Bandwidth Reservation in Multihop Wireless Networks: Complexity and Mechanisms

Leonidas Georgiadis, Philippe Jacquet, Bernard Mans

▶ To cite this version:

HAL Id: inria-00071707

https://hal.inria.fr/inria-00071707

Submitted on 23 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

I N R I A

Bandwidth Reservation in Multihop Wireless Networks: Complexity and Mechanisms

Leonidas Georgiadis — Philippe Jacquet — Bernard Mans

N° 4876

THÈME 1

Rapport de recherche

Bandwidth Reservation in Multihop Wireless Networks: Complexity and Mechanisms

Leonidas Georgiadis[*] , Philippe Jacquet[†] , Bernard Mans[‡] [§]

Abstract: We show that link interferences in wireless networks make multihop bandwidth reservation in such an environment an NP-complete problem. This is in sharp contrast to bandwidth reservation in wireline networks where efficient polynomial algorithms exist. We also consider the problem of slot allocation according to bandwidth requirements in a wireless slotted environment. In the context of Mobile Ad Hoc Networks, we propose simple heuristics applicable to the OLSR routing protocol in order to find routes that satisfies Quality of Service requirements without impairing the performance of other connections.

Key-words: Mobile ad hoc, bandwidth reservation, QoS, complexity, NP completeness

Réservation de bande passante dans les réseaux sans fil à routage par sauts multiples : complexité et mécanismes

Résumé : Nous montrons que les interférences entre les liens d'un réseau mobile ad hoc font que la réservation de bande passante sur plusieurs sauts devient un problème NP-complet. Ce résultat montre un fort contraste avec les résultats dans les réseaux filaires où des algorithmes de réservation optimaux sont polynomiaux. Nous considérons aussi le problème de l'allocation des slots en complément de la réservation de bande passante dans le contexte spécifique des réseaux sans fil de type slotté. Nous regardons aussi l'applicabilité de ces résultats dans les réseaux mobiles ad hoc et en particulier sur le protocole de routage OLSR. L'objectif est d'effectuer un contrôle d'admission des demande de connexion en cherchant une route qui satisfasse la qualité de service de la nouvelle connexion sans altérer les qualité de service des anciennes connexions.

Mots-clés : Mobile ad hoc, réservation de bande passante, QoS, complexité, NP complétude.

1 Introduction

Multimedia flows need to satisfy certain performance requirements, such as, bandwidth guarantees and upper bounds on packet losses and end-to-end delay in order to function properly. These requirements are generally known as Quality of Service (QoS) guarantees. Satisfaction of QoS guarantees has been an active area of research in wireline networks for several years. The same issue becomes more challenging in radio mobile networks where capacity is a scarce resource and packet transmissions are more prone to errors, noise and interference than wireline networks. This is especially true in wireless ad hoc networks, where packets are relayed by mobile nodes, rather than a base station as in cellular systems.

Two common mechanisms of providing QoS guarantees are, admission control and resource reservation. When a new flow is to be set between a source and a destination the source examines whether there are enough resources in the network so that the flow can be admitted and be provided with the required QoS, without affecting the QoS of the already admitted connections (Admission Control). If these resources can be found, they are reserved for the flow (Resource Reservation) and transmission begins.

In this work we are concerned with bandwidth as a performance requirement of a flow. We assume that a flow needs a certain bit rate in order to satisfy its QoS. We are interested in providing these bandwidth requirements to flows in a wireless ad hoc network.

In wireline networks, each link is physically isolated from the other links even if they are attached to the same node. Therefore admission control for a flow consists in finding a path from the source to the destination such that the path links have enough remaining capacity to satisfy the flow's bandwidth requirements. By "remaining capacity " is meant the link capacity minus the bandwidth already reserved for existing flows. Each node can advertise the remaining capacity of its links by periodically broadcasting this information in the network. Finding a path that satisfies the bandwidth requirements of a flow in a wireline network can be done using a simple modification of Dijkstra's shortest path algorithm in $O(n \ln n + m)$ time, where n is the number of nodes in the network and m the number of links.

In a wireless ad hoc network environment, the situation is different and much more complicated. The links are not isolated: traffic carried by neighbor links may interfere, especially if they are operated on the same channel-frequency, as it frequently happens with existing WLANs. A transmission interferes with nodes that are within a number of hops, H_I, from the transmitter and receiver, depending on the signal to noise ratio required for a correct reception. More specifically, for a successful transmission from node i to node j, it must be ensured that no node within H_I hops from receiver j are *transmitting* at the same time. Moreover, in order to ensure that i 's transmission does not interfere with other ongoing transmissions, it must be ensured that no nodes within H_I hops from i are *receiving* at the same time. Therefore reserving a unit of bandwidth for transmission from i to j requires reservation a unit of bandwidth from all receiving nodes within H_I hops from i and all transmitting nodes within H_I hops from j. This constraint is the key that greatly complicates the task of the flow admission control in the wireless ad hoc network.

In this paper we address the problem of bandwidth reservation in wireless networks. We show that when $H_I > 0$ (the case $H_I = 0$ corresponds to a wireline network), this problem is NP-hard. Moreover, it cannot be approximated in polynomial time within n^ε, $\varepsilon > 0$, provided that $P \neq NP$. For a slotted wireless system we describe a mechanism for admission control and bandwidth reservation along a given path, that ensure that each flow receives its requested bandwidth once it is admitted by the network. Finally, we present some applications that are adapted to the mobile ad hoc routing protocol such as OLSR and hint at further studies.

It was already known [1] that the global optimization of flows admitted in a network, assuming that all flow requests are available at the beginning of time, is an NP-hard problem, as it reduces to the multiknapsack problem. However the complexity of admission of a new request was not known. The novel element brought by the present paper is the fact that admission of a single new flow in the network is also NP-complete.

