BRAWN: Bandwidth Reservation over Ad-hoc Wireless Networks

Rafael Guimarães and Llorenç Cerdà
Technical University of Catalonia
Computer Architecture Dept.
Jordi Girona 1-3,
E-08034 Barcelona, Spain

March 4, 2008

Abstract

In this report we describe a QoS reservation mechanism for *Multirate* Ad-hoc Wireless Networks (AWNs) that allows bandwidth allocation on a per flow basis. By multirate we refer to those networks where wireless nodes are able to dynamically switch among several link rates. This allows nodes to select the highest possible transmission rate for exchanging data, independently for each neighbor.

1 Introduction

In this report we propose a simple, yet effective method to compute the available bandwidth at a node in AWNs. We use this method to propose a reservation based QoS mechanisms, which we call BRAWN (Bandwidth Reservation over Ad-hoc Wireless Networks). Our proposal not only guarantees certain QoS levels, but also naturally distributes the traffic more evenly among network nodes (i.e. load balancing).

Our mechanism takes into account the multirate capability of wireless networks, i.e., it considers that wireless nodes are able to choose among several modulation schemes, providing different transmission rates, in order to accommodate to different channel conditions. We provide a set of *QoS constraints* that must be satisfied for the ongoing QoS flows to consume an overall bandwidth at any node smaller than or equal to a certain threshold. Along the report we shall refer to this threshold as Q. It may be understood as the percentage of time that the channel can be busy at any given node, because it is transmitting, receiving or listening to traffic that belongs to QoS flows. We propose a set of CAC rules that, upon the assumptions listed in the following subsection, can satisfy the *QoS constraints*.

Our scheme can be easily integrated in the many of the routing protocols that have been proposed for AWNs. For instance, in [1] we describe how could it be integrated in *Optimized Link State Routing Protocol* (OLSR) [2], and in *Ad-hoc On-demand Distance Vector* AODV [3]. We have evaluated a preliminary version of BRAWN, and some enhancements in [1], [4], and [5]. In this report we give a detailed description of BRAWN, and we add the proof that BRAWN satisfies the stated QoS constrains.

1.1 Our proposal

We treat the problem of achieving end-to-end bandwidth reservation. Our mechanism, which we call BRAWN (Bandwidth Reservation over Ad-hoc Wireless Networks), is based on the computation of the available bandwidth seen by a given node and the use of this value to verify whether new flows can still be routed through this node.

Our scheme is based on the following assumptions: (i) QoS-aware applications are able to request the appropriate bandwidth when establishing a connection. (ii) The nodes know the capacity of the wireless links that is available for QoS flows. Besides this, we assume that the MAC used is able to isolate traffic classes, in such a way that QoS traffic has priority over non-QoS traffic (we could, for instance, use 802.11e). This allows nodes to fix the previously introduced Q threshold. (iii) A pure Carrier Sensing Medium Access (CSMA) protocol is used. Thus, whenever a node is transmitting, all its neighbors will remain silent. Through the report we shall refer as neighbors each pair of nodes that are in the receiving range of each other. Note that we are not considering a MAC using RTS/CTS, although it could be easily supported, as we proposed in [4]. (iv) Nodes are able to reach all their neighbors through broadcast packets.

Of course, the previously described assumptions are not exact in real wireless networks. For instance, the available capacity for QoS traffic may be influenced by non-QoS traffic and other network conditions. To cope with that, a conservative value shall be used for Q, or it may be made adaptive, as we proposed in [5]. Furthermore, changes in the network conditions, which can be very frequent in AWNs, make the information used by nodes to compute the available bandwidth to be uncertain. Therefore, after a flow is accepted, its QoS parameters (end-to-end delay, packet loss, etc.) should be constantly monitored in order to react to congestion. This could be done by re-routing or even dropping some of the involved flows. We will not deal with these issues, in order to keep the report focused on the reservation mechanism.

Note that a reservation mechanism approach is more appropriate for wireless ad-hoc networks with fixed nodes (e.g. wireless mesh networks [6]) or where mobility is not very high (e.g. pedestrian networks). If nodes constantly move with high speeds (vehicular networks, for instance), changes on the topology are very frequent, thus, the reserved path should be constantly updated. For this reason we use the term AWN (Ad-hoc Wireless Networks) and not MANET (Mobile Ad-hoc NETworks). MANET is commonly used in literature to remark the mobility characteristic of the AWN under consideration.

