Local Node Stability-Based Routing for Wireless Mesh Networks

Mustapha Boushaba¹, Abdelhakim Hafid¹, Michel Gendreau²
¹Network Research Laboratory
University of Montreal, Montreal, Canada
{boushamu, ahafid}@iro.umontreal.ca
²CIRRELT and MAGI
École Polytechnique de Montreal, Montreal, Canada
{michel.gendreau}@cirrelt.ca

Abstract—Thanks to their flexibility and their simple installation, Wireless Mesh Networks (WMNs) allow a low cost deployment of a network infrastructure. They can be used to extend the wired network coverage allowing connectivity anytime and anywhere.

Network stability is a key performance metric in supporting real time communication over the network. Because of high bandwidth demand and dynamic traffic variation, several paths in WMNs are expected to be unstable. High levels of network instability can lead to interferences, packet losses and high delays.

In this paper, we address the stability problem of WMNs; instability in these networks is caused mainly by link quality fluctuations and frequent route flapping. Indeed, most routing protocols try to optimize a routing metric locally or globally without considering network stability. First, we present the key factors that may cause network instability; then, we propose a new technique, called Local Node Stability-based Routing (LNS), using the entropy function (known as a measure of the uncertainty and the disorder in a system) to define a node stability.

Simulation results show that the stability can be improved in WMNs using LNS compared to other routing schemes namely RLBDR, MIC and ETX.

Keywords: WMN; Stability; Routing; Interferences;

I. INTRODUCTION

In the last several years, wireless communication has emerged as promising technology for next-generation networks. Among those networks, Wireless Mesh Networks (WMNs) [1] have become quite popular allowing communication to the Internet, anytime and anywhere. WMNs have the potential to eliminate cables and thus dramatically reduce installation and maintenance costs and improve adaptability, flexibility and scalability.

A WMN can be seen as a multi-hop Mobile Ad-hoc Network (MANET) with extended connectivity; the key difference is that WMNs are characterized by a relatively static architecture and low mobility. WMNs combine wired and wireless networks with wireless Mesh Routers (MRs) as backbone and mobile stations as users. The role of MRs is to relay information from one node to another via multi-hop communications. Usually, MRs send traffic to a gateway (GW) that connects nodes to the Internet. In the case of a WMN with a single gateway, the gateway selection problem becomes simple; indeed, all upstream/downstream traffic flows traverse the same GW to the Internet; thus, the GW is more likely to become the bottleneck/single point of failure in the network [2]. To mitigate this problem, multiple gateways are installed to distribute load and improve performance. However, increasing the number of GWs does not necessarily increase the capacity of WMNs. Indeed, network capacity is closely related to network connectivity and the placement of GWs; these issues are out of scope of this paper.

WMN performance is highly impacted by interferences and (GW) congestion causing considerable packet losses and higher delays. To improve WMN performance, many schemes using directional and smart antennas [3], multiple input multiple output (MIMO) systems [4, 5], and multiple radios and multiples channels [6, 7] have been proposed. Particularly, routing schemes can play a major role impacting (positively or negatively) the network performance; indeed, the main goal of routing is to find better routes, according to specific routing metrics, from sources to destinations. To achieve a good network performance, a routing metric should (a) not impact negatively the network stability; (b) capture the characteristics of mesh networks; (c) compute paths in polynomial time; and (d) avoid forwarding loops [8]. Network stability is considered one of the most determinant network performance metrics which has not been considered by most routing protocols. Network instability occurs whenever the frequency of rerouting increases considerably; this increase is generally caused by one or more links that exhibit considerable quality fluctuations.

