Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 192 (2008) 43–53
www.elsevier.com/locate/entcs

A Clustering-based Group Key Agreement Protocol for *Ad-Hoc* Networks

Maarit Hietalahti[1]

Department of Computer Science
Helsinki University of Technology
Espoo, Finland

Abstract

Efficient authenticated group key establishment is the pre-requirement for having group-wide encrypted communications in wireless ad hoc networks. Clustering has brought scalability to ad hoc networks in many ways, now we look at its benefits to group key agreement. In this paper, several solutions for authenticated key establishment in clustered ad hoc networks are surveyed and a new efficient solution for clustered group key establishment is presented. It is based on the AT-GDH and the Burmester-Desmedt broadcast group key protocols.

Keywords: Group Key Establishment, Diffie-Hellman, *Ad Hoc* Networks

1 Introduction

Constructing group wide keys in a large *ad hoc* network is a complicated task that may be unachievable due to the dynamic nature of *ad hoc* networks. Splitting the problem to pieces, clustering the network, is a usual solution suggested for routing already in [12]. Clusters are supposed to have more stable internal connections due to the greater amount of links between nodes in a same cluster. Therefore, contributory group keys are easier to establish and manage inside clusters. On the other hand, clusters are assumed to stay together longer than the nodes do in average, which makes inter-cluster key agreement more sensible. Thus, clustering may bring the necessary scalability into key establishment in very large networks.

In *ad hoc* networks, every pair of nodes cannot reach each other within one hop. This issue of restricted topology, what H. Shi and M. He [18] call *the neighbours communication problem* can be alleviated by a careful choice for the graph structure that can be found in an arbitrary topology. A key agreement protocol using a

[1] Email: maarit.hietalahti@hut.fi

doi:10.1016/j.entcs.2008.05.005

spanning tree was presented by the author [9] and Di Crescenzo, et al., [4]. A clustered hybrid protocol using the protocol in [9] in connection with the Burmester-Desmedt key agreement (BD) [3], will be presented in this paper.

Rest of this paper is organised as follows. Section 2 explains the concepts of clustering, along with (non-clustered) group key agreement, lists security requirements for a group key agreement and briefly describes the circumstances of an *ad hoc* network environment and their effect to the task. Section 3 goes through some related clustered group key establishment methods and Section 4 presents the new clustered group key agreement protocol. Section 5 concludes the paper and sketches some lines for future work.

2 Background

2.1 Clustering and Hierarchical routing

A cluster is a collection of nodes (geometrically) close together. Clusters can be formed deliberately for a common cause or they can form as a reaction to a factor that is common to the nodes.

A hierarchical structure in a network is composed of nested groupings (clusterings) of nodes, forming a tree topology. Hierarchical structures are often used in routing see, e.g., [12], where optimal clustering structures are determined so as to minimise the size of the routing tables. Many algorithms need the knowledge of the whole network topology, while others perform the computations knowing only the neighbouring nodes and their possible cluster-memberships [1,21].

Clustering algorithms differ in what types of clusters they produce. Many clustering algorithms choose special nodes, cluster-heads, that take care of the cluster formation and later of the maintenance of the cluster [1], sometimes routing too. Cluster-heads are not always necessary, some clustering protocols do not use them at all. Gateways are border nodes that relay messages from one cluster to another. If clusters are allowed to overlap, a gateway usually belongs to more than one cluster. Some clustering algorithms form cliques, i.e., clusters where every node is at a one hop distance from every other node [13].

More information on clusterings can be found, for example, in [19].

2.2 Group Key Agreement

The purpose of key establishment is to create a common key for a group of two or more participants to be used for encryption and authentication of their communications. For two participants, the Diffie-Hellman key exchange [6] is often the most convenient choice. The multi-party case requires a generalisation of a two-way key exchange.

There are *distributory* and *contributory* group key protocols. A contributory protocol means that all participants take part in the key generation and guarantee for their part that the resulting key is fresh. Key distribution, on the other hand, means that the key is generated by one party and distributed to the other partici-

pants. This cannot be done without the help of a previously agreed-on secret that is used in encrypting the new session key. There is also a method called key predistribution, whereby the key is completely determined by the previously agreed-on initial key material.