2 How much bandwidth is available for reservations?

The BRAWN mechanism is based on the computation of the available bandwidth (AB) by each node in the network in a distributed way. By knowing its available bandwidth, a node is able to accept or reject a new reservation. So, the first step we should take in order to define our mechanism is to compute the AB of each node.

If we want to compute the amount of bandwidth that is available for a given node to use for new reservations, we should first investigate the amount of bandwidth that is already being consumed by active flows. By knowing this value, we may just subtract it from the total bandwidth dedicated to QoS traffic in order to obtain the currently available bandwidth.

The first issue that we should notice is that a transmission between two nodes does not consume bandwidth only from these nodes but also from the whole neighborhood, since no other neighbor is able transmit at the same time (at least using the same channel) in order to avoid collisions. In fact, the exact knowledge of which nodes suffer the interference of a given transmission depends directly on the MAC protocol that is being used. For this reason, we assume the use of a Carrier Sensing based (CSMA-like) protocol for the analysis that we will present throughout the report.

In order to know how much bandwidth is available for a node to use, we must take into account all transmissions that directly affect its opportunities to transmit. In the case of a CSMA-based wireless MAC protocol, the bandwidth of a node is consumed whenever:

case 1) It transmits data to a neighbor;

case 2) One of its neighbors is transmitting data (if the node senses that the medium is being used, it remains in silence);

Representing this in an analytical way, we may state that the load impact of all transmissions on a node i is given by:

$$l_i = \left| \underbrace{x_i}_{Case\ 1} \cup \underbrace{\bigcup_{j \in \mathcal{N}_i} x_j}_{Case\ 2} \right| = \left| \bigcup_{j \in \mathcal{N}_i^{+}} x_j \right| \tag{1}$$

where:

- l_i is the load impact of all transmissions (in bps) on node i;
- x_i is the total traffic (in bps) that node i wants to transmit (either if node i is the source of the traffic or if it is just forwarding);
- $\mathcal{N}_i$ is the set of neighbors of node i;
- $\mathcal{N}_i^{+}$ is the set of neighbors of node i and node i itself;

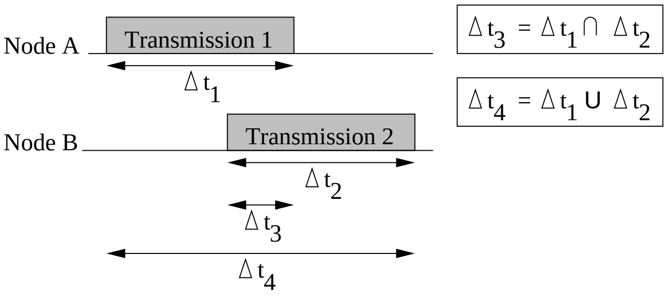

Figure 1: Example of "time-based" union and intersection operators

- The union operator ∪ represents a "time-based union", i.e., intersections represent parts of the transmissions that takes place simultaneously. See figure 1 for an example of the appliance of this operator over two transmissions that overlap in time.

The formula derived before can be generalized for wireless multirate networks, i.e., networks where nodes can communicate to each other at different transmission rates, depending on the wireless medium conditions. To do so, all these values that were represented in bps above must be normalized, dividing them by the transmission rate used:

$$L_i = \left| \bigcup_{j \in \mathcal{N}_i^+, \forall k} x_{jk}/v_{jk} \right| \tag{2}$$

where:

- L_i is the normalized load impact on node i. From now on, we will consider in this report only the multirate case (since the single-rate can be seen as a particular case of a multirate network, where all transmission rates are the same). We shall use capital letters for referring to normalized values.
- x_{jk} is the total traffic (in bps) that node j wants to transmit to node k.
- v_{jk} is the transmission rate used between nodes j and k.

Since the equation is normalized, if the node is not overloaded, L_i should be a value between 0 and 1.

The use of the union operator states that some transmissions in the neighborhood may overlap in time. This can happen in CSMA-based networks whenever these transmissions do not interfere with each other, as shown by figure 2. In this example, transmissions *a* and *b* can overlap in time.

