside details of the Switcher module is shown in Fig. 4. Suppose a multi-homed node presented in this paper is used as the switcher/router in a simulated heterogenous network similar to that shown in Fig. 1. Each interface of this multi-homed node is connected with a subnet. All interconnected subnets are assumed to have non-overlapped address spaces, so any two nodes in this heterogenous network can be told apart by their addresses.

<sup>2</sup>A multi-homed node can have multiple such Core Application modules depending on the number of applications simulated on this multi-homed node.

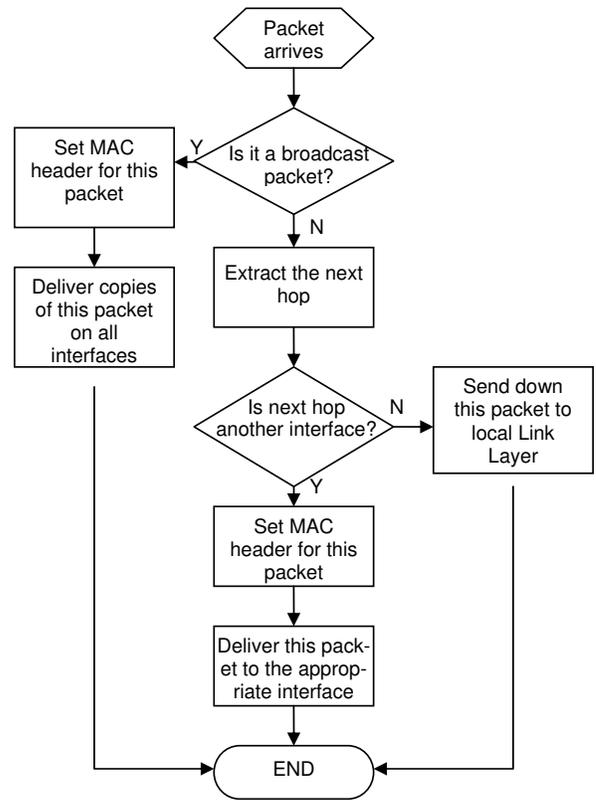


Fig. 4. The flowchart inside the Switcher module

When a packet delivered by the routing agent arrives at the Switcher module, it firstly check whether this packet is a broadcast such as a routing request packet. If the arriving packet is a broadcast, a copy of it will be delivered on every connected interface, including the local interface. If the arriving packet is not a broadcast, then the next hop of this packet must have been assigned by the routing agent before the packet is sent down to the Switcher module. In this case, the next hop information will be extracted (the next hop information is saved as an item in the common header), and compared with the address of every connected interface (noting each interface may have a separated address according to the multi-homing definition). If the next hop is found to be one of the other interfaces in the multi-homed node, the packet will be delivered to the corresponding interface, otherwise it will be sent down to the local link layer. A problem on directly delivering a packet to another interface is that the packet won’t pass the MAC layer, so the MAC header information will be left empty. Because some routing protocols may use the former hop information in the MAC header to build a reverse route path, the MAC header is preferred to be set by the Switcher module before the packet is delivered to another interface. Generally, the interconnected subnets can have different radio channels, different MAC protocols and routing protocols, etc. But if different protocols are used, the packets must be translated inside the Switcher module before they are forwarded from one interface to another interface

