Reliable and Semi-reliable Communication with Authentication in Mobile Ad Hoc Networks

Falko Dressler
Autonomic Networking Group, Dept. of Computer Sciences, University of Erlangen
Martensstr. 3, 91058 Erlangen, Germany
dressler@informatik.uni-erlangen.de

Abstract* - *A typical characteristic of wireless ad hoc sensor networks is the error-proneness and, therefore, the unreliability of communication paths. The search for reliable communication methodologies in this area in combination with appropriate security mechanisms has become a main research activity in the networking community. In this paper we describe a new approach and an according protocol for usage in ad hoc networks that provides reliable as well as semi-reliable communication services. Additionally, the proposed methodology allows to ensure data integrity and message authentication. The main aspects during the development were the limitations of typical sensor nodes in terms of available resources such as storage, processing power, and energy. Our solution provides the capability of acknowledging correct receptions as well as the check of data integrity and message authentication in a single step. Therefore, a low overhead solution was created providing all the mentioned communication goals.

Keywords* - *Mobile ad hoc networks, wireless sensor networks, reliable communication, message authentication

I. INTRODUCTION

In recent years, many efforts have been made in developing algorithms and methodologies for building efficient network mechanisms for reliable communications in mobile ad hoc networks [13]. This work is mainly driven by the spreading of wireless network technologies. The primary requirements are efficiency, adaptability, and scalability. New communication paradigms are needed for the forthcoming pervasive networking world. For example, the research fields of ad hoc wireless sensor networks (WSNs) require mechanisms and technologies for achieving optimum data rates while addressing issues such as power-consumption and reorganization during the data transfer [1, 2, 6].

Many groups are working on reliable communication in ad hoc networks. This effort is mainly driven by the key idea to adapt the mechanisms known from transport protocols such as TCP (transmission control protocol, [9]). Semi-reliable or partially reliable transport protocols in the Internet were designed to overcome the drawbacks of TCP in error-prone networks. The best-known example is SCTP (stream control transmission protocol, [11]) and its partial reliable extension [12]. Such protocols fail in large-scale ad hoc networks due to the immense resource requirements in terms of memory to store state information and sometimes complete messages and in terms of computational complexity. Other approaches are required that are primarily focused on ad hoc networks ad their capabilities, e.g. Obraczka et al. [8] provided a flooding approach for reliable group communication in multi-hop ad hoc networks. The applicability of TCP over ad hoc networks was analyzed in various studies such as [5, 7].

Questioning the requirements on communication system in today’s ad hoc networks, we find similar characteristics in WSNs, pervasive computing environments, and WPANs (wireless personal area networks). Such networks are built of small entities with little available resources in terms of processing power, memory, and energy. Therefore, the quite big communication protocols developed for the Internet are difficult or not applicable. In this paper, we present an approach for providing reliable and semi-reliable communication services in ad hoc networks based on the typical properties of reliable communication protocols, e.g. having sequence numbers and transmission windows. In addition, the same methodology allows to include security services such as data integrity checks, message authentication, and address verification. Because the latter two services are based on shared secrets known to both communication end points, confidentiality can be provided using the same keys and a low-overhead symmetric encryption algorithm. The proposed methodology allows a (online) tuning of all parameters to ensure an optimal utilization of the available resources at each communicating node. Therefore, it was possible to develop an adaptive, self-organizing methodology well suitable for WSNs.

The rest of the paper is organized as follows: in section II, the basic objectives and some motivation for this work is provided. In section III, the methodology is presented and the resulting RAC (reliable authenticated communication) algorithm is depicted in section IV. The parameters of the algorithm and the resource requirements are analyzed in a comparative simulation in section V. Finally, some conclusions summarize the paper.

0-7803-9466-6/05/$20.00 ©2005 IEEE

II. MOTIVATION AND OBJECTIVES

Among others, the following issues have to be addressed in the research area of mobile WSNs: data storage, data aggregation, and communication between individual nodes for data exchange and management tasks. Typically, WSNs are composed of multiple, independent, autonomously working nodes. These individual entities form a self-organizing compound which is able to solve tasks described at a higher level. A typical sensor network is shown in Fig 1. Pervasive communication entities are forming an ad hoc network be discovering the environment, setting up neighborhood relations, and using some kind of routing methods to perform end to end communication.

