Liveliness Evaluation of a Cooperation and Accounting Strategy in Hybrid Networks

Attila Weyland, Thomas Staub and Torsten Braun

Institute of Computer Science and Applied Mathematics
University of Bern
Neubrückstrasse 10, 3012 Bern, Switzerland
Email: {weyland|staub|braun}@iam.unibe.ch

***Abstract*— We propose a cooperation and accounting scheme for multi-hop cellular networks, which stimulates cooperation among nodes by making it a rewarding alternative to selfishness. The paper describes an architecture with highly decentralized security and accounting mechanisms. Our scheme charges senders and rewards forwarders, supports both sender- and receiver-based payments and coexists with ad hoc only traffic. We use of service stations deployed throughout the network to offer nodes a possibility to refill their money accounts. We present the results of simulation runs, where we investigated the liveliness of the proposed scheme. We find that the number of service stations and their distribution correlate in different ways.**

I. Introduction

Multi-hop cellular networks (also called hybrid networks) increasingly attract interest in the research community. They appear to be a promising combination of the advantages of two worlds: the dynamics of mobile ad hoc networks and the reliability of cellular networks. The motivation to overcome the single-hop limit between mobile and access point in (infrastructured) wireless networks and use multiple hops instead, comes from the gained ability to dynamically adapt the network topology to the respective needs. A better and larger coverage area and reduced installation costs are advantages for a wireless network provider in that case. Nodes can reduce their energy consumption for transmitting packets due to shorter next-hop distances. In the context of hybrid networks new possibilities to deal with the weaknesses of mobile ad hoc networks become available. We think that besides the security and routing issues the cooperation among nodes is of great importance. We propose a cooperation and accounting scheme for hybrid networks called CASHnet, which takes into account the availability of a reliable network infrastructure and stimulates cooperation by making it a rewarding alternative to selfishness.

The application scenarios of mobile ad hoc networks have become broader over time. Starting with nodes acting on behalf of a single authority in emergency situations, today we see civilian (and commercial) scenarios where each node is its own authority. When we attribute individuality to each node of a mobile ad hoc network, we have to deal with a node's individualism, especially in the form of selfishness. A node expresses its selfishness in the refusal of cooperation, i.e. an individual node might refuse forwarding packets from other nodes in order to save energy for transmitting its own packets. This behavior leads of course to a malfunctioning mobile ad hoc network.

II. Related Work

The first generation of proposed cooperation schemes for mobile ad hoc networks apply rather restrictive control mechanisms. Besides technical issues, restriction neglects the node's freedom of decision (to participate or to not participate) to a certain extent. In civilian scenarios it seems more suitable to achieve cooperation among nodes by means of rewards instead of penalization. Moreover, by using the available infrastructure in a multi-hop cellular network, it becomes possible to offer an architecture which corresponds with this notion. So far most of the cooperation schemes target mobile ad hoc networks only. Recently, few approaches have been taken to address the problem of cooperation among nodes in the context of multi-hop cellular networks.

Several proposals have been made to stimulate cooperation among nodes. The first approaches were aimed at mobile ad hoc networks and enforced cooperation by threat of punishment. In the Nuglet [1] scheme a node can only transmit self-generated packets when it has forwarded enough packets from its neighbors before. In the CONFIDANT [2] approach the behavior of a node is monitored by its neighbors and a selfish node will be isolated from the network. In both concepts a node can be excluded from participating in the network without itself being at fault (starvation or collective false accusation).

With the Sprite [3] scheme rewards have been introduced as incentive for cooperation in mobile ad hoc networks. Nodes report their forwarding activities to a central authority reachable via an overlay network. In conjunction with the missing security mechanisms this scheme seems highly vulnerable to attacks and transmission errors. In [4] the authors suggest the usage of rewards in multi-hop cellular networks and let a central authority collect and analyze reports to decide about rewards and punishments. However, the authors assume a single-hop down-link (from the base station to the node), which might not be available easily.

