Authenticated Reliable and Semi-reliable Communication in Wireless Sensor Networks

Falko Dressler
Autonomic Networking Group, Department of Computer Science, University of Erlangen-Nuremberg
Martensstr. 3, 91058 Erlangen, Germany (Email: dressler@informatik.uni-erlangen.de)

(Received Aug. 4, 2006; revised and accepted Oct. 3, 2006 & Nov. 8, 2006)

Abstract

Secure communication in wireless ad hoc sensor networks is a major research concern in the networking community. Especially the few available resources in terms of processing power and available memory are challenging factors. Additionally, a typical characteristic of wireless ad hoc sensor networks is the error-proneness and, therefore, the unreliability of communication paths. 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 in combination with security services. Thus, the proposed methodology allows to ensure data integrity and message authentication. The main aspects during the development were the limitations of typical sensor nodes used in ad hoc networks in terms of available 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 applicable to wireless sensor networks was created providing all the mentioned communication goals.

Keywords: cross-layer design, message authentication, network security, reliable communication, wireless sensor networks

1 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 [21]. 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 sensor networks and body area networks require mechanisms and technologies for achieving optimum data rates while addressing issues such as power-consumption and reorganization during the data transfer [1, 3, 10].

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, [17]). 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, [19]) and its partial reliable extension [20]. Such protocols fail in large-scale ad hoc networks due to the immense resource requirements in terms of memory to store state information as well as complete messages and in terms of computational complexity. Other approaches are required that are primarily focused on ad hoc networks ad their capabilities. One example is the flooding approach by Obraczka et al. [13] for providing reliable group communication in multi-hop ad hoc networks.

Questioning the requirements on communication system in today's ad hoc networks, we find similar characteristics in wireless sensor networks, pervasive computing environments, and WPANs (Wireless Personal Area Networks). Such networks are built of small entities with little available resources in term of processing power, memory, and energy. Therefore, the quite big communication protocols developed for the Internet are difficult or not applicable.

Focusing on ad hoc networks, several proposals have been published to compensate the characteristics of TCP to determine the network congestion by measuring the packet loss ratio. Wireless communication and mobility often lead to single link failures while this is not a sign of network congestion. Several surveys of TCP performance in ad hoc and mobile ad hoc networks are available, which summarize these approaches [2, 9, 11].

Secure communications in wireless sensor networks is still an underestimated research area [4]. Classical approaches are not applicable due to the low resource capabilities and novel approaches as described in [16] are still work in progress. For example, Perrig et al. developed a security architecture called SPINS that focuses on several security problems in sensor networks [16]. Nevertheless, such approaches do not focus on reliable communication.

Instead, often such reliability is presumed to be provided by the network. A prerequisite for any security solution is an efficient key management. Several groups already proposed solutions that can be applied [8, 12, 14]. For our work, we assume that such a key management infrastructure is available.

In this paper, we present a methodology that allows to offer security services such as data integrity checks, message authentication, and address-verification to wireless sensor networks. 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 symmetric encryption algorithm. In addition, the approach provides 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. The proposed methodology allows a (online) tuning of all parameters to ensure an optimal utilization of the available resources at each communicating node. We see the main advantage of this approach in the combination of security services, i.e. message authentication, and (semi-)reliable communication resulting in a very low resource consumption and, therefore, in its applicability for wireless sensor networks.

The contributions of this paper can be summarizes as follows. First, we discuss the need for simultaneously available secure and reliable communication. We propose a cross-layer design that uses well-known techniques such as message authentication to address the objectives. As we rely on symmetric cryptographic solutions, the performance of single operations will be adequate [15]. The flexible design of our approach allows to incorporate performance enhancements developed for TCP in ad hoc networks such as adaptive slow start and enhanced congestion control [9, 11]. Thus, our approach will perform similar to these solutions.

The rest of the paper is organized as follows: in Section 2, the basic objectives and some motivation for this work is provided. In Section 3, the methodology is presented and the resulting RAC (reliable authenticated communication) algorithm is depicted in Section 4. The parameters of the algorithm and the resource requirements are analyzed in a comparative simulation as shown in Section 5. Finally, some conclusions summarize the paper.

2 Motivation and Objectives

Among others, the following issues have to be addressed in the research area of mobile wireless sensor networks: data storage, data aggregation, and communication between individual nodes for data exchange and management tasks. Typically, sensor networks are composed of multiple, independent, autonomously working nodes. These individual entities form a self-organizing compound which is able to solve required tasks described at a higher level. A typical sensor network is shown in Figure 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.