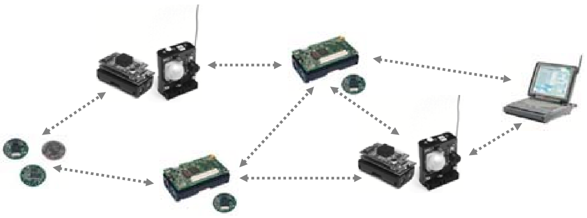

Fig 1. A typical wireless, multi-hop ad hoc sensor network consisting of heterogeneous mobile nodes

As already mentioned, we focus on communication aspects in WSNs, i.e. in the inter-node communication. For an effective self-management, current information about the state of individual network nodes is required, typically about neighborhood relationships and relevant distribution systems. Applications are for example routing mechanisms (ad hoc, pro-active, store-and-forward), the detection of failures, and the management of tasks and resources. The reduction of this state information itself provides interesting research aspects as shown by Dressler et al. [4].

The objective of this paper is to analyze the end-to-end communication in ad hoc networks and to provide a new solution to the requirement of reliable and semi-reliable communication. Additionally, security issues such as data integrity check and message authentication are solved by the same methodology. The key requirements for the algorithm were:

- scalability, i.e. the overhead due to the algorithm should be negligible (message count, message size, memory and processing requirements)
- flexibility, i.e. the optional selection of needed functionality such as reliability vs. semi-reliability and data integrity check vs. full message authentication
- configurability, i.e. the option to adapt the parameters to the capabilities of the particular entities involved in the communication
- extensibility, i.e. the possibility to implement new functionality such as data encryption to provide confidentiality

Additionally, the mechanism should not essentially contribute to the congestion in the network, e.g. by building live-locks due to unnecessarily high message rates. Congestion in WSNs is an important issue not yet solved by means of Internet congestion control mechanisms and novel approaches are required [3].

III. RELIABLE AND AUTHENTICATED COMMUNICATION IN AD HOC NETWORKS

In this section, the main properties of the reliable and authenticated communication mechanism are depicted. We start with the presentation of the corresponding model followed by a detailed overview of the hash-based reliability check and message authentication methodology.

A. *Model and Assumptions*

The underlying model of the proposed methodology is shown in Fig 2.

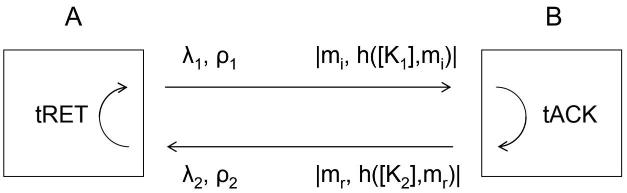

Fig 2. Model for (semi-)reliable and authenticated communication

Two communication end points (A and B) are involved, but only the unidirectional message transmission from A to B is shown. At A, messages (m_i) are created and transmitted with a rate λ_1. The communication channel has a loss probability of ρ_1. Each message sent from A to B is padded with a signed hash value (using key K_1) of the complete message (payload). At B, the hash values are stored for a while (tACK) and sent back to A in a compressed form (m_r), i.e. multiple hash values in a single message. Again, a signature is created (using key K_2). The resulting acknowledge rate is depicted as λ_2. The channel from B to A has a loss probability of ρ_2. Retransmissions as used in full-reliable mode are triggered by a retransmission timeout (tRET) individually for each message. Back-off mechanisms as provided by current TCP variants might be employed for fine-tuning of the tRET parameter.

