BRuIT: Bandwidth Reservation under InTerferences influence

Claude CHAUDET, Isabelle GUÉRIN LASSOUS
Laboratoire de l'Informatique du Parallélisme – ENS Lyon – 46 allée d'Italie – 69007 Lyon
{Claude.Chaudet ; Isabelle.Guerin-Lassous}@ens-lyon.fr
Contact author: Claude CHAUDET, phone: +33 4 72 72 84 70, fax: +33 4 72 72 80 80

ABSTRACT
This paper deals with the bandwidth reservation problem in ad hoc networks and with the influence that interferences between signals have on this problem. We show that interferences could decrease the applications rates. This can be a real problem for applications that need guarantees. We propose a distributed protocol (called BRuIT) for bandwidth reservation in ad hoc networks that takes into account the existence of interferences from far transmissions. The protocol is analyzed through simulations carried out under NS: we evaluate the signaling overhead required for maintaining the knowledge of existing interferences ; we show that this knowledge reduces delays in case of congestion ; we measure the time for rebuilding broken routes ; and finally we show that this protocol maintains the rate of accepted applications.

INTRODUCTION

The ad hoc networks are more and more studied and with the existence of commercial products like wireless network cards, such networks are built in places where wiring is expensive or impossible or when mobility is needed. Moreover the achieved rate of wireless cards is such that it is now possible to realize high rate applications on these networks. This kind of applications often requires guarantees on available bandwidth, small delays and few packets loss. To ensure these constraints, quality of service should be added to the network.

The IETF working group MANET concentrates essentially on routing protocols. Many multicast protocols have also been proposed in the literature. On the other hand, few works have been carried out on the subject. This lack of studies on the subject may be explained by the dynamic aspects of these networks and their bandwidth constraints ([9]). Most of the proposed works concerning QoS aspects in ad hoc networks concentrate on the bandwidth availability (as far as we know, only a single work deals with the delay constraint): the goal is to find and to use a route in the network that meets the application's bandwidth requirement. All the proposed protocols consider the features of the one hop neighborhood in order to provide quality of service: each mobile studies the available bandwidth, the delay and/or the stability with its one hop neighbors to build and maintain the route(s) that provides the requirements specified by the application. But none considers the existence of interferences with signals emitted by mobiles located farther than our transmission range that may have an impact on the quality of the protocol. However, we think that this "extended neighbor" traffic can have a strong influence on the behavior of each mobile (also called node henceforth) of the network: a mobile A can see its rate decreased although no other mobile communicates at the same time in its communication area, but because a distant mobile(with which A can not directly communicate) in the interference area accesses the radio medium at the same time. This is not a problem in the case of applications with no constraints, but guaranteed applications can not be unaware of this phenomenon for fear of being degraded.

In this article, we study the impact of interferences: we propose a bandwidth reservation protocol for ad hoc networks that takes into account the existence of these interferences. We called it *BRuIT* for Bandwidth Reservation under InTerferences influence. We first concentrate on the bandwidth problem, because it is one of the main network parameters. Moreover the bandwidth parameter may affect other parameters like the delay or the jitter for instance. *BRuIT* is a distributed protocol that does not require any control entity. Each mobile periodically determines a set of mobiles that can interfere with it and their respective bandwidth reservation. With this knowledge, each mobile is able to accept (reject resp.) a traffic that will (will not resp.) have an ensured rate all its execution long. The main difficulty lies in the setting up of the interference area of each mobile, i.e. the set of mobiles that interfere with it: how to identify a mobile that interferes but of which the transmission can not be decoded? To begin (note that this work presented here is an ongoing research and has not achieved its final form), we propose to consider for each mobile all the mobiles being at at most k hops, k being a parameter of our protocol. We will discuss this choice in the article, but the carried out simulations show that it is a first step towards the knowledge of interference areas. To evaluate *BRuIT*, we simulate it with the widely used *NS* simulator [8]. With these simulations, we are able to analyze the main features of our protocol.