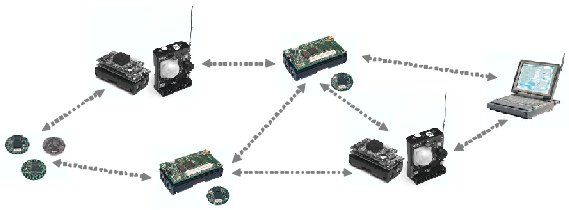

Figure 1: A typical wireless, multi-hop ad hoc sensor network consisting of heterogeneous mobile nodes

As already mentioned, we focus on communication aspects in wireless ad hoc sensor networks, i.e. in the internode 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, proactive, 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. [7, 6].

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 secured communication in terms of ensuring the integrity of sent messages and message authentication. Additionally, reliable and semi-reliable communications 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 of unnecessarily high message rates. Congestion in ad hoc networks is an important issue not yet solved by means of Internet congestion control mechanisms and novel approaches are work in progress [2, 5, 9, 11].

3 Reliable and Authenticated Communication in Ad Hoc Networks

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

3.1 Model and Assumptions

The underlying model for the proposed methodology is shown in Figure 2. Two communication end points (A and B) are involved, but only the unidirectional message transmission from A to B is shown.

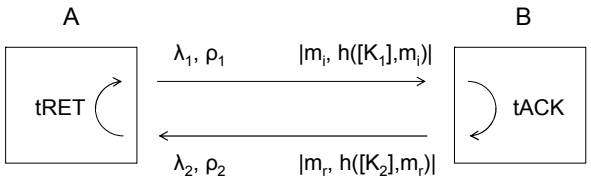

Figure 2: Model for reliable authenticated communication

At A, messages 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 hash value 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, i.e. multiple hash values in a single message. 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:

$$\lambda_1\prime = \lambda_1 * (1 - \rho_1).$$

Number of received messages in the acknowledge window:

$$|m_{recv}| = \lambda_1\prime * tACK.$$

Acknowledge rate using amax (maximum number of acknowledges per message):

$$\lambda_2 = (|m_{recv}| * a_{max})/tACK.$$

3.2 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. is 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 a few advantages compared to sequence numbers. First, there is no possibility to overflow the number space. Also, 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 requirement 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 $(x, x\prime)$ with $x \neq x\prime$ such that $h(x) = h(x\prime)$, 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 [18] due to its high performance [15], but any other hash function which fulfils the mentioned requirements can be used as well.

3.3 Data Integrity 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 $x\prime$ with $x \neq x\prime$ such that $h(x) = h(x\prime)$. 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 next section.

4 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 Figure 3. All the involved objects and data flows are depicted. Shortly, the algorithm works as follows.

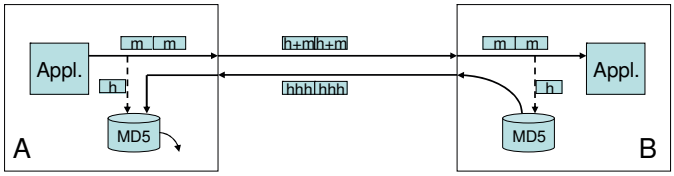

Figure 3: Involved objects and data flow of the RAC algorithm

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.

The operation modes of RAC are shown in Figure 4. It allows reliable and semi-reliable transmissions (with/without retransmissions) as well as integrity checked and authenticated messages (with/without shared secrets). The complete algorithm as provided in Figures 5 and 6 allows a direct implementation in a lab environment or in a simulation. Figure 5 depicts the behavior of the sender. For each message, it calculates the hash value h_m, collects the current timestamp t_m, and stores a tuple containing both values in the local database. Finally, the message can be sent. Periodically, the database is checked whether there are any messages that have not been acknowledged in the time interval tRET (i.e. $t_m + tRET > t_c$). Such messages are retransmitted. Accordingly, Figure 6 depicts the receiver behavior. After receiving a message, first, the hash value is verified. If the message was not altered during the transmission, the current time tm and the hash value are stored in the local database of the receiver. Periodically (tACK), an acknowledge message is sent that contains all hash values received so far.

