Achieving a good trade-off between complexity and enhancement in cross-layer architectures

Lamia Romdhani and Christian Bonnet
Mobile communication department, Institut Eurécom
2229 Route des Cretes BP 193, F-06904 Sophia Antipolis Cedex, France.
{romdhani,bonnet}@eurecom.fr

Abstract

The cross-layer design approach is an important concept in mobile ad-hoc networks which is adopted to solve several open issues. It aims to overcome MANET performance problems by allowing protocols belonging to different layers to cooperate and share network status informations while still maintaining separated layers. While cross-layer model could enhance the performance of the applications and achieve better QoS support, there is a lot of proposed models that have to be compared and optimized. In this paper, we investigate the different requirements to achieve a good cooperation. Then, we compare two cross -layer models dealing with delay sensitive application constraints and stability issues in MANET. The simulation results demonstrate that the performance of the proposals is strongly depending on the selective shared parameters.

Keywords: mobile ad hoc networks, cooperation, complexity, quality of service, MAC layer, routing layer.

1. Introduction

Quality of Service (QoS) support is critical to wireless home networking, video on demand, audio on demand and real-time voice IP applications. Time-bounded services such as audio and video conference typically require some specified bandwidth, delay and jitter guarantee. The problems of wireless links are mainly due to the high cost of channel access negotiation and transit network characteristics.

An ad hoc wireless network consists of a group of mobile nodes and all communication is carried out through wireless links in a distributed fashion without a centralized controller. It has different properties when operating in different nodal movement patterns, performing different tasks and carrying varieties of patterns of traffic. The topology of an ad hoc network varies as a result of the mobility of its mobile hosts and the links break down and set up more frequently. A number of factors such as limited transmission range and power limitations, force long-distance communication in ad hoc networks to go through multi hops and each intermediate node is not responsible for the traffic it relays. Routing in ad hoc networks has to adapt to the unexpected link breakage and topology changes. To discover and maintain the routes in ad hoc networks requires more control traffic, which makes the task of performing ad hoc network routing more complex and less efficient. Indeed, due to the random movement of nodes, the bandwidth and power limitations, and the lack of fixed infrastructure, the development of efficient protocols to support the various networking operations in mobile ad hoc networks (e.g., routing, resource allocation, quality of service (QoS) support, etc.) presents many issues and challenges. In the past, a lot of research have been conducted to address these issues separately. One new research direction to optimize data transfer in ad hoc networks is the cross-layer design without respecting the original layered design approach in which each layer operates independently. The main issue of the proposed models in the literature is the determination of what information could be shared and how is it used in a cross-layer architecture to provide QoS enhancement and enable an efficient resource utilization? Hereafter, we describe the different requirements to achieve a trade-off between performance enhancement and complexity. Then we compare the performance of two proposed cross-layer models [2], [1] under different network scenarios.

The remainder of this paper is organized as follows. In Section, 2we give the most important remarks that have to be addressed when introducing a new cross-layer model. The description of two compared proposed cross-layer routing protocols is given in Section .3Simulation methodology and performance evaluation of our proposal are detailed in Section. 4Section 5concludes the paper.

2. Analysis and Remarks about cross-layer architecture issues

The cross-layer model is introduced mainly to enhance the performance of the real time applications and achieve better QoS support. However, the proposed cooperative algorithms and parameters have to be rigorously selected, compared, and optimized. In the most cases, we have to take into account the benefits of each model that provides layer cooperation comparing to its complexity. Indeed, there are some proposals that compute global or local metrics which are used to make decisions for route establishment, scheduling, tuning transmission rate, etc. However, using these metrics in a cross-layer model could be not efficient because they are have sometimes inaccurate values which do not reflect the real situation around a given node. Moreover, since a node moves with an arbitrary speed and toward an arbitrary destination, the computed metrics (according to the participation of the node in communication and the traffic load level around it) could change during the time. So, other nodes that consider the metrics of that node to build routes for example, could have an inaccurate information since this later change according to mobility, traffic, and capacity.