2.3 Security Requirements for Group Key Establishment

In the context of group key exchange, implicit key authentication means that a participant can be sure that no-one outside the group can learn the key without the help of a dishonest participant. Key confirmation means that after the key has been established, the participants are assured that all legitimate participants do share the same key. As this would require many all-to-all messages, which may not even be possible in a sparse connection *ad hoc* network, achieving key confirmation is not practical. Explicit Key Authentication means that both implicit key authentication and key confirmation hold, i.e., all participants are assured that all legitimate participants know the key and no outsiders do.

An active adversary should not be able to mislead honest participants as to the final outcome. A compromise of past session keys should not allow a passive adversary to find out future session keys and should not allow impersonation by an active adversary in the future. Independence of long term and short term secrets is important when there is an additional long term secret present, for example, private keys of a public-key algorithm or passwords used in authentication.

2.4 The Challenges of Group Key Establishment inAd Hoc Network Environment

The limitations of *ad hoc* network environment pose some drastic demands on the group key establishment protocols. First, a global broadcast is most probably out of the question, that is, it is not probable that an arbitrary node will have direct connections to all other participant nodes. But on some occasions, a local broadcast from a node to its neighbours is feasible. Also, no fixed topology, such as a ring or a star can be assumed. Consequently, protocols requiring a specific topology either cannot be used at all or become inefficient.

In other words, every pair of nodes cannot reach each other within one hop. The issue, what H. Shi and M. He call *the neighbours communication problem* can be solved with the help of graph theory. This paper makes use of a spanning-tree algorithm, see [9] and [4]. There are also algorithms for generating more balanced spanning trees. See, for example, a survey by Gärtner [7].

The lack of infrastructure means that there are initially no third parties that can be trusted to calculate a random key safely and to distribute it. A lack of common history implies the lack of previously agreed shared secrets.

2.5 Group Key Agreement Protocols

For the definitions of trees and other graph theoretic notions, see, for example, [5].

The BD Broadcast Protocol

The broadcast protocol [3] assumes that every node is at a one hop distance from another. The protocol is accomplished with only two broadcasts per node.

BD Protocol Steps:

G is a finite cyclic group and g is a generator of G.

(i) Each node m_i selects a random exponent r_i and broadcasts $z_i = g^{r_i}$

(ii) Each node m_i computes and broadcasts $x_i = (z_{i+1}/z_{i-1}1)^{r_i}$

(iii) Each node computes the session key $k_i = z_{i-1}^{nr_i} x_i^{n-1} x_{i+1}^{n-2} \cdots x_{i+n-2}$.

TGDH

Tree-based Group Diffie-Hellman (TGDH) [11] employs Diffie-Hellman key exchanges in binary key trees. The described key structure results from the dynamic group key operations such as join, leave, merge and partition. There is no initial key agreement protocol.

The key structure in TGDH is very general, it can be used to describe the key structure of any bipartite group Diffie-Hellman key agreement where the resulting keys are used recursively as the new exponents. For example, the key structure of the protocol Hypercube [2] is the same as that of TGDH with perfect binary tree where all leaves are at the bottom level. Paper [14] extends TGDH protocol to improve the computational efficiency by utilising pairing-based cryptography. They use bilinear pairings in a ternary key tree which applies to any two-party and three-party key agreement protocol.

AT-GDH

AT-GDH (Arbitrary Topology Generalisation of Diffie-Hellman) [9] employs a spanning tree. A spanning tree contains only the (one hop) links used in initial key agreement. This avoids the neighbours communication problem, as the Diffie-Hellman key exchanges are done only with one-hop neighbours. The operations propagate over the network along the spanning tree. AT-GDH can be used in any connected network topology with bidirectional links, because a spanning tree can always be constructed in such a network.

All leaf nodes (nodes with no children) start by selecting a random secret exponent and blind it and send the result to their respective parents. After a node has received the blinded keys from all its children, they select their exponents and form Diffie-Hellman-type keys with their children repeatedly using the resulting key as the new exponent. The nodes do not send these keys to the children yet. The secret formed with the last child serves as the node's new private key, which the node blinds and sends to its parent. When this parent has received similar messages from all its children, it can repeat the same computation. This continues until the root has received all the blinded keys of its children. The root repeats the same kind of computation as all the other parent nodes. The secret key formed thus between the

root and its last child (and all other nodes) will be the shared session key material for the entire network. In the last phase of the protocol, the blinded keys needed for extracting the group key are propagated up the tree from the parents to their children starting from the root.

