Evaluation of MANET charging protocols in hotspots scenarios

João Paulo Barraca, Susana Sargento, Rui L. Aguiar
Insituto Telecomunicações - Aveiro, Universidade de Aveiro, Portugal
jpbarraca@av.it.pt, {ssargento, ruilaa}@det.ua.pt

Abstract — **This document compares several proposals for charging and rewarding mechanisms in ad-hoc networks, in terms of its real-time charging efficiency and impact on the overall network throughput. We notice that the introduction of charging and rewarding mechanisms in the ad-hoc network may have impact on the network performance, and then these proposals must be carefully chosen.**

I. INTRODUCTION

One of the major concerns in future networks is how to provide Internet access all around the world without any limitations and with reduced costs for the users and providers. As a consequence of this growth willing, hotspots are appearing all around the globe and in the most different and remote places. This is very profitable for the provider, which increases their revenues, and for the user, which can be connected to the Internet anytime and anywhere.

Since radio range is much reduced in closed spaces or areas with dense radio interferences, the multi hop characteristics of mobile ad-hoc networks are very appropriate to provide extra radio range with a low cost and easy deployment. This is especially attractive in places with high concentration of nodes like urban areas or shopping centres. In such scenarios network providers are responsible for maintaining the contracted services and police the network for misbehaved nodes. The network infrastructure will grow dynamically without any additional cost or intervention by the operator, while revenues will increase.

These virtual operators are already popular nowadays and allow users to share their internet connection to other users. This interaction and the creation of delegated mini-service providers result in some reward to the user providing the service, connectivity for users connected to the hotspot, and ultimately some profit to the operator responsible for managing and authenticating users. Extending the existing concept will result in a cooperative network where all nodes are united with the purpose to increase connectivity and overall network throughput. Although nodes are mainly cooperating, the services each user accesses and the set of applications used are very different. Also, the expectations and the requirements of users change.

The network we envisage in this document is depicted in Fig. 1. We assume that there is at least one ad-hoc access network connected, through an Access Router (AR), to the infrastructure network. The infrastructure network is managed by a network operator and all equipments are trusted: the AR, the Public Key server which stores keys both of the users and the operator, the Authentication, Authorization, Accounting and Charging (AAAC) server that collects and verifies the proofs, and the Internet gateway which connects the infrastructure network to other networks.

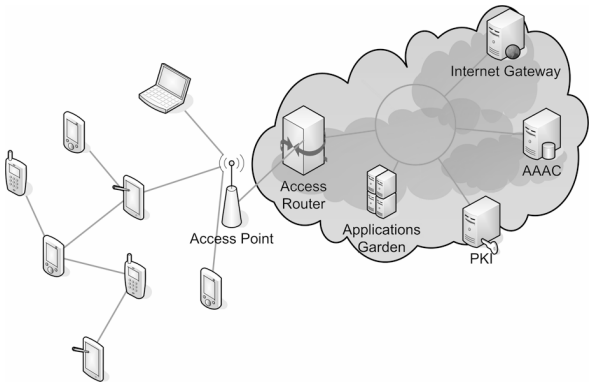

Fig. 1 - An Ad-hoc Network integrated in commercial hotspot.

We assume that users have a contract with the network operator, specified in a contract, and a pair of public and private keys. The equipments in the ad-hoc network are considered to be non-trusted and potentially selfish, not wasting resources for others' profit.

There are in the literature some proposals for the charging and rewarding processes in ad-hoc networks connected to infrastructure networks ([1], [2], [5], [4]). In this document we describe some of the existing proposals addressing both the issue of charging and rewarding. We will also present the results of the simulations preformed to evaluate the behaviour of the described solutions. We notice that the introduction of charging and rewarding mechanisms in the ad-hoc network may have impact on the network performance, and then these proposals must be carefully chosen.

This work is in part supported by the EU FP 6 for Research and Development. Daidalos (IST-2202-506997).

The paper is organized as follows. Section 2 presents the concepts of charging in mobile ad-hoc networks. Sections 3 to 6 briefly present each of the studied charging and rewarding schemes. Section 8 addresses the comparison between these schemes and evaluates the achieved results, and section 9 provides the final conclusions of this study.

II. Charging in Mobile Ad-hoc Networks

One of the major requirements for the integration of ad-hoc networks in hotspot environments is the charging of the services being accessed through the network. Notice that, due to the dynamic nature of ad-hoc networks, with nodes dynamically joining and leaving, the charging of nodes is not trivial.

