Heuristics for Bandwidth Reservation in Multihop Wireless Networks

Géraud Allard, Philippe Jacquet

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Heuristics for Bandwidth Reservation in Multihop Wireless Networks

Géraud Allard — Philippe Jacquet

N° 5075

January 2004

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--5075--FR+ENG

Heuristics for Bandwidth Reservation in Multihop Wireless Networks

Géraud Allard , Philippe Jacquet

Thème 1 — Réseaux et systèmes
Projet Hipercom

Rapport de recherche n° 5075 — January 2004 — 14 pages

Abstract: We propose three heuristics to compute QoS routes in a multihop wireless networks considering interferences constraints. It has been proved that reservation under such conditions is an NP-complete problem. Our heuristics are based on Dijkstra's shortest path algorithm in which we integrate the notion of bandwidth capacity in order to satisfy flows requirements.
We show with several simulations that these heuristics not only allows computation of routes that save bandwidth of nodes with low capacity but also that network can handle more QoS-flows.

Key-words: Ad hoc network, QoS, bandwidth reservation, heuristic

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Heuristiques pour la réservation de bande passante dans les réseaux sans fil à routage par sauts multiples

Résumé : Nous proposons trois heuristiques permettant le calcul de routes avec Qualité de Service dans un réseau mobile ad hoc soumis à des contraintes d'interférences. Il a été prouvé que la réservation de bande passante dans de tels réseaux est un problème NP-complet. Nos heuristiques sont basées sur l'algorithme de plus court chemin de Dijkstra dans lequel nous avons intégré la notion de bande passante afin de satisfaire les besoins des flux de données. Nous montrons avec plusieurs simulations que ces heuristiques ne permettent pas seulement d'économiser la bande passante des nœuds ayant une faible capacité, mais également de permettre au réseau de supporter un plus grand nombre de flux avec Qualité de Service.

Mots-clés : Réseau ad hoc, QoS, réservation de bande passante, heuristique

1 Introduction

Mobile ad-hoc networks have recently concentrated a massive research effort. Particularly, since mobile ad-hoc networks consist of wireless nodes that does not require any pre-existant infrastructure to communicate, one must provide routing protocols in order to make communication between nodes possible. Many protocols have been proposed, but they essentially provide *best-effort* routing mechanisms since they compute routes without any performance constraints.
However many applications (e.g. multimedia applications) require some performance guarantees : a phone application may need a low routing delay or jitter, packets sent by a video application may have certain bandwidth guarantees, etc... All these requirements are often gathered under the term of *Quality of Service*. But *best-effort* routing mechanisms are not sufficient since they cannot handle such requirements.
Some works have been done for supporting Quality of Service for ad-hoc networks. QoS protocols are designed to compute admissible routes which satisfy one or several QoS requirements. For example, some protocols propose to perform limited parallel routes explorations in order to find a satisfactory route (TBP [1]), other protocols try to extend existing protocols to handle QoS (QoS-AODV [2], MP-DSR[3] or QOLSR [4]).
Many wireless ad-hoc protocols essentially use CSMA/CA MAC protocols to communicate and this implies that radio medium is shared between nodes. Thus, when a node n_1 sends data to a neighbor node n_2, the transmission interfers with all the nodes located within a certain *interference range* from n_1. If we take bandwidth as our performance parameter, it is crucial to be aware that links between nodes are not isolated and that interferences may occur within several hops from the transmitter. If the transmitting node n_1 interfers with another node n_3, n_3 cannot either send or receive any data when n_1 transmits.
Although taking interferences into account for QoS routing sounds essential, only few protocols consider this constraint (BRuIT [5]). The authors of BRuIT proposed a reactive approach for the bandwidth reservation based on a request/reply mechanism. However, a proactive dialog must be carried out between nodes to send bandwidth capacity informations based on interferences evaluation. Whenever a node wishes to transmit a QoS flow, it broadcasts a request packet, containing its bandwidth requirement, to perform an admission control within the network. Only nodes satifying this bandwith requirement forward the request. When the request packet reaches the destination, it sends back a reply message through the route to perform effective bandwidth reservation.
It has been proved in [6], that bandwidth reservation in ad-hoc networks considering interferences constraints (*Path with Remaining Capacity Problem*) is an NP-complete problem and then heuristics must be found to avoid this issue. To be precise, it is an NP-complete problem even in its incremental form when a route has to be found for a new flow that does not disturb existing connexion. In this work we propose some heuristics to compute admissible routes in such networks. The organization of the paper is as follows. In Section II we expose our network model, Section III describes heuristics and in Section IV simulation results are exhibited. We conclude the paper in Section V.