Section 1 gives a brief state-of-the-art on the QoS aspect in ad hoc networks. Then the simulation presented in Section 2 shows the impact of interferences on on-going applications in terms of rate. To better take into account the interference phenomenon, the protocol *BRuIT* is described in Section 3. The protocol is analyzed through simulations carried out under *NS*. To conclude, we discuss the perspectives to give to our protocol to ensure a very good quality of service in ad hoc networks.

1 QUALITY OF SERVICE IN AD HOC NETWORKS

Some protocols have been proposed for the QoS issue in mobile ad hoc networks. In [11], a first synthesis presents some of these protocols. According to the authors, the proposed protocols can fall into four categories: the QoS models, the QoS MAC (Medium Access Control) protocols, the QoS routing protocols and the QoS signaling protocols.

A QoS model defines the type of services that can be offered in the network and the mechanisms required to realize these services. As far as we know, *FQMM* ([12]) is the only model that has been proposed for ad hoc networks. It mixes the well-known *IntServ* and *DiffServ* approaches: the priority traffic uses the per-flow granularity of *IntServ*, whereas the other traffics use the per-class granularity of *DiffServ*.

A QoS MAC protocol offers QoS guarantees in addition to solve medium collisions and other problems that arise in radio like hidden/exposed mobiles problems. In [1], a differentiation of services is added in the IEEE 802.11 protocol: the authors give a priority level to each frame (in modifying the backoff function or/and in assigning different Inter Frame Spacing). In *MACA/PR* protocol the real-time flows only use the RTS/CTS (Request to send/Clear to send) mechanism of 802.11 once at the beginning of the transmission whereas ordinary flows use one of these exchanges for each packet.

The goal of QoS routing is to find a route between the source and the receiver that meets the constraints specified by the application. The constraints can be the delay, the bandwidth or the transmission cost. In [4], the TDMA (Time Division Multiple Access) medium access mode is used to find a route with sufficient bandwidth available. The number of free TDMA units of the route corresponds to the available bandwidth of it. To solve the hidden mobiles problem, two adjacent links with two different traffics use different time slots. *CEDAR*[10] uses the CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance) medium access mode. This protocol is based on the dynamic election of a core in the network. This core provides information like the bandwidth availability and computes the routing. The use of the core limits the computations and the flooding. Flooding impact reduction is also the goal of the *Ticket Base Probing* protocol [3]. It uses the concept of tickets (the yellow ones seek for routes that meet the constraints whereas the green ones seek for routes with low cost) to limit the number of paths to seek when discovering routes. Therefore, the number of tickets associated to each application corresponds to the priority of the application.

The QoS signaling provides a way to propagate control information through the network. *INSIGNIA* is an inband signaling protocol that reserves bandwidth [6]. The control information (e.g. reservation requests) is included in the IPv4 header of each data packet. Periodic reports are sent by the receiver to allow the source to adapt its rate according the state of the used route. *dRSVP* reserves bandwidth for adaptative applications [7]. All the applications specify the lowest bound on the bandwidth required and the upper bound representing the maximal bandwidth that can be achieved. The reserved bandwidth for an application can be modified during the execution by the network (if resources get scarce or are released) or by the application (to release resources for other traffics).

In all the protocols mentioned here, each mobile accepts or rejects traffic according the state of its one hop neighborhood, i.e. according the available bandwidth, the delay and/or the stability with its one hop neighbors. None of these protocols consider the interferences phenomenon that can occur in ad hoc networks and that can have a strong impact on the rate of the applications as shown in Section 2.

2 INTERFERING TRANSMISSIONS

Transmissions in ad hoc networks are subject to many problems which are on one hand due to the radio interface (radio signals can be absorbed, can fade, can interfere with each other, . . .) and on the other hand due to the lack of centralized administration (routing and multiplexing of transmissions are mainly distributed). When designing a quality of service protocol in ad hoc networks, all these parameters should be considered. Bandwidth reservation protocols need to precisely evaluate how much bandwidth is available in order to be able to accept a reservation request. Otherwise, requests could be accepted while they cannot be properly satisfied. The information collected from the one hop neighbors is insufficient. Indeed, depending on the network density and on the environment, transmissions can interfere at distances up to 4 or 5 times the emission range.