Another main concern in ad-hoc networks is the requirement of mobile nodes cooperation in the traffic forwarding. Due to the bandwidth, battery and computation power requirements, the forwarding nodes may behave selfishly, not making available its resources to forward the traffic. Therefore, beyond charging, rewarding mechanisms need also to be in place to reward the forwarding nodes for their resources availability. However, if the charging process is a challenging task, rewarding is much more complex, since the overall path of the packets needs to be known in the infrastructure network. Also, security mechanisms need to be in place to make sure that the information of the nodes in the path is not modified in transit.

The creation of efficient charging and rewarding mechanisms needs to take into account not only its efficiency in the charging and rewarding procedure, but also the impact of these additional mechanisms operating in the network. Some existent proposals simply reduce the total throughput, increase the delay or jitter due to active marking of packets or by forcing routes to cross a point of profile enforcement. Other proposals may increase the end-to-end delay because of the required processing time at each node. Additional processing will deplete batteries of portable devices and consume the already scarce, processing power of mobile devices. If the impact of the charging protocol becomes too high, nodes will tend to act maliciously simply forwarding traffic without performing the mechanisms specified by the operating charging protocol. Secure devices like SIM cards present a solution to secure the charging algorithms. However, they present limitation in the amount, complexity and performance of the algorithms implemented in the secure device.

In the literature, several proposals exist for charging in ad-hoc networks without relying on tamper proof hardware. Some of them assume the existence of an entity in the infrastructure network that collects the proofs of each flow or packet, while others consider micro-payments where nodes can pre-buy some credits in a “bank” and use tokens directly to pay for the traffic they produce.

In the second group, the proposals focus on creating credit units called nuglets [6], [7] that nodes can buy from a bank and later exchange for services. Nodes can store each others nuglets and use them as well. As an example, a forwarding service will require some credit units which must be released by the sending node and distributed to the forwarding nodes.

Considering the first group of proposals, which is the one that will be dealt in this paper, we highlight the following approaches:[1], in which all traffic needs to cross the access point to provide correct charging and rewarding of the packets; [2] that optimizes the procedure by having only the last forwarding node collecting the information; [5] which enhances [2] by reducing the network overhead using end-to-end sessions; and [4] which proposes a novel strategy to reduce overhead and processing requirements at the nodes by using an algebraic approach. The next following sections briefly detail each of the proposals addressed and compared in this paper.

III. SALEM - Charging and rewarding scheme for packet forwarding in multi-hop cellular networks

This solution was first proposed in [1] as a protocol capable of charging and rewarding traffic in cellular ad-hoc networks connected to a commercial hotspot. According to the proposal, traffic is always obliged to cross the Access Point (AP) no matter the location of the source and destination nodes. The AP, being trusted both by nodes and network provider, is capable of enforcing individual user profiles and account for all traffic produced.

According to [1], when a node wants to send a packet, first it must create a session to the AP (also gateway) by means of a Session Setup Request message (Fig. 2). The gateway then creates a session with the destination point or signals another AP to start the session, in the case the destination node is outside the local ad-hoc network. If the destination node accepts the session, it replies with a Session Setup Response to the AP. The AP them starts the charging procedure and sends a Session Setup Configuration (SSCONF) to both the sending and destination nodes. During this process, if any forwarding node does not want to participate in the session, it will issue an error message. We should notice that a session is really composed by two sub-sessions: between source node and AP, and between AP and destination node. After the sending node

receives the SSCONF, it starts sending packets to the AP. These packets have a header indicating the Session Identifier (SID) negotiated during the setup phase and both forwarding nodes and the AP process packets based on this ID.

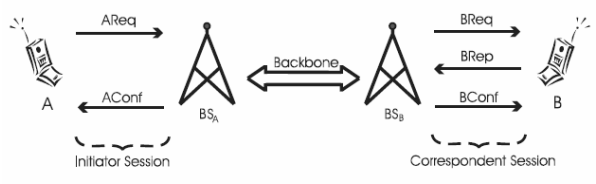

Fig. 2 - Session setup procedure in SALEM

Since the list of forwarding nodes is established during the setup phase, if the route changes, a new sub-session must be created, sharing information with the previous session. This session re-created is the one between the source node and the AP, or the AP and the destination node, depending in which segment the route changed. To trigger the setup of a new sub-session, nodes monitor network changes and notify the sending node (or AP) if they detect a change.

