Estimating Average End-to-End Delays in IEEE 802.11 Multihop Wireless Networks

Cheikh Sarr, Isabelle Guérin-Lassous

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Estimating Average End-to-End Delays in IEEE 802.11 Multihop Wireless Networks

Cheikh Sarr — Isabelle Guérin Lassous

N° ????

July 2007

Thème COM

Rapport de recherche

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

Estimating Average End-to-End Delays in IEEE 802.11 Multihop Wireless Networks

Cheikh Sarr * , Isabelle Guérin Lassous †

Thème COM — Systèmes communicants
Projets ARES

Rapport de recherche n° ???? — July 2007 — 21 pages

Abstract: In this paper, we present a new analytic model for evaluating average end-to-end delay in IEEE 802.11 multihop wireless networks. Our model gives closed expressions for the end-to-end delay in function of arrivals and service time patterns. Each node is modeled as a M/M/1/K queue from which we can derive expressions for service time via queueing theory. By combining this delay evaluation with different admission controls, we design a protocol called DEAN (Delay Estimation in Ad hoc Networks). DEAN is able to provide delay guarantees for QoS applications in function of the application level requirements. Through extensive simulations, we compare performance evaluation of DEAN with other approaches like, for instance, DDA.

Key-words: Average end-to-end delay estimation, *ad hoc* networks, IEEE 802.11, Quality of Service

* Inria Ares / Citi, Insa de Lyon – 69621 Villeurbanne Cedex – France
† Université de Lyon / INRIA – 69364 Lyon Cedex 07 – France

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Estimation du délai moyen de bout-en-bout dans les réseaux ad hoc basés sur IEEE 802.11

Résumé : Nous proposons une nouvelle méthode analytique pour estimer le délai moyen de bout-en-bout dans un réseau ad hoc basé sur IEEE 802.11. Ce modèle fournit une expression du délai de bout-en-bout en fonction des trafics d'arrivée et de service. Chaque nœud est modélisé par une file M/M/1/K à partir de laquelle nous pouvons dériver une expression du temps moyen de séjour à l'aide de la théorie sur les files d'attente. En combinant cette évaluation du délai avec différents contrôles d'admission adéquats, nous mettons en place un protocole appelé DEAN (Delay Estimation in Ad hoc Networks). DEAN est en mesure de fournir des garanties sur le délai aux applications QoS en fonction des exigences spécifiées. A travers des simulations intensives, nous avons comparé les performances de DEAN avec d'autres approches telles que DDA.

Mots-clés : Délai moyen de bout-en-bout, réseaux *ad hoc*, IEEE 802.11, Qualité de Service

1 Introduction

Using Voice-over-IP services in wireless mesh networks is becoming a reality [7]. Such VoIP applications are delay sensitive and require some delay guarantees. Offering and maintaining delay guarantees is a difficult task in wireless mesh networks and more generally in multihop wireless networks.

Recently, the works on quality of service (QoS) guarantees in ad hoc networks have attracted more and more attention. Most of these works assume that the underlying technology used in these networks is the IEEE 802.11 technology [5]. This technology is widely available, not so expensive and provides a distributed radio medium access that can be easily used in an ad hoc network. On the other hand, the random radio medium access provided by IEEE 802.11 offers few control on the emissions and makes the radio medium sharing difficult in a multihop context [3]. Many works offer quality of service to IEEE 802.11-based ad hoc networks by providing either throughput guarantees or delay guarantees or both. Among these studies, most of them have focused on throughput guarantees (like for instance [14, 4, 2, 13]) and few of them have proposed solutions for delay guarantees.

If solutions for providing throughput guarantees are still not perfect, they are nonetheless more and more efficient. The complex radio medium sharing in IEEE 802.11-based ad hoc networks is better and better integrated in the QoS solutions. In turn, these solutions can provide accurate available bandwidth estimations and thus guarantee the throughput efficiently. Ensuring the delay seems to be a more challenging task. As mentioned in [15], it is very difficult to predict the expected delay due to the strong dependency between the flows in a wireless multihop setting. In this article, the authors claim that it is very difficult to design measurement-based admission control protocols for the delay parameter compared to the throughput parameter.

In this article, we propose a new protocol to achieve delay guarantees in wireless multihop networks. With this study, we show that it is possible to design an efficient measurement-based admission control protocol for the delay parameter. The proposed protocol, called DEAN (Delay Estimation in Ad Hoc Networks), is based on a *a priori* estimation of average end-to-end delay. This estimation is derived from a simple model of IEEE 802.11 nodes and from an accurate evaluation of each link's collision probability. By combining this estimation with accurate admission controls, the estimated delay is guaranteed after a new flow starts. Such guarantees depend mainly on a strong correlation between the estimated delay and the available bandwidth as an efficient estimation of available bandwidth. This latter is estimated with the protocol ABE (Available Bandwidth Estimation) that provides an accurate evaluation [13]. Moreover, our protocol DEAN is not costly in terms of overhead since it uses the control packets required for the estimation of the available bandwidth and provided by ABE and thus does not add any overhead. Finally, extensive simulations show that our protocol DEAN is very efficient to provide delay guarantees.

The remainder of this paper is organised as follows: Section 2 presents related work. Section 3 presents succinctly the protocol ABE used for available bandwidth estimation. Section 4 describes our end-to-end delay estimation mechanism. Section 5 describes the protocol DEAN and simulation results are presented in Section 6. Section 7 concludes our work. Finally, notations for the entire paper can be found in Appendix.