In the example described in Figure 1, two couples of nodes try to communicate using all the available bandwidth. The transmitters are distant of nearly two times the transmission range. This scenario has been simulated in NS[1] using a modeling of the Wavelan 914 MHz cards, which have a maximum link bandwidth of 2 Mb/s. Until the fourth second, the transmission between A and B is alone and gets the whole bandwidth. At time 4s, the second communication between C and D starts and from this time, the two transmissions share the bandwidth even though the nodes are too far away to directly communicate.

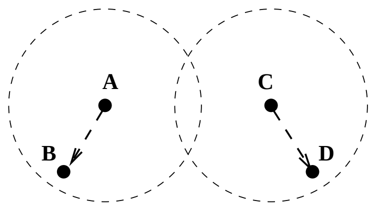

Figure 1: Two pairs of nodes communicate

In wireless networks with access points as well as in cellular networks, bandwidth is shared by allocating different times slots, frequencies or codes to each transmission. This dynamic allocation process is performed by the access points which administrate the local zones. In ad hoc networks, this kind of mechanism is hard to implement due to the lack of central administration. Instead, in order to prevent transmissions to interfere, the nodes

[1]NS: The Network Simulator – http://www.isi.edu/nsnam/ns/

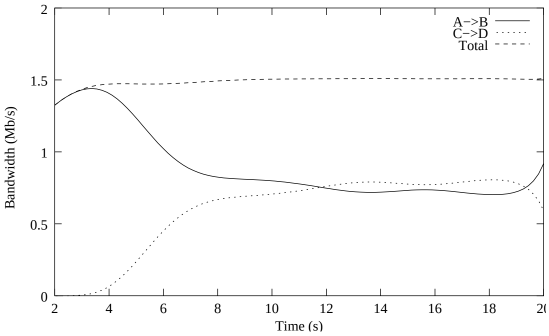

Figure 2: Interferences influence on bandwidth

share the medium using CSMA / CA protocol which prevents nodes that are in communication range to interfere but does not solve the distant interferences problems. If tuning the carrier sense threshold can allow to share the medium at a larger distance, jammers cannot be identified and the available bandwidth cannot be computed. Therefore it is not accurate enough for bandwidth reservation protocols since it does not provide good estimations of available bandwidth.

3 BRuIT – PROTOCOL DESCRIPTION

In [2], we proposed a bandwidth reservation model for ad hoc networks that takes into account the whole knowledge of interferences. We showed that this problem is NP-complete and we gave some heuristics to solve it with the associated theoretical evaluations. The proposed model assumes that each mobile has a whole knowledge of the network. Thus, the protocol that can be directly derived from this model requires a central administration that can induce large delays in ad hoc networks. In this paper, we deal with the distributed context of these networks.

In order to solve the problem of interferences caused by distant nodes, we tried to bring a knowledge of the neighborhood to the nodes of ad hoc networks. *BRuIT* is a distributed signaling protocol which achieves this goal by periodically sending messages containing information on bandwidth availability and provides a mechanism to reserve bandwidth for transmissions. *BRuIT* was implemented over a simple reactive routing protocol, but the signaling system can easily be adapted over a proactive or hybrid routing scheme.

3.1 Neighborhood knowledge

The first task performed by *BRuIT* is to provide to the nodes information about their neighbors. Each node periodically broadcasts a message (called *Hello* packet) to every other node that can hear it (i.e. that is in its communication range), as shown on Figure 3. This packet contains the address of the transmitter and the total bandwidth that it will use to route the already accepted privileged flows.

Because communications can interfere from much farther than the transmission range of nodes, we need to propagate information precisely, i.e. on an area larger than the one hop neighborhood. That is why each *Hello* packet not only includes information about the transmitter but also about every node at a distance of k hops from the transmitter. k, width of the extended neighborhood that we consider (in other words the propagation range of the information) is a parameter of the protocol. The *Hello* packets are propagated within two hops in Figure 4.

Upon reception of such a message, a node can compute the remaining bandwidth it can use for new flows. Therefore, the admission control process which decides if a new request is accepted or refused can be executed more accurately.