2   Problem Specification and NP-completeness

2.1   Node Bandwidth Capacity

Formally, consider an undirected graph $G(V, E)$ modelling a wireless network. Link (i, j) means that nodes i and j can communicate. Let $N(j)$ be the neighbors of node j. Denote by $N_j(h)$ the set of nodes that are at most h hops away from node j (including node j), *e.g.*, $N_j(1) = \{\{j\} \cup N(j)\}$. Node i has bandwidth C_i, i.e., it can send a message to its neighbors at rate C_i.

In the general case, when node i transmits a packet to its neighbor j, no nodes in $N_i(H_I)$ can receive and no nodes in $N_j(H_I)$ can transmit packets. Therefore, reserving b_f units of bandwidth on nodes i and j for the transmission from i to j, results in reserving units of bandwidth on the nodes in $N_a(H_I) \cup N_b(H_I)$. Determining the minimal units that need to be reserved is complicated and depends on the selected end-to-end flow path (see Section 3 for some of the difficulties involved). Instead, in this section we consider a simplified bandwidth reservation process determined by the following rule.

> **Rule A:** Whenever b_f units of bandwidth need to be transmitted between
> nodes i and j, reserve b_f units of bandwidth on all nodes in $N_i(H_I) \cup N_j(H_I)$.

As will be seen in Section 3 this rule may overestimate the bandwidth requirements and does not always guarrantee the existence of an appropriate mechanism that schedules packets according to the bandwidth reserved for the flow to which the packets belong. However, an extension of this rule, does guarrantee the existence of appropriate scheduling mechanisms. Moreover, using this extension, relatively simple bandwidth reservation mechanisms can be implemented without the need to carry inordinate amount of information throughout the network. We use Rule A in our presentation of NP-completeness results in order to simplify the discussion.

2.2 Incremental check

The network is supposed to correctly function if the remaining bandwidth is positive on each of its nodes. Therefore if traffic of a new flow f of bandwidth b_f is to be carried over the link (i, j), one must check that the capacities of all nodes in $N_i(H_I) \cup N_j(H_I)$ are not exhausted. To this end one must remove from the remaining bandwidth of each of the nodes in $N_i(H_I) \cup N_j(H_I)$ the quantity b_f. The path is acceptable if this operation leaves all remaining capacity of all nodes non-negative. For the correct functioning of the network one may need that the remaining capacity is always above a non-zero fraction of the nominal capacity, which is easy to check too.

Assume that a path $\Pi = \{v_1, \cdots, v_k\}$ from $s(= v_1)$ to $t(= v_k)$ is established in order to transmit a unit of bandwidth. It is easy to see that for this established path, the bandwidth b_i required to be reserved on node i is equal to the number of edges in the established path Π that are reachable within one hop from i. Indeed, each of such edges is involved in a transmission and requires one unit of bandwidth to be reserved.

Formally, for any node i, $b_i = |E_\Pi(i)|$ where $E_\Pi(i) = \{(v_j, v_{j+1})\}, 1 \le j < k$, such that $\{v_j, v_{j+1}\} \subseteq \Pi$ and $(v_j \in \{\{i\} \cup N(i)\}$ or $v_{j+1} \in \{\{i\} \cup N(i)\})$.

Assume that node i has capacity C_i. We must have (basic constraint)

$$C_i \ge b_i, \ i \in V \tag{1}$$

We can now re-define formally our Bandwidth Admission Control into a "Path with Remaining Capacity problem" as follows.

Definition 1 *The Path with Remaining Capacity problem (RC) is defined as:*
***Instance:** A Graph $G = (V, E)$, two vertices s and t from V, and a Capacity $C_i \in \mathbb{N}$ for each vertex i from V.*
***Question:** Is there a simple path from s to t in G that satisfies the constraint $C_i \ge b_i, \ i \in V$?*

In the following section, we prove that, because of constraint (1), this problem is NP-complete.

2.3 NP-completeness

We first recall the definition of the *Path with Forbidden Pairs problem (PFP)*.

Definition 2 *The Path with Forbidden Pairs problem (PFP)(see GT54 in [3, 2]) is defined as:*
***Instance:** A Graph $G = (V, E)$, two vertices s and t from V, and a collection $C = \{(a_1, b_1), \ldots, (a_m, b_m)\}$ of pairs of vertices from V.*
***Question:** Is there a simple path from s to t in G that contains at most one vertex from each pair in C?*

The PFP problem is known to be NP-complete (see GT54 in [3]). Variants of this problem exist where a measure is the length of the path (*i.e.*, the number of edges in the

path): the shortest feasible PFP and the longest feasible PFP problems. Not surprisingly both variants are NP-complete and also NPO PB-complete [4], *i.e.*, in polynomial time, they cannot be approximated within n^ε for some $\varepsilon > 0$, where n is the size of the input, provided that $P \neq NP$.

Theorem 3 *With $H_I = 1$, the Path with Remaining Capacity problem (RC) is NP-complete.*

Proof. It is easy to see that $RC \in NP$, because a nondeterministic algorithm needs only to guess a path and check in polynomial time that the constraint $C_i \geq b_i$, $i \in V$ is valid. (Of course, checking the constraint for the vertices of the path and for their neighbors is sufficient.)

We now give a polynomial reduction from this problem to the Path with Forbidden Pairs problem (PFP).

We first transform an instance $(G = (V, E), s, t, C)$ of the PFP problem in an instance $(G' = (V', E'), s, t, C')$ of the Remaining Capacity problem by formally defining:

- $V' = V \cup \{v_{xy} | (x, y) \in C\}$,

- $E' = E \cup \{(x, v_{xy}), (y, v_{xy}) | (x, y) \in C\}$,

- s and t are unchanged,

- C' is the capacity defined as: $C'_i = 2$ for $i \in \{v_{xy} | (x, y) \in C\}$ and $C'_i = |V|$ otherwise.