We believe that cross-layer a QoS model is a somewhat “danger”. In one hand, the modification that we have to add in the protocol stack and the complexity in introducing a new parameters and new algorithms to provide a ”good” layer cooperation are usually introduce a high complexity risk. In the other hand, this could be very interesting given that it captures the characteristics of the capacity, the expected behavior of node load to choose the ”best route” between sources and destinations in a way to achieve a global load balancing, and in other cases have knowledge about neighbor density and ”quality” to adapt transmission rate and to use scheduling strategies in an efficient manner.

So, if we recapitulate, cross-layer is a promised solution to address QoS support and service differentiation in mobile ad hoc networks, but it is affected by mobility and so the “lifetime“ of the availability of the accurate available informations. We recommend the following requirements to efficiently design a QoS cross-layer model which leads to the architecture shown in Figure1 :

1. **Choosing the metrics:** choosing of a very useful and efficient metrics such as battery level, available bandwidth, and mobility rate.
2. **Computing the metrics:** the way of computing these metrics regarding one path (energy, lifetime of nodes, throughput, delay, etc.) have to be decided. The well-known approach is to minimize a cost function for a given link in the path between a source and a destination then consider the different costs computed for all links in the path. Depending on the nature of the metric, the cumulative value could be additive, concave and multiplicative. Other techniques could be also used such are variance and max-min. Computation and complexity costs should always be taken into account.
3. **Adapting metrics' values:** an adaptive method should be used to update the measured metrics: They could be updated even more when mobility increases and less in a stable network while taking into account traffic load variation and application requirements.
4. **Deciding to use or not the metrics:** As shown in Figure 1, considering the information useful for model selection, the more efficient model has to be chosen according to the two following parameters:

(a) **Regarding to the network behavior:** in some cases, when the traffic load and its characteristics change rapidly (high mobility), it is very difficult to compute accurate values of the metrics that can be used to address QoS. Hence, the complexity of the cross-layer model becomes too high comparing to the expected performance enhancement and it is recommended in this case to use the legacy layered approach.

(b) **Regarding to the user application:** each layer of the protocol stack responding to local variations and informations from other layers. We have to evaluate the benefits and the disadvantages of the cross layer model for each specific user application.

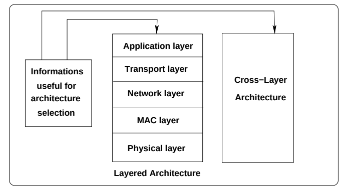

Figure 1.New architecture design

To investigate the evoked remarks, we compare two cross-layer models that we introduced in [2], [1]. We evaluate the application performance of these models under various network scenarios. The simulations results show the importance of considering the above presented requirements.

3. Description of two proposed cross-layer models

Hereafter, we describe two cross-layer models. They are based on the interaction between MAC and routing layers. In [2] the proposed mechanism aims to route the data packets over the most stable routes. In [1], the new architecture allows sensitive-delay applications to achieve a low delay.

3.1. A cross-layer on-demand routing protocol for delay-sensitive applications

3.1. 1.Short overview

In this scheme [1], the MAC layer computes the transmission delay which is simply the ACK's reception time by the MAC layer minus the packet's reception time from the routing layer. This delay takes into account queuing, transmission, and propagation delays, and is computed for all transmitted packets during a configurable *period* called T. At the end of each period T, the node updates the average transmission delay D_{avg}. The obtained time also takes into account the eventual retransmission retries at the MAC layer.

• **Including delay information in routing control packets**

The routes are established based on the end-to-end estimated delay transmission cost. Each RREQ packet includes the traversed path cost. Then each node maintains for each reverse route this cost in the routing table so that routes are built based on the route's cost, defined as follows:

$$Cost_{(s,d)} = \sum_{c \in path(s,d)} D_{avg}$$

Before sending a route reply, the destination has to include the computed route's cost which is mentioned in the routing request that it received. Note that in the basic AODV [4], intermediate nodes can send back to the sender routing reply message when they have already stored information about routes to each the destination. In this proposal, we disable this feature as we want the sender to receive an up-to-date information about the estimated end-to-end delay. Hence, all route replies are sent by the destination.