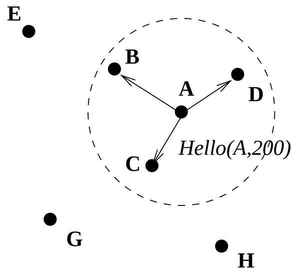

Figure 3: Node A locally broadcasts information on its identity and used bandwidth

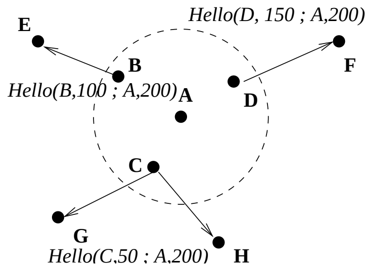

Figure 4: The information transmitted by node A is forwarded to nodes at two hops from A

3.2 Bandwidth reservation protocol

Using the information gathered by the reception of *Hello* packets, the nodes can perform admission control on each bandwidth reservation request.

Whenever a node wishes to reserve bandwidth for a flow, if floods the network with a route request containing the amount of bandwidth desired. Admission control is made on each node that receives the request and nodes do not forward it if they have not enough bandwidth available.

The forwarding process of such a message described above does not actually reserve any bandwidth in the intermediate nodes. Each node only takes note that a route request has been sent from a certain transmitter towards a receiver and stores the address of the node from which it received the route request.

Multiple techniques may be applied to limit the impact of this flooding ([5], [3]). Note that we do not intend to propose a new routing protocol, but our goal is to use a simple routing protocol that allows us to concentrate on the interferences impact. The proposed reservation scheme can be adapted to more accurate routing protocols for ad hoc networks (and this is one of our future work).

When the route request message reaches the destination, the same check is performed. If there are enough free resources, the receiver emits a route reply message towards the source of the flow that travels the way back to the transmitter, reserving resources in the intermediate nodes on its way. Upon reception of a route reply message, each node checks if it still can handle the bandwidth request (otherwise the reply is dropped), decreases its free bandwidth counter, stores the address of the next node on the route and forwards the message. When the transmitter finally receives the route reply notification, the transfer of data begins.

3.3 Soft state maintenance

Many issues in wireless ad hoc networks are due to nodes mobility. When nodes move, routes can be broken and resources cannot always be explicitly released. To deal with mobility, every information maintained by BRuIT has an expiration date.

The nodes cannot forecast their movements. Moreover, we cannot rely on the power of received signals to anticipate nodes disappearing because a node can suddenly get out of sight by moving behind a wall. That is why nodes regularly send *Hello* packets. A node can conclude that another node is not its neighbor anymore when it has not received any *Hello* packet from it for a certain time, corresponding to the loss of a certain number of *Hello* packets. This delay should not be too small because the loss of a signaling packet can occur frequently in wireless networks even though the transmitter has not disappeared. For broadcasted messages, there is no RTS/CTS scheme, so two *Hello* packets can sometimes collide. The delay should neither be too long otherwise the nodes will have an out-of-date view of the network topology. We set this value equal to half a second (as we will see in Section 4 it corresponds to the loss of five *Hello* packets; we think that this value is not too high, so the protocol reacts fast enough to mobility issues, but it is also not too low otherwise it would react too quickly to radio phenomena).

When a node in a used route moves, it can go out of the range of the previous node on the route. In this case, the route is broken and must be rebuilt. The bandwidth that was used for this flow should be released. Routes are maintained by soft-state too. When no data has been sent during a certain time, the intermediate nodes conclude that the route is not in use anymore and automatically release the used bandwidth. Here again, the value of the timeout should be chosen carefully. A too small value is not suited because transmitters should keep the ability not to transfer all the time. A too great value will cause reservations requests to be dropped because the nodes believe the bandwidth is used whereas it has been released by previous flows. The actual implementation of *BRuIT* discards routes if no data has been sent for longer than half a second (the same value as the one used for the setting up of the neighborhood).