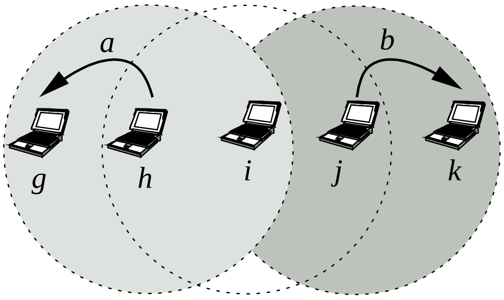

Figure 2: Simultaneous transmissions in the neighborhood of node *i*

Once we have computed the load impact on each node of the ad-hoc network and after defining the amount of normalized bandwidth dedicated to QoS traffic as Q, we are able to state the following *QoS constraint* that should be respected in order to provide QoS guarantees for real-time flows:

$$L_i \leq Q, \forall i \in \mathcal{S} \qquad (0 \leq Q \leq 1) \tag{3}$$

Where $\mathcal{S}$ is the set of nodes that are transmitting or receiving QoS traffic, i.e. the nodes having at least one QoS reservation. In the rest of the report we shall refer $\mathcal{S}$ as the QoS set. By guaranteeing condition (3), we can guarantee that the channel occupancy due to the QoS traffic observed by any node of the QoS set is never greater than Q. This condition should guarantee that there is enough capacity to accommodate all QoS flows.

Note that Q can be understood as the percentage of time that the channel can be busy at any node, because either it is transmitting or receiving traffic that belongs to the QoS flows. We shall assume that the MAC is able to restrict non-QoS traffic, such that the normalized capacity Q will be always available for QoS traffic. This could be achieved e.g. using 802.11e, or 802.11 with some additional mechanism, e.g. SWAN [7], that regulates non-QoS traffic. Of course, due to collisions, impact of non-QoS traffic and other reasons, the amount of normalized capacity Q available for QoS traffic may vary. To cope with that, a conservative value shall be used for Q, or it may be made adaptive, as we proposed in [5].

3 The basis of BRAWN

As previously mentioned, BRAWN is based on the computation of the available bandwidth (*AB*) in each node of the network. The goal of our *bandwidth reservation mechanism* is to provide rate allocation (e.g. peak or sustainable rate) and, at the same time, remain as simple as possible. The solution should provide QoS and yet introduce as little overhead as possible in the network. In order to do that, it should only make use of the information about its 1-hop neighborhood. Since most of the available ad-hoc routing protocols already provide 1-hop signaling, e.g. HELLO messages, any additional information that may be necessary can be piggybacked on these signaling messages.

In order to provide a simple mechanism that is feasible to implement, some simplifications must be done. The first of them is related to the computation of the load impact on each node of the ad-hoc network. The use of the union operator, as shown by equation (2), is not possible, since a node has no idea of the "degree of simultaneity" of the transmissions on the neighborhood. For this reason, we simplify the equation by using a simple sum instead, since it is always more restrictive than using the union (figure 1), what still guarantees the QoS requirements. Thus, the load on a node i will be computed as:

$$L_i = \left| \bigcup_{j \in \mathcal{N}_i^+} X_j \right| \approx \sum_{j \in \mathcal{N}_i^+} X_j \tag{4}$$

where:

$$X_j = \sum_{\forall k} \frac{x_{jk}}{v_{jk}} \tag{5}$$

is the normalized amount of QoS traffic that node j wants to transmit (either if node j is the source of the traffic or if it is just forwarding). In BRAWN each node would reserve bandwidth for this traffic, thus, X_j can also be interpreted as the total reserved bandwidth at node j. Using a sum in equation (4) to represent a union may be pessimistic in some scenarios. In [8] the approximation given by equation (6) has been proposed.

$$\left| \bigcup_{j \in \mathcal{N}_i^+, \forall k} X_j \right| \approx \sum_{j \in \mathcal{N}_i^+} X_j - \sum_{j,k \in \mathcal{N}_i^+} \left| X_j \cap X_k \right| \tag{6}$$

However, in 802.11-like networks two transmissions cannot overlap in time whenever either the sender or the receiver of one transmission is a neighbor of either the sender or the receiver of the other one. Therefore, in order to accurately compute which intersections from equation (6) are not null, a node would need individual information about every flow in the neighborhood, so that it would be able to identify those that may take place simultaneously. Exchanging this information would introduce too much overhead in the protocol. Consequently, we have considered equation (4) as a convenient approximation for the load demand.