- **Selecting the best path**

The basic AODV routing protocol uses the minimum hop count criteria to establish routes between sources and destinations. However, considering the new cross-layer model, we follow a new route discovery scheme. Indeed, the source uses the first path retrieved while still accepting other routing replies during the user session lifetime. When a new routing reply arrives, the source observes the estimated delay included in this reply. If it is less than the stored delay of the current available route, it updates the metric of that route entry and uses it for the next packets. This policy allows to reduce the time that the source waits to send the data packets as it cannot know in advance how many route replies it will receive.

- **MAC layer buffer management interaction with routing layer**

The mechanism consists of reordering audio packet transmissions in the queue according to two parameters. The first, is the total delay that packet experienced during the route until arriving at the current node. To avoid synchronization problems, we incorporate in the packet header the sum of times that the packet experienced in the previous intermediate nodes. The second parameter is the estimated delay that the packet has to experience before it reaches the destination. This parameter is available in the routing table and it is shared with the MAC layer. If the sum of the two delays close to the maximum tolerated end-to-end delay, the packet has to be enqueued at the head of the audio class'queue. Audio packets are then inversely ordered according to their remaining lifetime. Moreover, packets that exceed their maximum tolerate end-to-end delay are dropped. By this way we alleviate the network congestion since these packets will not be considered at the destinations.

3.2. A cross-layer stability-based on-demand routing protocol for MANET

A neighbor is considered stable with regard to a given node when they are stable or they are moving to the same direction. However, this cannot be known without monitoring frames sent by each node's neighbor in absence of GPS-based localization system.

3.2. 1.A short overview

Each node in the mobile ad hoc network maintains and a stability vector which measures the stability of neighbors. This vector is updated at each given period. When a node i receives a route request from its neighbors j, it updates its next hop to the source according to the stability rate link (i,j) in order to use the most stable next hop for the reverse path. On the other hand, when a node receives a route reply, it updates the stability information field by taking into account the stability rate of the node from which it received this message. The **minimum stability** rate value of intermediate node is maintained for each route. At MAC layer, the node computes the average load of all the neighbors by observing received packets even the received node is not the destination. Hence, a field in a `load rate vector` is maintained for each neighbor. This load ratio will be applied as a weigh to select two routes that have the same `stability rate.` The `stability rate` is computed as follows:

- **Computing the stability vector**

Each node monitors the control packets received from its neighbors. These control packets include not only routing period packets but also each frame that the MAC layer receives.

A stability vector is maintained by each node which is indexed by the neighbors link layer 2 identity. Hereafter, we detail how this vector is computed and updated and explain their effects on route lifetime and so on the end-to-end QoS guarantees in mobile ad hoc networks.

Let's first define some useful parameters. Denote by $\triangle_{up}$ the update period after which the stability vector V_i at node i will be updated and let $n_i$be the number of neighbors of node i.

Assume $n_{ij}(\triangle_{up})$ be the number of Hello messages received from node j at node i during $\triangle_{up}$. Note that $n_{ij}(\triangle_{up}) \leq n_{HELLO}^{max}$, where n_{HELLO}^{max} is the expected maximum number of Hello messages that can be sent during $\triangle_{up}$. Indeed, as Hello messages are broadcasted periodically, we can know the maximum number of these messages that could be sent by any node in the network in a given period.

Given the parameters defined above, the stability rate of node j at node i is then $V_i(j) = s_{ij} = \frac{n_{ij}(\triangle_{up})}{n_{HELLO}^{max}}$, $0 \leq s_{ij} \leq 1$.

- **Selecting the stable reverse path by intermediate nodes**

Before forwarding a RREQ message, a node has to include in the `stability information` field of RREQ, its stability rate to the source. When a node i receives a route request message `RREQ(s,d)` from a neighbor j. It computes a new stability rate for the source as follows:

$$SR_{new}(s,i) = \min\left(SR(s,j), s_{ij}\right)$$

If the node has already a routing entry to the source node s through its neighbors k $(k \neq j)$, it has to consider the one that has the **maximum stability rate** and to update the routing table according to the obtained result. If the next hop to the source s has been modified, the node i has to broadcast the RREQ packet to its neighbors.