Informally, G' contains all the vertices of G as well as m vertices representing each forbidden pairs. Let us define F as the set of m vertices in G' representing each forbidden pairs of C. Each vertex of F is only connected to its two respective "forbidden" vertices and is assigned a capacity equal to 2. All other vertices are assigned a capacity equal to $|V|$ that is larger than any possible vertex bandwidth induced by a given established path. Of course, without loss of generality, we assume that (s, t) is not a forbidden pair.

We now prove that a solution of this instance of the RC problem is a solution if and only if it is a solution for the original instance of the PFP problem. Let first assume the length of this solution path is at least 3.

It is easy to see that a solution path Π from s to t for the RC problem in $(G' = (V', E'), s, t, C')$ does not include any of the vertices of F, as each vertex v of the path (except, possibly, s and t) requires $b_v \geq 3$. Hence this path Π is also a path Π in G.

Furthermore, none of the forbidden pairs are included in the solution path. Otherwise there would exist a pair of vertices a and b (with $(a, b) \in C$) that both belong to the solution path. This would imply that the vertex x_{ab} is reaching more than 2 edges of the path within 1 hop and that the basic capacity constraint for the vertex x_{ab} is not valid as the bandwidth

$$b_{x_{ab}} \geq 3 > C_{x_{ab}} = 2,$$

thus leading to a contradiction. Hence a solution for the RC problem in $(G' = (V', E'), s, t, C')$ is a solution for the instance $(G = (V, E), s, t, C)$ of the PFP problem.

Conversely, given a solution path Π' for the instance $(G = (V, E), s, t, C)$ of the PFP problem, we can verify that the path Π' is a feasible solution path for the the RC problem in $(G' = (V', E'), s, t, C')$. Obviously, Π' is a simple path of G' as G is a subgraph of G'. Hence, we just need to verify that the bandwidth induced by the path respects the capacity constraint. It is clear from the reduction that only the nodes of F may jeopardize the feasibility of the solution, as they may not have sufficient capacity. Again, none of them belong to the path. A vertex of F can be a neighbor of a vertex of Π', incurring a bandwidth of at most 2. However, a vertex of F cannot be a neighbor of two nodes in the paths as it can only be neighbor of its forbidden pair which would contradict the feasibility of Π' for the instance $(G = (V, E), s, t, C)$ of the PFP problem.

Finally, in the case that the length of the solution path is smaller than 3, it is easy to check the two remaining possibilities. When the path is direct from s to t, the only forbidden pair of interest is (s, t) which is trivial. When the path is of length 2, say $\Pi = \{s, v, t\}$, we must also check that (s, v) and (v, t) are not forbidden pairs. All cases take a constant time to check. ∎

As the proof follows a polynomial reduction from this problem to the Path with Forbidden Pairs problem (PFP), an immediate corollary follows.

Corollary 4 *With $H_I > 1$, the Path with Remaining Capacity problem (RC) is NP-complete.*

For other considerations, we may also request that each path reserved is a shortest path. Using the result of Kann [4] on the shortest feasible PFP, it is easy to deduce the following corollary.

Corollary 5 *With $H_I > 1$, Shortest Path with Remaining Capacity problem (SPRC) remains NP-complete and is also hard to approximate (i.e., NPO PB-complete).*

2.4 Variants and Hints for Heuristics

According to Definition 1, bandwidth reservation is limited to either determine a solution path that respects the capacity constraint or reject the reservation if none is found (although one may exist; according to Theorem 3 it is unlikely to device a polynomial algorithm that always finds a solution if one exists). Of course, the answer to a new request depends on the previous requests. In particular, at the initialization of the network, it is unlikely that the first requests will be rejected as the capacity is maximal in each node of the network. However, the protocol heuristic must prevent as much as possible the possible rejection of future requests by attempting to consume as small node capacity as possible and leaving the remaining capacity at each node as large as possible.

Two measures of interest are: the remaining capacity on node i, $C_i - b_i$ and the total consumed capacity $\sum_{i \in V} b_i$. These lead to the two following optimization problems:

1. Least Remaining Capacity (LRC): $\max \min_{i \in V} \{C_i - b_i\}$.

2. Total Consumed Capacity (TCC): $\min \sum_{i \in V} b_i$.

By Theorem 3, both are NP-hard and hence optimal solutions are unlikely to be found. However, as will be indicated in this section there is the possibility for the development of heuristics that work well in practice.

For simplicity in the discussion we assume that $H_I = 1$ and that a path capable of transfering a unit of bandwidth is required. Let us consider problem TCC first. Without the capacity constraint or when the capacity in each node is large enough, the Problem TCC is easy to solve: associate with edge (i,j) cost $c_{(i,j)} = |\{i\} \cup \{j\} \cup N(i) \cup N(j)|$ and find the minimum cost path Π from s to t. As by definition it holds for any choice of the path p,

$$\sum_{l \in p} c_l = \sum_{i \in V} b_i,$$

path Π is also a solution to TCC, provided that $C_i \geq b_i$ for all $i \in V$.

Remark. It should be noted that as a unit of bandwidth transmission from a to b requires $|N_a(H_I) \cup N_b(H_I)|$ units of bandwidth reservation, the bandwidth reservation weights on each link respect the triangular inequality. Indeed, suppose that three nodes a, b and c are pairwise neighbors, a path passing from a to c then to b must consume more bandwidth (at least $|N_a(H_I) \cup N_b(H_I) \cup N_c(H_I)|$) than a path going directly from a to b. However, it is clear that a shortest-hop path may not be an optimal for problem TCC.