All traffic is secured by a stream cipher using a key which is based on the current SID and a key unique to the sending node.

Another aspect of this proposal is the support for destination acknowledgment and the possibility to reward nodes forwarding packets in both sub-sessions.

This solution can be very efficient when used in scenarios without traffic internal to the ad-hoc like a commercial hotspot. The drawbacks are the sub-optimal routing and the fact that the throughput of all nodes is limited by the throughput of the gateway.

IV. SCP – Secure Charging Protocol

The Secure Charging Protocol (SCP) was proposed in [2] and solves some issues of charging in ad-hoc networks using a distributed, yet secure algorithm. This proposal allows for traffic to be routed directed from the sending node to the destination node using the best route provided by the routing protocol. A charging header is added to every packet with information to securely charge the sending node and, using the Source Routing (SR) header, rewarding the forwarding nodes.

In this proposal, each node needs to include its IP address in the charging header, in the SR header, in the case of SR protocols, or in a different field in the packet header if non-SR protocols are used. With this information, each node in the path is able to recognize the route of the packet and the forwarding nodes, the ones that need to be rewarded.

The protocol makes use of hash chains to secure the path in the charging header and asymmetric Elliptic Curve Cryptography to secure protocol messages. Nodes are expected to have certificates which need to be exchanged in order to verify the MAC included in the charging header (Fig. 3).

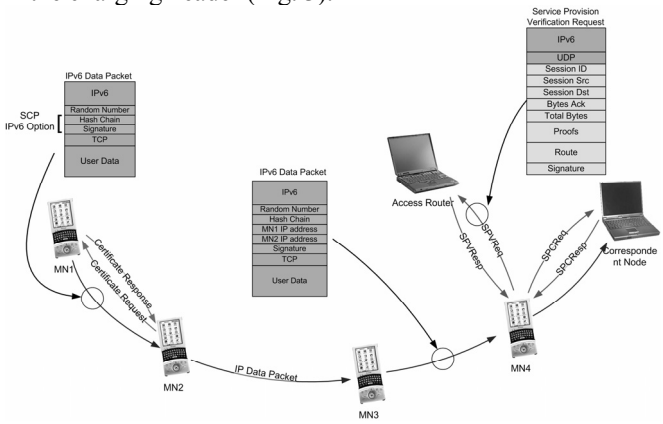

Fig. 3 - Secure Charging Protocol phases and related messages

The last forwarding node is expected to collect the charging headers in the packets and verify the reception with the destination node using a ServiceProvisionConfirmationRequest message.

After the receiving node acknowledges the batch of packets (or rejects) in a ServiceProvisionConfirmationResponse, the result of this step together with the path and hash chains is sent to the charging entity in the gateway of the ad-hoc network using a ServiceProvisionVerificationRequest message. The AAAC verifies the integrity of the reported information and the identity of the forwarding nodes, and issues a ServiceProvisionVerificationResponse with a status code. This status can notify the user to block further flows from that user or to continue the charging process.

In a more recent publication [3], a novel mechanism to build the route is described enabling SCP to operate in networks using other routing protocols like AODV [8] or OLSR [10]. Also, this mechanism reduces the network overhead resulting of the charging headers and increase robustness to mobility and new nodes arriving at the MANET. The authors of SCP in [3] also add the support for QoS based charging and rewarding.

Notice that although packets in SCP are authenticated, data is not secure from inspection from a malicious forwarding, or just listening, node. Also, no mechanism is described supporting enforcement of user profile parameters like access restrictions in traffic between nodes in the same ad-hoc network.

Traffic crossing the gateway is always admitted and shaped according to the sender's user profile.

V. SACP – Session Aware Charging Protocol

SACP [5] is based on the ideas proposed by SCP relating to marking of packets and report of proofs. As in SCP, all packets have a charging header and the last forwarding node is responsible for reporting the proofs to the network operator. As opposed to SCP, proofs are not acknowledged by the receiving node and only some of the packets need to include its route.

SACP proposes enhancements to previous proposals by describing a model in which packets are authorized even without forcing sub-optimal routes. In order to minimize network overhead of the marking process, especially due to SR headers, SACP makes use of two distinct phases for each flow. A setup phase and a forwarding phase.