The authors of [5] and [6] propose similar charging schemes, where cooperative nodes get rewarded in a multi-

hop cellular network environment. They both heavily rely on centralized accounting and security mechanisms. To remunerate intermediate forwarding nodes, both schemes require the complete route information from the sender to the receiver (e.g. using source routing). However, source routing does not scale well under high node mobility. Also, both schemes do not support cost sharing between sender and receiver, when both of them reside in different ad hoc networks. The sender also has to pay for the distance from the gateway to the destination. To better cope with misuse the authors of [5] require all the network traffic to go via the operator's access points, which leads to inefficient routes for traffic within the same ad hoc network. [6] requires an existing AAA infrastructure, which might not be available for all multi-hop cellular network scenarios. In a recent proposal [7], the authors extended their work from [5]. They introduced a local Nuglet counter for each node to address the issues of inefficient routes in pure mobile ad hoc networks and a central auditing entity [4] to better cope with abuse. The weaknesses of the Nuglet scheme, such as the unresolvable starvation of selfish nodes due to a single counter and the unsuitability for civilian (commercial) applications because of neglecting the node's freedom of choice (to cooperate or to not cooperate) remain as well as the single-hop down-link.

With our scheme we provide decentralized accounting and security mechanisms to the largest extent possible in a multi-hop cellular network environment. We support initiator- and receiver-based payments and we do not require full route information from the sender to the receiver. Also, our approach coexists with ad hoc only traffic in the sense that nodes get neither charged nor remunerated for this kind of traffic. We think that ad hoc only communication should be free since the provider has no cost in terms of network traffic. Although he provides the security mechanisms via the smart cards, ad hoc only communication would still be possible without him. The charge for the security service could be based on a subscription fee. Because we target multi-hop cellular networks in civilian use, where each node can be seen as its own authority, we leave the choice of cooperation to the node. But by providing monetary rewards we make cooperation among nodes a gainful alternative to selfishness.

III. CASHnet Architecture and Operation

In our scheme we assume - similar to the Nuglet [1] approach - the existence of a tamper resistant device, such as a smart card in each node. This device ensures a protected environment, where the functions of our schemes can be executed safely. Also, we assume the availability of a routing algorithm, which provides the hop count to the base station (e.g. AODV or DSR). Additionally, we require sufficient processing power and memory on the node. For our scheme we define an architecture as displayed in Fig. 1.

The CASHnet charging and rewarding mechanism works as follows: Every time a node wants to transmit a self-generated packet (i.e. node O), it has to pay with *Traffic Credits*. Every time a node forwards a packet (i.e. nodes N_{A1} - N_{A3} and N_{B1}), it gets *Helper Credits*. Traffic Credits can be bought for real money or traded for Helper Credits at service stations. Gateways provide the interconnection between the fixed networks and the mobile ad hoc networks.

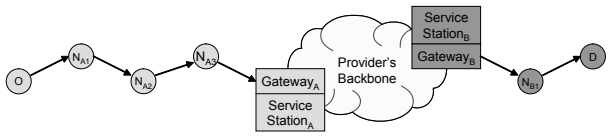

Fig. 1. Example Scenario

Our security mechanisms are based on public key cryptography. Nodes authenticate themselves using certificates issued by the provider. To avoid the creation of bogus nodes, we give a short lifetime to the certificates forcing the node owner to regularly visit a provider's service station. Transmitted messages are digitally signed to provide non-repudiation (data integrity and data origin authentication).

The operation of CASHnet is described in the following paragraphs. Fig. 1 shows an example scenario to which all the defined steps can be applied. The following notation is used: each paragraph describes a coherent phase of the operation process. A phase consists of several enumerated actions, which are executed consecutively. The processing of a phase can be terminated by a reference to another phase "⇒" or by a termination command "□". Numbered list entries in the form of questions indicated a forking of the processing path. Either the "[Yes]" or the "[No]" path is executed. The nested numbered elements of the chosen path are again executed consecutively.

a) Setup Phase: Before a node N can participate in the hybrid network belonging to operator P, node N has to perform the following steps:

1) Obtain a personal smart card from provider P which contains node N's unique identifier, node N's public/private key pair K_N/KP_N, a certificate $Cert_P(ID_N, K_N)$ issued by the provider, as well as the provider's public key K_P (one-time action).
2) Update node N's certificate $Cert_P(ID_N, K_N)$ (as necessary).
3) Load the Traffic Credits account at the provider's service station by paying with real money and/or by transferring Helper Credits (as necessary).

b) Initial Authentication Phase: Before a node can engage in the communication as a packet originator O in the hybrid network, it has to initially authenticate itself once to all nodes participating in its communication (intermediate nodes N and destination node D). This is done by sending an AUTH Request message to the destination. This message contains O's identifier ID_O, its public key K_O and the certificate $Cert_P(ID_N, K_N)$. Each node N along the path verifies the certificate $Cert_P(ID_N, K_N)$ and - if valid - saves O's identity ID_O and public key K_O as a pair in an AUTH list. After the successful validation of an AUTH Request message, the destination sends back an AUTH Reply message to the

originating node O. When node O receives the AUTH reply message, it knows that a path with cooperative node exists and can start with the transmission of self-generated data packets.