Consider now Problem LRC. Provided $C_i \geq 4$, $i \in V$, this problems can be easily approximated within 4 from the optimal. To see this, take any shortest-hop path $\Pi = \{v_1, \cdots, v_k\}$ from $s(= v_1)$ to $t(= v_k)$. Notice first that since Π is a shortest-hop path, there can be no link of the form (v_i, v_j) for $|i - j| > 1$. This implies that for the transmission of a unit of bandwidth on the path, the following bandwidth needs to be consumed by the nodes, assuming $k \geq 5$:

- Node v_1: 2 units

- Node v_2: 3 units

- Node v_i, $3 \leq i \leq k - 2$: 4 units

- Node v_{k-1}: 3 units

- Node v_k: 2 units

Hence nodes on the path Π will require at most 4 units of bandwidth for the transmission of 1 unit of bandwidth from s to t. Consider now the possible neighbors of the nodes on the path. Since Π is a shortest-hop path, the only possible arrangements are the ones shown in Figure 1. Because of the manner bandwidth consumption is defined, a node k that is not on the path requires the following bandwidth according to the arrangement in the figure.
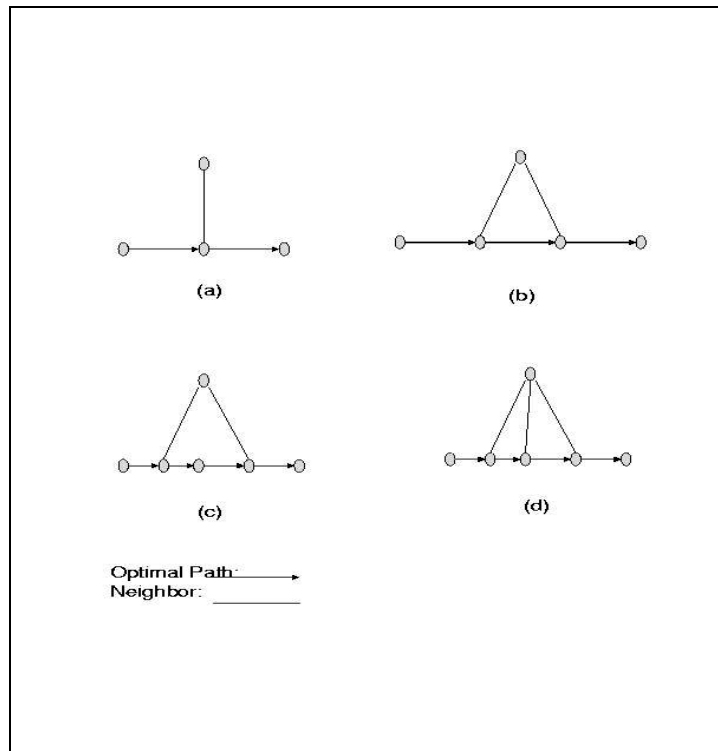
- In part (a): 2

Figure 1: Possible Neighbors of a Shortest Path

- In part (b): 3

- In part (c): 4

- In part (d): 4

Hence, the largest number is again 4. Therefore, picking the shortest-hop path Π, ensures that the bandwidth reserved on any of the nodes in no more than 4. However, this solution may be far from optimal when the TCC problem is of concern. As overall bandwidth consumption is of concern when nodes have large enough capacities, it seems appropriate to first attempt to solve Problem TCC, and in case a feasible solution is not found, to attempt to solve Problem LRC.

Based on the previous observations and assuming the C_i are large (say 100), one can proceed in the following way for the determination of an appropriate path.

- Find a solution to problem TCC. If the solution is feasible (the node capacity constraints are satisfied), provide this solution.

- If the solution to problem TCC is not feasible, solve problem LRC. If the solution is feasible provide the appropriate path. If not, the state that a solution cannot be found. Note that if all node remaining capacities are larger than 3, then according to the previous discussion a solution will be found. Hence a possible feasible solution will be missed only if some of the nodes have capacity at most 3. From an engineering point of view this may not be a big problem, since the difficulty starts only when some nodes reach their capacity limits.

3   Bandwidth Reservation and Slot Allocation

3.1   Slotted Model

In this section we assume that the channel is slotted, i.e., time is divided in slots of equal length, equal to the length of information packets. Moreover, time slots are divided into frames of length L. This channel may operate in parallel with a separate collision resolution channel. Transmissions that do not require reservation, as well as bandwidth reservation requests, take place on the collision resolution channel, while flows that require bandwidth reservation are transmitted in the slotted channel without collisions.

There are several implementation issues concerning the previously described model, one of them being the manner in which slot synchronization can be maintained throughout the network. This can be accomplished by having a system like GPS issue synchronization signals. However, we do not dwell further into these issues, as our main objective at this point is to show how a proposed method of bandwidth reservation, coupled with a relatively simple distributed algorithm for reserving time slots, can ensure that all admitted flows receive their requested bandwidth.

Assume that a new flow f arrives at network node s, and it is decided to use path p_f to transmit its packets to its destination d. When we say that a flow is given a bandwidth of $b_f \leq L$ units, we mean that some of the network nodes reserve slots so that all nodes along the path p_f are able to transmit or receive b_f of flows' f packets within each frame and that these transmissions can take place without interference, i.e., they can be safely delivered to the next hop (in the absence of errors due to channel noise).

We consider a proactive routing scheme where nodes obtain, through periodic information exchange, knowledge of network topology. Our objective is to devise a scheme by which the node where the new flow arrives will be able to decide on the path p_f based on available knowledge of topology and on the "available bandwidth" on each node, representing the number of unreserved slots in a frame. The latter quantity (available bandwidth) should be a single number in order to avoid inordinate amount of information exchange throughout the network during the periodic updates.

In order to motivate our particular method of bandwidth reservation we consider first some examples. In Figure 2 (a) assume that the frame length is $L = 2$ and that $H_I = 1$. Assume that a first flow f_1 requiring 2 slots per frame is established between nodes f and g. Thus nodes f and g fill-up all the available slots in a frame and announce zero available

bandwidth. The same is true for node e, since this node cannot receive packets during a frame, due to node f's transmission. Assume next that a new flow request f_2 arrives at node a requiring again bandwidth 2, and with destination node e. Since node e has available bandwidth 0, flow f_2 cannot be established, as is indeed the case.