3.1 The Available Bandwidth in each node

Once each node is able to compute the load demand on itself, this value can be used to establish which part of the total bandwidth dedicated to QoS connections is still available for reservations. By using just the information locally known by a node (the pre-established Q value and the computed load impact), we define a new value that represents this availability for new flows to be established, which we call the Maximum Available Bandwidth (*MAB*).

$$MAB_i = Q - L_i \tag{7}$$

This value is simply the amount of bandwidth available for QoS flows minus the amount of bandwidth already consumed under the point of view of this node, i.e., its load impact. By looking at equation (3) it is quite simple to notice that we may re-write the QoS constraint using this new value.

$$MAB_i \geq 0, \forall i \in \mathcal{S} \tag{8}$$

However, knowing the local *MAB* of a node is not enough for the node to decide if new flows can be accepted. This is because the available bandwidth of a given node i is also affected by transmissions of its two-hop nodes that have one of the neighbors of i as a receiver. In figure 3, for example, the transmission from g to h only causes an impact on the computation of MAB_g and MAB_h, although when it takes place, node i is not allowed to transmit (notice, however, that $MAB_i = 1$). That means that a node also needs to take into account its neighbors restrictions.

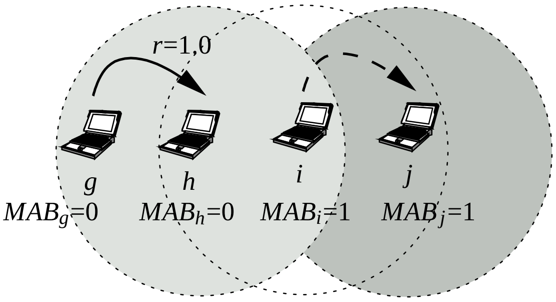

Figure 3: The Maximum Available Bandwidth and restrictions imposed by neighbors

We, thus, propose to estimate what we call the Available Bandwidth of a node i (AB_i) as the minimum value of the *MAB*s in its QoS set neighborhood:

$$AB_i = \min\{MAB_j\}, j \in \mathcal{N}_i^+ \cap \mathcal{S} \tag{9}$$

This value can also be understood as a more complete view of the node about the impact of new transmissions on the neighborhood. It is, in fact, the amount of bandwidth available for new transmissions over a given node.

Now, the QoS constraint given by equation (8) can be rewritten in terms of the available bandwidth as we state in the following theorem:

Theorem A. *Guaranteeing that the AB given by equation* (9) *of every node that takes part in a reserved path is non-negative, is equivalent to guaranteeing that the MAB of every node of the QoS set is non-negative.*

See the proof of this theorem in appendix A. In other words, the QoS constraint given by equation (8) can be rewritten as:

$$AB_i \geq 0, \forall i \in reserved\ paths \tag{10}$$

Summing up, BRAWN requires that the nodes know the normalized amount of traffic (X_j) and the maximum available bandwidth (MAB_j) of their neighbors belonging to the QoS set ($\mathcal{S}$). These values should be periodically exchanged among neighbors belonging to $\mathcal{S}$. Nodes that do not belong to $\mathcal{S}$ would compute the MAB_j, which could be needed in the CAC of future QoS reservations, but they would not send it. Each node i uses X_j to compute the load L_i using equation (4), and MAB_i using equation (7). Finally, the available bandwidth (AB_i) is computed using equation (9).

3.2 Call Admission Control

After defining the distributed mechanism to compute the *available bandwidth* at each node of the network, we will use this value to decide whether a new connection of r bps fits or not in a given node.

The first step toward the definition of a Call Admission Control (CAC) is realizing which transmissions cannot take place while a node i is transmitting towards a node j. As we have discussed before, if we are using a CSMA-like protocol, none of the i's neighbors nor the j's neighbors are allowed to transmit while i is transmitting to j. Therefore, the following CAC should be checked in every node along a candidate path:

$$AB_i \geq \left| \bigcup_{y \in \left(\left(\mathcal{N}_i^+ \cup \mathcal{N}_j^+ \right) \cap path \right)} \frac{r}{v_y} \right| \tag{11}$$

where

- i represents the current node in the path;
- j represents the next node in the path (to which i will transmit);
- r represents the bandwidth required by the new connection;
- v_y is the transmission rate from node y toward its next hop in the path.