Key Agreement Protocol for an Arbitrary Tree:

Initialisation:
Let G be a finite cyclic group of order q, and let α be a generator of G. The participants are assumed to pick their secret exponents randomly from $\mathbb{Z}_q$. The mapping $\varphi : G \longrightarrow \mathbb{Z}_q$ is a bijection (always exists). Here the participants are identified with their universal address in the tree. c_x is the number of x's children. h is the height of the tree.

Round 1
For all nodes $x = y.i$ with $c_x = 0$

(i) x selects a random $k_x \in \mathbb{Z}_q$

(ii) $x \rightarrow y : \alpha^{k_x}$

Rounds $2 \dots h$
For all nodes x with $c_x \neq 0$

(i) x selects a random $e_x \in G$

(ii) x waits to receive $\alpha^{k_{x.j}}$ for all $j = 1, \dots, c_x$

(iii) x calculates $k_x = \varphi(K(x, c_x))$ from

$$K(x, 0) = e_x$$
$$K(x, j) = \alpha^{k_{x.j}\varphi(K(x,j-1))} \text{ for } j = 1, \dots, c_x$$

(iv) $x \rightarrow y : \alpha^{k_x}$

Rounds $h + l$**,** $l = 1, \dots, h$
For every node $x.i$ on level l,
$x \rightarrow x.i$: $M_{x.i}$, where

$$M_{x.i} = \langle M_x, \alpha^{\varphi(K(x,i-1))}, \alpha^{k_{x.(i+1)}}, \alpha^{k_{x.(i+2)}}, \alpha^{k_{x.c_x}} \rangle$$

with M_ϵ being empty.
The resulting common key is $K(\epsilon, c_\epsilon) = k_\epsilon$

AT-GDH does not contain group key management mechanisms, or authenticate the resulting key explicitly. The number of synchronous rounds AT-GDH needs to gather and distribute the blinded keys is twice the height of the tree. The height of the tree is assumed to be logarithmic to the number of nodes in the network, depending on the spanning tree algorithm and the topology of the underlying network links.

Other Protocols

Di Crescenzo, et al., [4] approach the problem of arbitrary topology from another angle than AT-GDH. They rigorously analyse the effect of physical topology on the actual performance of some key agreement protocols. These include GDH.2 and the BD broadcast protocol. In their analysis, they apply a topology-driven simulation of the logical network over any arbitrary *ad hoc* network graph. In connection with the key agreement protocols, they use auxiliary protocols in order to generate efficient embeddings of logical networks over arbitrary *ad hoc* networks. The auxiliary protocol suggested for generating a spanning tree is the same as in [9] and is explained in the Appendix.

3 Related Group Key Establishment Schemes for Clustered *Ad Hoc* Networks

A generic model for key establishment in clustered *ad hoc* networks works along these lines: First, nodes form clusters with some clustering method. Then a backbone or a key-tree is formed from the clusters, sometimes the tree extends inside clusters, sometimes the clusters are considered as single vertex in the tree. After this, the initial key agreement begins. Usually keys are established in subgraphs first, and then combined for a whole group wide key. The Diffie-Hellman key exchange (bipartite or tripartite) is typically used recursively as a basis for the group keying. A group key is constructed so that every node can calculate it using its own secret and the blinded secrets of others, or combinations of them. In some scenarios the messages are signed and key confirmation messages are sent for authentication purposes.

Rhee, et al., [17] present an architecture for key management in hierarchical mobile *ad hoc* networks. They use implicitly certified public keys (ICPK) [8], an ID-based public key scheme where the public key of each participant is derived from its identity. It provides computationally efficient implicit authentication. A key confirmation message added to the key agreement protocol makes the protocol explicitly authenticated. A two layered hierarchy is prompted by a physically two-layered network, ground nodes and unmanned aerial vehicles. The layers use different key management methods, the clusters of nodes below use a centralised system, while the aerial vehicles use TGDH. The centralised system inside clusters is not contributory.