During the setup phase the header has a complete SR route and a hash chain to securely identify nodes forwarding the flow. After, only the hash chain is updated at each node and the route omitted from the header.

When a node generates a packet, it checks if it has already information of this flow and its current destination, if it is in a setup phased or forwarding phase. This information is denoted by session. The information about this session is stored in the node. If this session does not exist, it creates a new one (storing its information in the node). Then, it activates a timer that, upon expiration, triggers the event of adding the route information in the outgoing packets at specified intervals. This route information is included in RouteUpdate messages. Moreover, the node sets the number of packets that may be sent before the requirement of a new route update message to a predefined value.

When a route changes, the node detecting the change informs the sending node to restart the session. An amount of packets is still forwarded even if the node is not included in the session. This will result in a small error in the rewarding process, which is shown in [5] to be much reduced. After a determined number of packets free forwarded, nodes may be unwilling to forward any more packet from that flow without being rewarded.

In scenarios where non-SR protocols are in use, this proposal proves to be very efficient in reducing the overhead and increasing the performance [5].

VI. PACP – Polynomial-assisted Ad-hoc Charging Protocol

PACP [4] is also based on the ideas proposed by SCP of marking and reporting of packets. All packets have a charging header and the last forwarding node is responsible for collecting and reporting the proofs (Fig. 4). Hash chains are used to secure the route information but, unlike previous proposals, PACP does not use a SR mechanism.

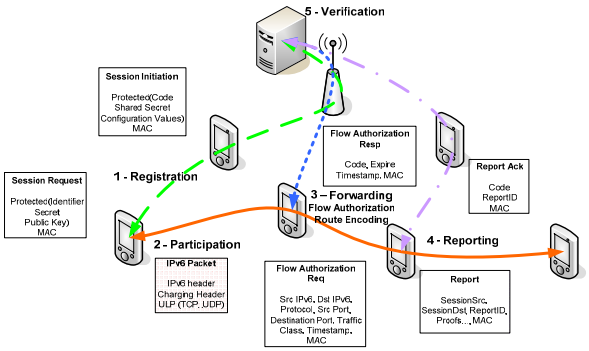

Fig. 4 - PACP phases and exchanged messages

As already referred, when non-SR routing protocols are used with SCP, the identification of each node is added interactively during the forwarding process. With this process, the packet size increases along the path at a rate of 16 bytes per hop. This brings some issues with upper layer protocols like TCP where the checksum must be recalculated. Also, there is a requirement to fragment packets if they exceed the MTU. Also, if some header above IP maintains some information regarding the content or size of the packet, that protocol will only interoperate with SCP if all nodes are able to interpret and adapt this protocol.

PACP solves these problems by proposing the usage of a fixed size header capable of identifying forwarding nodes and still charge communication end-points. This is accomplished by encoding the path using a polynomial approach, in a packet header field always with fixed size.

The idea behind the polynomial encoding is that for any polynomial $f(x)$ of degree d in the prime field $GF(p)$, with p being the smallest prime greater than 2^d-1, it is possible to recover $f(x)$, given $f(x)$ evaluated at $d+1$ unique points. If IP_i represents the IP address of the i forwarding node, and x_i an unique packet identifier in the route R with n nodes, the AAAC Server can recover all the IP addresses if it receives $Fr(x)=IP_1x^{n-1}+IP_2x^{n-2}+\ldots+IP_{n-1}x+IP_n$. Recovering this information is done using inversion of Vandermonde [11] matrixes under a prime field $GF(p)$ for a number of packets $d \geq n$. The complexity of the

problem is $O(n^2)$ and it can be solved in real-time for the average route length of ad-hoc networks. The recovering process through the Vandermonde matrix is depicted in the following expression.

$$\begin{bmatrix} x_1^{n-1} & \cdots & x_1^3 & x_1^2 & x_1 & 1 \\ x_2^{n-1} & \cdots & x_2^3 & x_2^2 & x_2 & 1 \\ x_3^{n-1} & \cdots & x_3^3 & x_3^2 & x_3 & 1 \\ x_4^{n-1} & \cdots & x_4^3 & x_4^2 & x_4 & 1 \\ \vdots & \iddots & \vdots & \vdots & \vdots & \vdots \\ x_n^{n-1} & \cdots & x_n^3 & x_n^2 & x_n & 1 \end{bmatrix} \begin{bmatrix} IP_1 \\ IP_2 \\ IP_3 \\ IP_4 \\ \vdots \\ IP_n \end{bmatrix} = \begin{bmatrix} Fr(x_1) \\ Fr(x_2) \\ Fr(x_3) \\ Fr(x_4) \\ \vdots \\ Fr(x_n) \end{bmatrix}$$