Based on these properties, the corresponding equations for calculating the acknowledge rate λ_2 can be determined:

- Message arrival rate at B: $\lambda_1' = \lambda_1 * (1 - \rho_1)$
- Number of received messages in the acknowledge window: $|m_{recv}| = \lambda_1' * tACK$
- Acknowledge rate using a_{max} (maximum number of acknowledges per message): $\lambda_2 = (|m_{recv}| * a_{max}) / tACK$

B. *Hash-based Reliability*

Typically, sequence numbers are used to provide reliable or semi-reliable communication. On the one hand, this mechanism allows to easily check the number of lost messages and to provide a window mechanism for available retransmissions. On the other hand, for semi-reliable transmissions, i.e. if only the reception of messages should be verified, there is no advantage of subsequent numbers identifying the individual messages. We decided to skip the classical sequence numbers and employ hash-based mechanisms instead. A hash value is calculated for each message (payload) and stored at the sender. If the message is acknowledged, the corresponding message was completely received at the destination. The change has some advantages compared to sequence numbers. First, there is no simple overflow of the number space. Depending on the distribution of the used hash algorithm, it is unlikely to have multiple identical hash values in a short period of time (see below). In this context, the low overhead created by the check of successful reception – If used in semi-reliable mode – has to be mentioned. Finally, it allows an all-in-one solution for reliable data communication, verification of data integrity, and message authentication.

There are a couple of requirements on the hash function to use in this methodology. It must be a collision-resistance hash function, i.e. it is computationally infeasible to find any pair (x, x') with x ≠ x' such that h(x) = h(x'), and it must be computable with low processing overhead. Research on such hash functions derived at least two useful functions: MD5 (message digest 5) and SHA-1 (secure hash algorithm 1). We decided to employ MD5 [10] but any other hash function which fulfils the mentioned requirements can be used as well.

C. *Data Integrity Check and Message Authentication*

The proposed solution for (semi-)reliable data communication already includes all necessary elements for offering security services such as data integrity check and message authentication. In addition to the requirements on the employed hash function presented in the last section, there are some more on a cryptographic secure hash function in order to ensure such security services. Such a cryptographic hash function must be pre-image resistance, i.e. for essentially all pre-specified outputs y, it is computationally infeasible to find an x such that h(x) = y, and 2nd pre-image resistance, i.e. given a x it is computationally infeasible to find any second input x' with x ≠ x' such that h(x) = h(x'). For example, the MD5 fulfils these requirements.

Finally, the hash function is used for two purposes:

1. verification of the successful reception of a particular message and
2. check of the data integrity during the transmission.

If a shared key is available between both communication end points, it can be used for complete message authentication using the same basic reliable and authenticated communication algorithm. Therefore, at least basic security services are embodied into the area of low resource mobile ad hoc networks. The complete algorithm is described in the following.

IV. RAC ALGORITHM

The main goal of this section is to provide a detailed view on the RAC (reliable and authenticated communication) algorithm. First, a schematic overview is given as shown in Fig 3. All the involved objects and data flows are depicted. In short, the algorithm works as follows.

For each message m to be sent from A to B, a message digest (hash value, h) is computed and stored in a local database. Additionally, it is padded to the message before actually transmitting it. In reliable mode the message m is stored together with h. Finally, a timestamp is stored in conjunction with h for maintaining retransmissions or recognizing lost data. At the receiver B, the hash value is cut from the message and stored in a local database that is used for combining multiple acknowledges into a single acknowledge message in order to reduce the number of acknowledge messages from B to A. If the number of acknowledges exceeds the maximum number of acknowledges per message or if the tACK timeout arrives, an acknowledge message is sent to A containing all received hash values. If A receives such an acknowledgment, it removes all included hash values from its database leaving only unacknowledged values remaining. Additionally, it maintains the tRET timeout which is used to inform the application about lost messages, i.e. messages unacknowledged for tRET, and to initiate retransmissions.