2 Related Work

One of the goals of VMAC [1] and SWAN [6] is to provide delay guarantees thanks to an admission control based on delay measurements. In [1], a mechanism for evaluating one hop delays in ad hoc networks is presented. This mechanism relies on an estimation of the channel occupancy. Let's consider that a specific node has monitored a channel occupancy equal to U during a measurement interval. If we denote by L the time to transmit successfully a whole packet on this channel, then the one hop delay d can be expressed by:

$$d = U \times d^{'} + (1 - U) \times L$$

If the channel around the node is always free ($U = 0$), there will be no collision and the one hop delay is equal to the transmission time L. However, when the channel is busy due to close transmissions ($U \neq 0$), the parameter $d^{'}$, corresponding to the time spent in the exponential backoff procedure and for retransmitting a collided packet, is taken into account in the delay evaluation. If this approach is effective for a one hop network, it can lead to a false evaluation in presence of hidden nodes configuration: for instance, U can be equal to 0 for an emitter but its transmission can experience collisions due to a hidden flow.

With SWAN [6], the delay requirements of real-time traffic are achieved with a rate control of best effort traffic performed at every node in a distributed manner without keeping per-flow state information. Each node regulates its best effort traffic via an Additive Increase Multiplicative Decrease algorithm. If the delay of best effort packets becomes greater than a threshold delay, then the emitter considers that the network is congested and decreases the rate of best effort traffic it transmits. In [15], it is shown that SWAN provides delay violation while reducing the overall throughput in multihop networks. This is mainly due to a certain level of inaccuracy in the available bandwidth estimation.

DDA [15] is also a stateless protocol that provides average delay guarantees for QoS applications. Each node is modeled as discrete G/G/1 queue: it allows an evaluation under general traffic arrival patterns. The main contribution of DDA are the following: i) the modeling of the closed-form relationship between contention window size and the distribution of the delay of a flow in an unsaturated network; ii) the regulation of the contention window size is based on locally available information and iii) the contention window allocation converges to a steady state if such an allocation exists. To achieve these goals, DDA relies on some assumptions, some of which are quite restrictive in ad hoc networks. For instance, the end-to-end delay is evenly broken down into per-hop delay requirements, which is not necessarily the case in wireless multihop networks. Or minimizing the impact of collisions in the computation of the contention delay is correct in a single-hop cell but may lead to a high inaccuracy in a multihop network.

Some other works model the delay on one hop link with queueing theory techniques, like for instance in [10]. Most of these works base their analysis on the modeling of the IEEE

802.11 access channel performed by Bianchi [8]. If these analysis work well in one hop cell where all the nodes are in a same communication area, it is not often the case in multihop wireless networks where collisions due to hidden node configurations can not be neglected. The goal of our work is to better integrate the features of ad hoc networks for achieving more accurate delay guarantees.

3 Available Bandwidth Estimation

For ensuring delay guarantees, our solution relies on an accurate available bandwidth estimation. Hereafter, we define the **available bandwidth** between two neighbor nodes as *the maximum throughput that can be transmitted between these two peers without disrupting any ongoing flow in the network.* This term should not be confused with the **link capacity** (also called **base bandwidth**) that designates *the maximum throughput a flow can achieve between two neighbor nodes*, even at the cost of other flows' level of service degradation.

For the available bandwidth estimation, we choose the protocol ABE (*Available Bandwidth Estimation*), first proposed in [12] and then refined in [13]. In [13], the authors show that ABE is more accurate than many protocols with the same goal while requiring a small overhead. By considering the overlapping of the silence periods of both emitter and receiver of a link, the collision probability that exists on the link and the backoff window size correlated to this collision probability, ABE provides an accuracy in the estimation that is often not achieved by the other protocols.

As our delay estimation depends strongly on this available bandwidth estimation, this section is devoted to the description of ABE. Of course, we can not include all the details of ABE that is not the novelty of our proposition. The interested reader can refer to [12, 13]. For providing an accurate evaluation, some phenomena need to be taken into account when the IEEE 802.11 MAC protocol operates:

- Carrier sense mechanism prevents two close emitters from transmitting simultaneously. Therefore, an emitter shares the channel bandwidth with all these close emitters. The channel utilization has to be monitored to evaluate the capacity of a node to emit a given traffic volume. As in many protocols, this channel utilization is computed by each node by monitoring the radio medium in its surroundings and measuring the total amount of time that is idle for emitting frames. Therefore, this method does not only take into account the bandwidth used in the transmission range of the nodes but also in the whole carrier sensing area.

- For a transmission to take place, both emitter and receiver need that no jamming occurs during the whole transmission. Therefore, the value of the available bandwidth on a link depends on both peers' respective channel utilization ratios but also on the idle periods synchronization. In [12], we propose a probabilistic method to estimate this synchronization. This estimation, for the link (s, r), is denoted $E(b_{(s,r)})$ in the following.

- No collision detection is possible in a wireless environment. Therefore, whenever collisions happen, both colliding frames are completely emitted, maximizing the bandwidth loss. It is thus necessary to integrate this bandwidth loss in the available bandwidth estimation. In [13, 11], we provide an estimation of the collision probability on each link. This estimation combines two approaches: i) A *on line* approach that computes the impact of the medium occupancy distribution at the receiver side thanks to the collision probability on *Hello* packets. These *Hello* packets are used in many ad hoc routing protocols and are required for computing the previous estimation $E(b)$ on each link; ii) A *off line* approach that takes into account the size of the packets sent by the source thanks to an interpolation. The goal of this last approach is to compute the collision probability that packets of known and fixed size will undergo on a link from the collision probability of *Hello* packets deduced from real measurements on the same link. This collision probability estimation is denoted p, in the following, and depends on the size of packets that will be sent.

- Finally, when collisions happen on unicast frames, the IEEE 802.11 protocol automatically retries to emit the same frame, drawing the backoff counter in a double-sized contention window. The time lost in additional overhead may also have an impact on the available bandwidth. In [13, 11], we compute the mean backoff according to p the collision probability computed in the previous estimation. It is then possible to deduce the proportion of bandwidth consumed by the backoff mechanism. This proportion is denoted by K in the following.