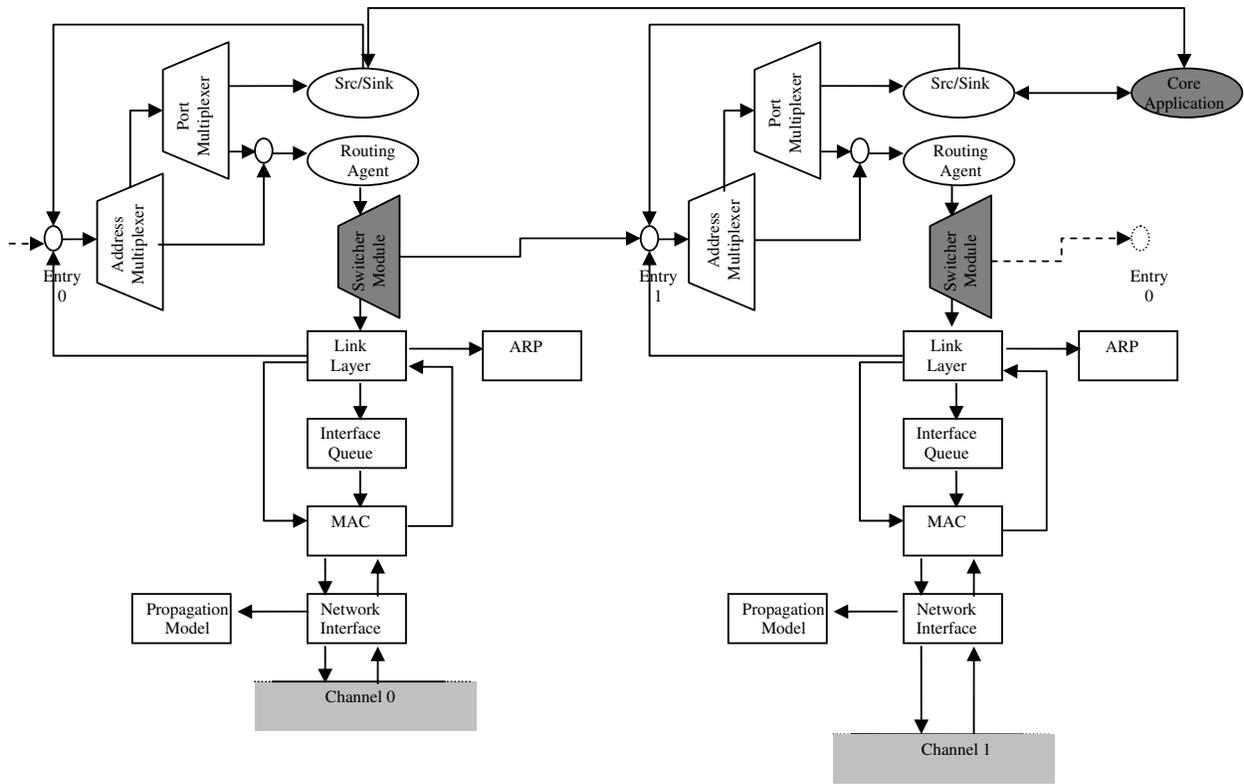


Fig. 3. An imitated multi-homed node using multiple “interface nodes”

within the multi-homed node. Given the packet forwarding among internal interfaces can be viewed as normal packet forwarding operated on a “perfect link”. The possible differences on tackling link failure problems by different kinds of protocols have no difference on a “perfect link”. Thus, only basic functions such as routing request and reply need to be translated, when different protocols are used.

Except for the above new modules need to be newly built and installed during the imitating of a multi-homed node, all original modules can be configured in the same way as those corresponding ones in the current wireless node design.

## V. MULTI-HOMING IMPLEMENTATION IN NS-2

According to former section IV, the implementation of the multi-homing extension mainly includes two parts: the implementation of the Core Application module, and the implementation of the Switcher module. For the Core Application module, the important thing is to make it possible of attaching multiple local agents (one agent corresponds to one “interface node”) simultaneously. We have implemented the Core Application module as an inherited class of Application. Besides the multiple local agents, we have also provided the Core Application module with member functions that can send and receive packets through any of the attached agents. For the Switcher module, it needs the ability of receiving down-going packets from the routing agent and deliver them to the link layer or to a peer “interface node” after a few judgements defined in Fig. 4. To minimize the work required

for an implementation, we have incorporated the Switcher module into the link layer and implemented the modified link layer as an inherited class from the original link layer class. By doing this, the Switcher module does not need to configure its own down-target and up-target in the network layer. When constructing a new node in ns-2, the Switcher module is automatically constructed and connected with other components inside the node providing the node is configured with this new link layer class. We have added a new member variable called “peer-target” to the new link layer; thus it can be connected to the entry point of a peer “interface node”.