Also, every intermediate node N participating in the communication needs to authenticate itself to the previous and the next node along the path. To reduce the delay caused by unauthenticated nodes on a forwarding path, each node in the hybrid network authenticates itself to all its one-hop neighbors. The identity and public key pairs of successfully authenticated neighboring nodes are also stored in the AUTH list.

If a route changes and a new node joins the path, it is already authenticated to its one-hop neighbors due to the periodic neighboring authentication, yet the new node has to authenticate the originator of the packet, which might cause a small delay.

c) Packet Generation Phase: When a node O wants to transmit a self-generated data packet to the destination D, node O performs the following steps:

1) Is the packet going to leave node O's ad hoc network via the gateway?
   - No. a) The packet classifies as ad hoc only traffic and therefore O does not get charged.
     b) Form a signed packet $Packet_O$ and transmit it to the next cooperative hop. □
   - Yes. a) Determine the transmission cost of the packet. (The transmission costs are related to the distance in hop counts to the gateway of O's ad hoc network.)
     b) Does O's Traffic Credits account allow to pay for the transmission cost?
     - No. O can not transmit a self-generated packet at this time. □
     - Yes. i) Debit O's Traffic Credits account according to the transmission cost (sender-based payment).
       ii) Form a signed packet $Packet_O$ and transmit it to the next cooperative hop. □

$$Packet_O = ID_O|Payload|Timestamp_O| \\ Sig_O(Payload, Timestamp_O)$$

d) Packet Reception Phase: When a node N receives a data packet $Packet_{N-1}$, it performs the following steps:

1) Does the digital signature from the received data packet Sig_{N-1} as well as from the encapsulated original packet Sig_O verify correctly?
   - No. Discard the packet. □
   - Yes. Proceed to the next check.
2) Does the packet originate from outside node N's ad hoc network and has the destination D been reached (node N equal node D)?
   - No. Proceed to the next check.
   - Yes. a) Determine the reception cost of the packet. (The reception costs are related to the distance in hop counts to the gateway of D's ad hoc network.)
     b) Debit D's Traffic Credits account according to the reception cost (receiver-based payment).
     c) Pass packet to the non-secured part of node N. □
3) Does the packet originate from within node N's ad hoc network and is it not going to leave node N's ad hoc network via the gateway?
   - No. Proceed to the next check.
   - Yes. Proceed to the Packet Forwarding Phase (no accounting for ad hoc only traffic). ⇒
4) Does the packet originate from the previous node $N-1$?
   - No. a) Form a signed ACK message ACK_N and send it to node $N-1$.
     b) Discard the information from the previous node $N-1$ to retrieve the encapsulated original packet $Packet_O$.
     c) Proceed to the Packet Forwarding Phase. ⇒
   - Yes. Proceed to the Packet Forwarding Phase (no reward for the packet originator). ⇒

$$ACK_N = ID_N|Timestamp_N| \\ Sig_N(Sig_{N-1}, Timestamp_N)$$

e) Packet Forwarding Phase: When a node N transmits a forwarded data packet, it performs the following steps:

1) Form a signed packet $Packet_N$.
2) Look up the next hop in the routing table towards the destination D.
3) Save the next hop identity ID_{N+1} and the signature of the packet to be forwarded Sig_N as a pair in a list.
4) Transmit the packet $Packet_N$ to the next hop.

$$Packet_N = ID_N|Packet_O|Timestamp_N| \\ Sig_N(Packet_O, Timestamp_N)$$

f) Rewarding Phase: When a node N receives an ACK message from a successor node $N+1$, node N performs the following steps:

1) Does the digital signature from the received ACK message Sig_{N+1} verify correctly?
   - No. Discard the message. □
   - Yes. Proceed to the next check.
2) Do the contained digital signature of the acknowledged packet Sig_N and the successor node's identity ID_{N+1} have a matching pair in the list?
   - No. □
   - Yes. a) Credit N's Helper Credits account.
     b) Remove the matching pair from the list. □

IV. Liveliness Evaluation

In our first simulation runs of the CASHnet protocol we investigated the issue of starving nodes. In particular, we are interested in the case when a node is unable to transmit self-generated packets because it has not enough Traffic Credits (i.e. it has run out of Traffic Credits or the hop distance of the destination is to high).