Solved in GF(p) (1)

Following the proposed approach, the sending node sets the RouteID field to 0 and Xi to a random value. Each forwarding node n computes RouteID(Xi)=[(RouteID(Xi)*Xi + IPn) mod p] and updates the RouteID field in the packet with the new value. This should be performed in chunks of 8 bits and using a p value of 257 (smaller prime larger than 28). The RouteID will be constructed iteratively in the forwarding nodes and the AAAC server will be then able to reconstruct the path.

The drawback of this solution is that the number of packets in each flow (n) needs to be at least equal to the number of hops traversed (d), in order to recover the overall path. The results in [4] however show that this error to be less than 0.6%, even for situations of very high mobility (170Km/h). Notice that the charging procedure does not incur any errors.

VII. Simulation Results

In this section we compare the performance of the presented charging and rewarding mechanisms. This comparison will be performed through ns-2 [12]. Due to the requirement of the Source Routing protocol in the SALEM proposal, we decided to perform the comparison using Dynamic Source Routing (DSR) [9] as the routing protocol.

In order to perform this comparison, we had to implement all the solutions presented. These implementations are according to the latest public document available for each solution and should reflect the state of the art.

A. Simulation Scenarios

In order to evaluate the behaviour of the proposals in different situations, two simulation scenarios were created. Both of them are composed by 30 ad-hoc nodes, where one of the nodes is considered to be the gateway.

One of the scenarios simulates a pure ad-hoc network where a central authority belongs to a trusted operator.

It simulates an area of 1Km square where the movement of the nodes is determined by a random waypoint model (RWP) and the speed varies according to a uniform distribution in the interval between 0 and 4 Km/h, to represent a general hotspot scenario as deployed in some commercial areas or airports. In this scenario, nodes only communicate with each other and no traffic goes to the gateway. In this case, the gateway only acts as a receiver for proofs and charging point.

The second scenario is similar to the previous one but traffic can be direct between nodes and also to a network outside the MANET. The number of flows to/from the outside is the triple as the number of direct flows between nodes. Flows simulate FTP (File Transfer Protocol) applications using TCP and the number varies between 5, 10, 15, 20 and 25 simultaneous flows. Simulations were divided in two phases: one phase from 0s to 500s and a second phase from 500s to 600s. During the first phase there is the coexistence of TCP flows and the charging protocol. After the 500s, TCP flows stop and only the charging protocols are in operation. These intervals were chosen both to evaluate the efficiency the impact of the charging protocol in the data flows and to evaluate the reporting procedure in SCP, SACP and PACP. The solution proposed in [1] does not requires this additional time since charging is performed directly by forcing traffic to cross the Access Point.

B. Network Throughput

We defined the impact on the network throughput as the performance penalty of a network running a charging protocol in comparison to a network without any other mechanism besides the Routing Protocol and the FTP applications.

Fig. 5 represents the results obtained by averaging several runs for different number of TCP flows in a pure ad-hoc network, where nodes directly communicate between each other. All proposals achieve similar results with the exception of SALEM. This is due to the suboptimal routing and the session establishment procedure it requires.

The results obtained for this scenario are very different from the ones obtained in [5] and [4], where a Source routing protocol was not used. In this scenario, SCP, SACP and PACP achieve similar throughputs. Notice that SCP greatly benefits from the DSR protocol since the route is already included in the routing protocol itself, and each node does not need to add its IPv6 address to the packet. When non-SR

protocols are used, the throughput of SCP much decreases in comparison with PACP and SACP.