These different estimations are then combined to estimate the available bandwidth on a wireless link, *i.e.* between an emitter s and a receiver r:

$$E_{final}\left(b_{(s,r)}\right) = (1-K)\cdot(1-p)\cdot E\left(b_{(s,r)}\right) \tag{1}$$

4 Mean delay estimation

Delay indicates the time to send a packet from a source to a destination node. Contrary to bandwidth, delay is an additive metric. Thus, the delay along a path is equal to the sum of the delays on the one-hop links of this path.

With the use of IEEE 802.11, the mean packet delay on a specific one-hop link, denoted by D, can be divided into three parts:

- The *mean queueing delay* which represents the interval between the time the packet enters in the queue of the link's emitter and the time that the packet becomes the head of line packet in this node's queue. We denote it by D_q.

- The *mean contention delay* is the interval between the time the packet arrives at the head of line and the time the packet is sent to the physical medium. We denote it by D_c. This interval reflects the fact that a node may contend to access to the channel because of other transmissions in its carrier sensing area.

- The *mean transmission delay* is the time to transmit the whole packet including possible retransmissions in case of collisions. We denote it by D_t.

Therefore we have the relation:

$$D = D_q + D_c + D_t \tag{2}$$

In the remainder of this section, we made some assumptions in order to simplify the analysis and to give an analytical expression for $(D_q + D_c)$ and secondly for D_t.

4.1 Assumptions

We model a IEEE 802.11 node as a discrete time M/M/1/K queue. The properties of this queue are:

- The packet arrival follows an exponential law of parameter λ.
- The service rate follows also an exponential law of parameter μ.
- The size of the queue is limited by the value K. When a new packet arrives and if there are already K packets in the queue, then this one is dropped.
- The queue is a classical FIFO (First in First Out).

We assume no use of RTS and CTS messages. The analysis can be easily extended for the cases where such messages are present. The parameter λ represents the number of packets arriving in the queue per second which depends on the application throughput (if such an application exists on the node) and the traffic routed by this node. The service rate μ represents the number of packets leaving the queue per second.

4.2 General idea

Our initial goal is to provide delay guarantees to delay sensitive flows. To this end, we need to estimate the mean delay that the packets of such a flow will achieve before transmitting this flow. Therefore, we need to estimate the service rate that can be offered to this flow on each node passed through by this flow. It is also important to remind that the acceptance of a new flow may impact the delays of existing flows. Our goal is then also to minimize such an impact in order to achieve the delay guarantees of existing delay sensitive flows. As for the available bandwidth estimation (see Section 3), we could define the **available service rate** of a node as *the service rate that can be offered to a new flow without increasing the delay of any ongoing flow in the network*.

In order to limit the impact on the mean delay of existing flows, a congestion control must be realized. Therefore, the service rate that can be offered by a node to a new flow is directly correlated to the residual bandwidth seen by this node. This residual bandwidth is equal to the medium occupancy seen by this node (including its own transmissions) multiplied by the capacity of this node. This value captures the effect that after the queuing process, a packet which arrives at the head of line of the MAC layer should wait until the channel is free in order to gain the access. More precisely, we model μ_{res}, the service rate that can be offered to a new flow, as the available bandwidth computed by the node rescaled in packets per second.

4.3 Estimating the mean queueing and the contention delay

In this section, we estimate $D_q + D_c$. When $\mu > \lambda$, the service rate of the node is higher than the arriving process and there will be no accumulation in the queue involving a queueing and a contention delay which are null.

$$\mu > \lambda \Longrightarrow D_q + D_c = 0$$

When $\mu \leq \lambda$. Let's denote by $p(n)$ the probability to have n packets in the queue ($n \leq K$). A packet arrives with rate λ and exits with rate μ. So:

$$p(n) = \frac{\lambda}{\mu} \times p(n-1) = \left(\frac{\lambda}{\mu}\right)^n \times p(0)$$
$$\text{Using } \rho = \frac{\lambda}{\mu} \Longrightarrow p(n) = \rho^n \times p(0)$$

The sum of the probabilities being equal to 1, we can simply express $p(n)$ in function of ρ and K:

$$p(n) = \begin{cases} \rho^n \dfrac{1-\rho}{1-\rho^{K+1}} & \text{if } \rho \neq 1 \\ \\ \dfrac{1}{K+1} & \text{if } \rho = 1 \end{cases}$$

The mean number of packets Q in the queue is therefore:

$$Q = \sum_{n=0}^{K} n \times p(n)$$

Using queueing theory and according to Little's law, the parameter $D_q + D_c$ is equal to the *mean waiting time*:

$$(D_q + D_c) = \frac{Q}{\lambda}$$

So the final expression is:

$$D_q + D_c = \begin{cases} \dfrac{\rho}{1-\rho} \dfrac{1-(K+1)\rho^K + K\rho^{K+1}}{1-\rho^K} \dfrac{1}{\lambda} & \text{if } \rho \neq 1 \\ \\ \dfrac{K}{2\lambda} & \text{if } \rho = 1 \end{cases} \tag{3}$$

We can also notice that, since the queue size is bounded, $D_q + D_c$ is bounded by a maximum value D_{max} such as:

$$D_{max} = \lim_{\rho \to +\infty} (D_q + D_c)$$

$$D_{max} \approx \frac{K \times \rho^{K+2}}{\rho^{K+1} \times \lambda}$$

As $\lambda = \rho \times \mu$ we have:

$$D_{max} \approx \frac{K}{\mu} \tag{4}$$

In this estimation, the contention delay only considers the time spent until the medium is free in order to gain the access to the radio medium to send the packet for the first time. We do not consider here the time that may be required to retransmit the packet. This time is included in our transmission delay, described in the next subsection.