Many applications (e.g., multimedia applications) require a stable wireless mesh network for acceptable quality. Indeed, instability (frequent route flapping) may cause out of order delivery, high jitter, packet losses and/or high delays which are unacceptable for multimedia applications. For better understanding, let us consider the network configuration shown in Fig. 1 where nodes $R1$, $R2$, $R3$, $R4$, $R5$ and $R6$ are MRs, nodes $R1$ and $R2$ are source routers optionally equipped with traffic aggregation devices and nodes $G1$ and $G2$ are gateways. In this example, we consider delay as a routing metric; thus, traffic from $R1$ to $G1$ follows the path $R1$-$R4$-$G1$. Now, let us assume that the estimation of the metrics of the links ($R6$-$G1$) and ($R3$-$R6$) change to 0.5 ms; thus, the delay to send packets from $R3$ to $G1$ through $R6$ is now 1 ms; in this case, the routing protocol selects the path $R1$-$R3$-$R6$-$G1$ to send traffic from $R1$ to $G1$. Hence, some packets (of the same flow) rerouted on $R1$-$R3$-$R6$-$G1$ will reach their destination $d$ in the Internet before the already transmitted packets on $R1$-$R4$-$G1$; consequently, many packets may be out-of-order at $d$ [27]. The problem becomes very critical when the route flapping is consistent.
The destination will be overloaded with handling the out-of-order packets and therefore delay and jitter will increase making the quality of some multimedia applications, such as video-conferencing, unacceptable.

Fig. 1 Route flap scenario

Wireless Mesh Networks rely on routing protocols to find optimal paths. Generally, these protocols do not consider the instability of the network which may occur when the volume of traffic and demand for wireless capacity increase. Unlike wired links, wireless links often have frequent bandwidth fluctuations. Factors, such as interference, fading and shadowing impact considerably the state of the link. Therefore, they increase the possibility of route breaks, during data transmission, which impact the network stability.

This paper proposes a new routing scheme, in WMNs, that takes into account network stability; indeed, the objective of the proposed scheme is not to impact negatively the network stability. The key contribution of the paper is the definition of a novel routing metric, called Local Neighborhood Stability (LNS) that quantifies the stability of Mesh routers. Using LNS, upon receipt of a packet, a MR selects for the next hop, towards the packet destination, a stable mesh router.

The remainder of the paper is organized as follows. In Section II, we present related work. Section III presents brief overview of network stability. In Section IV, we present the details of the proposed Local Node stability-based routing scheme. Section V evaluates the performance of our proposal. Section VI concludes the paper and presents future work.

II. RELATED WORK

Many routing schemes have been proposed in the literature for WMNs [9, 10]. These schemes could be roughly divided into three categories: (1) proactive routing protocols (e.g., Optimized Link State Routing OLSR [11]); (2) reactive routing protocols (e.g., Ad hoc On-demand Distance Vector AODV [12]); and (3) hybrid protocols (e.g., Temporally-ordered routing algorithm TORA [13]). Based on the aforementioned routing techniques, several routing metrics have emerged. In general, a routing metric is used, by a routing protocol, to select the path having the highest throughput, the lowest delay and/or the lowest packet loss ratio. However, most routing protocols do not take into consideration the network stability. We believe that network stability has a considerable impact on network performance; indeed, a stable network outperforms an instable network.

Stability is hard to define; it has been used in different domains and networks to indicate a state or quality of the system under consideration. In the open literature, several contributions [14, 15, 16, 17, 29] have been made to study stability in ATM networks, Ad-hoc networks, wireless sensor networks and wired networks, however, few in WMNs. In [14], the authors, in the context of ATM networks, stated that a network is stable if and only if all packets experience a bounded delay. More specifically, they compute the upper delay bounds (of a cell/packet) per server; then, they compute the end-to-end delay of a connection as the sum of the upper delay bounds of servers (ATM switches) used by the connection. The end-to-end delay (of all packets) should be below a threshold to consider the network stable.

For Ad-hoc networks [15, 29], Associativity-Based Routing (ABR) [15] is considered as one of earliest works whose fundamental objective is to find longer-lived routes. The authors define a link between two neighbor nodes as stable if a node receives constantly signals from it neighbor and the number of hello messages exceeds a threshold. Otherwise, the link is unstable when any signal is received during a threshold time. Signal Stability Adaptive Routing (SSA) [29] estimates link stability based on the measured signal strength. The authors define a strongly connected link as a link with signal strength exceeds a certain predefined threshold. Otherwise the link is weakly connected. Thus, a stable link is the one which exhibits the strongest signals for the maximum amount of time.