3.4 Filtering the flows

Our bandwidth reservation scheme relies on the knowledge of the amount of bandwidth used by the flows in the network. If an application asks for a certain bandwidth on a route and uses more during its execution, the information propagated though the network will be false and interferences can happen again. That is why every transmitter is required to filter its flows. The mechanism used in the actual implementation of *BRuIT* is token bucket filters.

4 BRuIT – SIMULATION AND EVALUATION

In order to evaluate our protocol, we simulated it under NS, a network simulator widely used in the scientific community. NS proposes an implementation of different radio propagation models including *Two-ray ground model*. If this model is rather simple compared to the real radio waves propagation scheme, it is accurate enough to bring out some properties of *BRuIT*. NS also offers a modeling of the 802.11 medium access layer and some routing protocols for ad hoc networks.

We give the first results that allow the evaluation of the main parameters of the protocol. To begin, we set the parameter k to the value 2. We based this choice according to the simulation presented Section 2 that also shows that each pair recovers the maximum bandwidth as soon as they are at a distant equal to two times the communication range. Nevertheless, we intend to carry out more simulations with other values for k.

4.1 Signaling overhead

The first characteristic we study is the bandwidth consumed by the sending of *Hello* packets. As mentioned in Section 3, *Hello* packets are locally broadcasted regularly by each node in order to react to mobility of nodes. Sending too few packets gives to the nodes an outdated view of the network topology and of the used bandwidth whether sending too often would consume too much bandwidth.

Figure 5 shows the maximum bandwidth obtained by a communication between two nodes alone in the network. Different routing protocols implemented under NS (AODV, DSDV, DSR and TORA) are tested: they all achieved a maximum rate of 1.5 Mb/s between the two nodes (the theoretical link bandwidth of the modeled interface cards is 2 Mb/s). *BRuIT* sends the data and the *Hello* packets between the two nodes. Figure 5 gives the achieved rate by *BRuIT* for the data packets without considering the *Hello* packets. If each node sends a single *Hello* packet per second, the maximum bandwidth is likely the same as when using routing protocols without bandwidth reservation support. When sending ten *Hello* packets per second, the maximum bandwidth is decreased by about 20 kb/s which represents 1.3% of the total bandwidth at application level. The value of one *Hello* packet per 100ms consumes an acceptable amount of bandwidth and allows the protocol to react quite well to nodes mobility. We will use it from now on.

Signaling overhead depends on the frequency of the broadcasts but it also depends on the size of the *Hello* packets. The denser the network is, the larger the *Hello*

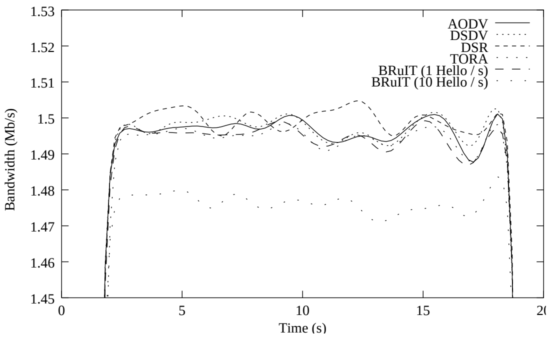

Figure 5: Maximum bandwidth between two nodes

packets will be. On Figure 6, we measured the signaling cost of the protocol on a communication between two nodes when each node has 0, 1 or 2 neighbors.

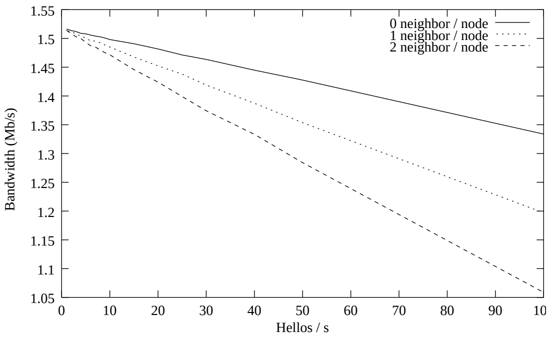

Figure 6: Maximum bandwidth between two nodes depending on the number of neighbors of per node

We can deduce that when a node sends *Hello* packets, it represents a loss of maximum bandwidth of approximately $(2 + 0.12 \times n) \times f$ kb/s where n is the number of neighbors of the node and f is the number of *Hello* packets sent per second. Note that with networks with at most ten neighbors per mobile and with a frequency of ten *Hello* packets per second, the signaling packets represent at most 10% of the maximum bandwidth.