4.4 Estimating the mean transmission delay

The mean transmission delay is the time to transmit the whole packet. In IEEE 802.11 DCF, when this operation is successful, a positive acknowledgement is sent back to the emitter. However, there is a chance, even for a single frame that when a packet is emitted, the medium is not idle at the receiver's side, provoking a collision. These collisions involve retransmissions of the same packet and increases the contention window size, all these phenomena resulting in an increase of the mean transmission delay.

4.4.1 Modeling the exponential backoff mechanism

As it has been mentioned in Section 3, we have already proposed in our protocol ABE ([13]) a mechanism to estimate the collision probability on each link.

Let us consider that an arbitrary wireless link suffers from collisions with a probability p (evaluated with the method given in [13]). For every frame, the transmission is successful at the first attempt with probability $(1-p)$. It succeeds at the second attempt with probability $p \cdot (1-p)$. After C unsuccessful attempts, C depending on the frame size, the IEEE 802.11 standard specifies that the frame should be dropped.

If we denote by X the random variable representing the number of attempts performed for the correct transmission of a given frame, we have:

$$P(X = k) = \begin{cases} p^k \cdot (1-p) & \text{if } k \leq C \\ p^k & \text{if } k = C + 1 \\ 0 & \text{if } k \geq C + 1 \end{cases}$$

The expected number of retransmissions n for a given frame can be expressed as follows:

$$n = \sum_{k=1}^{+\infty} k \cdot P(X = k) = \sum_{k=1}^{C+1} k \cdot P(X = k)$$

$$n = \sum_{k=1}^{C} k.p^k(1-p) + (C+1)p^{(C+1)}$$

Now, we need to compute the expected backoff that impacts the delay transmission. First, let us consider that there is no collision. Then the backoff is drawn according to a uniform law in the interval $[0; CW_{min} - 1]$, CW_{min} being determined by the MAC protocol specification. On a large observation window, the backoff can be approximated by its average value $\frac{CW_{min}-1}{2}$.

When collisions happen, the exponential backoff mechanism is triggered. After each unsuccessful transmission, the contention window size is doubled up to a maximum value denoted by CW_{max}. In this situation, the average backoff value increases way above $\frac{CW_{min}-1}{2}$ and it is necessary to model the time consumed by the exponential backoff process.

The expected number of backoff slots decremented until the end of transmission attempts for a single frame is therefore:

$$\overline{backoff} = \sum_{k=1}^{+\infty} P(X = k) \cdot \frac{\min(CW_{max}; 2^{k-1} \cdot CW_{min}) - 1}{2}$$

To simplify the expression, let us suppose that $CW_{max} = 2^c \cdot CW_{min}$ with $c \leq C$:

$$\overline{backoff} = \left(\sum_{k=1}^{c} P(X = k) \cdot \frac{2^{k-1} \cdot CW_{min} - 1}{2} \right) + \left(\sum_{k=c+1}^{C} P(X = k) \cdot \frac{CW_{max} - 1}{2} \right)$$

$$\overline{backoff} = \frac{1-p}{2} \cdot \left(\frac{1-(2 \cdot p)^c}{1 - 2 \cdot p} \cdot CW_{min} + \frac{p^c - p^C}{1-p} \right)$$

4.4.2 Mean transmission delay computation

The different points mentioned above can be combined to estimate the mean transmission delay on a wireless link, i.e. between an emitter and a receiver. To summarize, the mean transmission delay between two neighbor nodes can be estimated by the following formula:

$$D_t = \overline{backoff}.T_{slot} + \sum_{k=0}^{n-1} T_c + T_m$$

$$D_t = \overline{backoff}.T_{slot} + n.T_c + T_m \tag{5}$$

where T_m is the time to successfully transmit a whole packet of m bytes with IEEE 802.11, T_c is the collision duration, n is the mean number of retransmissions depending on collision probability, $\overline{backoff}$ is the expected number of backoff slots and T_{slot} is the duration of a slot.

To sum up, the mean delay D of a one-hop link consists of:

- the mean delay experienced by a packet on the link's emitter ($= D_q + D_c$). It corresponds to the waiting time before the first transmission of the packet;
- the mean delay experienced by a packet during the transmission ($= D_t$). It includes the potential retransmissions induced by collisions.[1]

5 The protocol DEAN

As our goal is to ensure delay for delay sensitive flows, we integrate the previous evaluation technique of the mean delay into a protocol. This protocol is called DEAN for *Delay Estimation in Ad hoc Networks.*

The protocol part, i.e. the setting up and maintenance of reservations, does not include any new or specific feature. It is based on broadcasted route request messages, admission control at each intermediate node and explicit reservation by a unicast route reply message issued by the destination. Our delay estimation requires an available bandwidth estimation. We use the technique designed in the protocol ABE (Section 3), since the obtained results show a high level of accuracy ([13]). The congestion control mechanism required to minimize the impact on the delays of existing flows will then be based on the available bandwidth estimation of ABE and will be performed via an admission control on bandwidth.

5.1 DEAN: Delay Estimation

In DEAN, neighboring nodes exchange their delay information through *Hello* messages. Every Δ seconds, each node locally estimates its medium occupancy ratio and includes this information in a *Hello* packet. The medium occupancy allows DEAN to estimate the available service rate μ_{res}.

Hello-based techniques generate additional overhead depending on the *Hello* emission frequency. Ideally, the frequency Δ of *Hello* packets should be adapted to the nodes mobility and/or to the flows dynamics. The larger Δ, the more stable the measurements, hiding fast variations in the medium load. However, Δ should also be small enough to allow fast

[1] Note that for simplicity in the writing, the time spent in $DIFS$ and in the backoff decrease before transmitting the packet for the first time is included in the transmission delay and not in the contention delay, as the definition lets think. Of course, if the reader thinks that it is more appropriate to include this time in the contention delay, a simple shift can be realized.

reactions to long-term load variation and to nodes mobility. In this protocol version of DEAN, we choose, in order to have meaningful comparisons, to fix this value to $\Delta = 1$ second. Similarly, all compared protocols are tuned accordingly to emit one information frame each second.