Assuming that the channel is slotted changes Rule A to

> **Rule I:** Whenever b_f packets per frame need to be transmitted between nodes a and b, reserve b_f slots in a frame on all nodes in $N_a(H_I) \cup N_b(H_I)$.

Note that this rule may overestimate the bandwidth requirements needed to establish a flow along a path, i.e., may not take advantage of all possibilities of spatial reuse. Assume for example in Figure 2 (a), that flow f_1 had origin node g and destination node f. Then node a will decide again that flow f_2 cannot be established. However, this time f_2 can be established since both nodes f and g will be receiving and therefore there will be no interference (recall that we assumed that $H_I = 1$). In order to take advantage of such a situation node a needs to know not only the available bandwidth on each node, but also (at least) the manner in which this bandwidth is consumed (for transmission or reception). As another example, consider the situation depicted in Figure 2 (b). Assume that $L = 4$, $H_I = 1$ and that flow f_1 arrives first with $b_{f_1} = 1$. Hence, one slot must be reserved on nodes i and k, two slots on node j (one for reception and another one for transmission) and two slots on node e (to avoid interference during transmissions i to j and j to k). Assume that the first two slots in a frame are reserved on e. When flow f_2 arrives next, requesting bandwidth $b_{f_2} = 1$, node e must reserve the last two slots in a frame (one for transmission from d to e and another one for transmission from e to f). If nodes a and b do not know about the manner slots are allocated at node e, then they may reserve the first two slots for transmissions a to b and b to c. In this case, node h must reserve all four slots in the frame, i.e. its bandwidth requirement will be 4, as will be computed by the Rule I above. If however, nodes a and b knew about the manner slots are allocated in node e, they could reserve the last two slots in a frame for transmissions a to b and b to c. In this case, only the last two slots in frame need to be reserved in a frame at node h. Again, this possibility of reducing the reserved bandwidth on a node comes at the expense of requiring knowledge of exact allocations of frame slots at each node.

As we saw above, using rules other that Rule I, one can reduce the reserved bandwidth. However, this comes at the expense of increasing the amount of information that needs to be carried over the network, as well as an increase in the complexity of bandwidth reservation mechanism, which may be unacceptable in wireless networks.

Unfortunately, the use of Rule I to determine a path p_f for a new flow f with bandwidth requirements b_f, does not always guarantee that there are b_f available slots in a frame at each node on the path p_f that can be dedicated for the correct transmission of flows' f packets. To see this, consider the example in Figure 3. Assume that $H_I = 1$, and $L = 2$. Assume that flow f_1 arrives first with $b_{f_1} = 1$. Nodes a and b reserve the first slot in the frame for the transmission of flow f_1 packets and hence according to Rule I, nodes f, a, b, c, reserve one unit of bandwidth. Next flow f_2 with $b_{f_2} = 1$ arrives and nodes c and d
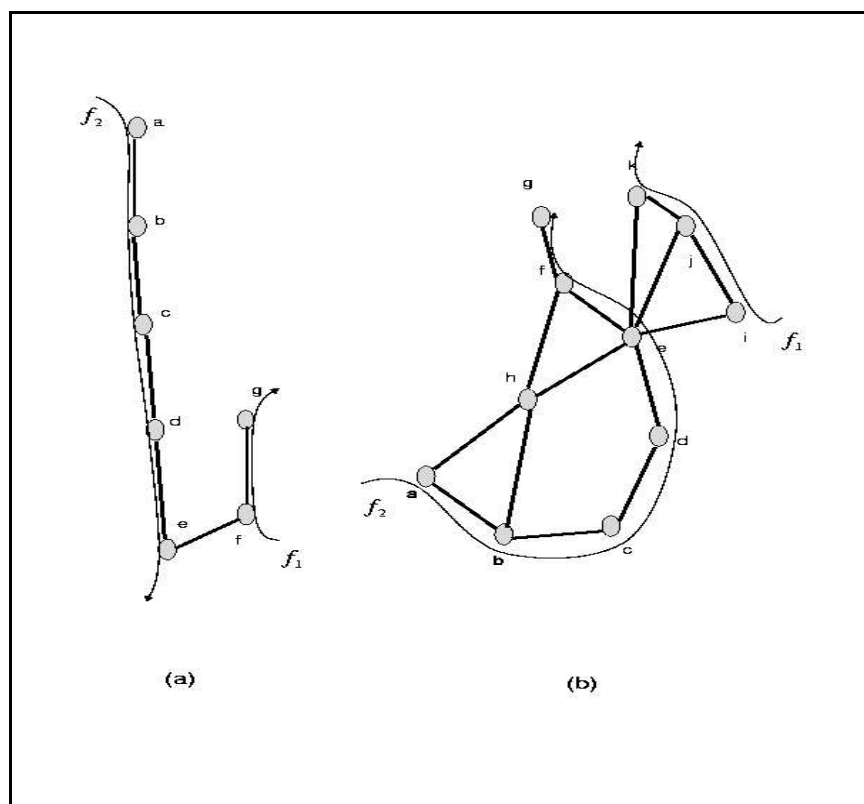
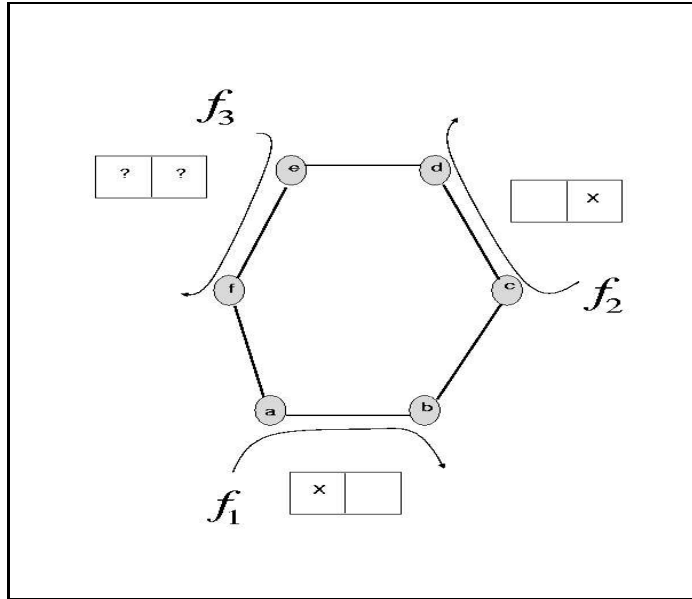Figure 2: Examples of Bandwidth Reservation