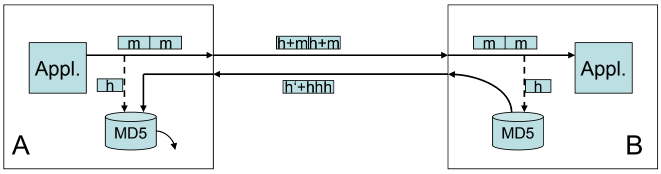

Fig 3. Involved objects and data flow of the RAC algorithm

The complete algorithm is provided next which allows a direct implementation in a lab environment or in a simulation:

Sender

Message sending

```
for each message m to be sent
do
        calculate $h_m$=h([K],m)
        get current time $t_m$
        store ($t_m$,$h_m$,[m]) in database
        transmit (m,$h_m$)
done
```

Periodically check for lost messages

```
get current time $t_c$
for each ($t_m$,$h_m$,[m]) in database
do
  if($t_m$+tRET>$t_c$)
  do
    retransmit m
    notice lost message
done
```

Receiver

Message receiving

```
for each received message (m,$h_m$)
do
        verify hm
    -> check data integrity /
        message authentication
        get current time $t_m$
        store ($t_m$,$h_m$) in database
done
```

Periodically send acknowledgments

```
get current time $t_c$
for each ($t_m$,$h_m$) in database
do
  if($t_m$+tACK>$t_c$)
  do
    acknowledge all $h_m$ in database
    finish
  done
done
```

The operation modes of RAC are shown in Fig 4. It allows reliable and semi-reliable transmissions (with/without retransmissions) as well as integrity checked and authenticated messages (with/without shared secrets).

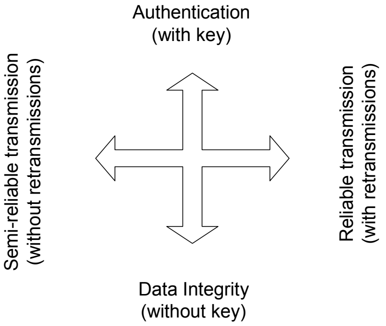

Fig 4. Two dimensional usage options of the hash-based mechanism

V. Simulative Analysis

For a detailed analysis of the parameters used in the RAC algorithm as well as to provide an overview to the message overhead caused by the mechanism, we realized the algorithm in a simulation. The individual results are depicted in the following. All the results were created using a model implemented in AnyLogic, a simulation environment for discrete simulations. The different measurements were taken from multiple runs of the same simulation with different parameters.

First, the required size of the retransmission buffer was analyzed. This buffer stores the hash values of each transmitted message, the timestamp, and, probably, the message itself. We analyzed the behavior of the global system by modifying, first, the loss ratio ρ shown in Fig 5, secondly, the tRET/tACK ratio shown in Fig 6, and finally, the message rate λ shown in Fig 7.

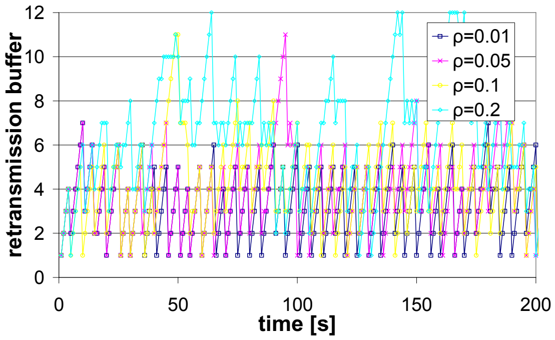

Fig 5. Analysis of the size of the retransmission buffer. tRET/tACK=10s/5s, λ_1=1, variable ρ

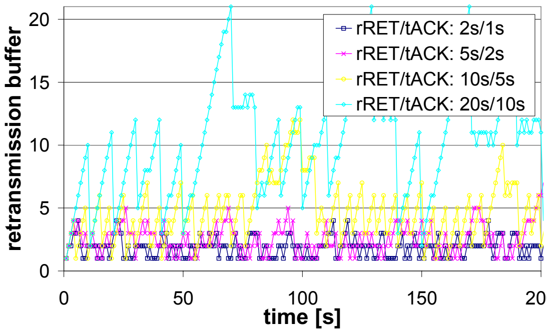

Fig 6. Analysis of the size of the retransmission buffer. λ_1=1, ρ=0.1, variable tRET/tACK