5.2 DEAN: Admission control and QoS routing

The DEAN routing protocol is a cross-layer routing protocol. The MAC layer of each node estimates proactively and periodically the mean delay of the neighboring links and the routing layer is in charge of discovering QoS routes complying to the applications demands, basing their decisions on the MAC layer result. We try to provide routes for which the end-to-end delay specified by the application level is higher than the mean value estimated along the path. Let's consider a path composed of K hops. The delay constraint can be expressed by the following inequality:

$$\sum_{i=0}^{K-1} D(i; i+1) \leq D_{appli} \tag{6}$$

where D_{appli} is the end-to-end delay specified by the application level and $D(i; i+1)$ is the one-hop delay between intermediate neighbor nodes i and $i+1$ on the path $(0 \leq i \leq K-1)$.

The routing process of DEAN is strongly inspired by AODV and consists of two major parts: route discovery and route maintenance.

5.2.1 Route discovery

The aim of the route discovery procedure is to find a route between the sender and the receiver that meets delay constraints specified by the application level. Therefore, two flows with the same source and destination can follow different routes depending on the network state.

DEAN performs on-demand route discovery like in AODV. When a source node has data to send, it broadcasts a route request (*RREQ*) to its neighbors. The *RREQ* packet contains the address of the sender, the bandwidth and delay requirements at the application level, the destination address, a sequence number and the cumulative delay computed along the path. Each mobile that receives such a *RREQ* performs three admission controls:

- The first one ensures that Equation 6 is still verified. The computed delay corresponds to the sum of the cumulative delay given in the RREQ packet and of the estimated delay on the link from which the RREQ packet is received. To estimate this latter, we use λ the throughput requirement of the application and μ_{res} the service rate that can be offered to the application by the link's emitter. The service rate that will serve the application on this node corresponds to $min(\lambda, \mu_{res})$.

- The second one ensures that throughput of the flow to be emitted (deduced from the service rate computed at the previous step) will not be degraded by close flows.
- The third one ensures that the emission of this flow on this link will not degrade the throughput of close flows which are in hidden nodes configuration. These two admission controls are performed in ABE [13] and we re-use them in DEAN.

Figure 1 represents the cross-layer mechanism and the different admission controls performed in DEAN.

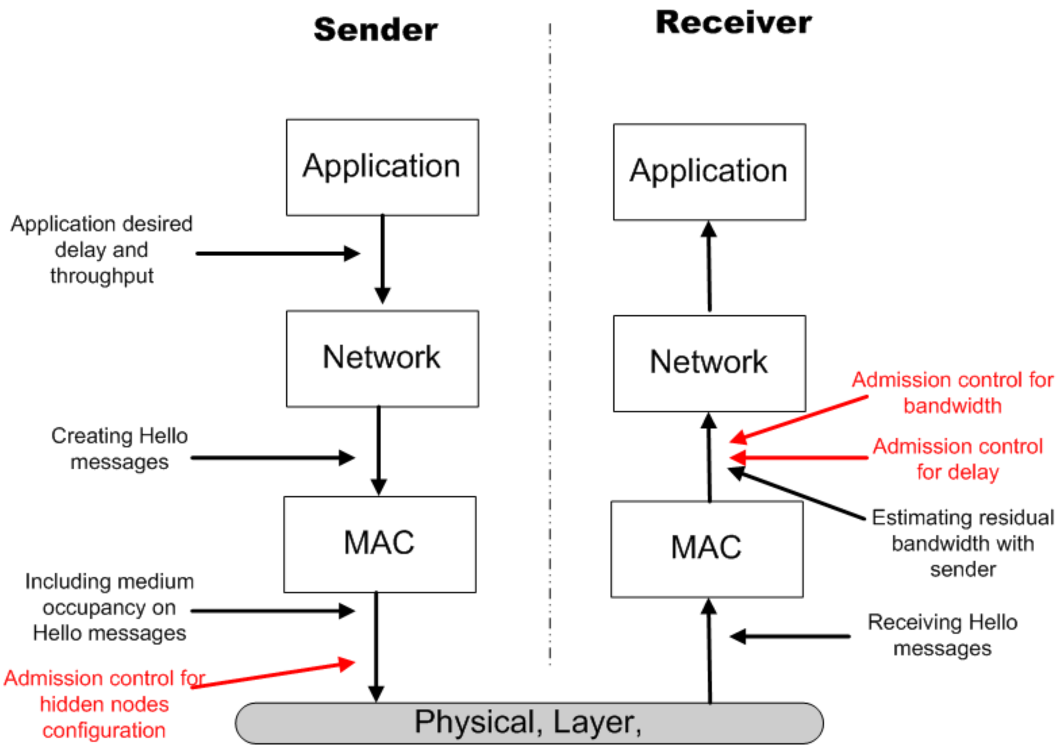


Figure 1: Cross-layer and admission controls in DEAN

If all these controls are positive, the node adds its own address and its mean delay to the cumulative delay of the route and then forwards the *RREQ*; otherwise it silently discards the message. Finally, when the destination receives a first *RREQ*, it sends a unicast route reply (*RREP*) to the initiator of the request along the reverse path. The resources in terms of service rate offered to this flow at each node are then reserved and the new QoS flow can be sent.

5.2.2 Route maintenance

A route maintenance process is essential, especially in case of mobility. We implemented a simple detection and reaction mechanism. DEAN detects a broken route by monitoring the *Hello* messages. If a node does not receive any *Hello* packet from a neighbor within a certain time interval (equal to the time to transmit 3 *Hello* packets in the evaluation part), or if one of its link does not meet the reserved delay any more, it sends a route error ($RERR$) to the source which subsequently rebuilds its route.