4.2 Effect on delays

Whenever a reservation request succeeds, a token bucket filter is created by the transmitter and used to be sure the flow does not exceed the bandwidth it obtained. Filtering the data flows allows to control and avoid congestions in the network. When the bandwidth is not fully used, the filtering has no particular effect on the delays, except that we do have a quite good knowledge of the traffic profile in the network. As long as there is no congestion, all the protocols (the routing protocols mentioned previously and *BRuIT*) show the same performances in terms of delay. Now, if we have a flow using enough bandwidth to swamp the network resources, the network interface queues will be more filled and packets will spend more time in each intermediate node. The results of the simulation of a flow trying to get full bandwidth between two nodes is show on Figure 7. When few packets are emitted, there is a delay with *BRuIT* that corresponds to the time required for the filtering. When there are more and more packets emitted, the congestion is partly avoided in *BRuIT* with the filtering and the delay remains constant, whereas the delay increases with the other routing protocols.

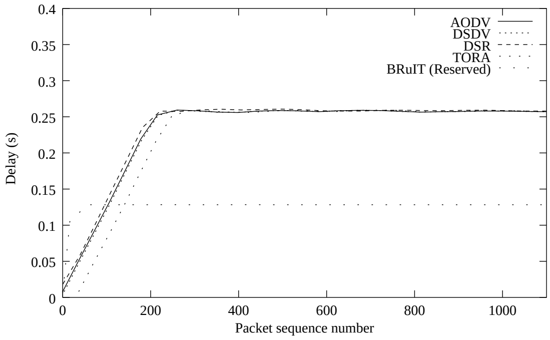

Figure 7: Filtering the flows has positive influence on delays when congestion appears

4.3 Rebuilding broken routes

One of the greatest issues in mobile ad hoc networks is routes breaking due to nodes mobility. When a route is broken, it should be rebuilt as fast as possible to avoid packets losses. This is a tricky part because we have to wait a little before concluding a node has disappeared otherwise routes will be rebuilt too often. As the routes establishment is a rather long process, it should not be too frequently done. When a node leaves, breaking a route, its predecessor on the route waits half a second before concluding it has vanished. Then, this node sends to the transmitter a message indicating that the route is broken. When receiving this message, the transmitter broadcasts a route request message again, initiating a new discovery process. To test this situation, we simulated the scenario presented in Figure 8. Node A transmits a 200 kb/s flow to node F. After five seconds, the node C gets out of the range of node B and returns quickly. We can see on Figure 9 a little loss of bandwidth. At date $t = 10s$, it goes out of the transmission range of B with a speed of 60km/h. The route request process is initiated and it takes two seconds to rebuild the route (D is inserted in the new route).

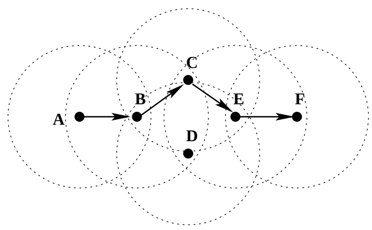

Figure 8: Scenario illustrating a route reconstruction due to mobility

4.4 Guarantees with *BRuIT*

In the previous sections, we have seen the main characteristics of the implementation of *BRuIT*. We have not shown so far how *BRuIT* enhances bandwidth reservation