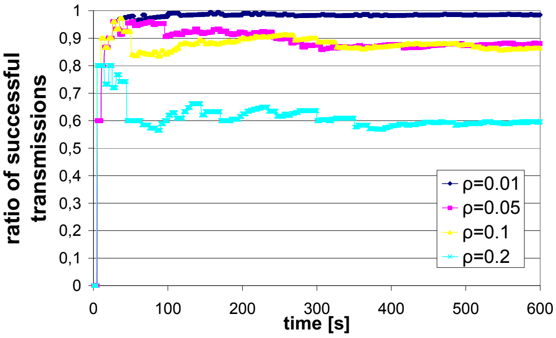

Fig 8. Analysis of the overall loss ratio as recognized at the sender of the messages. tRET/tACK=10s/5s, λ_1=1, variable ρ

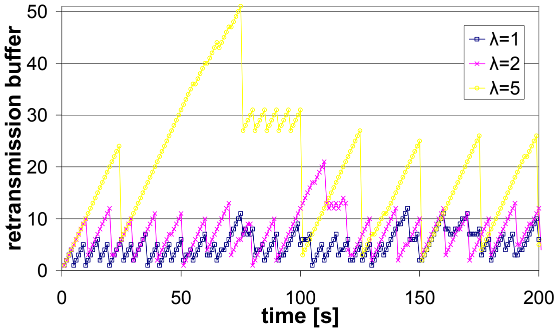

Fig 7. Analysis of the size of the retransmission buffer: tRET/tACK=10s/5s, ρ=0.1, variable λ_1

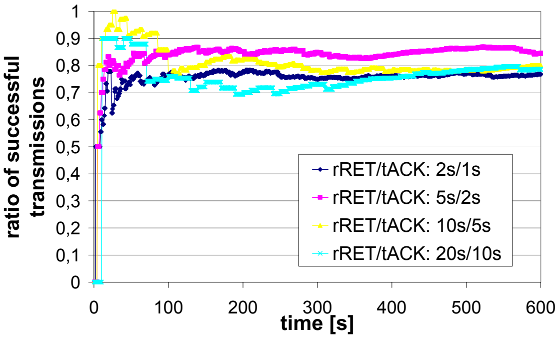

Fig 9. Analysis of the overall loss ratio as recognized at the sender of the messages. λ_1=1, ρ=0.1, variable tRET/tACK

Discussion of the analysis of retransmission buffer size: obviously, the size of the buffer for hash values, timestamps, and (possibly) the messages themselves depends mainly on the retransmission timeout tRET and the message rate λ_1. Essentially, the maximum can be specified by the product of tRET and λ_1. For the implementation and deployment of the algorithm, the mean and deviation depending on the parameters of the algorithm are of interest. As shown in Fig 5, it can be seen that the size can be adapted based on the currently assumed loss ratio.

Secondly, the loss ratio as estimated at the sender of the message stream was analyzed. This overall loss ratio includes lost messages sent from A to B as well as messages assumed as lost due to lost acknowledgments. Again, we analyzed the behavior of the global system by modifying, first, the loss ratio ρ shown in Fig 8, secondly, the tRET/tACK ratio shown in Fig 9, and finally, the message rate λ shown in Fig 10.

Discussion of the ratio of successful transmissions between A and B: as shown in Fig 8, the loss ratio on the link between A and B ρ (in the simulations we configured $\rho = \rho_1 = \rho_2$) is the main factor for the amount of messages from which the sender assumes that they have been received successfully. The tRET/tACK pair does not induce any deviation on the loss ratio. Interestingly, the simulation shows that the message rate seems to be an important factor as shown in Fig 10. This is the result of the algorithm running on the receiver. It accumulates as much has values as possible before sending an acknowledgment. Thus, the number of acknowledge messages stays the same while changing the message rate but the number of acknowledged messages per acknowledge increases and, therefore, the number of lost acknowledged hash values reduces.

