Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 29 (2006) 2661–2670
computer communications
www.elsevier.com/locate/comcom

Comparison of motivation-based cooperation mechanisms for hybrid wireless networks

Attila Weyland *, Thomas Staub, Torsten Braun

University of Bern, Institute of Computer Science and Applied Mathematics, Neubrückstrasse 10, 3012 Bern, Switzerland

Received 9 January 2006; accepted 9 January 2006
Available online 10 March 2006

Abstract

Today's public Wireless LANs are restricted to hotspots. With the current technology, providers can only target a small audience and in turn charge high prices for their service to generate revenue. Also, providers cannot react appropriately to dynamic changes in the demand. With multi-hop cellular networks the coverage area can be increased and the installation costs as well as investment risks for the provider can be reduced. However, the individual customers play an important role in such networks and their participation must be ensured. Therefore, we propose a cooperation and accounting scheme which introduces monetary rewards. We compare our scheme called CASHnet with the Nuglet scheme using simulations under the criteria of network liveliness, goodput, overhead, and packet drop reasons as well as cash flow.

Keywords: Cooperation; Accounting; Incentives; Charging; Rewarding; Resale; Service station; Hybrid wireless; Multi-hop cellular; Mobile ad hoc networks

1. Introduction

The current wireless network installations consists of a number of access points deployed in selected areas, where they are expected to serve a minimum amount of customers to bring revenue to the provider, e.g., at airports or railway stations. Potential customers outside the area covered by the access points cannot be served. Besides the financial risk limiting the deployment of access points, location properties can also be restricting factors.

With multi-hop cellular networks, also called hybrid networks, the single-hop limit does not exist any more. The computing devices of the customers participate in the packet forwarding process (like in mobile ad hoc networks) and a gateway offers the connection to the Internet. This gives the provider a greater coverage area with more customers and reduces the network installation costs. Customers get connectivity outside hot spot areas. However, the advantages of mobile ad hoc networks come together with disadvantages such as maintaining accurate route information, protecting customers from attacks as well as the need to ensure cooperation among customers for keeping the network alive. If customers do not forward packets from other participants, the network falls apart.

Although individual customers may have a common interest in obtaining connectivity, they tend to prioritize their self-generated packets over packets to be forwarded from other customers when energy and bandwidth are regarded as precious, limited goods. Thus, cooperation among selfish individuals cannot be taken for granted, but must be ensured. In the literature, two approaches to cooperation exist: enforcement and encouragement. In the first, detection-based approach, the nodes in the network monitor each other and punish any detected uncooperativeness by (partially) excluding the respective nodes. In the latter, motivation-based approach, the transmission of self-generated packets is subject to charges and the forwarding of packets from other nodes is rewarded via the

* Corresponding author. Tel.: +41 31 631 86 48; fax: +41 31 631 32 61.
E-mail address: weyland@iam.unibe.ch (A. Weyland).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.01.011

distribution of some (financial) incentive. The charging and rewarding mechanisms require an accounting infrastructure.

The challenge in the enforcement schemes lies in the reliable detection and punishment of uncooperative behavior. A node might get punished when it is unable and not unwilling to cooperate, e.g., a low battery or a full buffer leading to packet drops. Also, an excluded (punished) node can change its identity in a mobile ad hoc network, in case there is no common (central), trusted instance. The challenge in the encouragement schemes lies in the reliable detection of cooperativeness and the protection of the accounting mechanisms. We think that motivation-based cooperation mechanisms are more suitable for commercial application scenarios, where individual customers with no pre-existing social links want to use the network.

The first detection-based cooperation mechanism [1] avoids uncooperative participants, but does not punish them. In the proposals that followed [2–8], nodes become (partially) excluded from the network in case of uncooperative behavior. The authors of [9] give an overview on the current state of cooperation enforcement schemes. The first motivation-based approach [10] introduces a virtual currency, which is used to charge and reward participants of the mobile ad hoc network for their respective actions. The accounting and security mechanisms are completely decentralized. In [11], the authors improve their initial work. The motivation-based approaches that followed [12–15] all use centralized accounting and security infrastructure.

A centralized design implies the authentication of participants and the collection of cooperation proofs at the gateway(s), which in turn requires source routing. This centralized administrative signaling puts more load on the links towards the gateway. A decentralized architecture implies, that the authentication of nodes as well as the charging and rewarding is performed inside the ad hoc network among nodes, which requires some tamper-resistant hardware. Further, the network provider has no direct control over the decentralized administrative signaling.

In order to combine the advantages of a centralized (control for the provider) as well as a decentralized (flexibility) accounting and security architecture and thereby retain the advantages of multi-hop communications, we proposed a hybrid motivation-based cooperation mechanism [16]. It is called Cooperation and Accounting Strategy in Hybrid Wireless Networks (CASHnet). CASHnet uses decentralized metering and charging on the node, decentralized rewarding among nodes and centralized refill and reward exchange at service stations. The authentication of nodes is based on public key cryptography. We also allow cost sharing between originator and receiver located in different subnetworks. Each of them pays an amount related to their respective distance in hops to the gateway or a globally fixed price. Cost sharing generates revenue in each subnetwork, irrespective of the role of the node, i.e., an originator or a recipient of packets. Distance related charges provide the highest adaptability to the cost, because the longer the route to the gateway, the more intermediate forwarding nodes need to be rewarded. A globally fixed price must at least be equal to the average route length to the gateway in the current multi-hop cellular network. Otherwise the provider will experience a loss.