Figure 3: Rule I does not always guarantee correct bandwidth to a flow.

reserve the second slot in the frame for transmission of flow f_2 packets. Hence nodes b, c, d, e, reserve an additional unit of bandwidth. Hence at this point in time, nodes a, d, f, e, have one unit of available bandwidth. Assume now that flow f_3 with $b_{f_3} = 1$ arrives. According to Rule I, there is available bandwidth on path (e, f) for flow f_3. However, no slot assignment can be found in a frame for this transmission: the first slot is prohibited because of transmission of flow f_1 packets and the second slot because of transmission of flow f_2 packets.

The implication of the above example is the following. A node a, using Rule I may select a path for a flow f using the available bandwidth at each node. However, when the selected path is examined (during the connection establishment process) it will be found that the required bandwidth is not available by some node along the path. Then the origin node a should be notified and a new path must be chosen. Although this situation may occur under any distributed method of path selection when two new flows arrive simultaneously at the network, it should be avoided whenever possible since it implies a large number of control message exchanges.

In order to avoid the faulty (in terms of available bandwidth) path selection possibility we use the following modification of Rule I.

Rule II: Whenever b_f packets per frame need to be transmitted between nodes a and b, reserve b_f slots in a frame on all nodes in $N_a(H_I+1) \cup N_b(H_I+1)$.

Next we show that Rule II, combined with local information that each node has about the use of frame slots by its neighbors in $N_a(H_I + 1)$, guarantees the correct selection of paths for flow requests that arrive in the system.

Assume that node a knows the slots in a frame that are used for transmission or reception of packets by any of the nodes in $N_a(H_I + 1)$. The available bandwidth A_a at node a is then the unused slots in the frame. If a request requiring b_f slots arrives and $b_f \leq A_a$, then node a can reserve any b_f of the A_a slots for transmission of flow f packets. This process ensures the correct reception of these packets by the next hop node, say node j. To see this, note that $N_b(H_I) \subseteq N_a(H_I + 1)$. Hence an unused frame slot at node i implies that no nodes in $N_a(H_I) \cup N_b(H_I)$ is using this slot for transmission of reception. This implies that the transmission of a flow f packet from a to b will take place without interference.

Nodes a and b must also inform their neighbors in $N_a(H_I + 1) \cup N_b(H_I + 1)$ that they are reserving the b_f slots in a frame for transmission and reception. This process will also be successful because Rule II was used for path selection.

3.2   Connection Establishment Mechanism

We now describe a distributed mechanism for slot reservation using Rule II, that results in collision-free transmissions. Node i in the network maintains a list $R_i[k]$, $k = 1, ..., L$. When $R_i[k] = 0$, then slot k in a frame can used for a new transmission or reception by node i, or to avoid interference with its neighbors in $N_i(H_I)$. When $R_i[k] > 0$, then slot k is unusable because this slot has been reserved for transmission or reception of already established flows, or in order to avoid interference with its neighbors in $N_i(H_I + 1)$. We call such a slot, an "occupied " slot. Any flow that causes a slot to be occupied is said to "occupy" the slot. Note that $R_i[k]$ can be larger than one in case the same slot is occupied to avoid interference with more than one neighbors. How to keep track of this is explained below. The "available bandwidth" A_i on node i is defined as number of unoccupied slots in a frame, i.e., the number of slots k for which $R_i[k] = 0$.

In addition to $R_i[k]$, node i keeps track of information for each flow f that occupies some of the slots in a frame, in the form of an array $F_i[f, j, k]$, where $F_i[f, j, k] = 1$ if node j requested slot k on i 's frame to be occupied in order to ensure the collision-free transmission of flow's f packets. Otherwise, $F_i[f, j, k] = 1$. Note that during the process of bandwidth reservation more than one nodes may request slots to by occupied by the same flow f. For example, in Figure 2 (a), node e needs to reserve two slots for flow f_1, one for the transmission of node i and one for the transmission of node j. In this example, $F_e(f_1, i, 1) = 1$, $F_e(f, j, 2) = 1$.

We assume that each node knows the available bandwidth of the nodes in the network. During the process of bandwidth reservation, the nodes exchange the lists $F_i[f, j, k]$ with their $H_I + 1$-hop neighbors.

3.2.1 Arrival of a New Flow

New Flow Arrival at the Source Node Suppose that a new flow with bandwidth requirements b_f arrives at node s from the outside world. Then node i does the following.

1. Attempts to find a path p_f that joins s to the destination d of the flow and satisfies the requirements of Rule II. Specifically,

   Path Constraints: Let $(i_0 = s, i_1, i_2, ..., i_M = d)$ be the path nodes. The available bandwidth on any node $j \in \cup_{m=0}^{M} N_{i_m}(H_I + 1)$ must be at least as large as the quantity r_j determined by the following algorithm

   (a) Initialize $r_j = 0$ for all nodes in $\cup_{m=0}^{M} N_{i_m}(H_I + 1)$

   (b) For each node i_m, $m = 0, ..., M - 1$, do

         i. $r_j \leftarrow r_{i_m} + b_f$; $j \in N_{i_m}(H_l + 1) \cup N_{i_{m+1}}(H_I + 1)$;

   (c) endo

   The quantities r_j are in effect the bandwidth that needs to be reserved on the nodes in $\cup_{m=0}^{M} N_{i_m}(H_I + 1)$ according to Rule II, in order to guarantee interference-free transmission.