The operations described above are done at each intermediate node, until the route request reaches the destination d. The source selects the best path when it receives more than one route reply by choosing the **maximum** of the minimum stability rate obtained values. There are more details about this model in [2].

4. Performance Evaluation

The above models are implemented in the ns-2 network simulator [3]. We have extended the AODV routing protocol and the Enhanced Distributed Channel Access (EDCA) scheme [5], to support our two proposed cross-layer algorithms. We report in this section the main comparison results of simulations we have done for various network scenarios.

4.1. Scenario description

The simulated scenarios consist of 50 nodes located in a uniform distribution within an area of 1500x300m forming a multi-hop network. These scenarios are generated by the enhanced random way-point mobility model [6]. The sources are CBR and generate UDP at 4 packets/second, each packet being 512 bytes. Note that the number of source nodes is 30 sources. The radio model is very similar to the first generation WaveLAN radios with nominal radio range of 250m. The nominal bit rate is 2 Mbps. In our simulation the nodes move at an average speed of $15m/s$. We provide simulations for different pause time.

Table 1.IEEE 802.11a PHY/MAC parameters used in simulation

SIFS	$16\mu s$
DIFS	$34\mu s$
ACK size	14 bytes
Data rate	36 Mbits/s
Slot_time	$9\ \mu s$
CCA Time	$3\mu s$
MAC Header	28 bytes
Modulation	16-QAM
Preamble Length	$20\mu s$
RxTxTurnaround Time	$1\mu s$
PLCP header Length	$4\mu s$

In the following simulations, we assume that each wireless station operates at IEEE 802.11a PHY mode-6, see network parameters shown in Table 1.

4.2. Performance metrics

We compare the performance of the two proposed models using the following metrics:

* **Packet delivery fraction**: The delivery fraction is measured as the ratio of the number of data packets delivered to the destination and the number of data packets sent by the source.

* **Routing overhead**: It is the total number of Bytes of transmitted routing packets.

* **Mean delay**: It is the average delay of all the flows. The average delay is used to evaluate how well the schemes can accommodate real-time flows.

4.3. Simulation results

We denote by `delay-model` the first described model and the `stab-model` the second one.