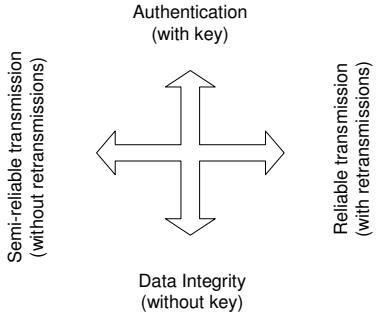

Figure 4: Two dimensional usage options of the hash-based mechanism

Some issues need to be discussed at this place: storage requirements, performance estimation, and communication overhead - a more detailed impression of the protocol behavior is provided in the following section.

Storage requirements: for each message that was transmitted, at least a tuple must be stored in a local database. In a typical implementation, this will require about 20 byte per packet. Assuming to have several KByte available, this seems to be feasible. In contrast, in reliable mode, the messages themselves need to be stored as well. Nevertheless, this cannot be prevented and this is also the case for any other reliable transport protocol.

Performance estimation: in addition to other reliable communication protocols, two hash operations need to be performed for each transmitted packet (and again for each retransmission). As shown in [15], such operations are not negligible. Thus, an additional end-to-end delay will be induced of two times a MD5 operation. Message authentication codes can be computed using the same hash functions (like MD5) but require typically three separate runs. Thus in authenticated mode, the additional end-to-end delay will be six times a MD5 operation.

Communication overhead: a hash value is appended to each message. Thus, about 16 byte (for MD5) per message must be transmitted. This might be a non-negligible overhead in particular scenarios where messages of only a few bytes need to be transmitted. Considering longer messages, the overhead obviously reduces. Compared to other proposals which rely on a sequence number only, the overhead is definitely higher. Nevertheless, the hash already contains an integrity check that is usually provided by a CRC that again will consume several bytes. Thus, the overhead of RAC compared to other protocols providing reliable communication is minimized - if not even smaller. If used in authentication mode, the overhead compared to other solutions that provide the mechanisms

```
Message sending
for each message m to be sent
do
  calculate hm=h([K],m)
  get current time tm
  store (tm,hm,[m]) in database
  transmit (m,hm)
done
```

```
Periodically check for lost messages
get current time tc
for each (tm,hm,[m]) in database
do
  if(tm+tRET>tc)
  do
    retransmit m
    notice lost message
done
```

Figure 5: RAC implementation for the sender

```
Periodically send acknowledgments
get current time tc
for each (tm,hm) in database
do
  if(tm+tACK>tc)
  do
    acknowledge all hm in
  database
    finish
  done
done
```

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

Figure 6: RAC implementation for the receiver

at different layers is dramatically reduced.

Finally, the mis-ordering issue needs to be discussed. RAC is intended to be used as a transport layer solution. Thus, we assume to have a network layer protocol responsible for message fragmentation. This network layer functionality will also provide sufficient re-ordering capabilities.

5 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 implemented 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 Figure 7, secondly, the tRET/tACK ratio shown in Figure 8, and finally, the message rate λ shown in Figure 9.

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

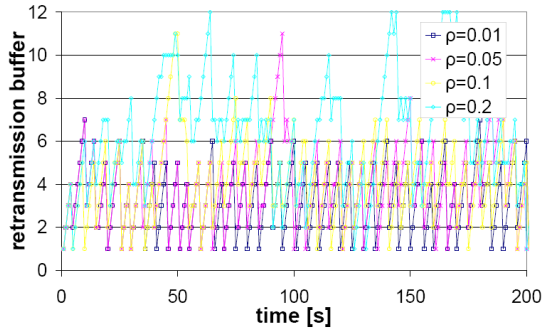

Figure 7: Analysis of the size of the retransmission buffer: tRET/tACK=10s/5s, $\lambda_1 = 1$, variable ρ

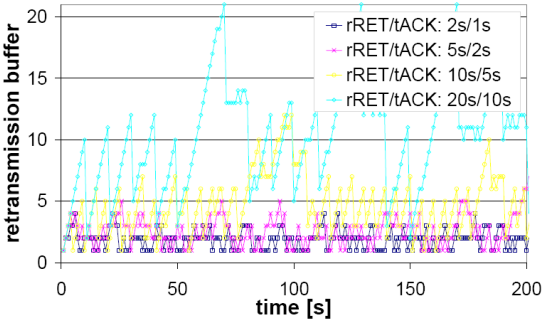

Figure 8: Analysis of the size of the retransmission buffer: $\lambda_1 = 1$, $\rho = 0.1$, variable tRET/tACK

