Security Aspects in the Sami Network Connectivity Project

Christian Rohner, Uppsala University

Abstract

The Sami Network Connectivity project is an extreme example of an intermittently connected network where lack of connectivity to other devices is default and latency is in the order of hours or days. Intermittent connectivity challenges security protocols. This document discusses the security challenges in the Sami Network Connectivity project based on the three aspects of network security, routing security, and transport and application layer security. Some solutions are proposed, but as the problems are complex some questions are left open for now.

1 Introduction

One goal of the Sami Network Connecitivity (SNC) project is to provide basic Internet access in areas where traditional communication infrastructure is missing. The idea is to use the mobility of persons, snowmobile, helicopters, etc. to disseminate data towards the destination.

In traditional way of living semi nomadic reindeer herding has great importance. The herders and their families are nomads in the mountains, living in remote sites close to their herds. Internet connectivity should improve family's possibilities to remain together in the remote site even under school periods, and to find new business opportunities on the Internet. Timeliness is not as important as eventual delivery.

The vision of the SNC project is to provide a standard networking infrastructure where sami can connect their laptop or PDA, and that mediates the connectivity through a delay tolerant network (DTN) consisting of mobile nodes opportunistically forwarding messages. The topology of the SNC project is illustrated in Figure 1. The network consists of different types of entities:

Vanilla Laptop An normal laptop (or PDA) with a wireless interface but without DTN specific software. This is the main device of a Sami. The laptop can connect to a site gateway via wireless interface.

Site Gateway A wireless base station with a DHCP server that allows a vanilla Laptop to obtain an IP address. A site gateway also offers application layer interfaces and translates them to DTN messages (or bundles). The gateway runs the Prophet routing protocol and represents the vanilla laptops connected to it.

DTN node A node has a wireless network interface, memory, and runs the Prophet routing protocol. Both, end-user computers and special devices only used to forward messages are considered to be nodes.

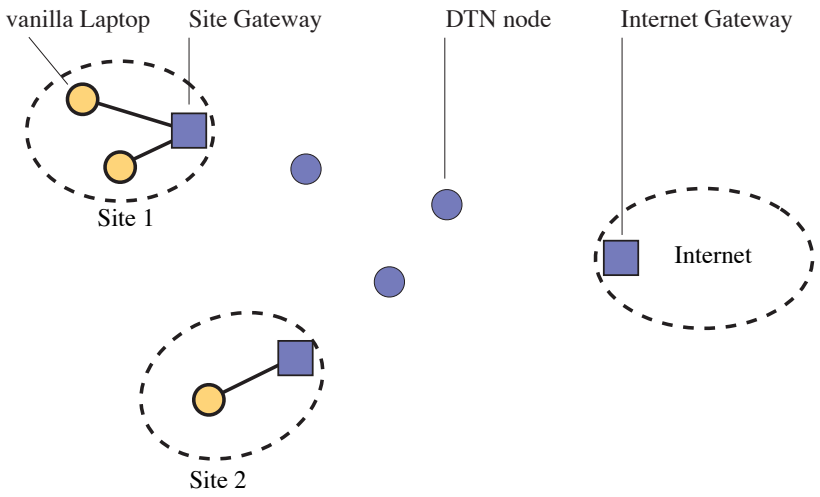


Figure 1: Network topology with two remote sites.

Internet Gateway Similar to a site gateway, but provides connectivity to the Internet. The important functionality of the Internet gateway is the translation between DTN messages and the Internet.

For the sake of simplicity and without loss of generality we can consider a DTN node as a site with one node, that is, a laptop with DTN aware gateway software.

1.1 Security Aspects

The main difference between traditional networks like the Internet, and delay tolerant networks is latency; all communication is asynchronous. As a consequence, protocol negotiations and access to supportive infrastructure like certificate authorities or policy servers take an unreasonably long time and have to be avoided. The challenge in security for delay tolerant networks is therefore to support off-line operation, based on careful security bootstrapping.