2 Network and interferences models

In order to understand the problem raised by interferences, let us consider the network as a undirected graph $G(V, E)$. Let $i, j \in V$, i and j can commmunicate (*i.e.* send packets to each other) if and only if $(i, j) \in E$. Denote by $N_i(h)$ the *h-hop neighborhood* of node i. Each node i have a certain bandwidth capacity represented by a number of *bandwidth units* C_i.
Unlike wired networks, links in an ad-hoc network are not isolated and a transmission between a node i and a node j generates interferences in the neighborhood of node i. More precisely, if we consider that interferences are propagated within H_I hops from the transmitter, nodes in $N_i(H_I)$ are disturbed by the transmission.

Let us see now how reservations must be locally performed in every nodes. Let i and j two nodes such as $(i, j) \in E$. Let f be a data flow transmitted from node i to node j which requires n units of bandwidth to be correctly handle. The three following rules must be observed to perform a correct bandwidth reservation for f :

- node i must reserve n units of bandwidth to transmit the data flow f. Its remaining capacity becomes $C_i - n$
- nodes in $N_i(H_I)$ must also reserve n units of bandwidth since CSMA/CA protocols like IEEE 802.11 forbid transmissions, from other nodes, in the interference range of i. As explained in [7], there are some cases where transmissions may be possible in the interference range of i, but this infers a modification of the MAC protocol. Thus, each node k in $N_i(H_I)$ have a remaining capacity $C_k - n$
- if a node k in $N_j(H_I)$ sends data during node i transmission , j will not be able to correctly receive the data from i since k transmission will interfere with the first one. So, each node k in $N_j(H_I)$ must reserve n units of bandwidth and then have a remaining capacity $C_k - n$

Notice that if a node meets several rules for a given transmission, it should only consider one of these rules.
To summarize, the following rule can be used :
nodes in $N_i(H_I) \cup N_j(H_I)$ must reserve n units of bandwidth to correctly handle a data flow between i and j which requires n units of bandwidth [6].
We can see on Figure 1 an example of bandwidth reservation for a data flow, from node A to node E, which requires one unit of bandwidth.
In this example, we take $H_I = 1$. As shown on Figure 1(a), each node starts with 10 units of bandwith. In order to handle the flow between A and B, nodes A, B, C, F and G must reserve one unit of bandwidth.
For B-C flow, nodes A, B, C, D and G must reserve one bandwidth unit, etc...
Admission of a route, for a new data flow that requires n units of bandwidth, will go in two steps :