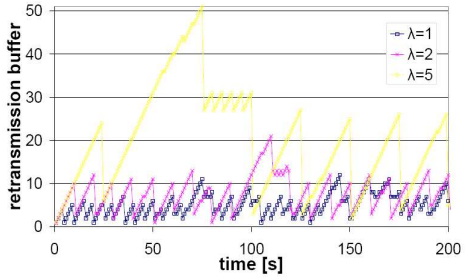

Figure 9: Analysis of the size of the retransmission buffer: tRET/tACK=10s/5s, $\rho = 0.1$, variable λ_1

depending on the parameters of the algorithm is of interest. As show in Figure 8, it can be seen that the size can be adapted based on the currently assumed loss ratio.

Secondly, the loss ratio as analyzed 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 Figure 10, secondly, the tRET/tACK ratio shown in Figure 11, and finally, the message rate λ shown in Figure 12.

Discussion of the ratio of successful transmissions between A and B: as shown in Figure 10, 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

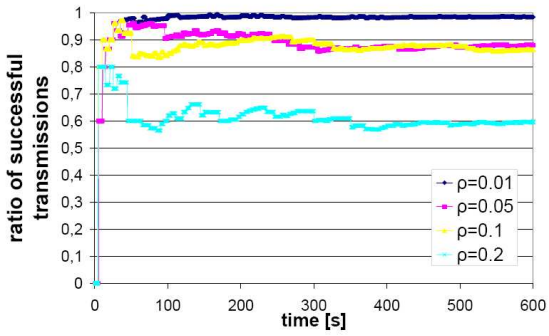

Figure 10: Analysis of the overall loss ratio as recognized at the sender of the messages: RET/tACK=10s/5s, $\lambda_1 = 1$, variable ρ

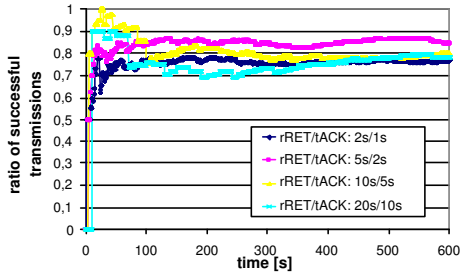

Figure 11: Analysis of the overall loss ratio as recognized at the sender of the messages: $\lambda_1 = 1$, $\rho = 0.1$, variable tRET/tACK

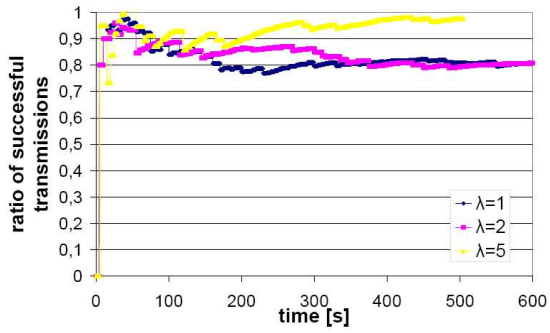

Figure 12: Analysis of the overall loss ratio as recognized at the sender of the messages: tRET/tACK=10s/5s, $\rho = 0.1$, variable λ_1

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 Figure 12. 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.

6 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. Here, we presented an according functionality to perform (semi-)reliable transmissions. Additionally, the same mechanism is employed for checking data integrity or even to allow message authentication. Especially in low-resource sensor networks, this allows a far 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 a 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 sensor networks and WPANs is a current research issue. We provided an algorithm for providing message authentication, a main security objective in typical applications of sensor networks. This security service is provided with low (or even zero) overhead because it is a main functionality of the transmission methodology.

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] V. Anantharaman, S.-J. Park, K. Sundaresan, and R. Sivakumar, "TCP performance over mobile ad hoc networks: a quantitative study," *Wireless Communications and Mobile Computing*, vol. 4, no. 2, pp. 203-222, Mar. 2004.

[3] D. Culler, D. Estrin, and M. B. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41-49, Aug. 2004.

[4] D. Djenouri and L. Khelladi, "A Survey of security issues in mobile ad hoc and sensor networks," *IEEE Communication Surveys and Tutorials*, vol. 7, no. 4, pp. 2-28, Dec. 2005.

[5] F. Dressler, "Locality driven congestion control in self-organizing wireless sensor networks," in *Proceedings of 3rd International Conference on Pervasive Computing (Pervasive'05): Workshop Software Architectures for Self-Organization: Beyond Ad-Hoc Networking (SASO'05)*, 2005.