In the context of SNC: - assume one administrative domain. Certificates. - etc

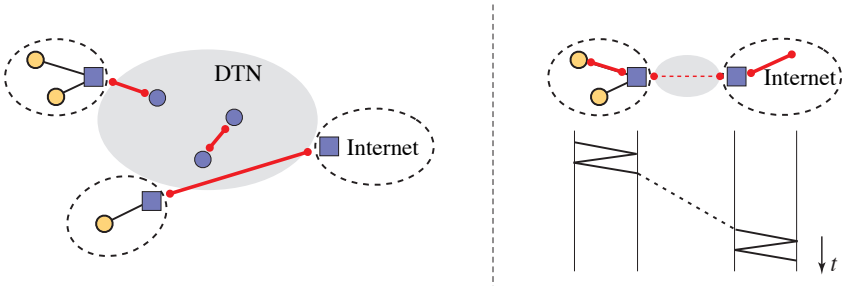
Looking at the different communication layers:

Network Security Messages in the SNC network are forwarded from node to node, while the involved nodes might not be trusted. Even if transported by devices of friends, it might be inconvenient to have them reading the messages.

Goal: Confidential message delivery through the DTN.

Verifiable access to the network prevents network overload or denial-of-service (DoS) attacks. In a first phase of SNC we do not provide such mechanisms, but cover some aspects in routing security.

Secure Routing The Prophet routing protocol depends on input values for the calculation of probabilities to deliver a message to a particular destination.



a) Network Security b) Transport Layer Security

Figure 2: Two security aspects in the Sami Network Connectivity Project.

Delivery acknowledgements support queue management. Attacks such as wrong input values or faked delivery acknowledgements degrade the efficiency of Prophet.

Goal: Investigation of the resilience of Prophet to attacks and the design of security mechanisms for protection.

Transport and Application Layer Security All security mechanisms that involve end-to-end protocol negotiation or several rounds of computations are not suited for SNC because of the latency. Problems appear in particular for applications not designed for asynchronous connectivity, as for example, Web access.

Goal: Termination of transport layer protocols (SSL/TLS, etc.) at the gateway and applications supporting direct secure transactions.

Figure 2 illustrates network security and transport layer security. In the following sections we discuss these three security aspects in more detail. Section ...

2 Network Security

The goal of confidential delivery of messages requires end-to-end encryption. Hop-by-Hop security would allow forwarding nodes to decrypt messages and read them. As a consequence, nodes have to have an authentic key of the destination. Note that if the destination of a message is outside the Sami Connectivity Network, the gateway node is considered as the security end-point for that message.

Another consequence of end-to-end encryption is that we cannot use shared group keys throughout the system. Every node must maintain a list of destination keys, and a key distribution mechanism must be in place. From a cryptographic point of view we have several possibilities to implement that. The next subsection presents a first proposal to build a public key infrastructure.

2.1 A Key Distribution Scenario

Every node (or user on a node) generates a public/secret-key-pair. Upon initialization of a new node with another node of the network, the public key of the

gateway node and optionally some other public node keys is transferred to the new node. These keys can be used to encrypt messages for the corresponding nodes. As a first action, an initialized node will send a message with its public key to the gateway node. If a node wants to send a message to a destination whose key is not known, it might relay the message through another node (e.g., the gateway node) whose key is known and who is assumed to know the key of the destination node. Such a relay node will then send a copy of the destination code to the message source. Over time, a node will therefore build up a list of keys of nodes it uses to communicate with.

Optionally, keys might be exchanged together with the destination probabilities whenever nodes encounter. However, it is important to only use keys which are believed to be authentic. More specific comments on that issue are to be added here...

2.2 Network Access

At the moment it is not a stated goal to restrict access to the Sami Connectivity Network. Every node that is willing to forward messages increases the chances to get own messages to the destination. As an incentive, such nodes can be allowed to send their own messages.

However, it is not clear what mechanisms that have to be used if we want to protect against the attacks against delivery probabilities and delivery acknowledgments.

3 Routing Security

A node in Prophet maintains a probabilistic metric that reflects the probability that another node will be encountered. The metric is updated upon three criterias:

- Update the probabilistic metric of a node upon encounter
  $P_{(a,b)} = P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}}) \times P_{init}$
- The probabilistic metric has a transitive property
  $P_{(a,c)} = P_{(a,c)_{old}} + (1 - P_{(a,c)_{old}}) \times P_{(a,b)} \times P_{(b,c)} \times \beta$
- Decrease the value as the metric ages
  $P_{(a,b)} = P_{(a,b)_{old}} \times \gamma^k$

A message is forwarded to all encountered nodes with higher probability than yourself for the given destination. Upon reception of an delivery acknowledge `ACK(m)` for a message m, a copy of that message in the own buffer is erased. The message buffer at a node is FIFO, that is, if the buffer gets full, the oldest message in the buffer will be erased.