- *computation* of a route with a given heuristic

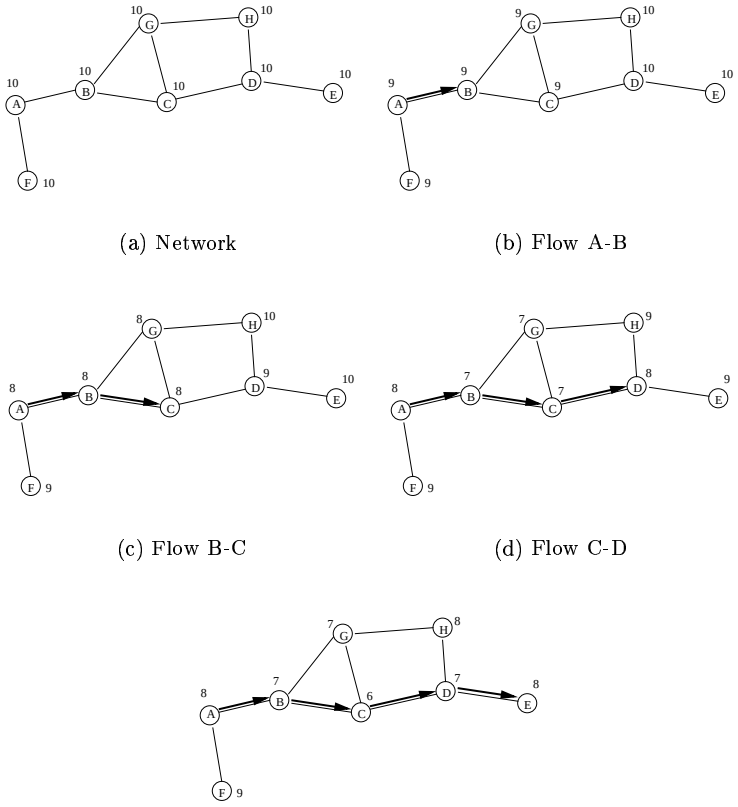


Figure 1: Reservation example for a data flow

- *admission control* performed for each node i located within H_I hops from the route to check whether the remaining bandwidth C_i meets the requirements of this flow. If $C_i - n \leq 0$ then the flow is rejected.

The main aim of this work, is to find QoS heuristics which computes *bandwidth-aware* routes in a wireless multihop network subjected to interferences.

3 Description of heuristics

Before beginning the description of QoS heuristics, it is important to point out what we expect from these heuristics. Many *best-effort* routing protocols use a shortest path algo-

rithm (*e.g.* Dijkstra's Algorithm) to compute routes, but such protocols might not suit our requirements since they do not take into account the constraint of nodes capacity.
First, QoS heuristics must offer a better routes admission capability than a common shortest path algorithm. In other words, we expect the network to handle more trafic flows with QoS heuristics than without.
Next, we want the QoS heuristics to save bandwidth of nodes that have low bandwidth capacity. So, it is suitable that QoS heuristics choose, as a priority, nodes with high capacity. Moreover, it is suitable that routes are not too long : routes may be longer than those computed with a shortest path algorithm (for instance, longer paths may be used to bypass some nodes with low capacity), but it is important to see that the longest a route is, the largest interferences potentially are.
In order to meet these issues, the three QoS heuristics we present in this paper are based on modifications of *Dijkstra's algorithm* in which we introduce the notion of nodes capacity. This implies that a node should have a vision of the network topology. This paper is not concerned with topology discovery but we can imagine an underlying mechanism similar to link-state routing protocols.

3.1 Heuristic based on the remaining capacity of forwarding nodes

In the common *Dijkstra's algorithm*, each edge $(i,j) \in E$ is associated with a weight $w_{(i,j)}$. Denote by $W_{\mathcal{P}}$ the sum of links weights along a path $\mathcal{P} = \langle n_1, n_2, \ldots, n_{l-1}, n_l \rangle$ (i.e. $W_{\mathcal{P}} = \sum w_{(n_i,n_{i+1})}$, $1 \leq i \leq l-1$). The goal of Dijkstra's algorithm is then to minimize $W_{\mathcal{P}}$.
The idea of this first QoS heuristic $H_1$is to choose nodes with regard to their current capacities and to maximize the bandwidth available along the path. In other words we want to minimize the inverse of nodes remaining bandwidth. More precisely, we consider that each node i is associated with a weight w_i equal to the inverse of its remaining bandwidth (*i.e.* $w_i = \frac{1}{C_i}$). The problem amount to *Dijkstra's shortest path algorithm* since the objective of H_1 is now to compute a route $n_1, n_2, \ldots, n_{l-1}, n_l$ which minimizes $\sum \frac{1}{C_{n_i}}$, $1 \leq i \leq l$. Thus, a large weight is allocated to nodes which bandwidth capacity is low. Since the heuristic tries to minimize $W_{\mathcal{P}}$, it is easy to see that computed routes will tend to bypass nodes with low bandwidth capacity. For instance, if $C_i = 0$ then $w_i = \infty$ and then i will never be chosen by H_1 to route QoS-packets.