In this paper, we compare our proposal with the Nuglet [11] scheme in terms of liveliness of the network, overall network performance as well as cash flow. We find that CASHnet outperforms Nuglet under high traffic load and that the self-perpetuating cycle of virtual money assumed by Nuglet is very difficult to sustain in the long run. We also find that the generated overhead is very high for CASHnet and that the immobility of the service stations is a disadvantage. With these results we discuss further improvements to CASHnet.

The rest of the paper is structured as follows. In Section 2, we present and compare the Nuglet and the CASHnet schemes. Section 3 describes our evaluation process. In Section 4, we discuss the results we obtained. We conclude our paper and give an outlook in Section 5.

2. Cooperation schemes

In the introduction, we presented some of the available cooperation schemes in the literature and we motivated our approach. The following two sections describe the CASHnet and the Nuglet scheme, which will be compared through simulations later on. From the available cooperation schemes we chose the Nuglet approach because its decentralized architecture and requirements are similar to CASHnet and therefore easier to compare.

2.1. CASHnet

The CASHnet [16] charging and rewarding mechanism works as follows: Every time a node wants to transmit a self-generated packet or receive a packet addressed to it, the node has to pay with *traffic credits*. The amount is either related to the current distance in hop counts to the gateway or a globally fixed price. Every time a node forwards a packet, it gets *helper credits*. Traffic credits can be bought for real money or traded for helper credits at service stations. A service station is similar to a terminal for loading prepaid cards and has a secure, low-bandwidth connection to the provider, which is used for authentication and payment operations.

For CASHnet, we extend each mobile device with a smart card, which we use to manage all critical information, e.g., the node's identity, cryptographic keys and credit accounts. Fig. 1 depicts an example scenario for CASHnet. It shows a multi-hop cellular network with several mobile nodes equipped with smart cards, two interconnected service stations and two interconnected base stations, which act as gateways to the Internet. The service stations and the gateways are connected to their respective backbone network. In CASHnet, all transmitted packets are digitally

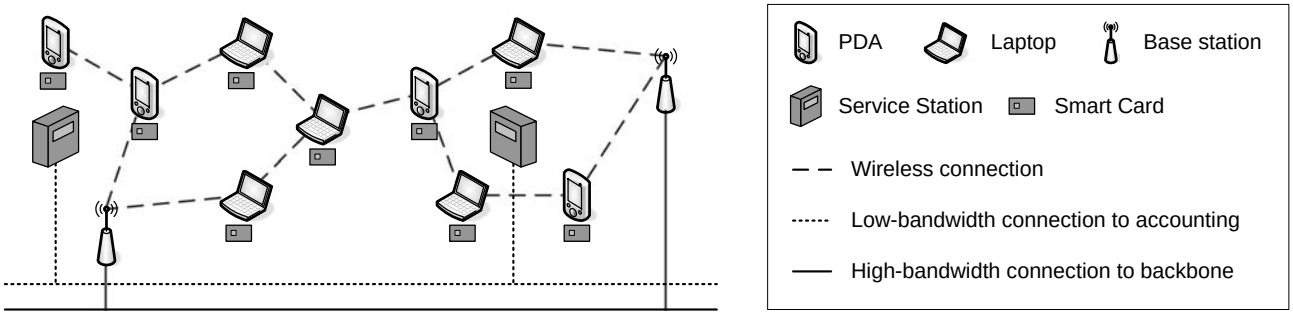

Fig. 1. CASHnet example scenario.

signed and verified upon reception to ensure non-repudiation, i.e., data integrity and data origin authentication.

The typical course of action for a customer, who wants to participate in a CASHnet-enabled multi-hop cellular network, consists of five steps: preparation, authentication, transmission/reception & charging, and forwarding & rewarding and refill. The first and the last step are performed at the service station, where the customer inserts her smart card. Fig. 2 illustrates step two to four as a message sequence chart for a scenario, where the originator and the recipient are located in different multi-hop cellular networks. The numbered gray markers refer to example positions of the actions from the following list. The rows at the bottom indicate the transactions on the traffic credits, TC and helper credits, HC accounts.

(1) *Preparation.* The customer obtains a smart card from the provider and loads the traffic credits accounts at the service station.
(2) *Authentication.* Preliminary to the normal communication with a recipient, the originator O sends a certificate advertisement $CADV_O$ to the recipient R. Thereby all intermediate nodes (*A*, *B*, and *C*) and the recipient obtain the authentication information of the originator. The recipient in turn replies with a certificate reply $CREP_R$ addressed to O. Now all intermediate nodes obtain the authentication information of the recipient.
(3) *Transmission/reception & charging.* Before the transmission of a self-generated packet, the originator's traffic credits account is charged (−3 TC) and the packet is digitally signed. Upon reception of a packet destined to the current node, the recipient's traffic credits account is also charged (−2 TC).
(4) *Forwarding & rewarding.* At the reception of a packet, the node rewards the previous forwarding node in case it was not the originator or a gateway by sending a digitally signed acknowledgement *ACK*. Receiving an *ACK* increases the node's helper credits account (+1 HC). The node also removes the digital signature of the previous node and adds its own before forwarding the packet. In addition, the node keeps the digital signature of each forwarded packet in order to validate the *ACKs*.
(5) *Refill.* After some time, the customer goes to a service station in order to refill her traffic credits account by exchanging available helper credits and/or buying traffic credits for real money.