Finally, it is interesting to note that DEAN, as it is described here, only guarantees the mean delay to applications. With slight modifications on the admission control phases, it is possible to guarantee both throughput and delay requirements.

6 Performance evaluation

In this section, we evaluate and compare, by simulation, the performances of our solution with other approaches. We use the network simulator 2 (NS-2.27) [2] and the IEEE 802.11 implementation provided with the simulator.

The parameters used for all scenarii are presented on Table 1. Unless specified, the queue of each node (K) can contain at mot 100 packets of 1000 bytes. We compare the performances of our delay estimation technique through the DEAN protocol described above with DDA [15]. We chose DDA because it is one of the most recent solutions that outperforms several measurement-based admission control protocols like SWAN for instance. As DDA relies on any existing protocol for throughput guarantees[3], we have integrated DDA in ABE. Thus, the comparison between DDA and DEAN is focused on the delay estimation and the delay guarantee since these two approaches are based on the same bandwidth admission control and the same available bandwidth estimation. We also compare DEAN with AODV as a baseline for comparison.

Parameters	**Values**
HELLO interval	1 s
Packet size	1000 bytes
Radio medium capacity	2 or 11 Mb/s
Communication range	250 m
Carrier sensing range	550 m
Grid size	1000 m×1000 m
C (Number of retransmissions)	6

Table 1: General parameters for simulations

[2] `http://www.isi.edu/nsnam/ns/`

[3] no protocol is specifically proposed in [15].

6.1 First results

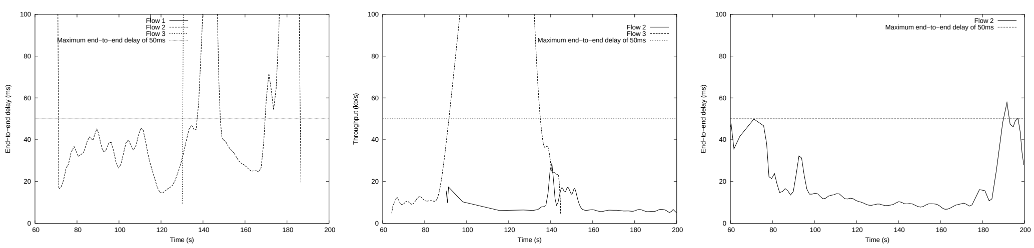
To compare the different protocols and illustrate the effectiveness of DEAN to provide end-to-end delay guarantees, we generate random topologies with random constant bit-rate flows (random source, random destination and random throughput with fixed 1000 bytes frames). For each of these protocols, similar scenarios (same number of nodes and same number of flows) lead to similar behaviors. Therefore, for this section, we give the results of one scenario and the presented results are obtained over 30 simulation runs with different random seeds. The scenario consists of a static network involving 20 randomly positioned nodes. The channel capacity is set to 2 Mb/s. We distinguish two particular cases:

- In the first case, the three delay-sensitive CBR connections are competing with five best effort CBR flows as background traffic.
- In the second case, these three delay-sensitive CBR connections are now competing with five best effort FTP flows as background traffic.

For each flow, the source is randomly chosen and the destination is also randomly chosen. The throughput of each delay-sensitive connection is uniformly drawn between 100 kb/s and 300 kb/s and the end-to-end delay requirement is fixed to 50 ms. The background traffic is first started (their throughput is also uniformly drawn between 100 kb/s and 300 kb/s). Then the delay sensitive flows are started every two seconds. Table 2 gives bandwidth and end-to-end delay requirements for the three delay-sensitive flows.

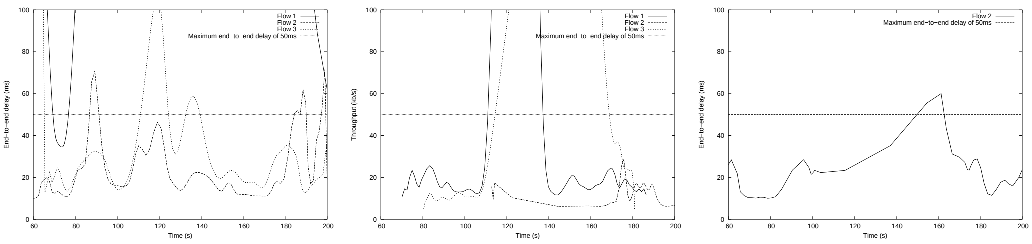
Flow identity	**Throughput (kb/s)**	**End-to-end delay (ms)**
Flow 1	149	50
Flow 2	237	50
Flow 3	212	50

Table 2: Bandwidth and end-to-end delay requirements for delay-sensitive flows



(a) AODV (b) DDA (c) DEAN

Figure 2: End-to-end delay obtained by AODV, DDA and DEAN for delay-sensitive flows and CBR flows as background traffic



(a) AODV (b) DDA (c) DEAN

Figure 3: End-to-end delay obtained by AODV, DDA and DEAN for delay-sensitive flows and FTP flows as background traffic

Figures 2(a) and 2(b) show flow delays during the simulation for AODV, DDA and DEAN. Both AODV and DDA show delay violations as the initial end-to-end delay requirement was fixed to 50 ms. The three delay sensitive flows are transmitted with AODV while only two flows are accepted with DDA. In this simulation, about 31% of packets undergo a delay violation with DDA and this value increases until 44% for AODV. With DEAN (Figure 2(c)), only one flow is accepted and transmitted and the delay violation is almost equal to 0.09%. This scenario illustrates the ability of DEAN to provide end-to-end delay guarantees in wireless multihop networks.