A. Implementation Details

For the simulation we use ns-2 [8], where we implemented a simplified version of the CASHnet scheme including the

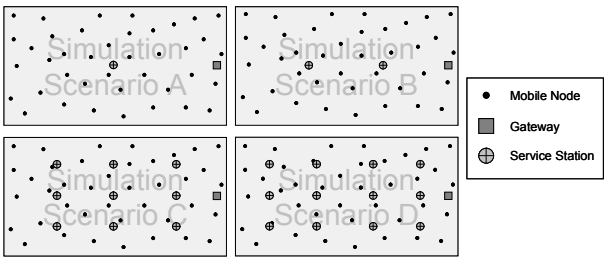

Fig. 2. Simulation Scenarios

charging and rewarding functionality without considering the security mechanisms. In particular, we used the wireless and mobility extensions [9] with an extended version of the AODV protocol called AODV+ [10], which adds Internet gateway discovery support.

We created a new ns node class called CashnetNode which inherits from the class MobileNode. The class CashnetNode contains the traffic and the helper credits accounts. Additionally, we implemented an agent at the source/sink, which is responsible for the rewarding procedure. It generates ACK messages and also evaluates them. The service stations are represented as fixed nodes derived from the class MobileNode. We keep a static list with the available service stations and their coordinates. The class MobileNode provides a method to calculate the distance to any other node in the network. Therefore, we are able to determine if a node is within range of a service station. To be able to follow the events from our CASHnet mechanisms we extended the class CMUTrace from the wireless and mobility extensions. Currently we are charging only for data traffic.

B. Simulation Scenario and Setup

Fig. 2 shows our simulation scenarios. We only consider one multi-hop cellular network in this simulation runs. The simulation area is 1500 x 800 meters. A gateway is placed at the border of the sector. 40 nodes are placed randomly inside the area and every node is communicating to the gateway. In the current simulation setup the movements of the nodes are pre-calculated using the random waypoint model with a minimum speed of 1 m/s and a maximum speed of 10 m/s. The pause time is uniformly distributed between 0 and 20 ms. The simulation runs for 900ms. The transmission range of a node is set to 250 meters. Table I shows the parameters which have to be specified for the CASHnet scheme. We give each node the same starting amount of 100 Traffic Credits, 500 Real Money units and no Helper Credits. The exchange rate for Traffic/Helper Credits is set to 1:1. When a threshold of 10 Helper Credits is reached, a node can exchange its Helper Credits. The distance required between a node and a service station to conduct an exchange of credits is set to 50 meters.

We vary the number and the distribution of deployed service stations (1, 2, 9 and 12 as shown in Fig. 2 as well as 9 and 12 service stations distributed randomly) as well as the packet interval at the CBR traffic sources (1 s, 5 s and 10 s). For each of the 18 simulation scenarios, 20 simulation runs have been conducted. We measured the frequency of occurrence of starving events and the duration of the starvation.

TABLE I

CASHNET PARAMETERS AND THEIR DEFAULT VALUES

Parameter	Value
Starting amount for Traffic Credits account [TC]	100
Starting amount for Real Money account [RM]	500
Starting amount for Helper Credits account [HC]	0
Traffic Credits/Helper Credits exchange rate	1:1
Exchange threshold at Service Stations [HC]	10
Distance threshold to Service Stations [m]	50

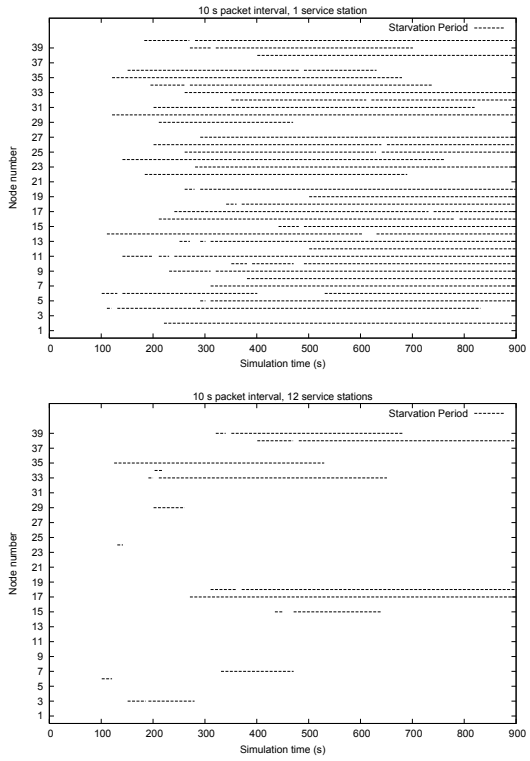

Fig. 3. Starvation periods for all nodes during a single simulation run for scenario A and D