Attacks specific to Prophet can try to manipulate the routing metric to prevent message forwarding, to fill the buffer of other nodes with irrelevant messages in order to erase old (relevant) messges, and inject acknowledgments to selectively erase messages. In the remainder of this section we will have a closer look to these two types of attacks. Attacking nodes are not expected to participate in the protocol, that is, we do not expect them to forwarding messages.

3.1 Attack on Delivery Probabilities

Node a forwards a message for node b to a node c if c's delivery probability is higher than the own. At the same time, the delivery probability $P_{(a,b)}$ is updated following the transitivity property. Having a high own delivery probability means that chances are small to encounter another node with higher delivery probability, and therefore it is less probable that a message gets forwarded to other nodes. On the other side, having a low delivery probability increases the chances to encounter nodes with higher delivery probabilities.

We look now at two different attack scenarios, namely trying to increase the delivery probability to avoid forwarding and trying to fill up buffers to erase relevant messages.

Avoid Forwarding Message forwarding can be prevented by trying to increase the delivery probability of static nodes that are not likely to encounter the destination. Such nodes will then no longer forward messages because of the own high delivery probability. Delivery probabilities of other nodes can be increased via the transitivity property by *claiming a high delivery probability*. Claiming a low delivery probability would not affect the probability metric of other nodes. The other possibility to increase an other node's delivery probability for a specific destination node is to *claim that destination node's identity*. Besides of increasing the delivery probability, this approach also causes all messages for that destination to be erased on the other node because they are believed to be delivered.

Filling Buffers The buffers of other nodes can by filled up – and thus old messages deleted – by *distributing messages* for unknown destinations. As all nodes will have low delivery probability for that messages, they will spread in the network and use valuable buffer space, possibly causing other messages to be erased.

The described attacks to Prophet are certainly more effective in large networks with a large number of hops between the source of a message and its destination. It is not clear if the attacks are effective when the number of hops is small (e.g., 2 hops) as we can expect in the sami connectivity network. We have to discuss this question and then decide if we want to apply some protection for some of the vulnerabilities.

3.2 Secure Delivery Acknowledgments

Delivery acknowledgments are used to free the network (i.e., buffers) from messages that are delivered. By *spreading acknowledgments for a message that is not yet delivered* causes other nodes to erase that specific message, and thus reducing the probability that this message will get to the destination.

In order to avoid attacks on the delivery acknowledge to selectively erase messages propagating in the network, it must be possible to verify the authenticity of an acknowledgment. An acknowledgment has to be generated at the destination of a message. Because the destination might be unknown for many of the transporting nodes, signatures and message authentication codes are not suitable. We propose to use a threshold secret sharing scheme (e.g., Shamir's secret sharing scheme) where one part of the shared secret is publicly distributed

together with the message, the second part is encrypted together with the message such that only the destination can read it. The destination attaches this second part in plaintext to the acknowledgment, and every node having both parts of the shared secret can reconstruct the secret to then decrypt an authenticator (e.g., a message authentication code) that was also sent together with the original message.

4 Transport and Application Layer Security

Because of the long latency, protocols with handshake or protocol negotiations are not suitable over DTN networks in an end-to-end fashion. The common approach (e.g., for TCP) is to terminate the protocols at the gateways, and re-establish the protocol at the other end of the DTN network (see Figure 2). The bundling has to be done for every protocol supported.

This way of breaking up a protocol between two end-nodes A and B can be modelled as a sequence of communication channels:

$$A \longleftrightarrow T_1 \longleftrightarrow T_2 \longleftrightarrow B \tag{1}$$

where T_1 and T_1 are the Internet and site gateways respectively. From security theory we know that a secure channel between node A and B can only be achieved if and only if every link in between has to be secure, and both, A and B trust T_1 and T_2 to only forward authentic information.

Even though individual secure links seem to be achievable, an arbitrary node on the Internet cannot be expected to trust the SNC gateways T_1 and T_2. We can therefore not achieve authentic message delivery between a node in the SNC network and a node on the Internet.

4.1 Secure Transactions

Secure transactions, for example in Internet banking, as used in the traditional Internet are not suitable in a DTN network because they often involve several rounds of interactions. The user has to click several times and provide passwords or secret numbers at various instances. These mechanisms are designed with freshness of secret information in mind.

For DTN it would be desirable to have one-way transactions, like a signed letter of authority telling the bank to pay a certain amount of money to a specific account. The required keys for signing such a message would initially have to be exchanged during a visit at the bank.

Issues to be further investigated are the separation of, for example, the ssh protocol, and how to handle caching of websites transported over https.