Routing, service and resource discovery capabilities are needed to enable applications and services over large scale WMNs. In this context, the authors in [16] propose a layered architecture to autonomously select a set of backbone nodes to create a quorum system. They propose an algorithm called ADB that consists of three components (a) neighbor discovery; (b) backbone selection; and (c) backbone connection. ADB explores stability information, maintained by each mesh node, that consists of two metrics: (1) nodeStability, is the estimated stability of the area surrounding a node. It consists of a set of probabilities (the cardinality of this set is equal to the number of the node’s neighbors); an element of this set represents the approximation of the probability P that a link from a node (denoted by i) to its neighbor will not break within the next specific time window; and (2) pathStability: is the estimated path stability from i to its current backbone. It is computed as follows: pathStability=\pathStability(parent of i)\P_i.

In [17], the authors have analyzed routing stability for static wireless mesh networks. They reported, after experiments on UCSB Meshnet [18] and MIT Roofnet [19], that the main reason of network instability is the fluctuating quality of links caused by the effect of multipath fading, interference and weather conditions. The analysis of routing stability is based on link quality information and uses three route-level characteristics [17]: (1) prevalence: represents the probability or the number of times that a path has been met in the past; (2) persistence: represents the probability that a route remains unchanged over a long time period; and (3) route flapping: represents the variation of the paths during a time period.

Based on the analysis of the persistence and the prevalence, the authors indicate that routes in wireless mesh networks are inherently unstable. Whereas, for route flapping analysis, a routing protocol that always flaps routes will likely achieve
only minimal gains (i.e., marginal improvement in throughput) in a large number of instances.

III. FACTORS LEADING TO INSTABLE NETWORKS

High levels of network instability can lead to packet losses and increase network latency. There are several factors that lead to network instability; these factors include: (1) Interferences: they are one of the major factors that impact network performance by creating instability. In fact, the shared nature of the wireless channel is the main characteristic of wireless networks. Thus, when the medium is shared among multiple nodes using the same channel and transmitting at the same time it can cause inter-flow interferences and intra-flow interferences; (2) Routing protocols: they have also an impact on the network stability. In particular, in link-state protocols, the inappropriate number of advertisement messages to update routes may consume an excessive amount of resources in the network and cause route flaps. Consequently, it degrades performance and creates an instable network; (3) Network topology and traffic patterns: they have an impact on the behavior of routing and network stability. In particular, the shortest path routing protocol, which has seen enormous success in wired networks, is inadequate in WMNs. In fact, shortest path routing, using congestion as the link cost is very prone to instability under heavy loads and bursty traffic; (4) Congestion: it is the result of MAC queue buildup at intermediate nodes. It can be observed by a forwarding node when packets arrival rate is greater than its forwarding rate. There are many factors that contribute to congestion, namely network topology, number of flows, the traffic characteristics of the flows and their routes, as well as channel capacity and the available transmission rate at the physical layer. Presence of congestion in a network impacts the network throughput, causes peer failures in routers and leads to routing failures and network instability; (5) route flapping: It occurs when a MR alternately advertises a destination network first via one route then via another (e.g., unavailable route and then available again). This phenomenon is undesirable in WMNs; it often forces a mesh router to compute a new preferred route to a destination which causes network delay and requires extra CPU cycles. Whenever route flapping occurs, it causes unnecessary delay in packets forwarding; and (6) Effects of exploration: One of the major difficulties in the development of reinforcement learning algorithms is managing the trade-off between the execution of the best known policy, the exploitation and the exploration. In a multi-agent environment, exploration can contribute to instability since the agents switch their respective action from time to time. The problem is even greater when the number of nodes that contribute to exploration is high.

IV. PROPOSAL DESCRIPTION

One of the symptoms of the network instability is the disappearance and the reappearance of routes in the routing table in an intermittent manner. Thus, the number of these disappearances/reappearances of routes may characterize the intensity of the perturbation in the network.