In this case, just like in the load demand computation (equation (4)), we use a simple sum approximation for the union operator.

$$AB_i \geq \sum_{y \in \left(\left(\mathcal{N}_i^+ \cup \mathcal{N}_j^+ \right) \cap path \right)} \frac{r}{v_y} \tag{12}$$

See the proof that this CAC condition guarantees the QoS constraint presented by equation (10) in appendix B.

Notice that in the case that nodes move, topology changes in the network may cause connections that were previously accepted by the CAC not to have their QoS requirements guaranteed after a while. Moreover, even if QoS can still be guaranteed over a given path, topology changes may cause more efficient paths to show up, and being able to use them may optimize the use of network resources. Therefore, in the presence of movement, the QoS mechanism should be adaptive. This could be achieved e.g. by periodically refreshing reservations, so that the network is constantly re-validating the admission control and searching for better routes for previously established connections.

4 Exemplifying BRAWN's behavior

In order to better understand the behavior of the BRAWN mechanism, we will take a step-by-step look at the ad-hoc network example depicted by figure 4. In this simple example all links between mobile nodes are 5 Mbps. Assume that in this network there is an established reservation for a QoS flow of 1 Mbps following the path $MN_A \rightarrow MN_B \rightarrow MN_E \rightarrow MN_F$. For simplifying the example, we shall also assume that the reserved capacity for QoS traffic is $Q = 1$.

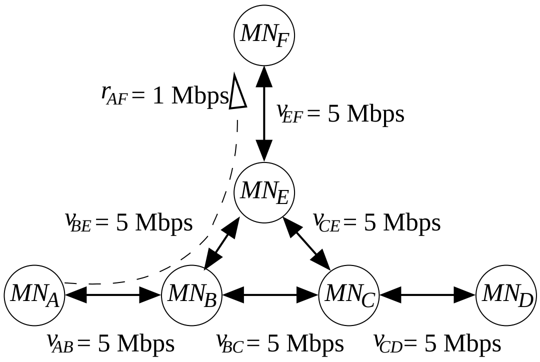

Figure 4: Network topology

The row X_i in table 1.(a) shows the normalized amount of traffic that would be advertised by the nodes. Upon receiving theses values, each node would compute the MAB_i shown in the corresponding row of the table. For instance, MN_B would receive $X_A = 0.2$, $X_C = 0.0$ and $X_E = 0.2$. Since $X_B = 0.2$, it would compute $MAB_B = 0.4$. Finally, upon receiving the MAB from their neighbors, nodes would compute the AB_i given in the table. Note that nodes MN_C and MN_D would not advertise their MAB, because they do not belong to the QoS set.

Assume that after this, node MN_C wishes to establish a new QoS flow of r_{CD} = 2 Mbps with node MN_D. The following CAC conditions would be checked: $AB_C \geq 0.4$ and $AB_D \geq 0.4$ (2 Mbps / 5 Mbps = 0.4). Thus, the flow would be accepted, and the values of X_i, MAB_i and AB_i would change as shown in table 1.(b).

		MN_A	MN_B	MN_C	MN_D	MN_E	MN_F
(a)	X_i	0.2	0.2	0.0	0.0	0.2	0.0
	MAB_i	0.6	0.4	0.6	1.0	0.6	0.8
	AB_i	0.4	0.4	0.4	1.0	0.4	0.6
(b)	X_i	0.2	0.2	0.4	0.0	0.2	0.0
	MAB_i	0.6	0.0	0.2	0.6	0.2	0.8
	AB_i	0.0	0.0	0.0	0.2	0.0	0.2

Table 1: Parameters computed by nodes using BRAWN. With flow r_{AF} (a), and with flows r_{AF} and r_{CD} (b).

We may intuitively check that, after accepting the flow r_{CD}, the available bandwidth computed by the nodes is correct: Whenever one of the nodes MN_A, MN_B, MN_C and MN_E send a packet, all the others in this set must remain silent. Since altogether send 5 Mbps, which is the link capacity, their available bandwidth is 0. Node MN_E must be silent whenever MN_B, MN_C or MN_F transmit. Since nodes MN_E, MN_B and MN_C transmit altogether 4 Mbps, the available bandwidth at node MN_F is 1-4/5 = 0.2. Similarly, we can derive that the available bandwidth at node MN_D is also 0.2.