The same observations can be done when the background traffic is composed by FTP flows as shown in Figures 3(a), 3(b) and 3(c).

6.2 Accuracy of the estimation

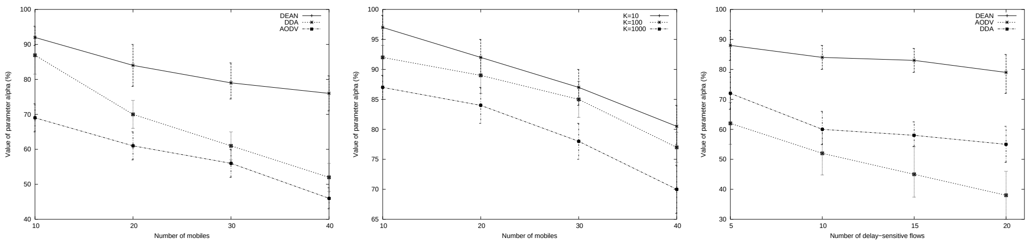
Let us now investigate more scenarii. To evaluate the accuracy of the end-to-end delay estimation in several scenarii, we use the following metric:

$$\alpha = \frac{n}{N}$$

where n is the number of packets that don't experience delay violation and N is the total number of packets correctly received. We consider that a packet, for which the end-to-end delay may be higher than 5% of the delay requirements when it is emitted, experiences a delay violation. This metric reflects the fact that a falsely admitted flow either degrades delay of close flows or is not able to achieve its own desired end-to-end delay.

We measure the value of α by simulation on networks composed of 10 to 40 nodes. The radio medium capacity is of 11 Mb/s. Each simulation lasts 100 seconds and three random sources try to establish delay-sensitive CBR connections towards three random destinations. These three CBR connections are competing with 5 best effort CBR flows, started at the beginning of the simulation. The throughput of each connection is uniformly drawn between

100 kb/s and 300 kb/s and the end-to-end delay requirements is fixed to 50 ms for all delay-sensitive flows. In the following, we evaluate the values for α in function of three parameters: the used protocol; the queue size and the number of delay-sensitive flows[4]. All the results presented below are the average of 30 simulations with different random seed.



(a) Value of α in function of the used protocol

(b) Value of α in function of queue size for DEAN

(c) Value of α in function of the number of delay-sensitive flows

Figure 4: Values of α

Influence of the used protocol: Figures 4(a) gives the values of α in function of the number of nodes present in the network. When the network is not too loaded, α is relatively high for both DEAN and DDA (respectively about 92 and 86% for 10 nodes). By increasing the density of the network, we see that DEAN is more accurate on delay guarantees than DDA. For example, when there are 30 nodes, the value of α is about 79% for DEAN and 61% for DDA.

DDA proposes a stateless approach, meaning that no estimation of residual resources are carried out. Each node of DDA regulates dynamically the contention window size in function of its local perception of close transmissions. This regulation is also based on some assumptions. The obtained results let us think that some of these assumptions are maybe too strong, especially when the density of the network increases. Indeed, when the density becomes higher, more collisions may appear involving packets retransmissions. Moreover, the delay on a path may very probably not evenly broken down among the hops of this path. Therefore, DDA handles less efficiently these collisions in the delay evaluation than DEAN. In all cases, the performances of DDA and DEAN are better than AODV for which no admission control and regulation are done.

Influence of queue size: Figure 4(b) shows the values of α in function of the queue size of all nodes in the network. Increasing the queue size decreases the value of α. For example, for a density of 10 nodes, α is equal to 97% for a queue size of K =10 and 87% for a queue size of K =1000. It can be explained by the fact that, with small queues, the queueing delays will be smaller as the variance in delay. Therefore, the delays are more stable when nodes have small queues.

[4]In this last case, the number of delay sensitive flows will not be three.

Value of K	10	100	1000
% of dropped packets	3.1	5.4	7.3

Table 3: % of dropped packets in function of queue size

Surprisingly, the rate of dropped packets also decreases with K as shown in Table 3. Here the results are the average of 30 simulations. These results tend to show that the stability in delays is important for α and for the rate of dropped packets and that DEAN is more efficient with a low value for K.

Influence of the number of delay-sensitive flows: For this evaluation, the number of nodes is fixed to 20. We increase gradually the total number of delay-sensitive flows from 5 to 20 and compute the values of α. As shown on Figure 4(c), for DDA, α decreases when the number of delay-sensitive flows increases. Actually, DDA only performs an admission control on bandwidth but not on delay and the regulation performed by DDA becomes more and more difficult when the total amount of traffic increases in the network. Therefore, these results show that an admission control on delay that eliminates potential routes is of some interest when the network becomes loaded. For example, when there are 20 delay-sensitive flows, α is equal to 85% for DEAN and 55% for DDA. This part illustrates the scalability of DEAN.

6.3 Overhead

Routes establishment and reconstructions require messages exchanges (*RREQ*, *RREP* and *RERR* packets). We evaluate the overhead introduced by AODV, DDA and DEAN which gives an indication on the convergence speed of the routing mechanism and on its stability. The tested random topologies consist of 20 nodes and 5 delay-sensitive flows competing with 5 CBR background traffic. We compute the overhead generated in the network, expressed in total number of exchanged control messages (*RREQ*, *RREP* and *RERR* packets) by all the nodes in the network.

Protocols	**Total number of control packets**	**Confidence interval**
AODV	1248	1206 - 1290
DDA	584	553 - 615
DEAN	368	342 - 394

Table 4: Routes establishment and maintenance-related messages overhead

Table 4 represents the total number of control messages exchanged for the routes setup and maintenance. Results are the average of one hundred simulations performed for each protocol. DDA an DEAN generate less overhead than AODV since the admission control phase eliminates routes that can not meet application requirements in term of bandwidth and end-to-end delay. Hence, the admission control reduces significantly the number of route request packets forwarded in the network. Moreover, AODV suffers from a lot of routes breakages and performs many route reconstructions. These extra requests increase the overhead. We can also see that DEAN generates lower overhead than DDA because in DDA, just one admission control on bandwidth is performed while for DEAN, this admission control on bandwidth is combined to an additional end-to-end delay admission control, leading to fewer route requests propagated in the whole network.