A. Network Model and assumptions

We consider multi-hop infrastructure WMNs with two sets of nodes: MRs and GWs. MRs form a multi-hop wireless backbone to relay traffic between users and Internet. To reach the Internet, traffic passes through GWs which are MRs with more functionality (i.e., more buffer size, wired and wireless interfaces). Each MR may be equipped with multiple wireless interfaces. We assume that the interfaces of a MR are assigned different channels. We formally model the backbone WMN as a graph $G=(V, E)$ where $V$ is the set of nodes (MRs and GWs) and $E$ is the set of links. Let $i_g$ denotes the link between two MRs $v_i$ and $v_j$ ($v_i, v_j \in V$). In $V$, there are GWs that provide connectivity to Internet. Like in [5, 6], we define the load $L_g[t]$, of gateway $g$, at time $t$ as the average interface queue length. It can be measured by monitoring the number of packets buffered in the GW interface over a time period. At every generated advertisement by GWs, the current load $L_g[t]$ is based on the previous estimated load $L_g[t-1]$ and the volume of traffic $V[t]$ that gateway $g$ has processed since the last advertisement:

$$L_g[t] = \alpha \frac{V[t]}{C_g} + (1-\alpha)L_g[t-1]$$

where $\alpha \in [0, 1]$ and $C_g$ is the maximum queue length of gateway $g$.

We assume that each node $v_i$ maintains routing table $RT_i$ that stores routing information to each destination $d_i$; $|RT_i| = N$ is the size of $RT_i$. For MR $v_i$ ($i = 1, 2, ..., |V|$) we denote by $RT_i^t$ its routing table at time $t$. Each entry in $RT_i^t$ includes $(d_i, f_1, f_2, ..., f_n, k_{ij}, v_j)$ where $n > 0$, $j <> i$, $k_{ij}$ is the channel used on link $l_{ij} \in E$, $v_j$ is next hop towards destination $d_i$, and $f_1, f_2, ..., f_n$ represent the features (e.g. Interference Ratio and Loss Ratio) of $l_{ij}$ when using $k_{ij}$. For each incoming packet, $v_i$ uses the information stored in its routing table $RT_i^t$ to assign the outgoing link to be used to forward the packet towards its destination.

Let us define $RT_i^{t+1}$ as the routing table at time $t+1$ of the node $v_i$. $RT_i^{t+1}$ is a combination of three sub routing tables: (1) $r_{nc}^{t+1}$: contains links that experience no change between $t$ and $t+1$; (2) $r_{e}^{t+1}$: contains links at time $t$ that experience some change at time $t+1$ (this change may be an improvement or a degradation of the quality of the link); and (3) $r_{n}^{t+1}$: contains new links observed at time $t+1$ that were not present at time $t$.

$$RT_i^{t+1} = r_{nc}^{t+1} + r_{e}^{t+1} + r_{n}^{t+1}$$

B. Link Quality Metric and Node stability

1) Link Quality Metric (LQM)

$LQM_i$ for a link $i$ is considered as one of the link features stored in the routing table. According to [21] $LQM_i$ is defined as a weighted function of two parameters: $IR$ (Interference Ratio) and $CL$ (Congestion Level). $IR$ estimates the interference level in the network through the Signal to Noise Ratio ($SNR$) and the Signal to Interference-plus-Noise Ratio ($SINR$). $IR$ for a node $u$ on link $l$ ($u$ and $v$ are the ends of link $l$ operating on the same channel) is defined as follows:

$$IR_l(u) = \frac{SNR_l(u)}{SNR_l(u)}$$

where $SNR_l(u)$ and $SNR_l(u)$ are defined in Eq. (4) and Eq. (5):
\[ SINR_t(u) = \frac{P_v}{\text{Noise} + \sum_{w \in N_{i-v}} \tau_w P_w} \]  
(4)

\[ SNR_t(u) = \frac{P_v}{\text{Noise}} \]  
(5)

where \( P_v \) is the signal strength of a packet, sent by node \( v \), at node \( u \), \( N_i \) is the set of nodes from which node \( u \) can hear a packet and \( \tau_w \) is the amount of time that node \( w \) occupies the channel.

\[ CL \] is closely related to link utilization \((U(l,t))\) during a time period \( t\).\( U(l,t) \) is defined as follows:

\[ U(l,t) = \frac{\sum_{i \in \text{Succ}(l,t)} \text{Size}_i}{B W_i} \]  
(6)

where \( \text{Succ}(l,t) \) denotes the set of packets that have successfully traversed link \( l \) during a period time \( t \) and \( \text{Size}_i \) denotes the size of packet \( i \) \((i \in \text{Succ}(l,t))\). \( BW_i \) is the bandwidth capacity of link \( l \).