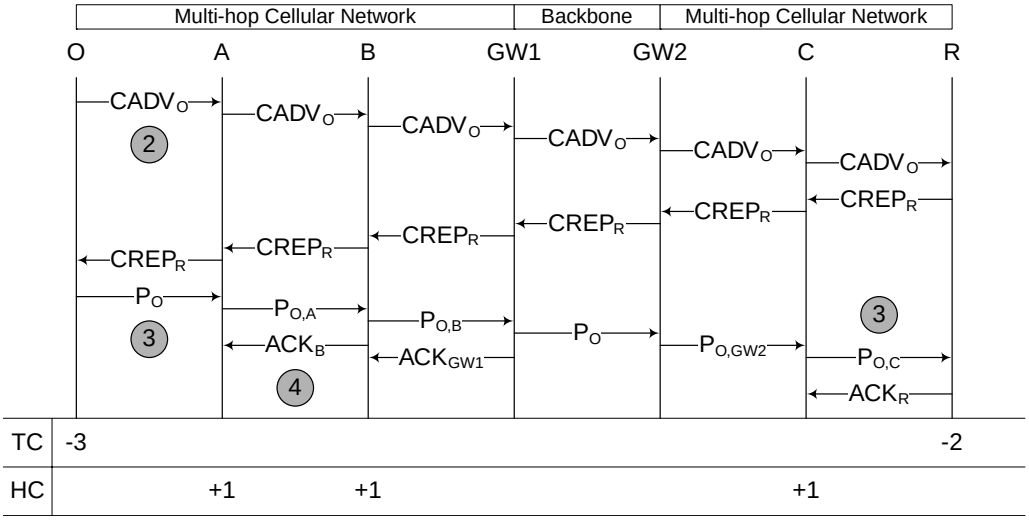

Fig. 2. CASHnet operation message sequence chart.

2.2. Nuglet

The Nuglet [11] cooperation mechanism is targeted at mobile ad hoc networks and has the following main principle: Every time a node wants to transmit a self-generated packet, it has to pay with *nuglets*. The amount corresponds to the estimated number of intermediate nodes between the originator and recipient. Every time a node forwards a packet it receives one nuglet.

In Nuglet, each mobile node obtains a smart card from the provider. The smart card provides the protected environment in which the Nuglet functions and the routing protocol run. A so called security module acts on behalf of the node and is responsible for establishing the security associations, i.e., a session key, with other modules. For each packet to be transmitted, a message authentication code is computed on a node-to-node basis using the respective session key of two neighbor nodes.

The typical procedure for a node, which wants to participate in a Nuglet-enabled mobile ad hoc network consists of four steps: preparation, authentication, transmission & charging, and forwarding & rewarding. The first step is performed only once. The flow of virtual money is expected to be self-perpetuating, i.e., nodes can forward enough packets to be able to cover the cost of transmitting self-generated packets. Fig. 3 shows step two to four as a message sequence chart for a scenario, where the originator and the recipient are located in a mobile ad hoc network. The numbered gray markers refer to example positions of the actions from the following list. The rows at the bottom indicate the operations on the nuglet, N and pending nuglet, PC accounts.

(1) *Preparation.* The customer obtains a smart card from the provider and loads the nuglet account.
(2) *Authentication.* Before a node can communicate with its one-hop neighbors, it tries to establish a security association (i.e., a symmetric session key). Only nodes with a security association are considered neighbors by the routing protocol.
(3) *Transmission & charging.* Before the transmission of a self-generated packet, the originator's nuglet account is charged (−2 N) and a message authentication code is added to the packet to ensure non-repudiation. The message authentication code is based on the session key, which the node has established with the next hop towards the recipient.
(4) *Forwarding & rewarding.* When a node receives a forwarded packet, i.e., the previous hop is not the originator, it increases the pending nuglet counter for the previous hop (+1 N). The node also removes the message authentication code computed by the previous hop and adds its own using the session key it has established with the next hop towards the recipient. To credit the pending nuglets to the respective nodes, a synchronization protocol runs periodically on each node. It transmits all pending nuglets of a certain node to this node and resets the pending nuglet counter.

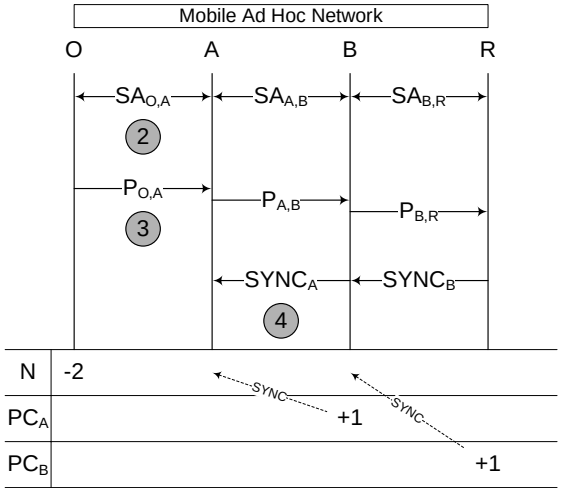

Fig. 3. Nuglet operation message sequence chart.

2.3. Comparison

Although the two motivation-based cooperation mechanisms were targeted at different networks, they have similar requirements because of their decentralized design. Both schemes rely on tamper-resistant hardware and public key cryptography. In addition, Nuglet uses symmetric key sessions among one-hop neighbors. CASHnet and Nuglet charge for the transmission of self-generated packets. If a node has not enough virtual money (traffic credits or nuglets), it is not allowed to transmit its own packets. In CASHnet the cost is related to the hop count to the gateway, in Nuglet to the number of intermediate nodes to the destination. Additionally, CASHnet allows cost sharing among nodes residing in different multi-hop cellular networks by charging for the reception of packets addressed to the current node.

Both schemes stimulate the cooperation among nodes (forwarding packets) through rewards. A node, which forwards a packet receives 1 or more helper credits or 1 nuglet, respectively. In CASHnet as well as in Nuglet, a node can earn its right for transmission, i.e., it must forward enough packets to be able to send its own packets. The CASHnet scheme additionally allows a node to buy its right for transmission, using additional infrastructure in the network (i.e., service stations). Thus, Nuglet actually enforces the cooperation of a node, as it has no other choice but to cooperate.