3.2 Extension of the first heuristic

Denote by H_n the generalization of the first QoS heuristic associated with the weight $w_i = \frac{1}{C_i^n}$, $i \in V$. The idea of the second heuristic is the following : whenever a route discovery fails with H_1 (*i.e.* no admissible route has been found with H_1), another try is performed with H_2. If a failure once again occurs with this new weight, another try is performed with H_3 etc.... One can set a bound B and say that if no admissible route has been found with H_B, the route discovery definitely failed.
In the rest of this paper this *incremental heuristic* is denoted by H_{Inc}.

3.3 Heuristic based on the capacity of forwarding nodes neighborhood

Heuristics H_n are only concerned with capacity of nodes that are located on the routing path. But if we now consider capacity of adjacent nodes, heuristic comes closer to the network model and then nodes have a more accurate vision of their neighborhood effective capacity.
Thus, we propose a third heuristic HN where nodes weights are :

$$w_i = \sum_{j \in N_i(1) \cup \{i\}} \frac{1}{C_j} \; with \; i \in V$$

If a node i has neighbors which bandwidth capacity is low, w_i becomes large and HN will avoid choosing i for routing. For example if $j \in N_i(1)$ and $C_j = 0$, then $w_i = \infty$ and HN will never choose i for QoS-routing. It is important to note that although this heuristic provides a more accurate vision of capacities, it implies a higher computation complexity. Moreover, one can denote by HN_n the heuristic were,

$$w_i = \sum_{j \in N_i(n) \cup \{i\}} \frac{1}{C_j} \; with \; i \in V$$

If we consider that interferences propagates within n hops from the transmitter, HN_n exactly match the interferences constraint of the network but its complexity is much more higher. So we have to find a compromise between computation speed and accurate model matching.

4 Simulation results

In this section we present results of simulations conducted to compare performances of the heuristics we presented on Section III.
In these simulations, we consider a 1000 $m \times 1000$ m flat area. Let n be the number of wireless nodes and r their communication range (*i.e.* a node n_1 can receive data from a node n_2 if the distance between n_1 and n_2 is less or equal to r). Denote by δ the average degree of the network graph and let $\pi = \frac{\delta}{n}$. Whenever π becomes close to 1, interferences tend to propagate in the whole network.
In all these simulations we consider that $H_I = 1$ (*i.e.* interferences propagates within 1 hop from the transmitter).
We conducted several simulations with $1000 \leq n \leq 1500$, $50 \leq r \leq 300$ and we compared performances of Dijkstra's shortest path algorithm, H_1, H_{Inc} and HN_1 heuristics.

4.1 Number of accepted flows

In the first simulations, we consider that all nodes in the network have 500 units of bandwidth. Then, we try to insert, one by one, 1000 QoS-flows, between random sources and

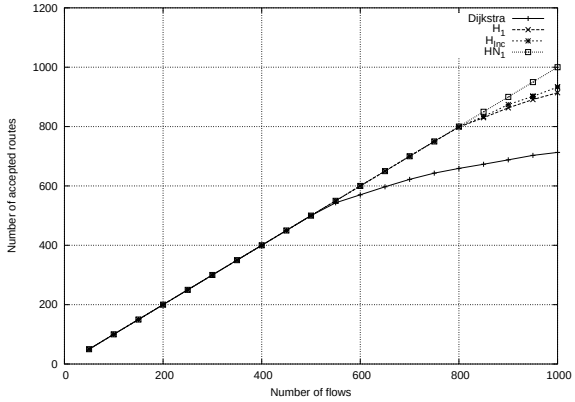


Figure 2: Insertion of 1000 flows for $n = 1300$, $r = 150$ and $\pi = 0.06$.