Thus, by using the moving average, we define \( CL \) as follows:

\[ CL(l,t) = \beta U(l,t) + (1 - \beta) CL(l,t-1) \]  
(7)

where \( \beta \in [0,1] \)

By grouping \( IR \) and \( CL \), \( LQM_i \) for link \( l \) operating on channel \( i \) is defined as follows:

\[ LQM_i = \left( 1 - \frac{1}{2^n} \right) \times IR_i + \frac{1}{2^n} \times CL_i \]  
(8)

where \( n \) is the number of 1-hop neighboring nodes that share channel \( i \) with the node computing its \( LQM_i \).

2) Node stability-based routing

Before defining the stability of a node let us discuss the stability index which we associate to a link in a routing table. The stability index of a link is computed using the link quality history including the current quality; a link is said to be acceptable (or unacceptable) when its \( LQM_i \) is smaller (or bigger) than a predefined threshold. It is clear that a link that changes frequently between acceptable and unacceptable will not be chosen to route packets; indeed, these changes will create oscillations in the network causing its instability.

We propose an algorithm, called Stability Index Algorithm (SIA) that computes link stability index using link oscillations between acceptable and unacceptable, link persistence, and link quality improvement/deterioration. In the following, we present the pseudo-code of the algorithm used to compute \( S_{t+1} \) (link stability index of \( l \) at time \( t+1 \)). The algorithm is executed by each node (mesh router) \( v_i \in V \).

The algorithm starts by adding acceptable or unacceptable links (bigger or smaller than a predefined threshold) in the routing table. Even if a link is unacceptable, it will be added in the routing table with a stability index equal to 0 (an acceptable link is added with stability index equal to 1) which corresponds to line 6 (vs. line 4) in the algorithm. Thus, based on these values at the initialization, the routing protocol will avoid selecting newly created links \((S^t_1 = 1): links having no history in the past) or bad quality links \((S^t_1 = 0)\). After each period where the link quality does not change (line 9 with a small \( \beta \) value) we increment by 1 the stability index of the link. If the link quality has improved relative to the past value, its stability index will be incremented by 2 (line 13 of the algorithm); otherwise, if the link quality has deteriorated but still acceptable \((< \varepsilon)\) we divide its accumulated stability index by 2 (line 16); thus, if link degradation continues, its index stability will exponentially approaches zero. In the case of a link quality above the threshold \((\varepsilon)\), stability index will be set to 0 (line 18).

Algorithm 1: Stability Index Computation by a node \( i \)

Variables

\( l \): link

\( LQM^t \) : Link quality of \( l \) measured at time \( t \)

\( LQM^{t+1} \) : Link quality of \( l \) measured at time \( t+1 \)

\( RT^t_i \) : Routing table at time \( t \) of the node \( i \)

\( RT^{t+1}_i \) : Routing table at time \( t+1 \) of the node \( i \)

\( S^t_i \) : Stability index of the link \( l \) at time \( t \)

\( S^{t+1}_i \) : Stability index of the link \( l \) at time \( t+1 \)

\( \varepsilon \) : Predefined threshold.