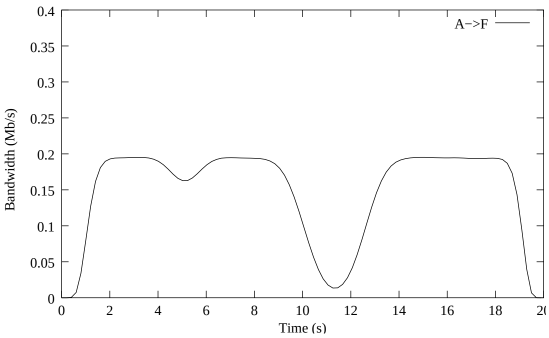

Figure 9: Time needed to rebuild a route broken due to mobility

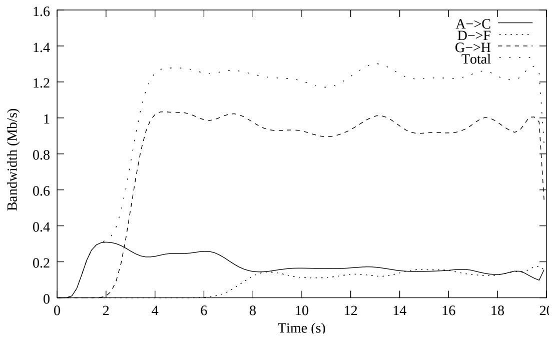

Figure 11: Simulation of the scenario of Figure 10 using a protocol without interferences knowledge

and how it deals with the interferences problem. To illustrate this, we simulate the scenario shown on Figure 10. In this simulation, three flows can interfere. First, node A makes a reservation for 300 kb/s to node C and sends a constant bit rate flow as soon as the reservation is granted. At time 2s, node G asks to have a reservation towards node H for a constant bit rate flow of 1Mb/s. Finally, at time 6s, node D makes a reservation for a 300 kb/s flow towards node F. The networks has a limited bandwidth at application level of about 1.6 Mb/s. At time 6s, we have four nodes sending (or forwarding) data at 300 kb/s and one sending data at 1Mb/s. The total throughput of the transmitters exceeds the medium bandwidth.

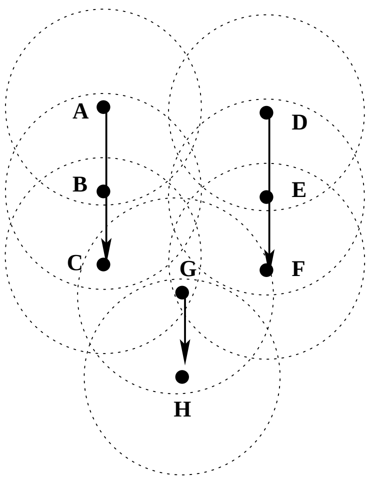

Figure 10: Three transmissions can interfere: A to C at 200 kb/s, D to F at 200 kb/s and G to H at 1Mb/s

If this scenario is executed with a reservation protocol that is not aware of the interferences problem, the three reservation requests will be accepted by the network, because the nodes have no knowledge of their "extended neighbors' flows". The results of the simulation of this scenario are shown on Figure 11. We clearly see that though the three reservations requests have been accepted, the bandwidth cannot be guaranteed, at least for A and D. This can be a real problem when an application adapts its emissions to the granted bandwidth information provided by the network.

Now, if we simulate the same scenario with *BRuIT*, the second reservation request (from G to H) is refused because the nodes are aware that they will not be able to route as much data. The reservation from D to F is then accepted. The results of this simulation are presented on Figure 12. When the network accepts a bandwidth reservation, applications can use this information because the bandwidth availability can be guaranteed.

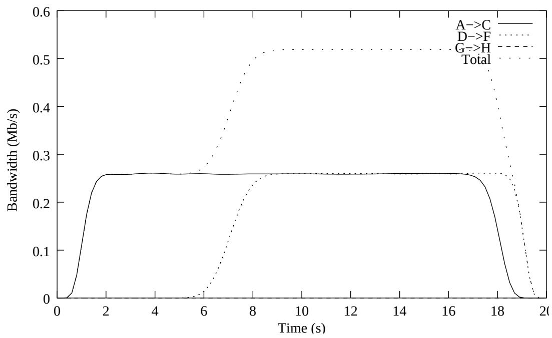

Figure 12: Simulation of the scenario of Figure 10 using *BRuIT*

CONCLUSION

In this paper, we presented the main features and some simulation results of our bandwidth reservation protocol *BRuIT* for ad hoc networks. *BRuIT* allows accurate bandwidth reservation by transmitting informations on the load of the radio medium.