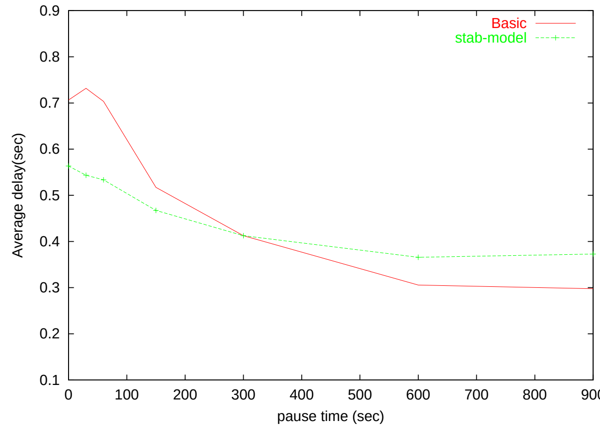

Figure 2.Mean delay results

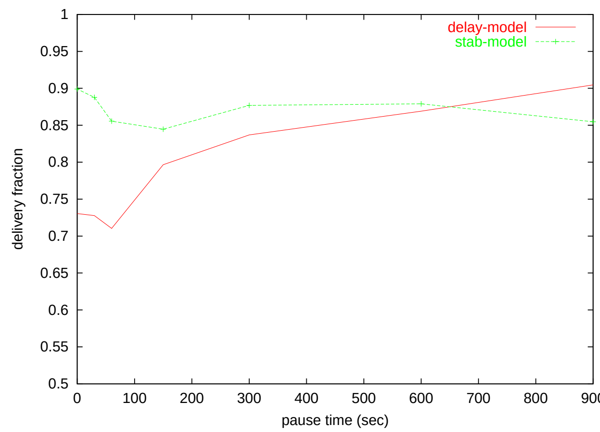

Figure 3.Total packet delivery fraction

In Figures 2, we plot the mean delay of the two mechanisms. It's obvious from the curves that the mean delay is improved well using `delay-model` when there is no mobility (pause time=900). Indeed, the model enables packet routing over less congested nodes. However, this good performance decreases when the node mobility increases. Hence, the `stab-model` performs better in such frequent scenario changes. This later, allows re-routing and refresh routes including new nodes that have better quality than in the old routes which improves the end to end delay. Moreover, we remark that the improvement on delay increases with high network mobility. Furthermore, we can also observe this difference on performance in Figure 3 and 4. Indeed, for the `delay-model` the packet delivery ratio increases when the nodes are more stable. The routing overhead results obtained with this mechanism decreases when considering low mobility. These results demonstrate the importance of the adequate selection of cross-layer parameters regarding both network metrics and application requirements. In one hand, the efficiency of the `stab-model` is shown with high link changes. This mechanism is able to select stable routes even with mobile nodes but they follow the same movement direction. In the other hand, with the `delay-model` the performance improvement is obtained only when considering a stable nodes. The reason is that there is any scheme was considered to overcome the high mobility problem. So, we can conclude that the performance results depend strongly on the network characteristics.

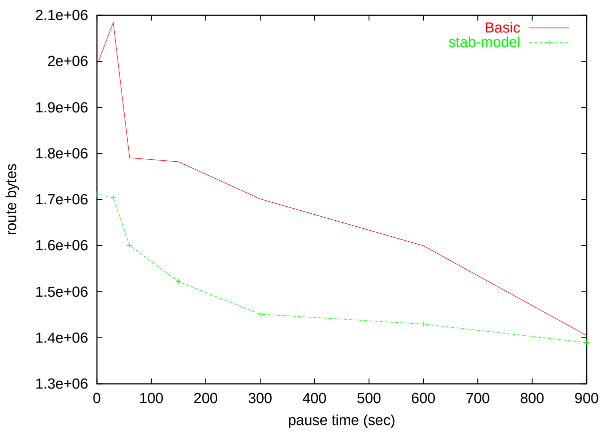

Figure 4.Routing overhead results

By this comparison study, we demonstrate that cross-layer models could not be efficient when network characteristics change. This let thinking about adding an intermediate model as we evoked in the Section. 2The aim of this model is to decide to consider or not the extra parameters that are introduced to provide layer cooperation.

5. Conclusion

The decision to use a cross-layer model is very coupled with the nature of the user application and the evolution of the network behavior. The very promising cross-layer design model consists in maintaining the layer isolation in the protocol stack while enabling a cross-layer interaction according to network and traffic characteristics. We have to specify and explain whether cross-layer paradigm is suitable for all types of wireless networks and applications or not. Even if the answer is yes, it is necessary to maintain the layered approach, while enabling interactions between various protocols at different layers. Unless, the complexity of the new architecture could be expensive and inefficient regarding the performance enhancement as we have shown in the compared examples that we evoked in this paper.

6. References

[1] L.Romdhani, and C.Bonnet, A cross-layer on-demand routing protocol for delay-sensitive applications, 16th Annual IEEE PIMRC 2005, Berlin, Germany.

[2] L.Romdhani, and C.Bonnet, A cross-layer stability-based on-demand routing protocol for mobile ad-hoc networks IWWAN 2005, International Workshop on Wireless Ad-hoc Networks, May 23-26, 2005, London, U.K.

[3] Network Simulator-2. www.isi.edu/nsnam/ns/.

[4] Perkins, *Ad hoc on demand distance vector (AODV) routing*, IETF, RFC 3561, Mai 2004.

[5] IEEE 802.11e WG. Draft supplement to Part 11: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: MAC Enhancements for Quality of Service (QoS)*, IEEE Std 802.11e/D9.0, August 2004.

[6] J.Yoon and al., *Sound mobility models*, International Conference on Mobile Computing and Networking archive Proceedings of the 9th annual international conference on Mobile computing and networking, San Diego, CA, USA 2003.