\( \beta \) : is a constant belonging to \([0,1]\)

Algorithm

1. For each \( l \in RT^{t+1}_i \) do

2. if \((l \not\in RT_i^t) /* new link*/

3. if \((LQM^t_i) > \varepsilon )/* accepted link*/

4. \( S^{t+1}_i \leftarrow 1 \)

5. else

6. \( S^{t+1}_i \leftarrow 0 \)

7. end if

8. else /* the link already exists*/

9. if \((|LQM_i^t - LQM_i^t| < \beta ) /* the link quality has not changed*/

10. \( S^{t+1}_i \leftarrow S^t + 1 \)

11. else /* the link quality has changed*/

12. if \((LQM_i^t < |LQM_i^t| ) /* the link quality has improved*/

13. \( S^{t+1}_i \leftarrow S^t_i + 2 \)

14. else /*the link quality worsened and */

15. if \((LQM_i^t) > LQM_i^t | /*

16. \( S^{t+1}_i \leftarrow S^t_i + 1 \)

17. else

18. \( S^{t+1}_i \leftarrow 0 \)

19. endif

20. endif

22. endif

23. end for

Let us define that a node is stable when all or most of links in its routing table are stable. To compute the stability of a node \( v_i \) we use (1) the stability index of the links that \( v_i \) maintains in its routing table (see SIA for details); and (2) the entropy function defined by [28] as a measure of the uncertainty and the disorder in a system.

Using Equation (10), that defines the probability of the stability of a link relative to other links (in the routing table of node \( v_i \)) having a non-zero stability index, we define the
entropy $H(\Delta_t)$ at node $v_i$ during the time interval $\Delta_t$ in Equation (11).

$$p_t = \frac{S_{i+1}^t}{\sum_{i \in RT(i)}(S_{i+1}^t)} \quad (10)$$

$$H_i(\Delta_t) = -\sum p_i \log p_i \quad (11)$$

where $N_i = |RT(i+1)|$ (number of entries in the routing table of $v_i$). Let us note that a high value of the entropy (varies between 0 and 1) means a more stable node.

At time $t$, every node computes its entropy (see Equation 11) and broadcasts it to its 1-hop neighbors. Upon receipt of this information, a node stores it in its routing table and uses it for best next hop to forward data packet to its destination. Indeed, a node will select a best path from a set of possible paths (in its routing table) to a given destination, that has the most stable node (i.e., the node with the biggest entropy) as next hop.

V. SIMULATION AND RESULTS

We study the performance of our proposed LNS based routing by using ns-2 with the module implementing multi-radio and multi-channel WMNs [24] and the simulation parameters shown in table I.

<table>
<thead>
<tr>
<th>Simulation Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Area</td>
<td>1000mx1000m</td>
</tr>
<tr>
<td>Link Data rate</td>
<td>11Mbps</td>
</tr>
<tr>
<td>Frequency</td>
<td>9.14e+08 Hz</td>
</tr>
<tr>
<td>Traffic Type</td>
<td>CBR (UDP)</td>
</tr>
<tr>
<td>Network Load</td>
<td>1000-3000 Kbps</td>
</tr>
</tbody>
</table>

The topology used for the backbone Wireless Mesh Networks consists of 16 MRs randomly placed in 1000x1000 area and 3 GWs (nodes 1, 2 and 3). The MAC layer protocol is IEEE 802.11 and the channel assignment is performed using the scheme in [25]. We use a realistic reception model by taking into consideration Bit Error Rate (BER) used in Intersil HFA3861B radio hardware [26] and the corresponding Frame Error Rate (FER). We use Constant Bit-Rate traffic (CBR) with UDP having a packet size of 1,000 bytes. Since most traffic is destined to the Internet, we assume that traffic is randomly generated from different source nodes and may cross several intermediate nodes (i.e., multi-hop routing) to reach the Internet through a GW.

We have implemented and compared the performance of LNS-based routing with RLBDR [21], MIC [22], ETX [23] and NEAREST-G (minimum number of hops to a gateway) based routing; in the rest of the section, metric-based routing (LNS-based routing) and metric (e.g., LNS) are used interchangeably. The performance is evaluated in terms of (1) network average throughput: represents the amount of successfully transmitted bytes in a given time interval (2) average end to end delay: represents the delay for sending CBR packets from source nodes one of the gateways to destinations; and (3) the loss rate:represents the average number of packets dropped inside the WMN over the number of sent packets. All the following results are obtained after 100 runs.