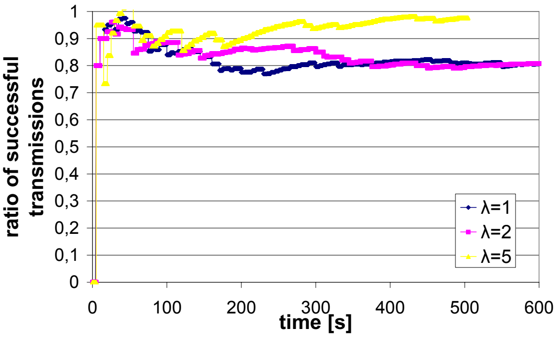

Fig 10. Analysis of the overall loss ratio as recognized at the sender of the messages. tRET/tACK=10s/5s, ρ=0.1, variable λ_1

VI. Conclusions

In conclusion it can be said that we were able to construct a novel communication methodology for (semi-)reliable and authenticated transmission of messages in wireless ad hoc networks. Unreliable data paths and time variations of the reliability affect the traditional network protocols and lead to unnecessarily high transmission overhead or impractical communication. In this paper, we presented a methodology to perform (semi-)reliable transmissions. Additionally, the same mechanism is employed for data integrity checks and message authentication. Especially in low-resource WSNs, this allows a more efficient utilization of available resources and leads to an improved quality of the global system.

We see the primary application for partial reliability, i.e. the application needs to be informed about loss ratio or about specific lost messages. The simulation results proved the applicability of the proposed algorithm and allow an on-time adaptation of the individual parameters depending on the current characteristics of the communication pathways. Additionally, the security in ad hoc networks, especially in WSNs and WPANs is a current research issue. This security service is provided with low (or even zero) overhead because it is a main functionality of the transmission methodology. The adaptability and self-organization properties have been the major objectives during the development.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393-422, 2002.

[2] D. Culler, D. Estrin, and M. B. Srivastava, "Overview of Sensor Networks," *Computer*, vol. 37 (8), pp. 41-49, August 2004.

[3] F. Dressler, "Locality Driven Congestion Control in Self-Organizing Wireless Sensor Networks," Proceedings of 3rd International Conference on Pervasive Computing (Pervasive 2005): Workshop Software Architectures for Self-Organization: Beyond Ad-Hoc Networking (SASO'05), 2005. (submitted)

[4] F. Dressler, B. Krüger, G. Fuchs, and R. German, "Self-Organization in Sensor Networks using Bio-Inspired Mechanisms," Proceedings of 18th GI/ITG/ACM International Conference on Architecture of Computing Systems - System Aspects in Organic and Pervasive Computing (ARCS'05): Workshop Self-Organization and Emergence, Innsbruck, Austria, March 2005.

[5] A. A. Hanbali, E. Altman, and P. Nain, "A survey of TCP over Ad Hoc Networks," Inria Research, Report RR-5182, May 2004.

[6] V. Handziski, A. Köpke, H. Karl, C. Frank, and W. Drytkiewicz, "Improving the Energy Efficiency of Directed Diffusion Using Pervasive Clustering," Proceedings of 1st European Workshop in Wireless Sensor Networks (EWSN), Berlin, Germany, January 2004, pp. 172-187.

[7] G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Seattle, Wasington, USA, 1999, pp. 219-230.

[8] K. Obraczka, K. Viswanath, and G. Tsudik, "Flooding for Reliable Multicast in Multi-Hop Ad Hoc Networks," *Wireless Networks*, vol. 7 (6), pp. 627-634, November 2001.

[9] J. Postel, "Transmission Control Protocol," RFC 793, September 1981.

[10] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321, April 1992.

[11] R. Steward, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream Control Transmission Protocol," RFC 2960, October 2000.

[12] R. Steward, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad, "Stream Control Transmission Protocol (SCTP) - Partial Reliability Extension," RFC 3758, May 2004.

[13] W. Ye, J. Heidemann, and D. Estrin, "A Flexible and Reliable Radio Communication Stack on Motes," USC Information Sciences Intitute, Technical Report ISI-TR-565, September 2002.