Another hierarchical key agreement is proposed in [22]. This is a multilevel hierarchy, where a node can have several cluster keys according to the cluster and its super-clusters it belongs to. However, it is not completely contributory. Keys are agreed among cluster-heads on the same level and then distributed to their respective clusters.

Hybrid key management [15] propose a clustered key establishment, where each cluster selects a cluster-head that makes a key agreement with other cluster-heads. After that, the cluster-head distributes the key to the cluster. Thus, other nodes in the cluster do not contribute to the key. Clustering is made according to the

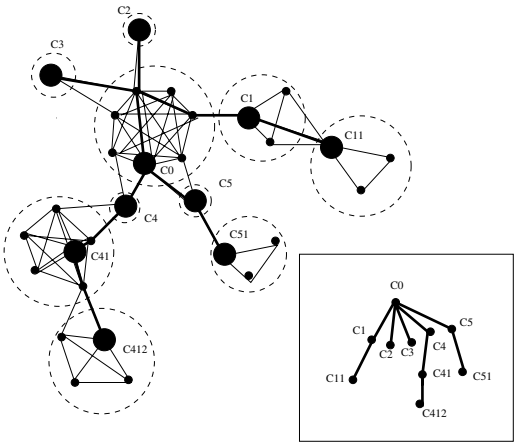

Fig. 1. Clustered AT-GDH

geometric locations of the nodes. The key agreement used can be any group key agreement protocol, for example GDH [20].

A cluster-tree-based group key agreement ACEKA is presented in [18]. ACEKA uses ternary trees with the Joux tripartite Diffie-Hellman key agreement [10]. There is a virtual backbone and virtual nodes in addition to the real nodes. ACEKA uses cluster-heads and "sponsors" for management. Authentication is done by signing every message using ID-based cryptography, with a variant of the ElGamal signature scheme.

4 The Proposed Group Key Protocol: Clustered AT-GDH

First, the network is divided into clusters with a clustering mechanism that creates very stable clusters. Nodes in a cluster are at a one hop distance from each other, i.e., cliques. In this kind of a cluster, the most efficient group key agreement protocol is the BD broadcast protocol explained before. It takes only two rounds of broadcasts, after which each node can calculate the common group key from its own secret exponent and the blinded shares of others.

When every cluster has a common secret key, the clusters agree a group key by AT-GDH protocol. Cluster-head can represent its cluster and use the cluster key as its secret exponent, instead of selecting a random $k_x \in \mathbb{Z}_q$ in Round 1. (see the box in Subsection 2.5). After the AT-GDH protocol run, cluster-heads distribute (broadcast) the last received message in their cluster, so that other nodes can also calculate the network wide group key. Figure 1 shows an example structure of the resulting cluster-tree.

Now that cluster-heads are not necessarily at a one hop distance from each other, the messages need to be relayed. The gateways relaying the messages are members of a cluster, and know the cluster secret already. However, it affects the communication complexity by adding extra links to the path.

4.1 Complexity Theoretic Analysis of Performance

This clustered group key agreement is efficient when cliques are large. Radio connections may create relatively large cliques. We evaluate the complexity in respect to synchronous rounds and number of exponentiations. A *synchronous round* means that every participant can send arbitrarily many packages concurrently within a single time tick (round) or receive arbitrarily many at the beginning of a round. The *number of exponentiations* means the total number (the sum) of exponentiations performed by all participants.

Every clique forms a group key in two communication rounds, i.e., constant amount of rounds. In the end, cluster-heads broadcast the key parts in one communication round. The complexity of a clustered key agreement is the complexity of AT-GDH in the number of clusters plus a constant. The resulting communication complexity is logarithmic to the number of clusters c.

The number of exponentiations can be estimated in the same way, using the figures for BD and AT-GDH. The amount of exponentiations in BD is $n^2+n = \mathcal{O}(n^2)$ and AT-GDH (with cluster-heads) $\mathcal{O}(c \log c)$. In the end, the exponentiations done after the final broadcast by a single node is the amount of clusters, in the worst case, and all leave nodes ($\leq n$) do a total of $\mathcal{O}(nc)$ exponentiations. The total number of exponentiations is $\mathcal{O}(n^2) + \mathcal{O}(c \log c)$.