Because our current implementation of the two schemes does not include the cryptographic functionality, the security mechanisms are left out from the evaluation. In Nuglet each pair of communicating nodes establishes a symmetric key session to reduce the computational overhead. CASHnet only uses public key cryptography. The high mobility in

ad hoc networks is a disadvantage for the session establishment in Nuglet, whereas the power constraints of mobile devices might be a minor disadvantage for the computationally expensive public key operations in CASHnet.

3. Simulation scenarios

We evaluate both mechanisms through simulations, where we measure the amount and frequency of starving events in the network, i.e., nodes that cannot transmit, because they run out of virtual money (traffic credits or nuglets) and use this as an indicator for the liveliness of the network. Also we give results on the overall packet delivery ratio as well as generated overhead and packet drop reasons. Further, we analyze the cash flow of the virtual money. We adjusted our schemes' parameters to match the Nuglet scheme to provide a solid foundation for the comparative evaluations.

For the simulation we use ns-2 [17], where we implemented a version of the Nuglet and the CASHnet scheme including the charging and rewarding functionality and leaving out the security mechanisms. In particular, we used the wireless and mobility extensions [18] with an extended version of the AODV protocol called AODV+ [19], which adds Internet gateway discovery support as defined in [20].

Fig. 4 shows our simulation scenarios. We only consider a single multi-hop cellular network to be compatible with the Nuglet scheme, which is targeted at mobile ad hoc networks. All nodes in the network send their packets to the gateway. The simulation scenario for Nuglet differs from the CASHnet scenario in two aspects: the service stations are removed and the gateway is replaced by a normal node.

Table 1 lists the parameters for the simulations. Except for the scheme specific properties, all parameters are identical. Within an area of 900×600 m we deploy 40 nodes. The nodes move according the random waypoint model using pre-generated movements files. We vary the number and the distribution of deployed service stations (none for Nuglet and 1, 2, 5, 9 for CASHnet) as shown in Fig. 4 as well as the packet generation interval at the CBR traffic sources (0.5, 1, and 2 seconds corresponding to a packet generation rate of 2, 1, and 0.5 packets/s). The packet length is 512 bytes. In total, we investigate 15

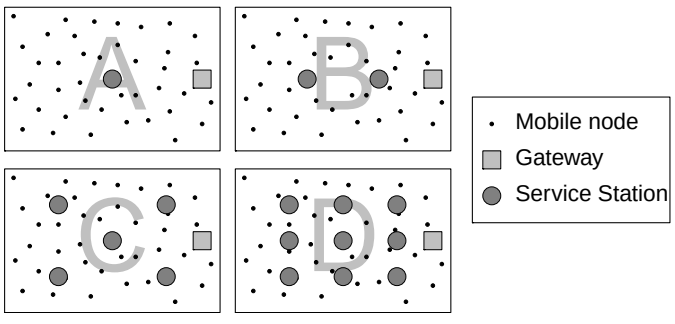

Fig. 4. Simulation scenarios.

Table 1
Simulation parameters

Parameter	Value	
	CASHnet	Nuglet
Space	900×600 m	
Number of nodes	40	
Transmission range	250 m	
Mobility model	Random waypoint	
Speed	Uni. dist. between 1 and 10 m/s	
Pause time	Uni. dist. between 0 and 20 s	
Packet generation interval	0.5, 1, 2 s	
Routing	AODV+	
Simulation time	900 s	
Initial virtual money account state [N/TC]	1000	1000
Initial real money account state [RM]	200	—
Nuglet synchronization interval [s]	—	5
Trade thresh. at service stations [HC]	10	—
Helper/traffic credits rate	1:1	—
Real money/traffic credits rate	1:20	—
Distance thresh. to service stations [m]	50	—
Number of service stations	1, 2, 5 or 9	—

(5×3) simulation scenarios and for each of the scenarios we conduct 20 simulation runs using 20 independent movement files.

In both schemes, the amount of initial virtual money is set to 1000 traffic credits, TC or nuglets, N. To reflect the ability of a CASHnet node, to buy its right for transmission from available service stations, each node also has a real money account initially set to 200. Real money does not exist in the Nuglet scheme and is not equal to virtual money, as it must be exchanged first. Therefore, we believe the comparison to be fair in a sense that both schemes have the same initial situation according to their abilities. In the Nuglet scheme, a node needs to find other nodes and forward their packets to earn nuglets. In the CASHnet scheme, a node needs to find a service station to exchange the helper credits and the real money against traffic credits. The problem of Nuglet lies not in the initial amount of virtual money, but in the lack of refill possibilities for virtual money.

The trade threshold defines the minimum amount of helper credits necessary before a node exchanges them into traffic credits at the service station. The distance threshold specifies the maximum distance between a node and a service stations to be able to exchange the helper credits.

4. Simulation results

We measured the frequency of occurrence of starvation events and the duration of the starvation for both schemes. We also analyzed the packet flow, in particular the overall goodput, the generated overhead and the reasons for dropped packets. In addition, we investigated the cash flow in the network.

In the following figures, we show mean values, which we obtained by averaging the results from the 15 simulation scenarios and – in case of the starvation and the cash flow – the 40 contained nodes. The packet flow results already represent the overall network performance. We also show the corresponding standard deviation. Each label marking on the *x*-axis consists of two lines. The first line indicates the number of service stations used (1, 2, 5, 9 for CASHnet and 0 for Nuglet). The second line lists the packet generation interval (0.5, 1, and 2 s). The three packet generation intervals are separated by vertical lines.