5 Conclusions

In this report we have described a bandwidth reservation scheme for ad-hoc networks that satisfies the following QoS constraint: “The load demand offered to the wireless media by the QoS traffic observed at any node in a path that is about to be established $\leq Q$”. Parameter Q is dimensioned in a way that delays are acceptable for QoS connections. Our reservation scheme is designed for networks where nodes can communicate to neighbors using different transmission rates depending on channel conditions (multirate ad-hoc networks) and only requires that nodes know the normalized bandwidth reservation and maximum available bandwidth of their neighbors. These quantities can be easily advertised by means of HELLO packets. We also give a CAC rule that nodes should apply to new connections requiring QoS.

Appendix A: Proof of Theorem A

Theorem A. *Guaranteeing that the AB given by equation* (9) *of every node that takes part in a reserved path is non-negative, is equivalent to guaranteeing that the MAB of every node of the QoS set ($\mathcal{S}$) is non-negative.*

Proof. The computation of the *MAB* of a given node takes into account only information about transmissions performed by the node and its 1-hop neighbors (see equation (7)). Consequently, a transmission between two nodes only impacts the *MAB* of the 1-hop neighborhood of the sender.

By observing this, we can state that whenever a new real-time flow is established over the network, only nodes that takes part in the flow path and their 1-hop neighborhood are affected by these new transmissions. All the other nodes throughout the network see no changes on their *MAB*. Thus, if they were non-negative before the flow was established, they would remain like that afterwards.

Then:

$$\begin{aligned} AB_i \geq 0, \forall i \in resvd\ paths \Leftrightarrow \\ \min_{j \in \mathcal{N}_i^+} \{MAB_j\} \geq 0, \forall i \in resvd\ paths \Leftrightarrow \\ MAB_j \geq 0, \forall j \in \mathcal{N}_i^+, \forall i \in resvd\ paths \end{aligned} \tag{13}$$

So, considering that in the beginning all nodes of the QoS set have non-negative *MAB*s and that nodes that are not in the 1-hop neighborhood of reserved paths do not see any changes on their *MAB*, we can conclude that:

$$\begin{aligned} MAB_j \geq 0, \forall j \notin \mathcal{N}_i^+, \forall i \in resvd\ paths \Leftrightarrow \\ MAB_j \geq 0, \forall j \in \overline{\mathcal{N}_i^+}, \forall i \in resvd\ paths \end{aligned} \tag{14}$$

By using the results of (13) and (14):

$$\begin{aligned} & MAB_j \geq 0, \forall j \in \left(\mathcal{N}_i^+ \cup \overline{\mathcal{N}_i^+}\right), \forall i \in resvd\ paths \Leftrightarrow \\ & MAB_i \geq 0, \forall i \in \mathcal{S} \end{aligned} \tag{15}$$

So, as we were willing to demonstrate:

$$AB_i \geq 0, \forall i \in resvd\ paths \Leftrightarrow MAB_i \geq 0, \forall i \in \mathcal{S} \tag{16}$$

$\square$

Appendix B: Proof that the CAC guarantees the QoS Constraint

The use of the CAC proposed in equation (12) guarantees that the QoS constraint defined in equation (8) is respected in every condition. In fact, as we demonstrate below, it is guaranteed in all cases when using a simple sum approximation for the union operator.

Proof. As we have previously demonstrated, in order to guarantee the QoS constraint presented in equation (8), we can limit ourselves to guaranteeing the condition presented by equation (10). Thus, we just need to demonstrate that the proposed CAC guarantees that after accepting a new flow of r bps, every node in the flow path present a non-negative AB, i.e., all the nodes in the flow path and their 1-hop neighborhood belonging to the QoS set, present a non-negative MAB. Since we are only concerned with the nodes belonging to the QoS set, in the following we shall refer to only this set of nodes.

Nodes in the flow path: for a given node i in the path, we want to guarantee that its MAB is non-negative in the moment t_1 just after the acceptance of the new flow (t_0 represents the moment just before the acceptance).

$$\begin{gathered} MAB_i(t_1) \geq 0 \Leftrightarrow \\ Q_i - L_i(t_1) \geq 0 \Leftrightarrow \\ Q_i - \sum_{y \in \mathcal{N}_i^+} X_y(t_1) \geq 0 \end{gathered}$$

Since the only new transmissions from t_0 to t_1 are the ones due to the accepted flow, we have:

$$Q_i - \underbrace{\sum_{y \in \mathcal{N}_i^+} X_y(t_0)}_{=L_i(t_0)} - \sum_{y \in \left(\mathcal{N}_i^+ \cap path\right)} \frac{r}{v_y} \geq 0$$