C. Results

Figure 3 displays each node's starvation period(s) during a single simulation run for scenario A and D. Fig. 4-7 contain the results for the simulation scenarios A (1 service station), B (2 service stations), C (9 service stations) and D (12 service stations) for three packet intervals (10 s - top row, 5 s - middle row and 1 s bottom row). Fig. 8 and 9 contains the results for two scenarios with 9 and 12 randomly distributed service stations respectively.

Each histogram categorizes starvation events according to their duration. The boxes display the mean values of the number of starvations for the 20 simulation runs. The longer the starvation period (x axis), the smaller the number of starvations should be (y axis). A high amount of long starvations is naturally much worse than many small starvations.

From the figures several conclusions can be drawn. In Fig. 3

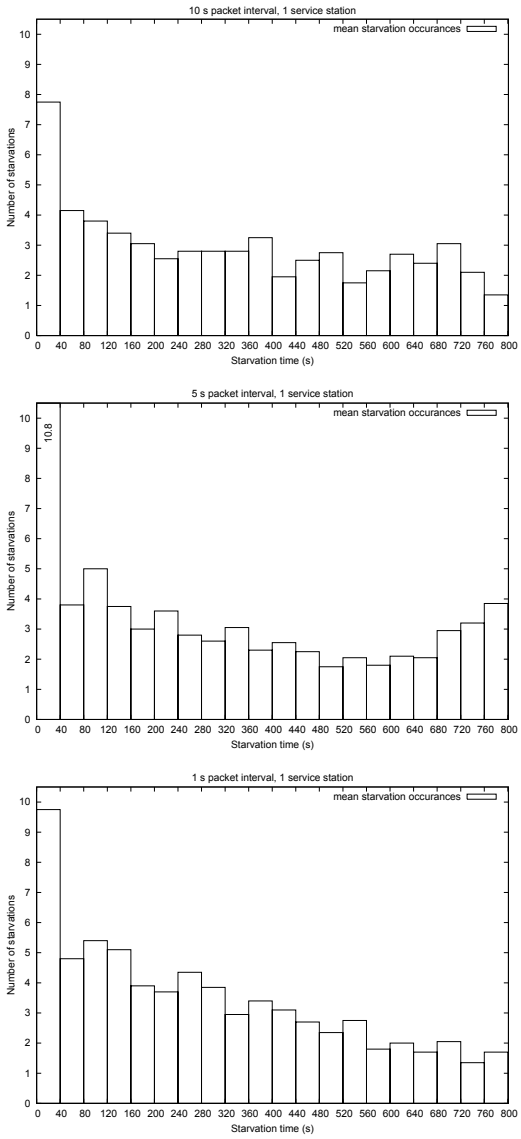

Fig. 4. Mean number of starvations per duration category for scenario A

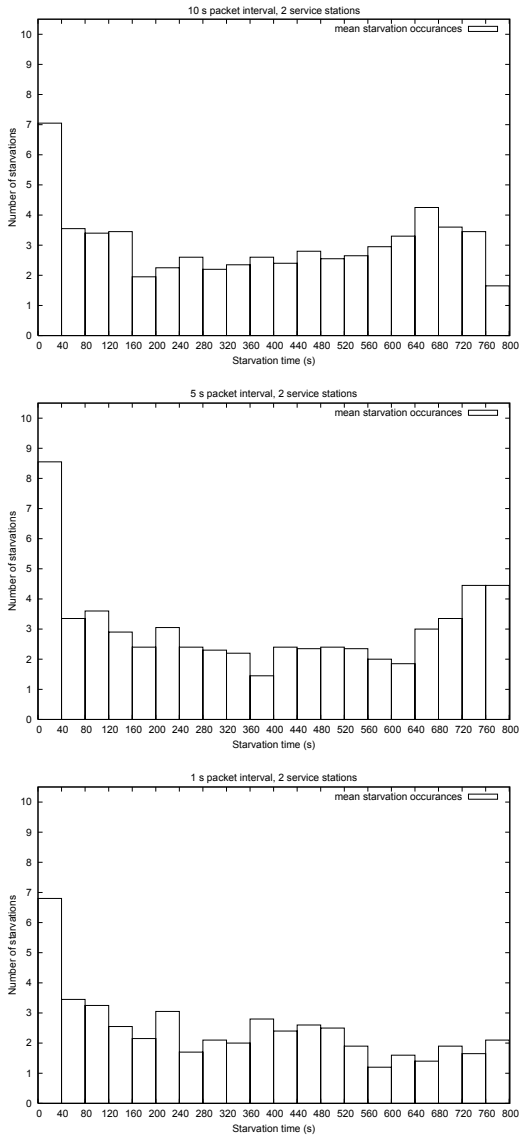

Fig. 5. Mean number of starvations per duration category for scenario B

the exhaustion of Traffic Credits can be seen when the first nodes start to starve at around 100 s. The difference between the top (1 service station) and the bottom (12 service stations) is very obvious. In the first, many nodes starve for the whole simulation period in the latter only few nodes starve and even fewer for a long time. The figures also show that it is difficult to achieve an equilibrium between generating and forwarding packets so that no starvation occurs. This emphasizes the need for separating the right for transmission from the amount of forwarded packets.

Increasing the number of service stations reduces the overall number of starvations and transforms long starvations into short ones (compare Fig. 4 with Fig. 7). This can be explained easily by the increased probability that a node can refill its Traffic Credits account by exchanging Helper Credits or paying with Real Money. When comparing the results of scenario A in Fig. 4 with those of scenario B in Fig. 5 we see a slight worsening of the result, although the number of service stations is higher. This could be caused by the nodes movement which have a higher probability of going through the center of the simulation area. Increasing the sending rate leads to an increasing number of starvations if the number of service stations is high (see Fig. 7). A high number of transmitted packets makes nodes run out of Traffic Credits faster. Thus the probability of a node not being able to transmit increases. Due to the high number of service stations, mostly the occurrences of short starvations increase. If the number of service stations is low, increasing the sending rate has a different effect (see Fig. 4). It mainly transforms the longer starvations into shorter ones. The increased number of transmitted packets leads to a higher amount of Helper Credits, which the nodes can exchange at the service station. Deploying services stations