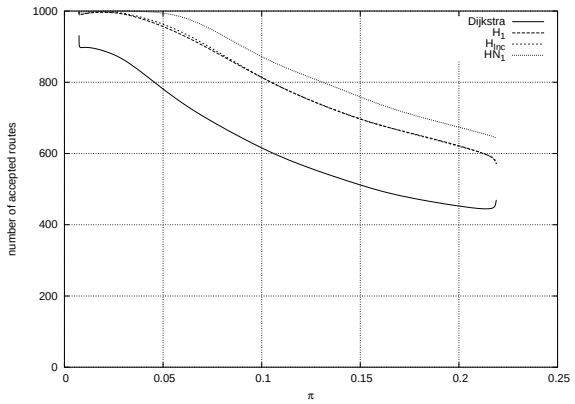
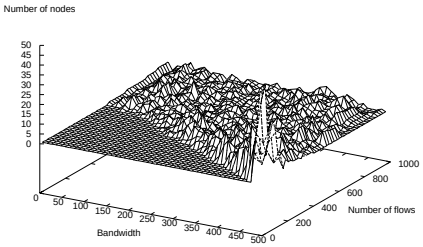
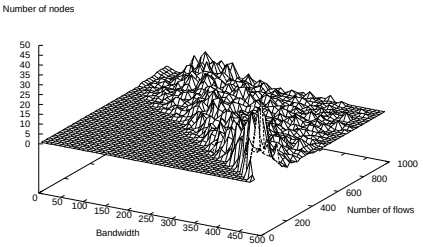


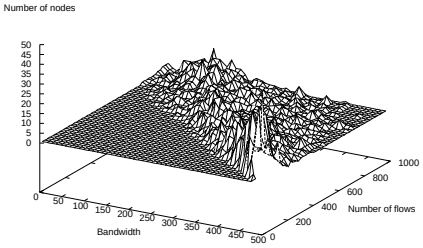
Figure 3: Number of inserted routes for different values of π



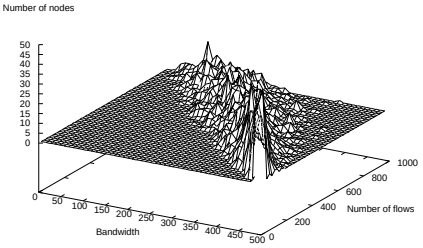
(a) Dijkstra's Algorithm



(b) Heuristic H_1



(c) Heuristic H_{Inc}



(d) Heuristic HN_1

Figure 4: Bandwidth distribution for Dijkstra's algorithm and the three heuristics

destinations. All these flows require the reservation of 1 bandwidth unit to be correctly handle.
Figure 2 shows simulation results for $n = 1300$ and $r = 150$ ($\pi \approx 0.08$). We can see that Dijkstra's algorithm begins to reject QoS-flows from the 400th flow insertion whereas H_1 and H_{Inc} accept flows without any rejection until 700 flows. HN_1 starts rejecting flows from the 900th insertion. At the end of the simulation 650 flows were accepted with Dijkstra's algorithm, almost 900 flows for H_1 and H_{Inc} and almost 980 for HN.
If we only consider the final number of accepted flows according to the parameter π for all our simulations, we achieve results presented on Figure 3. We can see that for all the network configurations we simulated, heuristic H_1,H_{Inc} and HN always handle more QoS-flows than Dijkstra's algorithm. Results for H_1 and H_{Inc} are very similar whereas HN always support more QoS-flows since it provides a more accurate vision of the interferences.

4.2 Bandwidth distribution

Another important point to consider is the evolution of nodes capacities. Figure 4 shows, for each algorithm, the bandwidth distribution within the nodes when new flows are inserted. When the number of inserted flows is low, all nodes have a large capacity. The first highest values have not been drawn on Figure 4 so as to make the representation more readable.
First, we can see that nodes using Dijkstra's algorithm reach a capacity equal to zero faster than with the other heuristics. The convergence to lowest capacity is slower with heuristics since they tend to use nodes with large capacity as depicted on Figure 4(b), 4(c) and 4(d). Dijkstra's algorithm tends to quickly spread bandwidth distribution : we have almost the same number of nodes that have low, medium or large capacity. This leads to creation of congestion areas where flows cannot be routed anymore. On the other hand, H_1, H_{Inc} and HN_1 slowly convergence to low capacities and then offer better routing capabilities. Particularly as shown on Figure 4(d) HN_1 concentrate the bandwidth distribution around a relatively high average value.