Bringing to the mobiles the knowledge of the used bandwidth by the other mobiles that may interfere with their transmissions allows the admission control procedure to be more exact. When a bandwidth reservation request is accepted by the network, the application has the guarantee that the bandwidth will be available as long as the route is valid.

The simulations showed that *BRuIT* generates few signaling overhead, has a positive influence on transmissions delay by controlling the network congestion and reacts quite fastly to the breaking of routes due to mobility. Moreover, it guarantees the rate of accepted applications.

Nevertheless, much can still be done to improve our protocol. First of all, if NS is a simulator widely used in the academic community, we need to do a real implementation of *BRuIT*. Radio propagation models included in NS are quite good models for outdoor environment but they are inaccurate for indoor propagation. Moreover, radio waves propagation cannot easily be modeled by a software due to the complexity of the phenomenon.

We also have to find a way to identify interfering nodes more precisely. Actually, we consider that two nodes distant of less than a certain number of hops can interfere. Ideally, we should identify interfering nodes by the received signal power. Nevertheless, this can be hard to do because a node cannot identify the transmitters outside of its receiving range.

Moreover, rebuilding routes is actually initiated by the transmitter of the flow but it could be done locally where the route is broken. We also have to find a better way to route best effort traffics which could skew our estimations on remaining bandwidth. Finally, as long as we locally broadcast topologic information, we could use this information in a proactive or a hybrid routing scheme.

Nevertheless, the first results obtained with BRuIT are encouraging and above all, they confirmed that long distance interferences are a real problem for bandwidth reservation schemes.

REFERENCES

[1] Imad Aad and Claude Castelluccia. Differentiation Mechanisms for IEEE 802.11. In *IEEE Infocom 2001 conference*, April 2001.

[2] K. Bertet, C. Chaudet, I. Guérin-Lassous, and L. Viennot. Impact of Interferences on Bandwidth Reservation for Ad Hoc Networks: a First Theoretical Study. In *accepted to IEEE Globecom'01*, San Antonio, Texas, USA, November 2001.

[3] Shigang Chen and Klara Nahrstedt. Distributed Quality-of-Service Routing in Ad-Hoc Networks. *IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad Hoc Networks*, 17(8):1488–1505, August 1999.

[4] Richard Lin Chunhung and Liu Jain-Shing. QoS Routing in Ad Hoc Wireless Networks. *IEEE Journal on Selected Areas in Communications*, August 1999.

[5] Philippe Jacquet, Paul Muhlethaler, Amir Qayyum, Anis Laouiti, Laurent Viennot, and Thomas Clausen. Optimized Link State Routing Protocol. Technical report, IETF Internet Draft (work in progress)., march 2001.

[6] Seoung-Bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, and Andrew T. Campbell. INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Network. *Journal of Parallel and Distributed Computing (Academic Press) – Special issue on Wireless and Mobile Computing and Communications*, 60(4):374–406, 2000.

[7] M. Mirhakkak, N. Schult, and D. Thomson. Dynamic Quality-of-Service for Mobile Ad Hoc Networks. In *IEEE MobiHoc 2000*, Boston, Massachusets, USA, August 2000.

[8] The VINT Project. The Network Simulator – ns-2.

[9] M. S. Scorson. Issues in Supporting Quality of Service in Mobile Ad Hoc Networks. In *IFIP 5th Int. Workshop on Quality of Service*, May 1997.

[10] Prasun Sinha, Raghupathy Sivakumar, and Haduvur Bharghavan. CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm. *IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad Hoc Networks*, 17(8):1454–1465, August 1999.

[11] Kui Wu and Janelle Harms. Qos Support in Mobile Ad Hoc Networks. Not yet published.

[12] Hannan Xiao, Winston K.G. Seah, Anthony Lo, and Kee Chaing Chua. A Flexible Quality of Service Model for Mobile Ad Hoc Networks. In *IEEE Vehicular Technology Conference*, pages 445–449, Tokyo, Japan, May 2000.