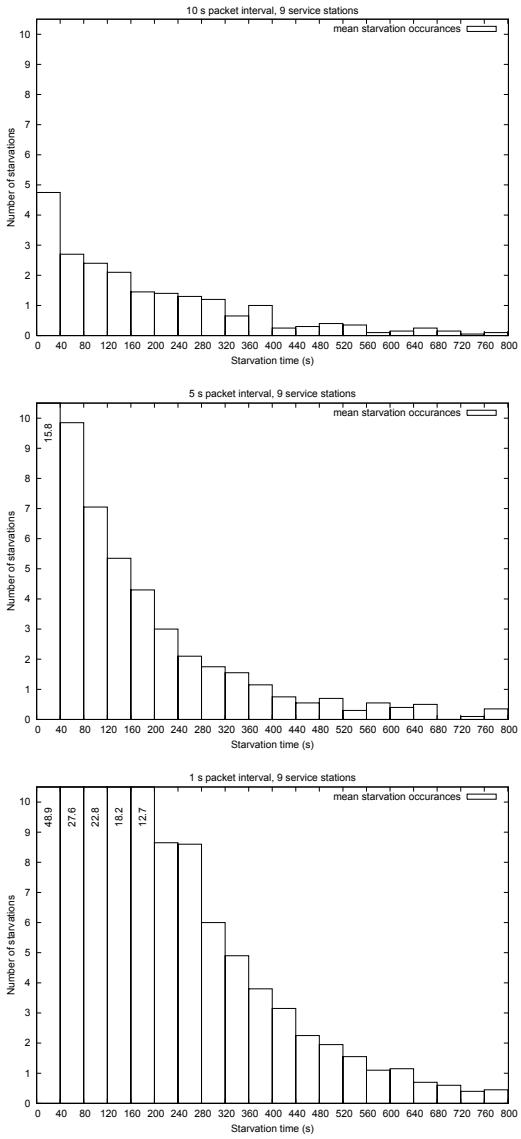

Fig. 6. Mean number of starvations per duration category for scenario C

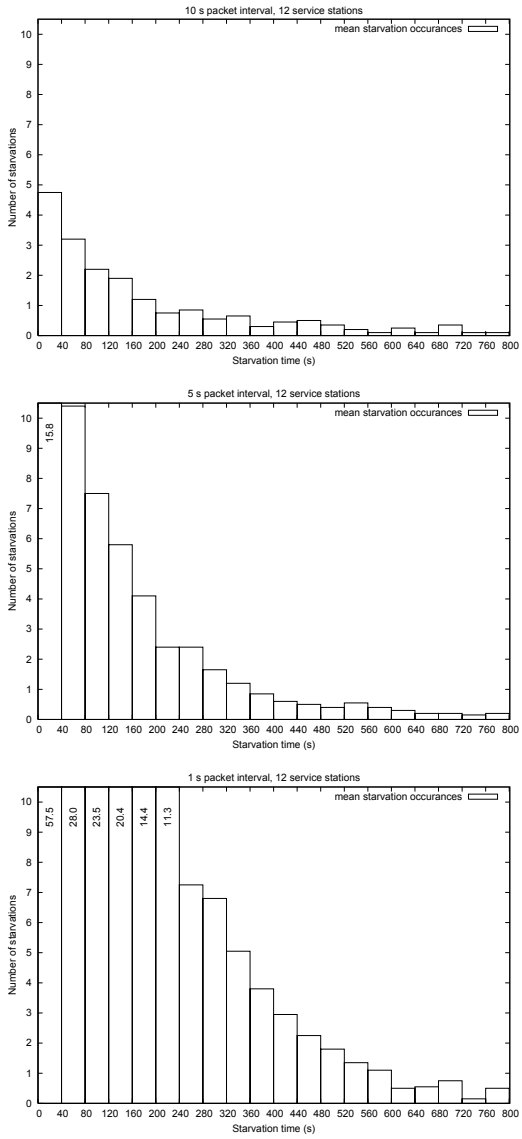

Fig. 7. Mean number of starvations per duration category for scenario D

randomly is not as effective as placing them in a aligned pattern (compare Fig. 7 with Fig. 9). Compared to the scenario D, the scenario with 12 randomly distributed service stations has a higher number of longer starvation periods. Thus an equal distribution of service stations increases their helpful effect on the overall network liveliness.

V. Summary and Outlook

We proposed a highly decentralized accounting and security architecture which provides a solid foundation for a cooperation scheme based on rewards and which is applicable to multi-hop cellular networks. In contrast to previous work we allow selfish nodes, but encourage them to participate in packet forwarding via rewards. Additionally, we allow initiator as well as receiver based payment which - to the best of our knowledge - is not possible in the available schemes. Last, we do not charge nor reward for traffic within the same multi-hop cellular network (ad hoc only traffic), while other schemes do not allow that. We validated our scheme with simulation runs and investigated the issue of starvation. Our simulation results show that it is difficult to achieve an equilibrium between generated and forwarded packets so that nodes do not starve. Our solution is to provide an additional mean of buying the right for transmission via the separation of accounts and the deployment of service stations. Future work will include the comparison with other incentive mechanisms, study of possible extensions (e.g. charging for ad hoc only traffic) as well as the optimization of the charging and remuneration relation.

References

[1] L. Buttyán and J.-P. Hubaux, “Stimulating cooperation in self-organizing mobile ad hoc networks,” *ACM Mobile Networks & Applications*, vol. 8, no. 5, Oct. 2003.