4.3 Nodes with different original capacities

For the simulations presented above, each node starts with 500 bandwidth units. We are now interested in the evolution of the bandwidth when nodes do not start with the same capacity. Moreover, we want to show how our heuristics bypass low capacity areas in order to save bandwidth of nodes in these areas.
We take the same 1000 $m \times 1000$ m flat area as depicted above. We draw a square in the middle of this map and we consider that nodes outside the square start with 500 bandwidth units and nodes inside the square start with 250 units of bandwidth.
Then we try to insert 500 QoS-flows, which all require 1 bandwidth unit, with Dijkstra's algorithm, H_1, H_{Inc} and HN_1.
In these simulations we take $H_I = 1$.

Figure 5 represents the insertion of 500 flows for $n = 1000$ and $r = 175$. Here 350 nodes start with 250 bandwidth units and 650 nodes with 500 units. We can see that Dijkstra's

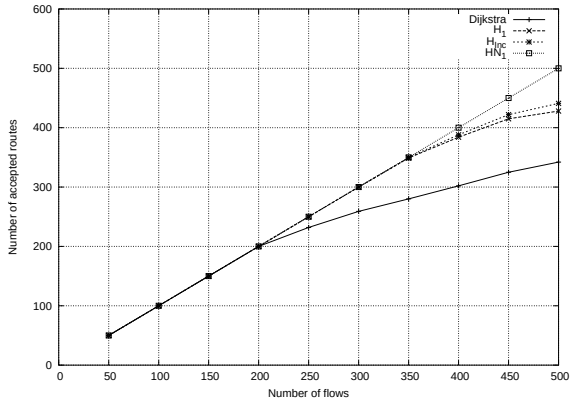
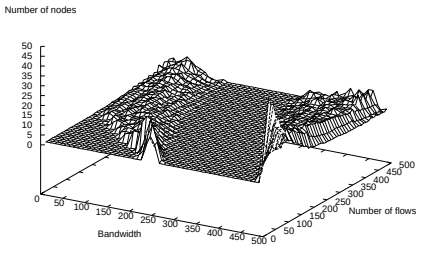


Figure 5: Insertion of 500 flows for $n = 1000$, $r = 175$ with different original capacities.

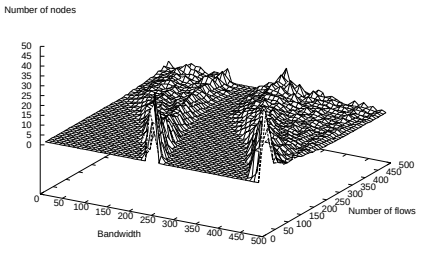
algorithm begins rejecting flows from 200 whereas H_1 and H_{inc} accept the 350 first flows. Heuristic HN_1 accepts almost all the flows. This can be explained by Figure 6 which represents the bandwidth distribution for these simulations. We also truncated the highest values to make the representation more readable. Figure 6(d) clearly shows how heuristic HN_1 saves bandwidth of nodes with low capacity : these nodes do not take part in routing as much as nodes with larger capacity since the lowest capacity value remains between 150 and 250. On the other hand, nodes capacities quickly converge to 0 with Dijkstra's algorithm. Although there are several nodes with large capacity, QoS-flows cannot be routed anymore because of the number of nodes with low capacity. H_1 and H_{Inc} also produce a slow convergence to low capacities.
Figure 7 represents the number of inserted routes for all the simulations we conducted. First, we notice that a lot of routes are accepted with HN_1 : for $\pi = 0.05$ for example, heuristics HN_1 accepts almost 450 routes whereas Dijkstra's algorithm accepts 200 routes.
H_1 and H_{Inc} accept more routes than Dijkstra's Algorithm. Also notice, in this case, the better performance of H_{Inc} in relation to H_1.