7 Conclusion

In this article, we propose a technique to evaluate *a priori* the delays that flows will achieve in a wireless multihop network. By combining this technique to admission controls based on available bandwidth and on delay, we design a protocol DEAN that can guarantee delays to delay sensitive flows with a high level of accuracy. Simulations compare the performance of AODV, DDA and DEAN and show that, under different configurations, DEAN is able to provide strict guarantees.

In the future, we plan to work towards three main directions. First, in the presented study, we haven't considered the co-existence of delay sensitive traffic and best effort traffic. If, in simulations, some best effort flows were present, they were transmitted before the starting of delay sensitive flows. Our solution is based on an admission control that ensures that congestion will not appear. It is therefore important to take care that best effort flows started while delay sensitive flows are transmitting will not impact the delays of these latter. To be efficient, it will be also important to let the possibility to use the whole bandwidth to best effort traffics if no delay sensitive flows are transmitted. In [9], we have already designed a mechanism to ensure the co-existence between best effort flows and bandwith sensitive flows. We plan to extend this approach to delay sensitive flows. Second, we plan to merge the different QoS solutions we have designed in order to provide a unique solution that will handle best effort traffic, bandwidth sensitive traffic and delay sensitive traffic. Our goal is to provide accurate guarantees while using the bandwidth efficiently. Finally, we plan to test our solution with real delay sensitive applications, like for instance, VoIP applications. Other mechanisms like VoIP packet aggregation and header compression, as mentioned in [7], should also be considered.

A Notation

Here are the different notations used in the paper:

1. λ: number of packets arriving per second
2. μ: number of packets leaving per second
3. ρ: ratio between λ and μ
4. K: maximum queue size
5. D: mean delay on one hop

6. D_q: mean queueing delay
7. D_c: mean contention delay
8. D_t: mean transmission delay
9. $p(n)$: probability to have n packets in the queue
10. Q: mean number of packets in the queue
11. D_{max}: maximum queueing and contention delay
12. p: collision probability on a link (it depends on the size of packets to be emitted)
13. CW_{min}: minimum contention window size
14. n: mean number of retransmissions for a collided packet
15. $\overline{backoff}$: mean number of backoff slots
16. Δ: *Hello* packets frequency
17. D_{appli}: end-to-end delay desired by the application
18. α: proportion of QoS packets which don't induce end-to-end delay violation.

References

[1] A. Veres, A. Campbell, M. Barry and L-H SUN. Supporting Service Differentation in Wireless Packet Networks Using Distributed Control. *IEEE Journal on Selected Areas in Communications*, 19:10, October 2001.

[2] H. Badis and K. Al Agha. QOLSR, QoS routing for Ad Hoc Wireless Networks Using OLSR. *European Transactions on Telecommunications*, 15(4), 2005.

[3] C. Chaudet, D. Dhoutaut, and I. Guérin Lassous. Performance Issues with IEEE 802.11 in Ad Hoc Networking. *IEEE Communication Magazine*, 43(7), July 2005.

[4] C. Chaudet and I. Guérin Lassous. BRuIT - Bandwidth Reservation under InTerferences influence. In *In Proceedings of European Wireless 2002 (EW2002)*, Florence, Italy, Feb 2002.

[5] IEEE Standard for Information Technology Telecommunications and Information Exchange between Systems. Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.

[6] Gahng-Seop Ahn, Andrew T. Campbell, Andreas Veres and Li-HsiangSun. SWAN: Service Differentiation in Wireless Ad Hoc Networks. In *IEEE INFOCOM*, 2002.

[7] S. Ganguly, V. Navda, K. Kim, A. Kashyap, D. Niculescu, R. Izmailov, S. Hong, and S. R. Das. Performance Optimizations for Deploying VoIP Services in Mesh Networks. *IEEE Journal on Selected Areas in Communications*, 24(11), November 2006.

[8] Giuseppe Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, Volume 18(3):pages 535–547, March 2000.

[9] S. Khalfallah, C. Sarr, and I. Guérin Lassous. Dynamic bandwidth management for multihop wireless ad hoc networks. In *VTC-Spring*, 2007.

[10] Mustafa Ozdemir and A. Bruce McDonald. An M/MMGI/1/K Queing Model for IEEE 802.11 Ad hoc Networks. In *Proceedings of Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, Italy, October 2004.

[11] C. Sarr, C. Chaudet, G. Chelius, and I. Guérin-Lassous. Improving accuracy in available bandwidth estimation for 802.11-based ad hoc networks. In *Third International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Vancoucer, Canada, October 2006. IEEE.

[12] C. Sarr, C. Chaudet, G. Chelius, and I. Guérin-Lassous. A node-based available bandwidth evaluation in ieee 802.11 ad hoc networks. *International Journal of Parallel, Emergent and Distributed Systems*, 21(6), 2006.

[13] C. Sarr, C. Chaudet, G. Chelius, and I. Guérin Lassous. Improving Accuracy in Available Bandwidth Estimation for 802.11-based Ad Hoc Networks. Technical Report 1, INRIA, June 2007.

[14] Y. Yang and R. Kravets. Contention Aware Admission Control for Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 4(4):363–377, 2005.

[15] Y. Yang and R. Kravets. Achieving Delay Guarantee in Ad Hoc Networks by Adapting IEEE 802.11 Contention Windows. In *Infocom*, Anchorage, USA, May 2007.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

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