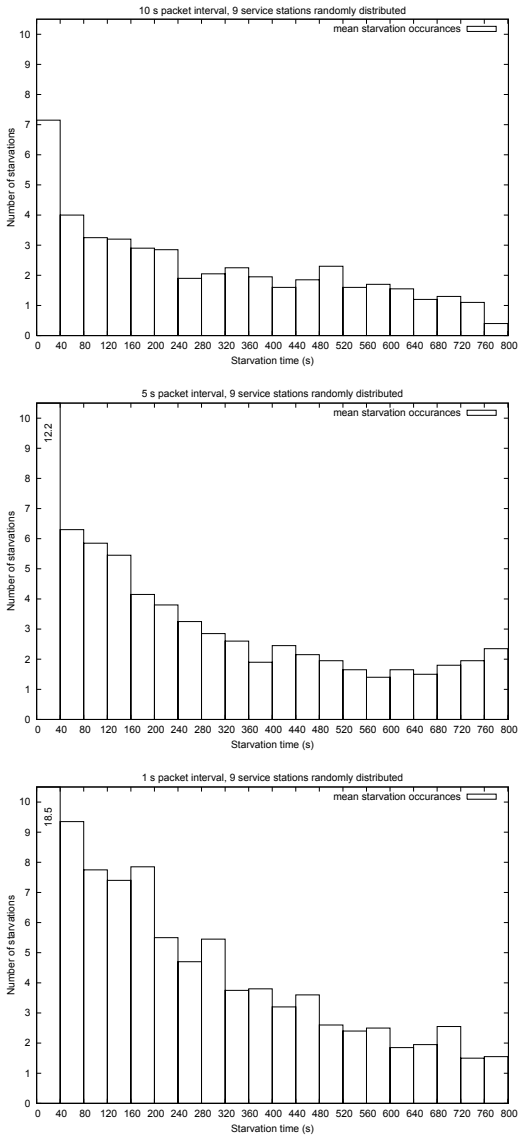

Fig. 8. Mean number of starvations per duration category for 9 randomly distributed service stations

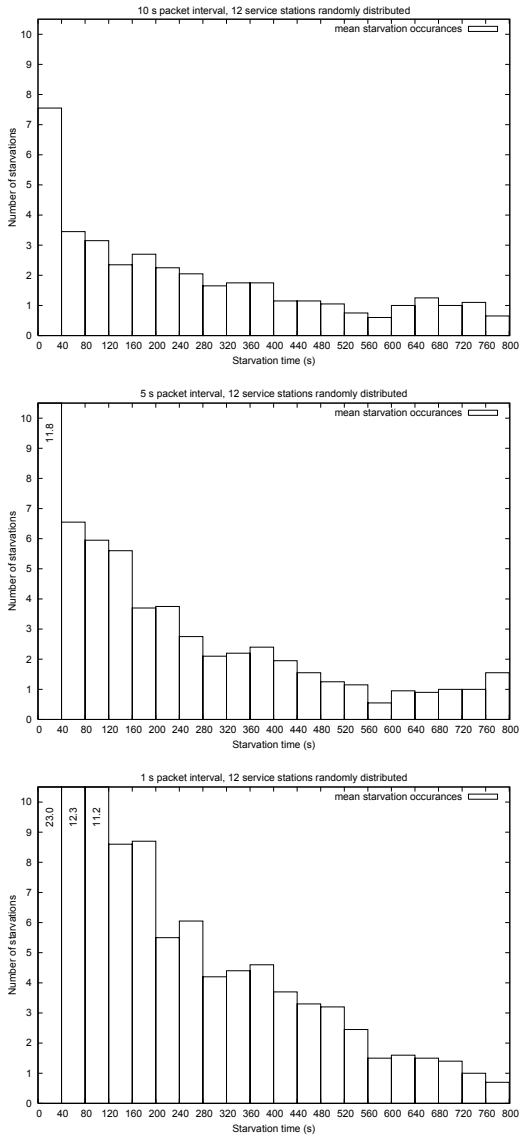

Fig. 9. Mean number of starvations per duration category for 12 randomly distributed service stations

[2] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the confidant protocol (cooperation of nodes - fairness in dynamic ad-hoc networks)," in *Proc. ACM Intl. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, Lausanne, Switzerland, June 2002.

[3] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Mar.-Apr. 2003.

[4] M. Jakobsson, J.-P. Hubaux, and L. Buttyán, "A micro-payment scheme encouraging collaboration in multi-hop cellular networks," in *Proc. Intl. Financial Cryptography Conf*, Gosier, Guadeloupe, Jan. 2003.

[5] N. B. Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson, "A charging and rewarding scheme for packet forwarding in multi-hop cellular networks," in *Proc. ACM Intl. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, Annapolis, MD, USA, June 2003.

[6] B. Lamparter, K. Paul, and D. Westhoff, "Charging support for ad hoc stub networks," *Elsevier J. Computer Communications*, vol. 26, no. 13, pp. 1504–1514, Aug. 2003.

[7] N. B. Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson, "Stimulating cooperation in ad hoc and multi-hop cellular networks," Poster Session of MICS Scientific Conference, Monte Verita, Switzerland, Oct. 2003.

[8] T. V. Project. (2004) The Network Simulator ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/

[9] Rice Monarch Project. (1999) Wireless and Mobility Extensions to ns-2. [Online]. Available: http://www.monarch.cs.cmu.edu/cmu-ns.html

[10] A. Hamidian. (2003) AODV+. [Online]. Available: http://www.telecom.lth.se/Personal/alexh/