	BD	GDH.2	AT-GDH	Clustered AT-GDH
syn. rounds	2	n	$2\lceil \log n \rceil$	$3 + 2\lceil \log c \rceil$
total exp.	$n^2 + n$	$\mathcal{O}(n^2)$	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^2) + \mathcal{O}(c \log c)$

Table 1
The efficiency measures for some key agreement protocols

The above analysis is done without considering the cost of embedding the protocol in an arbitrary *ad hoc* network topology. Thus, each protocol is evaluated in its optimal network topology. However, AT-GDH requires as its embedding only a protocol for generating a spanning tree, which can be done rather efficiently. The protocol described in [9] adds only h communication rounds, h equalling the height of the tree, i.e., the eccentricity of the initiating node.

In this case, the assumption of a balanced spanning tree that has a logarithmic communication complexity may be too optimistic. Using the Unit Disk Graph model and assuming a dense evenly distributed network we can approximate the height of the resulting spanning tree in another way. In the unit disk model, the nodes have same transmission radii, one unit. If the node density is δ, the average number of neighbours is the number of nodes in a unit disk $\delta\pi$. The average distance between neighbouring nodes is $d = 1/\sqrt{2}$. In a dense network, the number of nodes within k hops, n_k can be estimated by counting the area of a disk with a radius kd. Then, $n_k = (kd)^2\pi\delta$. The maximum path length from the root to the leave is

then $k = \frac{1}{d}\sqrt{\frac{n}{\pi\delta}} = \sqrt{\frac{2n}{\pi\delta}}$. Hence, k grows to the square root of n and the resulting complexity for clustered AT-GDH is the square root to the number of clusters.

4.2 Adding Authentication

Previously group key agreements, like the authenticated GDH, A-GDH, relied much on the implicit key authentication property: The group key can not be constructed without the secret share of one of the participants. However, Pereira and Quisquater [16] showed that it is impossible to design a scalable authenticated group key agreement protocol on the same building blocks as A-GDH. Hence other authentication methods are needed. Authentication with ID-based crypto, such as the ICPK [8] public keys, with key confirmation messages, could be used here, as it is independent of the group key establishment method used.

5 Conclusions and Future Work

Clustering is a versatile solution in *ad hoc* networks, its benefits can be seen in routing and other operations requiring efficient gathering and propagation of information among the network. It was seen here that clustering can also help in creating a symmetric group key for fast encrypted communications. A new group key protocol was proposed. The cluster-based extension of AT-GDH combined to the broadcast group key protocol turned out to be very efficient. Using UDG:s, the number of rounds was found to be comparable to the square root of the number of clusters.

Clustered AT-GDH could be more efficient with tripartite key exchange realized with bilinear pairings as in [14]. Manipulating the form of the tree and clusters may also affect the efficiency of the group key establishment. However this needs more research. A security proof of the new combined protocol is also left for future work.

In an *ad hoc* network where nodes are mobile, a mere group key establishment is not always enough. The group key needs maintenance. At least the key should be updated when nodes join or leave the network, to preserve its contributory property. Neither AT-GDH nor the clustered extension proposed here have group key maintenance which is outside the scope of this paper and left for future work.

References

[1] Basagni, S., *Distributed clustering for ad hoc networks*, in: *Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN)*, IEEE, Perth, Australia, 1999, pp. 310–315.

[2] Becker, K. and U. Wille, *Communication complexity of group key distribution*, in: *Proceedings of the ACM Conference on Computer and Communications Security (CCS)* (1998), pp. 1–6.

[3] Burmester, M. and Y. Desmedt, *A secure and efficient conference key distribution system*, in: *Advances in Cryptology – Proceedings of EUROCRYPT*, LNCS **950** (1994), pp. 275–286.

[4] Di Crescenzo, G., M. Striki and J. S. Baras, *Modeling key agreement in multi-hop ad hoc networks*, in: *Proceeding of the 2006 International Conference on Communications and Mobile Computing*, Vancouver, British Columbia, Canada, 2006.

[5] Diestel, R., ""Graph Theory"," Springer, New York, USA, 1997, xiv+289 pp.

[6] Diffie, W. and M. E. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory **IT–22** (1976), pp. 644–654.