[6] F. Dressler, *Self-Organization In Ad Hoc Networks: Overview And Classification*, University of Erlangen, Dept. of Computer Science 7, Technical Report 02/06, Mar. 2006.

[7] F. Dressler, B. Krüger, G. Fuchs, and R. German, "Self-organization in sensor networks using bio-inspired mechanisms," in *Proceedings of 18th ACM/GI/ITG International Conference on Architecture of Computing Systems - System Aspects in Organic and Pervasive Computing (ARCS'05): Workshop Self-Organization and Emergence*, pp. 139-144, Innsbruck, Austria, Mar. 2005.

[8] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of 9th ACM Conference on Computer and Communication Security (ACM CCS'02)*, pp. 41-47, Washington, DC, Nov. 2002.

[9] A. A. Hanbali, E. Altman, and P. Nain, *A Survey Of TCP Over Ad Hoc Networks*, Inria Research, Report RR-5182, May 2004.

[10] V. Handziski, A. Köpke, H. Karl, C. Frank, and W. Drytkiewicz, "Improving the energy efficiency of directed diffusion using pervasive clustering," in *Proceedings of 1st European Workshop in Wireless Sensor Networks (EWSN'04)*, pp. 172-187, Berlin, Germany, Jan. 2004.

[11] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 219-230, Seattle, Wasington, USA, 1999.

[12] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of 10th ACM Conference on Computer and Communications Security*, pp. 52-61, Washington D.C., USA, Oct. 2003.

[13] K. Obraczka, K. Viswanath, and G. Tsudik, "Flooding for reliable multicast in multi-hop ad hoc networks," *Wireless Networks*, vol. 7, no. 6, pp. 627-634, Nov. 2001.

[14] J. Park, Z. Kim, and K. Kim, "State-based key management scheme for wireless sensor networks," in *Proceedings of 2nd IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS'05): International Workshop on Wireless and Sensor Networks Security (WSNS'05)*, Washington, D.C., USA, Nov. 2005.

[15] M. Passing and F. Dressler, "Experimental performance evaluation of cryptographic algorithms on sensor nodes," in *Proceedings of 3rd IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS'06): 2nd IEEE International Workshop on Wireless and Sensor Networks Security (WSNS'06)*, pp. 882-887, Vancouver, Canada, Oct. 2006.

[16] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "Spins: security protocols for sensor networks," *Wireless Networks*, vol. 8,no. 5, pp. 521-534, Sep. 2002.

[17] J. Postel, *Transmission Control Protocol*, RFC 793, Sep. 1981.

[18] R. Rivest, *The MD5 Message-Digest Algorithm*, RFC 1321, Apr. 1992.

[19] 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, Oct. 2000.

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

[21] W. Ye, J. Heidemann, and D. Estrin, "*A Flexible And Reliable Radio Communication Stack On Motes*," Technical Report ISI-TR-565, USC Information Sciences Intitute, Sep. 2002.

Falko Dressler Dr. Dressler is an assistant professor leading the Autonomic Networking Group at the Department of Computer Sciences, University of Erlangen-Nuremberg. He teaches on self-organizing sensor/actuator networks, network security, and communication systems.

Dr. Dressler received his M.Sc. and Ph.D. degree from the Dept. of Computer Sciences, University of Erlangen in 1998 and 2003, respectively. From 1998 to 2003 he worked at the Regional Computing Center at the University of Erlangen as a research assistant. In 2003, Dr. Dressler joined the Networking Group (Chair for Computer Networks and Internet) of Prof. Dr. Georg Carle at the Wilhelm-Schickard-Institute for Computer Science, University of Tuebingen as an assistant professor. In 2004, he joined the Computer Networks and Communication Systems Group of Prof. Dr. Reinhard German at the Department of Computer Sciences, University of Erlangen-Nuremberg.

Dr. Dressler co-authored more than 60 reviewed research papers. He was co-chair and PC member for various international conferences (ACM, IEEE, GI, ITG). He is a member of ACM, IEEE, IEEE Computer Society, and GI (Gesellschaft fr Informatik). Dr. Dressler is actively participating in several working groups of the IETF. His research activities are focused on (but not limited to) Autonomic Networking addressing issues in Wireless Ad Hoc and Sensor Networks, Self-Organization, Bio-inspired Mechanisms, Network Security, Network Monitoring and Measurements, and Robotics.