4.1. Starvation

With starvation we describe the nodes inability to transmit self-generated packets due to lack of virtual money. Both cooperation schemes charge for sending self-generated packets and reward for forwarding packets. In Nuglet, a node has only one source of income for virtual money (nuglets): it has to forward packets from other nodes. In CASHnet, a node has two sources of income for virtual money (traffic credits): it can exchange the virtual money earned while forwarding packets from other nodes (helper credits) or pay with real money. We find that a self-perpetuating cycle of virtual money, which is assumed by the Nuglet scheme, is difficult to achieve. In such a cycle, each node always receives enough virtual money to be able to transmit self-generated packets. Fig. 5 shows the mean starvation duration and occurrences for the 15 scenarios, which we previously described. The figures on starvation show the mean duration as bars and the occurrences as points with standard deviation in solid and dashed lines, respectively.

For CASHnet we see that under high traffic load (packet generation interval of 0.5 s) and 1 service station, the mean starvation duration per node is around 2/3 of the simulation time (348 s) with an average of 1.2 starvation occurrences (periods). With Nuglet, the average starvation duration per node is half of the simulation time (453 s) with 15 occurrences. The mean starvation duration in Nuglet is high, because the self-perpetuating cycle of Nuglets cannot be established in the network. To the contrary, many nodes run out of nuglets and are therefore unable to transmit self-generated packets. The increased number of starvation occurrences in Nuglet is due to the periodic synchronization protocol, which transfers pending nuglets to the respective nodes every 5 s. In CASHnet, a node can only refill the traffic credits at a service stations. Because the nodes move according to the random waypoint mobility model, an encounter with a service station does not occur often. Increasing the number of service stations to 9, reduces the average starvation duration in CASHnet to 58 s.

When we increase the packet generation interval to 1 s, the mean starvation duration in CASHnet with 9 service stations drops to 4 s, while in Nuglet, a node starves 140 s on average. In the case of low traffic load (packet generation interval of 2 s), CASHnet performs worse than Nuglet for scenarios with 1 and 2 service stations. We attribute this to the reduced probability of meeting a service station. Also, the reduced number of generated packets and the regular transfer of pending nuglets almost eliminates starvation in Nuglet. With 5 or 9 service stations, CASHnet achieves similar or better results than Nuglet.

The high standard deviation for all scenarios indicates that some nodes almost never starve, while others starve for almost the double of the average duration. This is due to the random movement of the nodes, which decides about their probability of receiving packets to be forwarded, of receiving pending nuglets or of meeting a service

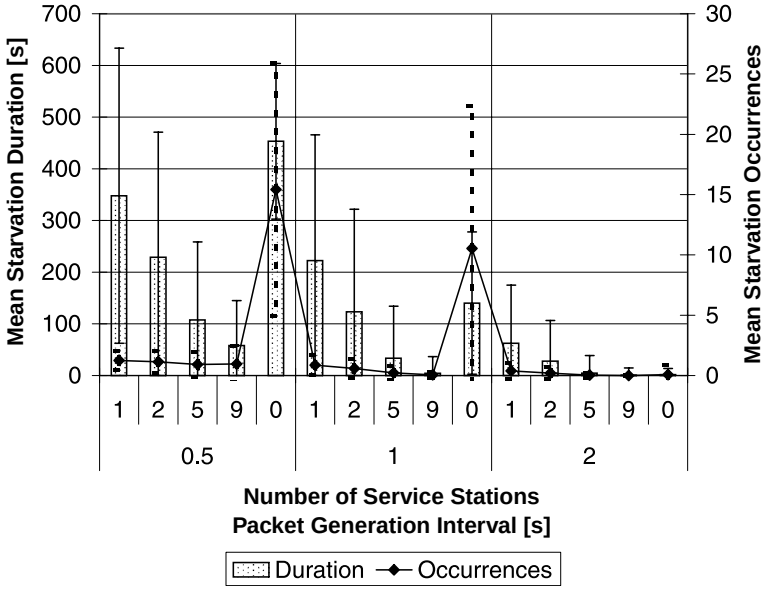

Fig. 5. Starvation in CASHnet & Nuglet.

station. The low number of average occurrences indicates that few and long starvation periods occur.

4.2. Packet flow

The packet flow describes the overall network performance of both cooperation schemes. We analyze the goodput, i.e., number of transmitted and received packets as well as the overhead, i.e., the amount and effectiveness of the signaling traffic. We also investigate the different drop reasons for the data packets given by the simulator.

Fig. 6 shows the goodput in both schemes. For a packet generation interval of 0.5 s, we measure a mean goodput of 42% for CASHnet with 1 service station and 39% for Nuglet. When we use CASHnet with 2 service stations, the goodput increases to 44%. However, increasing the number of service stations to 5 or 9 decreases the goodput slightly. This is due to congestion as we will see in the analysis of the drop reasons. When we use a packet generation interval of 1 s, we measure a goodput of 78% for CASHnet with 9 service stations and a goodput for Nuglet of 73%. For a packet generation interval of 2 s, we obtain 82% goodput for Nuglet and 80% for CASHnet with 9 service stations. We attribute this to the fact, that the immobile service stations in CASHnet are only encountered randomly by the nodes to refill their traffic credits account, while the periodic transfer of nuglets provides for a regular refill. However, the total number of nuglets in the network decreases over time as we will see in the analysis of the cash flow, which renders the long-term operation of Nuglet impossible.

Fig. 7 illustrates the actual overhead introduced by both schemes. It shows the amount of signaling messages (CASHnet ACK/Nuglet SYNC) transmitted and dropped. In CASHnet, the overhead is much higher than in Nuglet because every packet is rewarded immediately on a per-hop basis by an ACK packet, which increases the nodes helper credits account. In Nuglet, each node collects the rewards for its neighbors in dedicated accounts and transfers this virtual money periodically to all reachable nodes. A former neighbor node, that is not reachable at the time of the synchronization looses all its earned virtual money on that node.