2. If a path satisfying the Path Constraints cannot be found then the flow is rejected. Else,

3. Node s picks b_f of the zero $R_s[k]$ and makes them one. Moreover, it sets $F_s[f, s, k] = 1$ if slot k is one of the slots picked in this manner.

4. Node s sends a *reservation request* that includes $F_s[f, s, k]$ to all its neighbors in $N_s(H_I + 1)$ and forwards the path p_f to the next-hop node i_1.

Processing a New Reservation Request Suppose that a reservation request $F_i[f, l, k]$ arrives to node j from a neighbor i. Then

1. If the elements $F_j[f, l, k]$ $k = 1, ..., L$ are not already set, then

2. Node j sets $F_j[f, l, k] = F_i[f, l, k]$, $R_j[k] \leftarrow R_j[k] + F_j[f, l, k]$, $k = 1, ..., L$.

3. If node j is the next hop on the path p_f, it does the following

4. Node j sends a *reservation request* that includes $F_j[f, l, k]$ to all its neighbors in $N_j(H_I + 1)$

5. If node j is not the last node on the path, then it acts as if the flow arrived from the outside world with the exception that the flow path p_f is the one received by node i. That is

(a) Node j picks b_f of the zero $R_j[k]$ and makes them one. Moreover, it sets $F_j[f, j, k] = 1$ if slot k is one of the slots picked in this manner.

(b) Node j sends a *reservation request* that includes $F_j[f, j, k]$ to all its neighbors in $N_j(H_I + 1)$ and forwards the path p_f to the next-hop.

The reservation in Step 4 ensures that all nodes in $H_I + 1$ neighborhood of the next-hop node reserve slots for the transmission of node l. The reservation in Step 5b ensures that all nodes in the H_I neighborhood of node j reserve slots for the transmission on node j (if j is not the last hop on the path). The combination of steps 4 and 5b may result in a node being charged twice for the transmission of the same node. For example, in Figure 2, node h must be charged only once for node a's transmission. However, it will receive the same request twice: first when node a send the request in step 5b and second when node b sends the reservation request at step 4. The check in Step 1 eliminates this possibility and makes sure that all nodes that belong to the intersection of the H_I neighborhood of nodes a and b are charged only once.

3.2.2 Completion of Flow

When a flow f completes, the established connection must be teared down and the reserved bandwidth must be released. The procedure followed in this case is the following.

Connection Tear Down at the Source Node The source node s does the following.

1. It sets $R_s[k] = R[k] - F_s[f, s, k]$, $k = 1, ..., M$ and removes the entry $F_s[f, s, k]$

2. It send a *connection tear down* message that includes $F_s[f, s, k]$ to all its neighbors in $N_s(H_I + 1)$.

Processing a Connection Tear Down Request A node j that receives a Connection Tear Down request $F_i[f, l, k]$ from a neighbor i does the following.

1. It checks whether there is the entry $F_j[f, l, k]$. If this entry does exist, then

2. It sets $R_s[k] = R[k] - F_j[f, l, k]$, $k = 1, ..., M$ and removes the entry $F_j[f, l, k]$

3. If node j is the next hop from node i on the path, then

4. Node j sends a Connection Tear Down message that includes $F_j[f, l, k]$ to all its neighbors in $N_j(H_I)$

5. If node j is not the last node on the path, then it does the following

6. Node j sets $R_j[k] \leftarrow R_j[k] - F_j[f, j, k]$, $k = 1, ..., M$ and removes the entry $F_j[f, j, k]$

7. Node j sends a Connection Tear Down message that includes $F_j[f, j, k]$ to all its neighbors in $N_j(H_I + 1)$

Optimizations of message exchanges and storage requirements are possible.

4 Application to MANET

The IETF standardization forum is addressing the problem of mobile ad hoc routing in the working group MANET. Several protocols have been proposed in experimental standards. There are two classes of routing protocols: the proactive protocols and the reactive protocols. The reactive protocols discover routes on demand and they do not seem suitable for the admission control we describe in this paper, since we need an a priori knowledge on the remaining bandwidth of each node in the network before the admission control process.

The OLSR protocol [6] is well suited for the admission control because it contains an embedded broadcast mechanism that optimizes the number of retransmissions. The broadcast mechanism uses specific relay nodes called MultiPoint Relay nodes (MPR) [7]. There exist discussions in order to extend OLSR to QoS management (QOLSR) [5], but these previous works did not address the foundamental problem of link interferences.

In the optimized version of OLSR protocol, nodes only advertise a subset of their adjacent links (i.e. their neighbor nodes). For the admission control to work properly, every node must know all the links between the other nodes in order to properly evaluate the radius of interference areas. We shall use F-OLSR, the version of OLSR where nodes advertise their full neighborhood, relayed by MPR's.

Nodes will advertise their remaining bandwidth in a specific control message, relayed by the MPR's. We can call this kind of message, Remaining Bandwidth Advertisement (RBA). This message will be generated periodically.

In order to compute their remaining bandwidth, nodes must know the status of the flows that go within H_I hops. To this end every node can broadcast within H_I hops the identification of the incoming and outcogoing flows as well as their bandwidth consumption. In order to perform the scheduling algorithm the nodes should also broadcast the schedule matrix of the flow. This would be contained in a message called Flow Schedule Advertisement message (FSA). This message will be generated periodically. It will be send with TTL equal to H_I in order to avoid proadcasting beyond the H_I neighborhood of a node. This message can also be used for connection termination advertisement.