5 Conclusion and future work

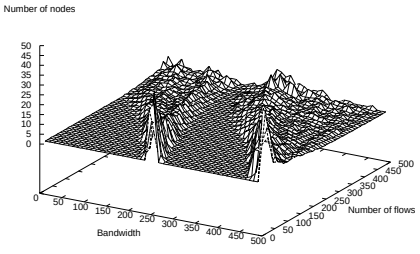
In this work we have presented three heuristics based on Dijkstra's Algorithm providing computation of QoS-routes in wireless ad hoc networks under interferences constraints. We have shown that these heuristics handle more QoS flows by saving low capacity nodes and bypassing low capacity areas.
We can now imagine a proactive QoS protocol where bandwidth informations are periodically broadcast within the network (using, for instance, an optimized flooding mechanism as in [8]) so as to compute routes with a given heuristic and where the QoS flows are source-routed towards the destination.



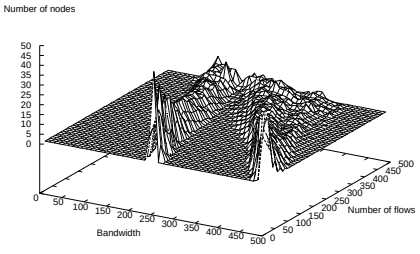
(a) Dijkstra's Algorithm



(b) Heuristic H_1



(c) Heuristic H_{Inc}



(d) Heuristic HN_1

Figure 6: Bandwidth distribution for Dijkstra's algorithm and the three heuristics with different original capacities.

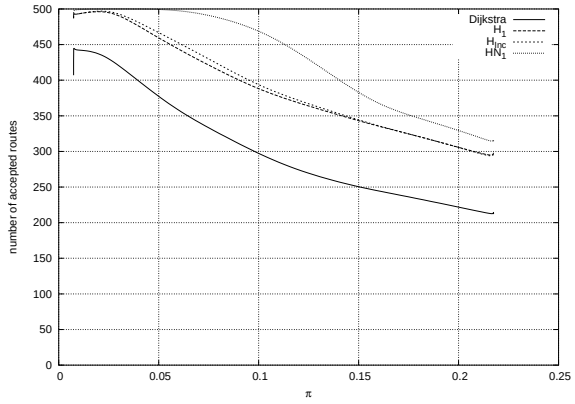


Figure 7: Number of inserted routes for different values of π

In this work we are not concerned with flow scheduling, but we have to find an effective solution to avoid collisions which may occur in such CSMA/CA networks.

References

[1] Shigang Chen and Klara Nahrstedt, "Distributed Quality-of-Service routing in Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, August 1999.

[2] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das, "Quality of Service in Ad hoc On-Demand Distance Vector Routing." IETF Internet Draft.

[3] R. Leung, J. Liu, E. Poon, Ah-Lot. Chan and B. Li., "A QoS-Aware Multi-Path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks," in *26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*, 2001.

[4] Anelise Munaretto, Hakim Badis, Khaldoun Al Agha and Guy Pujolle, "QOLSR : Routage avec QoS dans OLSR," in *AlgoTel 2003*, (Banyuls-sur-mer, France), INRIA, May 2003.

[5] C. Chaudet and I. Guérin Lassous in *European Wireless 2002 (EW2002)*, (Florence, Italy), February 2002.

[6] Leonidas Georgiadis, Philippe Jacquet and Bernard Mans , "Bandwidth Reservation in Multihop Wireless Networks : Complexity and Mechanisms," Tech. Rep. INRIA Research Report RR-4876, INRIA, http://www.inria.fr, July 2003.

[7] A. Velayutham and H. Wang, "Solution to the Exposed Node Problem,"

[8] Philippe Jacquet, Anis Laouiti, Pascale Minet and Laurent Viennot, "Performance analysis of OLSR multipoint flooding in two ad hoc wireless network models," Tech. Rep. INRIA Research Report RR-4260, INRIA, http://www.inria.fr, 2001.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399