The functioning of our multi-homing implementation was verified by running a ns-2 simulation test with a simple network scenario. In this simple network scenario, there were four nodes simulated. Two of them were configured with one radio channel and the rest were configured with another radio channel. Thus, those nodes formed two isolated networks. To bridge those two isolated networks, we picked one node from each network and combined them into an imitated multi-homed node. Finally, the unpicked node in one network could communicate with the unpicked node in another network through this multi-homed node. It was AODV [10] that served as the routing protocol in our test.

Our multi-homing implementation is available on NS2-MIUN website [11], where ns-2 extensions on simulating wireless sensor networks and on simulating network attacks are also available. Currently, we still consider the networks bridged by a multi-homed node use the same routing proto-

col. Routing protocol interpretation among different routing protocols may be implemented if it is necessary.

## VI. CONCLUSION

The multi-homed nodes are usually recognized for their ability in enhancing the transmission reliability, security, and even efficiency in a deployed network. Now, it's becoming more recongized for its ability in interconnecting heterogenous networks, since more and more originally independently deployed networks are waiting to be interconnected. To fulfill the requirement on the simulation of such a multi-homed node that can interconnect networks, we propose a multi-homing extension for the wireless node design in ns-2, which is the mostly used Network Simulator in the simulation of MANETs. We do that by using several ordinary wireless nodes in current design to imitate the multiple interfaces in a multi-homed node. All "interface nodes" participating in the multi-homed node's imitation are associated by sharing common mobility scenario, common application layer, and also by the introduction of internal routing among interfaces. The use of the multi-homed node proposed in this paper does not suppress the use of the existing ordinary wireless nodes in ns-2, thus it can be simply added in the existing ns-2 network simulations.

## REFERENCES

- [1] The Network Simulator - NS-2. [www.isi.edu/nsnam/ns](http://www.isi.edu/nsnam/ns).
- [2] CMU Monarch project, Computer Science Department, Canergie Mellon University, Pittsburgh. "The CMU Monarch project's wireless and mobility extensions to ns", 1999.
- [3] Stream Control Transmission Protocol - RFC 4960, available at [www.rfc-editor.org/rfc/rfc4960.txt](http://www.rfc-editor.org/rfc/rfc4960.txt).
- [4] D. Curren, "A Survey of Simulation in Sensor Networks," available at [www.cs.binghamton.edu/~kang/teaching/cs580s/david.pdf](http://www.cs.binghamton.edu/~kang/teaching/cs580s/david.pdf).
- [5] R. A. Calvo and J. P. Campo, "Adding Multiple Interface Support in NS-2," available at [personales.unican.es/aguerocr/files/ucMultifacesSupport.pdf](http://personales.unican.es/aguerocr/files/ucMultifacesSupport.pdf).
- [6] S. Kurkowski, T. Camp and M. Colagrosso, "MANET Simulation Studies: The Incredibles," *Mobile Computing and Communications Review*, vol. 9, 2005, pp. 50-61.
- [7] K. Fall and K. Varadhan, "The ns Manual," *The VINT Project*, UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2006
- [8] L. Paquereau and B. E. Helvik, "A Module-Based Wireless Node for NS-2," in *1st Workshop. on NS-2: The IP Network Simulator (WNS2)*, Italy, 2006.
- [9] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi and M. Zorzi, "ns2-MIRACLE: a Modular Framework for Multi-Technology and Cross-Layer Support in Network Simulator 2," in *2nd International Conference on Performance Evaluation Methodologies and Tools*, France, 2007.
- [10] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, Feb. 1999, pp. 90-100.
- [11] Q. Wang, "NS2-MIUN," in [www.miun.se/personal/qinghua.wang/resources.htm](http://www.miun.se/personal/qinghua.wang/resources.htm), 2009.