With the knowledge of the remaining bandwidth of every node and the knowledge of the links of the network, every source node can perform the admission control described so far. OLSR protocol naturally maintains a routing table for best effort traffic. This routing table should not be affected by the QoS management. Instead admitted connections should be routed on the basis of source routing (packets are forced to be routed on the route mentionned in their headers). That way two different connections with same source and destination may take different routes in order to not affect each other.

In some situations a full distributed implementation of the admission control protocol may not be desirable. For example, the knowledge of all existing active connections in the networks may pose security problems. In this case one may envision a centralized Admission Controller (AC) that collects all needed informations from other nodes. In this case the RBA packet maybe a secured unicast packet addressed to the AC instead of a broadcast packet addressed to the whole network. In the same vein OLSR maybe used in its most optimized version where only partial topology is advertised via broadcasted Topology Control (TC) messages. Since only MPR links are advertised, the overhead of traffic control is greatly reduced. However this partial topology may not be sufficient in order to make an efficient admission control since some links may miss in order to evaluate the effect of candidate route on the remaining bandwidth of the nodes which H_I hop away. Nodes that have additional links that are not advertised in the TC messages can sent them in a separate unicast message to the AC. If the implementation is fully distributed, one has to use the version of OLSR – called F-OLSR, where TC messages conveyed the identifiers of all the links adjacent to the originator node. This version still benefit from the optimized flooding mechanism that uses MPR nodes.

5 Further study

In Section 2.4, some heuristics were oulined for the path selection problem. These heuristics, as well as other variants need to be generalized. Moreover, the performance of the heuristics needs to studied either analytically or through simulations.

5.1 Per link Remaining bandwidth

In the previous sections we assumed that the remaining bandwidth is computed on a per node basis. It turns out that this approach may be too conservative, since in the slot scheduling process one has to define the remaining bandwidth of a node as the nominal bandwidth minus the bandwidth emitted or received $H_I + 1$ hops away. In fact it turns that a more accurate evaluation of the remaining bandwidth must be done on a per link basis. In this case the remaining bandwidth of the link will the nominal bandwidth of the link (i.e. the node) minus the bandwidth emitted H_I hops from the receiver and the bandwidth received H_I hops from the sender. Notice that with this definition the remaining bandwidth of a link is not equal to the remaining bandwidth of the reverse link.

The proof of NP completeness should hold since the counterpart of the forbidden pair problem would be the forbidden link pair problem. It is clear that both problem belongs to the same class since they can be derived from each other by polynomial transformation.

5.2 Remaining Bandwidth computation in CSMA/CA, hidden nodes

When IEEE 802.11 wireless interfaces are used the protocol is not slotted. Instead DCF protocol of IEEE 802.11 is CSMA with Collision detection. However such protocol actually

generates collisions between packets in presence of so-called hidden nodes. These collisions will very likely impact the bandwidth reservation mechanism since they lead to packet retransmission. In particular that if on a link the experienced probability collision is $p_c < 1$, then the bandwidth occupation of the connexion will be magnified by a factor $\frac{1}{1-p_c}$.

In order to estimate the probability of collision on a link one must split the link bandwidth into two component:

- the visible busy bandwidth b_v, that interferes with the transmitter (i.e. seen by the transmitter);

- the hidden bandwidth b_h, that interferes with the receiver but not with the transmitter.

Notice that the hidden bandwidth is likely to be the source of most collisions.

The probability of collision p_c is something that would depend on b_h and on the remaining bandwidth R_w. For example: unrealistic assumption) could be

$$p_c = \frac{b_h}{b_h + R_w}$$

or

$$p_c = \exp(-L \times b_h)$$

where L is the packet length. Of course one can imagine more sophisticated estimate of the collision probability using more parameters.

Remark: if IEEE 802.11 RTS/CTS option is used, then the bandwidth transmitted by neighbors of the receiver (and not detectable by the transmitter) should be removed from b_h to b_v.

The main change with the old bandwidth admission control scheme is that the new flow may increases the collision rate of other flows and amplify again their bandwidth cost.

Assuming that the new flow is added on a given path.

For any link we have

$$b_h = \sum_{\ell \in \text{hidden links}} \frac{1}{1 - p_c(\ell)} \text{flow}(\ell)$$

A hidden link is a link which contributes to hidden bandwidth. We can use the expression of p_c in order to get a fixed point equation. For example $p_c = \exp(-Lb_h)$.

References

[1] K. Bertet, C. Chaudet, I. Guérin Lassous, L. Viennot (2001): Impact of Interferences on Bandwidth Reservation for Ad Hoc Networks: a First Theoretical Study, *GLOBECOM SAWN'2001, IEEE Symposium on Ad-Hoc Wireless Networks*, 2001. (Also INRIA-RR-3895, 2000.)

[2] H.N. Gabow, S.N. Maheshwari, L.J. Osterweil, On two problems in the generation of program test paths, *IEEE Transactions on Software Engineering*, SE-2(3), 1976, 227-231.

[3] M. Garey, D.S. Johnson (1979), *Computers and Intractability: a guide to the Theory of NP-Completeness*, Freeman 1979.

[4] V. Kann (1994), Polynomially bounded minimization problems that are hard to approximate, *Nordic Journal of Computing*, 1(3), 317–331.

[5] A. Munaretto, H. Badis, K. Al Agha, Guy Pujolle (2003): QOLSR : Routage avec QoS dans OLSR. *AlgoTel 2003, 5eme Rencontres Francophones sur les aspects Algorithmiques des Telecommunications*, Banyuls-sur-mer, France, May 12-14, 2003, 109–114.

[6] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum and L. Viennot. Optimized Link State Routing Protocol. *IEEE INMIC*, Pakistan 2001

[7] P. Jacquet, A. Laouiti, P. Minet, L. Viennot. Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models, INRIA research report RR-4260, 2001.