In the first set of simulations, we study the performance of LNS compared to other metrics when increasing the data rate. Fig. 2 shows the average throughput; we observe that LNS performs better than RLBDR, MIC, ETX, NEAREST-G. In fact, at high data rate (3000 kbps), where the number of interference is usually large, the throughput performance is better than the above metrics by around 7%, 17%, 36% and 41%, respectively.

<table>
<thead>
<tr>
<th>Data Rate (kbps)</th>
<th>Throughput (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>LNS &gt; RLBDR, MIC, ETX, NEAREST-G</td>
</tr>
<tr>
<td>1500</td>
<td>LNS &gt; RLBDR, MIC, ETX, NEAREST-G</td>
</tr>
<tr>
<td>2000</td>
<td>LNS &gt; RLBDR, MIC, ETX, NEAREST-G</td>
</tr>
<tr>
<td>2500</td>
<td>LNS &gt; RLBDR, MIC, ETX, NEAREST-G</td>
</tr>
<tr>
<td>3000</td>
<td>LNS &gt; RLBDR, MIC, ETX, NEAREST-G</td>
</tr>
</tbody>
</table>

![Fig. 2. Average throughput](image)

In the second set of simulations, we study the performance in terms of delay. Fig. 3 shows clearly that LNS outperforms MIC, ETX, NEAREST-G. In fact, at 1500 kbps, the delay performance is better than the above metrics by around 59%, 72%, 69%, respectively. However, for a data rate between 1000 kbps and 2000 kbps, RLBDR outperforms LNS by around 25%. This is explained by the fact that LNS is not interested in the number of hops but the stability of nodes. From 2500 kbps, when interferences increase, creating instability nodes, LNS outperforms RLBDR (i.e., at 2500 kbps, LNS performs RLBDR by 36%).

<table>
<thead>
<tr>
<th>Data Rate (kbps)</th>
<th>Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>LNS &lt; RLBDR, MIC, ETX, NEAREST-G</td>
</tr>
<tr>
<td>1500</td>
<td>LNS &lt; RLBDR, MIC, ETX, NEAREST-G</td>
</tr>
<tr>
<td>2000</td>
<td>LNS &lt; RLBDR, MIC, ETX, NEAREST-G</td>
</tr>
<tr>
<td>2500</td>
<td>LNS &lt; RLBDR, MIC, ETX, NEAREST-G</td>
</tr>
<tr>
<td>3000</td>
<td>LNS &lt; RLBDR, MIC, ETX, NEAREST-G</td>
</tr>
</tbody>
</table>

![Fig. 3. Delay vs. Data Rate](image)

In the third set of simulations, we study the impact of data rate variation on the packet loss ratio. In Fig. 4 shows the total loss rate variation while Figures 4 and 5 show loss rate variation due to collisions and IFQ respectively. We observe that LNS outperforms RLBDR, MIC, ETX and NEAREST-G especially when data rate increases. This performance is due to the use of the node stability principle. It is obvious that increasing data rates increases interferences (Fig 5) and packet loss rate due to Interface Queue (IFQ) overflow (Fig 6); this creates oscillations making the network unstable.
LNS in multi-channel WMNs. We propose a new metric, called stability of its nodes.

Currently, we are working to improve the network performance. The objective of the proposed scheme is not to impact negatively network stability, and thus improve the network performance.

Thus, a routing protocol must consider not only routing metrics that capture the effect of interferences and packet losses but also metrics representing stability of nodes. By considering the stability of nodes we can avoid/reduce the well known problem of oscillations in WMNs (i.e., increase network stability), and thus improve the network performance.

VI. CONCLUSION

In this paper, we investigate the impact of routing metrics in multi-channel WMNs. We propose a new metric, called LNS, which takes into account network stability; indeed, the objective of the proposed scheme is not to impact negatively the network stability. Simulations results show that LNS outperforms RLBDR, MIC, ETX and NEAREST-G. In terms of throughput, delay and loss rate. Currently, we are working to improve the estimation of the network stability based on the stability of its nodes.

REFERENCES

[27] B. Ye, A. P. Jayasumana and N. Piratla, "On Monitoring of End-to-End Packet Rendering over the Internet" Int. Conference on Networking and Services (ICNS’06), Santa Clara, CA, June 2006