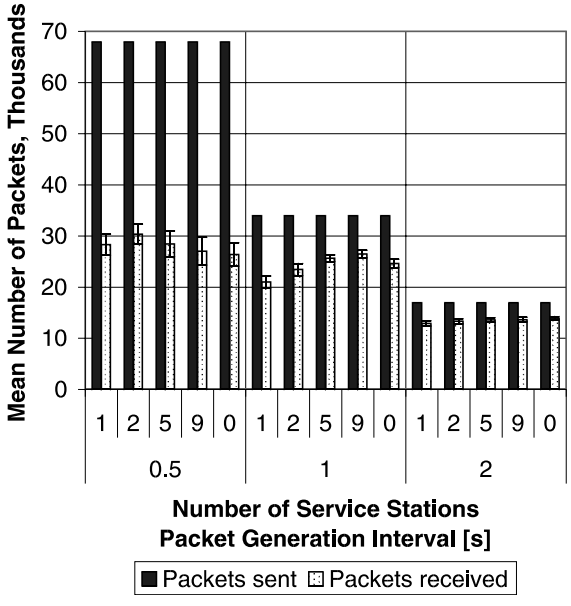

Fig. 6. Goodput for CASHnet & Nuglet.

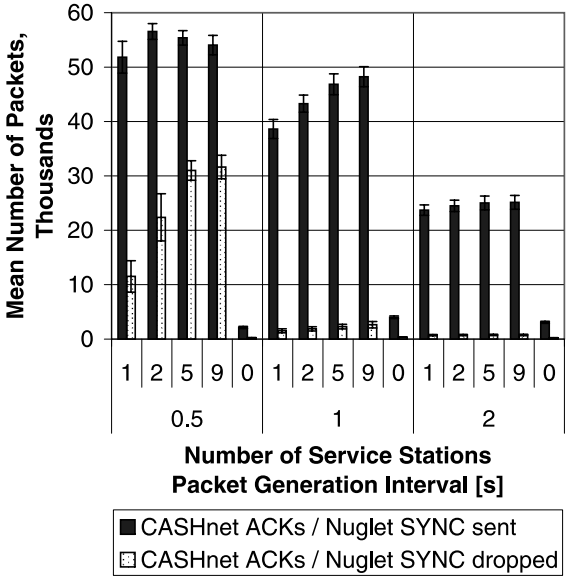

Fig. 7. Overhead for CASHnet & Nuglet.

For CASHnet we observe, that the number of dropped acknowledgements increases with the number of service stations under high network load even though the number of transmitted acknowledgements does not increase. We attribute this phenomena to the congestion. With more service stations, the nodes are able to transmit more packets. As the network is already congested, these packets place an additional burden on the interface queues and thus block the transmission of acknowledgements.

The different reasons for dropped packets are displayed in Fig. 8. The drop reasons, which occurred in our simulation runs were lack of money (No Cash), no available route (No Route), routing loop (Loop), link break detection at the routing layer (Callback), a full buffer in the address resolution protocol, and a full buffer in the interface queue. When a route is not available in AODV, a route request is sent and the packet will be retained until the route requested succeeds or times out, which leads to the No Route error. The Callback error indicates that the link detection from AODV has noticed a link break. When the address resolution protocol resolve an address for a packet by sending a request, it buffers the packet until the reply arrives. If in the meantime, to many new requests arrive, the buffer is full and the unanswered packets are dropped.

In both schemes, the main reason for packet drops is the lack of virtual money. In Nuglet, it is difficult to generate enough traffic to build up a self-perpetuating cycle of virtual money. In CASHnet, a node has the possibility to

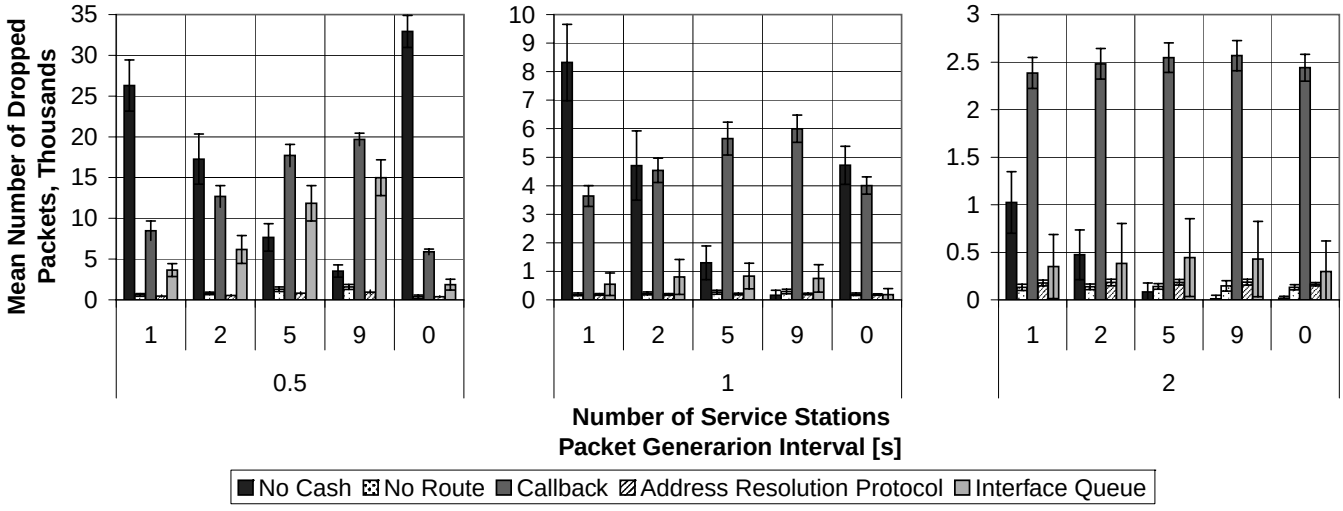

Fig. 8. Packet drop reasons for CASHnet and Nuglet.

buy its right for transmission. However, due to the dependency on the fixed service stations as the only place for obtaining traffic credits, the positive effect of having two sources of income for virtual money is reduced. The second major packet drop reason is the unavailability of a route. In the simulation runs we used an extended version of AODV. When a route is not available in AODV, a route request is send and the packet will be retained until the route requested succeeds or times out.