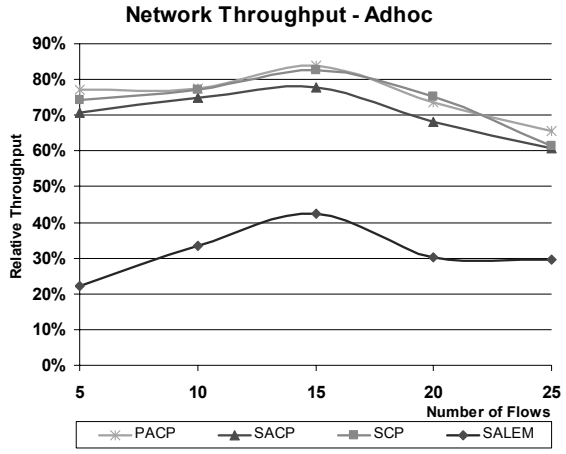

Fig. 5 - Relative Throughput of different proposals in a pure Ad-hoc scenario

In Fig. 6 results consider the hotspot scenario where some traffic is directed to and from the access router. Therefore, they are more favourable to the SALEM proposal. However, the network throughput penalty is usually the double when compared to other proposals. This fact is explained by the process of end-to-end sessions establishment and the required restart of the session in SALEM. A restart suddenly pauses the TCP flow which results in TCP slowing the throughput. SACP, although being a session-aware protocol, is not affected by this problem because nodes allow the forwarding of a small amount of packets before the flow is blocked. Also, one third of the flows have both end points inside the ad-hoc network, and this will heavily affect the increase in the hop count in SALEM. The hop count increase also increases the delay of the packets.

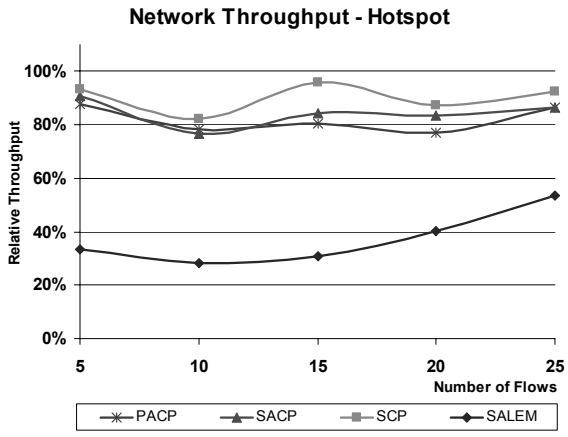

Fig. 6 - Relative throughput of different proposals in a hotspot scenario

C. Charging Rate

To compare the efficiency of the charging process, we defined a metric denoted as charging rate. It is defined as the percentage of proofs received at the AAAC server from the total of proofs collected, by the forwarding nodes:

$$\text{ChargingRate} = \frac{\text{NumberProofsReturned}}{\text{TotalNumberProofs}} \quad (2)$$

Since in the simulations the nodes stop moving after the traffic flows stop, there is some probability of having nodes with stored proofs that do not have a route to the AR, and then, to the A4C. Therefore, a low charging rate means that proofs were collected, but it was impossible to report those proofs in the specified time. This does not apply to SALEM proposal where all packets are directed to the AP and thus charged. A charging rate of 100% is then guaranteed.

In the ad-hoc scenario (Fig. 7), SCP, SACP and PACP achieve similar charging rates. As the number of flows increase, all these solutions decrease the efficiency of the charging procedure. With a higher number of flows, more packets are produced and more proofs need to be reported. The radio link near the AP can become saturated with the flood of reports resulting in loss of report packets. However, mechanisms are proposed to provide the retransmission of lost proofs. The minimum value registered is close to 95% which still represents an acceptable charging rate. Also, notice that these values reach 100% with increased simulation time.

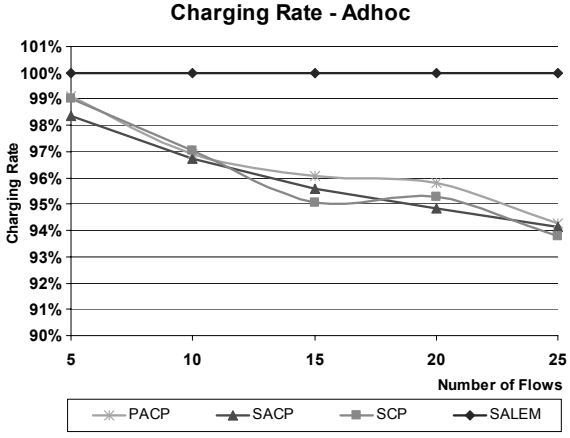

Fig. 7 - Charging rate of the different charging proposals in an ad-hoc scenario

In the hotspot scenario (Fig. 8), the SALEM proposal also achieves a charging rate of 100% while other proposals achieve very similar results around 99%. In SACP, SCP and PACP, the gateway directly

charges for all outgoing traffic which explains the high efficiency of theses solutions in this scenario. Only traffic direct between nodes and the packets entering the hotspot need to be stored by the last forwarding node and then reported back to the gateway.