We then have:

$$MAB_i(t_0) \geq \sum_{y \in \left(\mathcal{N}_i^+ \cap path\right)} \frac{r}{v_y} \tag{17}$$

So, in order to be correct, the CAC must guarantee equation (17). Our CAC, in simple sum approximation form, expressed by equation (12) guarantees the following condition:

$$AB_i(t_0) \geq \sum_{y \in \left(\left(\mathcal{N}_i^+ \cup \mathcal{N}_j^+\right) \cap path\right)} \frac{r}{v_y} \tag{18}$$

Since we know that $AB_i(t_0) \geq MAB_i(t_0)$, we may also say that the CAC guarantees that:

$$MAB_i(t_0) \geq \sum_{y \in \left(\left(\mathcal{N}_i^+ \cup \mathcal{N}_j^+\right) \cap path\right)} \frac{r}{v_y} \tag{19}$$

And finally, since equation (17) is more restrictive than equation (19) (note the additional terms in the former, as well as the sum over a more restricted set of nodes), we may say that the CAC satisfies the desired conditions.

Nodes in the 1-hop neighborhood of the flow path: There are basically two different cases that should be taken into account:

- Node n that is a neighbor of a node i in the path and all its other neighbors in the path are in the neighborhood of i or j (considering that i transmits to j).
- Node whose neighbors in the path are "more spread".

In the first case, we have that:

$$\begin{aligned} & MAB_n(t_1) \geq 0 \Leftrightarrow \\ & MAB_n(t_0) \geq \sum_{y \in (\mathcal{N}_n^+ \cap path)} \frac{r}{v_y} \end{aligned} \tag{20}$$

Since we know that

$$\begin{gathered} AB_i(t_0) \leq MAB_n(t_0) \\ \left(\mathcal{N}_n^+ \cap path\right) \subset \left(\left(\mathcal{N}_i^+ \cup \mathcal{N}_j^+\right) \cap path\right) \end{gathered}$$

and since the CAC guarantees equation (18), we are also able to guarantee the condition expressed by equation (20).

Since transmissions that take place outside the neighborhood of the sender and the transmitter may overlap in time, the second case may, in fact, be seen as a combination of several non-correlated occurrences of the first case. So, if the CAC guarantees the QoS constraints for the first case, it will guarantee for the second as well.

□

Acknowledgments

This work was supported by CAPES - Brazil, the Ministry of Education of Spain under grant CICYT TSI2007-66869-C02-01, the Fund for Scientific Research Flanders under Scientific Research Community Broadband communication and multimedia services for mobile users, by DWTC Belgium under project IAP P5/11 MOTION (Mobile Multimedia Communication Systems and Networks), by IWT under project 020152.End-to-end QoS in IP based Mobile Networks, the European NoE EuroFI and the European project WIDENS.

References

[1] Llorenç Cerdà and Michael Voorhaen and Rafael Guimarães and José-M. Barceló and Jorge García and Chris Blondia, Vol., 3427 of Lecture Notes in Computer Science (LNCS), Ch., pp., 176–188.

[2] T. Clausen and P. Jacquet, RFC 3626: Optimized Link State Routing Protocol (OLSR) (Oct 2003).

[3] C. Perkins and E. Belding-Royer, S. Daas, RFC 3561: Ad hoc On-Demand Distance Vector Routing (Jul 2003).

[4] R. Guimarães and Ll. Cerdà, Bandwidth Reservation On Wireless Networks with RTS/CTS Signalling, in: Proceedings of the 12th European Wireless Conference (EW2006), 2006.

[5] Rafael Guimarães and Llorenç Cerdà, Vol., 4396 of Lecture Notes in Computer Science (LNCS), Ch., pp., 102–112.

[6] Akyildiz and Ian F. and Wang and Xudong and Wang and Weilin, Wireless mesh networks: a survey, Computer Networks 47 (4) (2005) 445–487.

[7] A. Veres and G. Ahn and A.T. Campbell and L. Sun, SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks, in: Proceedings of the Conference on Computer Communications (IEEE Infocom), 2002.

[8] Y. Gao and D. Chiu and J. C. S. Lui, Determining the End-to-end Throughput Capacity in Multi-Hop Networks: Methodology and Applications, in: Proceedings of the SIGMetrics/Performance’06, 2006.