For CASHnet, we find that while the number of packets dropped due to lack of money (No Cash) decreases with more deployed service stations, the packets dropped in the interface queue and the initiated drops by the MAC layer (Callback) increases. We found, that the link layer trigger in ns-2 interprets congestion as link break and commands AODV to search for a new route, which worsens the situation.

4.3. Cash flow

The cash flow provides important information about the distribution of virtual money in the network. We analyze the average final account states of the virtual money per node. In addition, we show the average amount of virtual money spent for the transmission of self-generated packets. Further, we present the amount of virtual money traded at the service station for CASHnet and received by other nodes for Nuglet.

Fig. 9 illustrates the cash flow for both cooperation mechanisms. We observe that the average final virtual

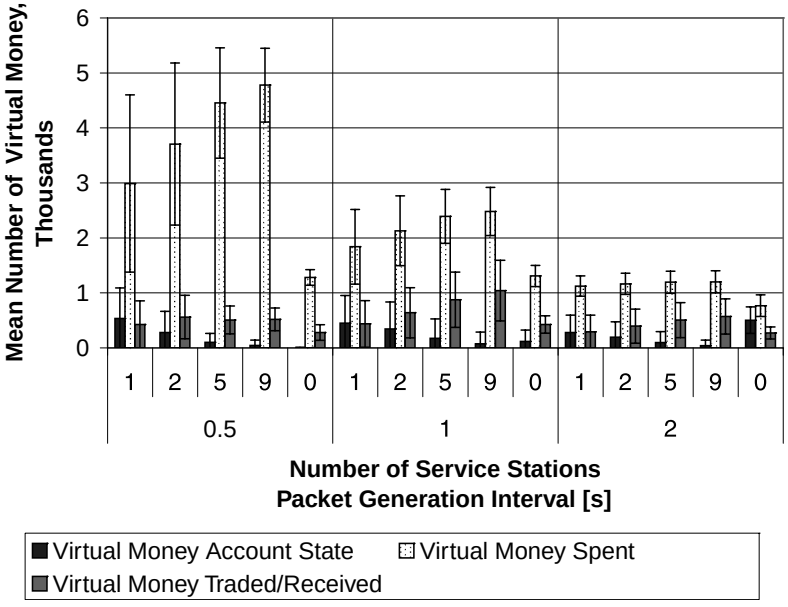

Fig. 9. Cash flow for CASHnet and Nuglet.

money account state in CASHnet with 1 service station is 533 traffic credits under high network load, while for Nuglet it is only 1 nuglet. This indicates that in CASHnet, the nodes were unable to refill their account in time, while in Nuglet, the nodes ran out of virtual money and had no possibility to earn additional nuglets.

As expected, the amount of traffic credits spent increases with the number of service stations, because the nodes have more opportunities to refill their account, which is also indicated by the reduced final state account of helper credits. However, as we previously saw the increased number of traffic credits spent does not automatically result in an improved goodput under high network load. Also, the number of traffic credits traded is very similar under high network load, which is caused by the high amount of acknowledgements lost due to congestion. The high standard deviation for CASHnet indicates that some nodes have a depleted traffic credits account, which is due to the random encounter with the service stations.

5. Summary and outlook

We presented CASHnet, our cooperation and accounting strategy for hybrid networks, which uses a highly decentralized accounting and security architecture. It allows selfish nodes and supports cost sharing between sender and receivers located in different subnetworks. To put the performance of our scheme in context with other work in this area, we compared the CASHnet with the Nuglet scheme, which was also explained in this paper. We implemented both schemes in ns-2 and evaluated them through simulation runs. We monitored the network liveliness, the overall network performance and the cash flow.

As a result from the evaluation, we see that the goal of the Nuglet scheme, that is a self-perpetuating cycle of virtual money, is difficult to achieve. We find that in Nuglet the amount of virtual money decreases over time, which makes the long-term operation questionable. CASHnet performs better than Nuglet in scenarios with high network load independent of the number of service stations. In scenarios with low network load, CASHnet requires 2 or 5 service stations to perform similar or better than Nuglet. The high protocol overhead and the immobility of the service stations in CASHnet weakens the positive effect of an additional source of income for virtual money.

For CASHnet, we see room for improvement in the granularity of the charging and rewarding mechanisms. This would help to reduce the overhead. In our current approach, we use service stations as low-bandwidth, low-cost terminals for buying and exchanging virtual money (similar to loading a prepaid card). We are currently investigating the possibility of introducing mobile service stations to make up for the random encounters caused by our mobility model.

Using other mobility models with more realistic user behavior and adapting the deployment of service stations accordingly could also greatly improve our schemes performance. Additionally, the generic behavior of the customers themselves could be made more realistic, e.g., when running out of virtual money, the movement direction changes to the closest service station.