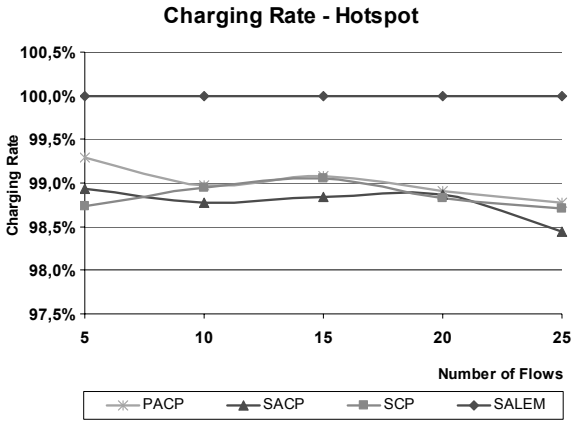

Fig. 8 - Charging rate of the different charging proposals in a hotspot scenario

VIII. CONCLUSIONS

This paper presented a comparison between charging and rewarding mechanisms in ad-hoc networks.

The SALEM proposal provides the most exact mechanism to charge and reward network traffic in ad-hoc networks in the same time the traffic is traversing the network. However, the impact on throughput due to sub-optimal routing and session maintenance is very high. Such high performance impact on the network should be reduced; otherwise, the benefits of ad-hoc networks will no longer exist. The other studied solutions still present acceptable charging rates in real-time and a very large network throughput.

Notice that, in the studies performed in this paper, the performance of SACP and PACP in scenarios using DSR is smaller from the results obtained in [5] [4], when compared to SCP. This is due to the fact that, in scenarios where Source Routing mechanisms are natively used, the improvements of the SACP and PACP solutions are less noticeable.

From the study conducted in this paper, we conclude that choosing a charging protocol must be accomplished by a first analysis of the scenarios envisioned and of the requirements in performance and in efficiency of the real-time charging procedure.

IX. REFERENCES

[1] Naouel Ben Salem, Levente Buttyán , Jean-Pierre Hubaux , Markus Jakobsson, "A charging and rewarding scheme for packet forwarding in multi-hop cellular networks", Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, June 01-03, 2003, Annapolis, Maryland, USA.

[2] B. Lamparter, K. Paul, and D. Westhoff. "Charging Support for Ad Hoc Stub Networks". Elsevier Journal of Computer Communication, Special Issue on Internet Pricing and Charging: Algorithms, Technology and Applications, 2003.

[3] João Girão, João P Barraca, Bernd Lamparter, Dirk Westhoff & Rui Aguiar "QoS-differentiated Secure Charging in Ad-hoc environments", International Conference on Telecommunications 2004 (ICT2004).

[4] João Paulo Barraca, Susana Sargento & Rui Aguiar, "Polynomial-assisted Ad-hoc Charging Protocol", IEEE Symposium on Computers and Communications (ISCC 2005).

[5] João Paulo Barraca, Susana Sargento & Rui Aguiar, "A Lightweight and Secure Session-Aware Ad-Hoc Charging Protocol", International Conference on Telecommunications 2005 (ICT2005).

[6] L. Butty´an and J.-P. Hubaux. "Enforcing Service Availability in Mobile Ad Hoc WANs". In Proceedings of MobiHocC, Boston, MA, USA, August 2000.

[7] L. Butty´an and J.-P. Hubaux. "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks". ACM/Kluwer Mobile Networks and Applications (MONET), 8(5), October 2003.

[8] RFC 3561, "Ad hoc On-Demand Distance Vector (AODV) Routing".

[9] draft-ietf-manet-dsr-10.txt, „The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", IETF draft

[10]: RFC 3626 "Optimized Link State Routing Protocol (OLSR)".

[11]: W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T.Vetterling. "Numerical Recipes in FORTRAN: The Art of Scientific Computing". Cambridge University Press, 1992.

[12] The Network Simulator NS2, http://www.isi.edu/nsnam/ns, as in November 2004.