[7] Gärtner, F., *A survey of self-stabilizing spanning-tree construction algorithms*, Technical report, EPFL (2003).

[8] Günther, C., *An identity–based key exchange protocol*, in: *Advances in Cryptology – Proceedings of EUROCRYPT*, LNCS **434**, 1989, pp. 29–37.

[9] Hietalahti, M., ""Efficient Key Agreement for Ad Hoc Networks"," Master's thesis, Helsinki University of Technology, Department of Computer Science and Engineering, Espoo, Finland (2001).

[10] Joux, A., *A one round protocol for tripartite Diffie–Hellman*, in: *Proceedings of Algorithmic Number Theory Symposium IV*, LNCS **1838** (2000), pp. 385–394.

[11] Kim, Y., A. Perrig and G. Tsudik, *Simple and fault–tolerant key agreement for dynamic collaborative groups*, in: *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, ACM, Athens, Greece, 2000, pp. 235–244.

[12] Kleinrock, L. and F. Kamoun, *Hierarchical routing for large networks; performance evaluation and optimization*, Computer Networks **1** (1977), pp. 155–174.

[13] Krishna, P., N. H. Vaidya, M. Chatterjee and D. K. Pradhan, *A cluster–based approach for routing in dynamic networks*, ACM SIGCOMM Computer Communication Review (1997), pp. 49–65.

[14] Lee, S., Y. Kim, K. Kim and D. H. Ruy, *An efficient tree–based group key agreement using bilinear map*, in: *Applied Cryptography and Network Security ACNS*, LNCS **2486** (2003), pp. 357–371.

[15] Li, X., Y. Wan and O. Frieder, *Efficient hybrid key agreement protocol for wireless ad hoc networks*, in: *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on*, 2002, pp. 404 – 409.

[16] Pereira, O. and J.-J. Quisquater, *Generic insecurity of cliques-type authenticated group key agreement protocols*, in: *Computer Security Foundations Workshop, 2004. Proceedings. 17th IEEE* (2004), pp. 16–29.

[17] Rhee, K. H., Y. H. Park and G. Tsudik, *A group key management architecture for mobile ad-hoc wireless networks*, Journal of Information Science and Engineering **21** (2005), pp. 415–428.

[18] Shi, H. and M. He, *Authenticated and communication efficient group key agreement for ad hoc networks*, in: *The 5th International Conference on Cryptology and Network Security CANS06*, Suzhou, China, 2006.

[19] Steenstrup, M., *Cluster–based networks*, in: C. E. Perkins, editor, *"Ad Hoc Networks"*, Addison Wesley, 2001 pp. 75–138.

[20] Steiner, M., G. Tsudik and M. Waidner, *Diffie–Hellman key distribution extended to group communication*, in: *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, ACM (1996), pp. 31–37.

[21] Virtanen, S. E. and P. Nikander, *Local clustering for hierarchical ad hoc networks*, in: *Proceedings of WiOpt: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks* (2004), pp. 404–405.

[22] Yao, G., K. Ren, F. Bao, R. H. Deng and D. Feng, *Making the key agreement protocol in mobile ad hoc network more efficient*, in: *Applied Cryptography and Network Security ACNS*, LNCS **2846** (2003), pp. 343–356.

A Appendix: Constructing a Spanning Tree

This way to construct a spanning tree so that the node initiating the protocol becomes the root was described in [9].

In the initial state it is assumed that the nodes know their neighbours and that all links are two-way. The initiator sends a message to each of its neighbours. It thereby becomes the root of the spanning tree and its neighbours become its children. After receiving a message, a node acknowledges it and sends a similar

message to all its neighbours, except to the parent. The nodes that acknowledge a message from a node become its children in the tree. When a node gets more than one of these messages, it acknowledges and processes only the message that it receives first, and consequent messages are ignored. This continues until every node has received a message. A leaf is a node that does not receive acknowledgements from any of its neighbours. The spanning tree has now been constructed and all nodes know their parent and their children.

This spanning-tree protocol can be used in a network where the connections are reliable. In an unreliable network, negative acknowledgements must be sent for the messages that are ignored. Local broadcast can be used to send the message to all neighbours and an acknowledgement to the parent all in one message.