Further work will include the implementation of a prototype of our CASHnet scheme under Linux using Smart Cards. We will analyze our security mechanisms in terms of effectiveness against different attack types and resource consumption. Also, we will study possible extensions to our scheme and optimize the relation between charging and remuneration.

Acknowledgment

This work is supported by the Swiss National Science Foundation under Grant No. 20-68086.02/1.

References

[1] S. Marti, T.J. Giuli, K. Lai, M. Baker, Mitigating routing misbehavior in mobile ad hoc networks, in: Proceedings of 6th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Boston, MA, USA, 2000, pp. 255–265.

[2] S. Buchegger, J.-Y.L. Boudec, Performance analysis of the confidant protocol (cooperation of nodes – fairness in dynamic ad-hoc networks), in: Proceedings of 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Lausanne, Switzerland, 2002, pp. 226–236.

[3] P. Michiardi, R. Molva, Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks, in: Proceedings of 6th IFIP Conference on Communications and Multimedia Security (CMS), Portoroz, Slovenia, 2002, pp. 107–121.

[4] K. Paul, D. Westhoff, Context aware detection of selfish nodes in dsr based ad-hoc networks, in: Proceedings of IEEE GLOBECOM, Taipei, Taiwan, 2002, pp. 178–182.

[5] H. Yang, X. Meng, S. Lu, Self-organized network-layer security in mobile ad hoc networks, in: ACM Workshop on Wireless Security (WiSe), Atlanta, GA, USA, 2002, pp. 11–20.

[6] S. Bansal, M. Baker, Observation-based cooperation enforcement in ad hoc networks, Tech. Rep. cs.NI/0307012, Stanford University (Jul. 2003).

[7] M. Frank, P. Martini, M. Plaggemeier, Cinema: cooperation enhancement in manets, in: Proceedings of 29th IEEE Conference on Local Computer Networks (LCN), Tampa, FL, USA, 2004, pp. 86–93.

[8] Q. He, D. Wu, P. Khosla, Sori: a secure and objective reputation-based incentive scheme for ad-hoc networks, in: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), Atlanta, GA, USA, 2004, pp. 13–15.

[9] S. Buchegger, J.-Y. Le Boudec, Self-policing mobile ad hoc networks by reputation systems, IEEE Communications Magazine 43 (7) (2005) 101–107.

[10] L. Buttyán, J.-P. Hubaux, Enforcing service availability in mobile ad-hoc wans, in: Proceedings of 1st ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Boston, MA, USA, 2000, pp. 87–96.

[11] L. Buttyán, J.-P. Hubaux, Stimulating cooperation in self-organizing mobile ad hoc networks, ACM Mobile Networks and Applications 8 (5) (2003) 579–592.

[12] M. Jakobsson, J.-P. Hubaux, L. Buttyán, A micro-payment scheme encouraging collaboration in multi-hop cellular networks, in: Proceedings of 7th International Financial Cryptography Conference, Gosier, Guadeloupe, 2003, pp. 15–33.

[13] S. Zhong, J. Chen, Y.R. Yang, Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks, Proceedings of 22nd IEEE INFOCOM, vol. 3, San Francisco, CA, USA, 2003, pp. 1987–1997.
[14] N.B. Salem, L. Buttyán, J.-P. Hubaux, M. Jakobsson, A charging and rewarding scheme for packet forwarding in multi-hop cellular networks, in: Proceedings of 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Annapolis, MD, USA, 2003, pp. 13–24.
[15] B. Lamparter, K. Paul, D. Westhoff, Charging support for ad hoc stub networks, Elsevier Journal of Computer Communications 26 (13) (2003) 1504–1514.
[16] A. Weyland, T. Braun, Cooperation and Accounting Strategy for Multi-hop Cellular Networks, in: Proceedings of 13th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN), Mill Valley, CA, USA, 2004, pp. 193–198.
[17] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, H. Yu, Advances in network simulation, IEEE Computer 33 (5) (2000) 59–67.
[18] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proceedings of 4th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Dallas, TX, USA, 1998, pp. 85–97.
[19] A. Hamidian, A study of internet connectivity for mobile ad hoc networks in ns 2, Master's thesis, Lund University, Sweden (Jan. 2003).
[20] R. Wakikawa, J.T. Malinen, C.E. Perkins, A. Nilsson, A.J. Tuominen, Global connectivity for ipv6 mobile ad hoc networks, Internet-Draft, [visited 8 Aug. 2005] (Jul. 2005). URL <http://www.ietf.org/internet-drafts/draft-wakikawa-manet-globalv6-04.txt/>.

Attila Weyland received his masters and Ph.D. degree in computer science from University of Bern in 2002 and 2005, respectively. His research interests include wireless networks and technologies with specific focus on cooperation and accounting in multi-hop networks. He also worked in the area of distance learning, in particular on infrastructures for remote network laboratories as well as didactic aspects.

Thomas Staub received his masters degree in computer science from University of Bern in 2004. He's a Ph.D. student in the Computer Networks and Distributed Systems research group at University of Bern. His research interest include wireless mesh networks and wireless sensor networks, in particular multi-path routing and quality of service.

Torsten Braun received his Masters and Ph.D. Degree from University of Karlsruhe (Germany) in 1990 and 1993, respectively. From 1994 to 1995, he has been a visiting scientist at INRIA, Sophia-Antipolis (France). He worked at the IBM European Networking Center at Heidelberg (Germany) from 1995 to 1997 as senior consultant and project leader. Since 1998, he has been full professor of computer science at the University of Bern (Swiztzerland) heading the research group on Computer Networks and Distributed Systems. Since 2000, he has been member of the SWITCH (Swiss research and education network) board. In 1994, he spent his sabbatical at INRIA Sophia-Antipolis and the Swedish Computer Science Institute at Kista.