Journal of Computer Science 6 (5): 566-571, 2010
ISSN 1549-3636

Security Challenges in Multicast Communication for Mobile Ad Hoc Network

[1]S. Gunasekaran and [2]K. Duraiswamy
[1]Department of Post Graduate Studies, Faculty of Computer Applications,
[2]Department of Computer Science and Engineering,
KSRangasamy College of Technology, Tiruchengode, Namakkal-Dist, Tamil Nadu-637215, India

Abstract: Problem statement: Multicasting communication network accepted a single message from an application and delivered copies of the message to multiple recipients at different locations. Recently, there has been an explosion of research literature on multicast communication environment. The objective of this study were to contribute the complexity of supporting current multicast applications, (i) the lack of reliable multicast transport mechanisms at the network level and (ii) the lack of network support for large scale multicast communication. The scaling problem of secure multicast key distribution compounded for the case where sender-specific keys need to be distributed to a group and required for sender-specific authentication of data traffic and minimize control overhead (iii) compare RC4, AES-128,RS(2) and RS(3) computation time of both algorithms. **Approach:** Algorithms were collected and performed computation time. In general the multicast key distribution scheme implemented for distributing 128 bit session keys. Thus the Maximum Distance Separable Codes (MDS Codes) needed for their encoding and decoding process. In rekeying scheme errors were occurred during over period of time or at a particular point of time and to eliminate all these errors in the level of encryption and decryption mechanism. The MDS codes played an important role in providing security services for multicast, such as traffic, integrity, authentication and confidentiality, is particularly problematic since it requires securely distributing a group (session) key to each of a group's receivers. **Results:** First we showed that internet multicasting algorithms based on reverse path forwarding were inherently unreliable and present a source-tree-based reliable multicasting scheme also. The new scheme proposed and used as an inter-gateway protocol and worked on top of the previously developed distance vector and link state internet routing schemes. Next, to support large scale applications, we presented a scheme for partial multicasting and introduced a new network level operation, called gather. The partial multicasting mechanism allowed messages to be delivered to subsets of multicast destinations, while the gather operation aids gateways in selectively suppressing redundant messages, thus reducing the message complexity. **Conclusion:** Hence the findings suggested the control overhead reasonably minimized and using simulations, we investigated the efficiency of our schemes in supporting scalable application domain based multicast communication.

Key words: Multicast, gather, inherently, redundant, unreliable

INTRODUCTION

With the growing social networking communication in the internet scenario (heterogeneous network), need for a robust multicast communication model in terms of scalability and reliability arises with high demand. In distributed applications, such as multimedia teleconferencing (Raghavan *et al*., 2007; Judge and Ammar, 2002), distributed database systems (Hardjono and Cain, 2000), factory automation (Kumar *et al*., 2004) and distributed games (Caesar and Rexford, 2005), group communication (Aslan, 2004), (i.e.,) one-to-many communication between groups of cooperating process is required. In resource location, clients seek out the services of a group of remote servers, by multicasting queries. In data distribution applications, such as software distribution and time management, copies of a message are delivered from one central site to multiple destinations concurrently. Many of these applications require reliable delivery and ordering of multicast messages. Furthermore, some applications, such as

Corresponding Author: K. Duraiswamy, Department of Computer Science and Engineering,
KSRangasamy College of Technology, Tiruchengode, Namakkal-Dist, Tamil Nadu-637215, India

manufacturing control systems and trading room systems for stock brokerage firms, involve large groups of entities communication.

Network level support for such communication has so far remained minimal, Network multicast schemes usually provide only a best-effort delivery service. For process group communication, however, a number of higher level schemes have been developed to provide both global ordering and reliable delivery of multicast messages in the presence of host and network faults. But these schemes invariably use multiple, reliable point-to-point connections (e.g., TCP) and network level multicast message transport (Fig. 1). Such an approach has the drawback of being wasteful in terms of communication overheads.

Apart from reliability, efficiency concerns arise in large scales multicast communication. Communication overheads increase with group size in many ways: first, when point-to-point network transport is used, the number of messages required to implement a reliable multicast increases. Second, even when the network supports multicasting, the number of messages increases, especially for those applications that need only a subset of group members to receive a multicast message. For instance, in a number of client-server applications, any one of the group of servers is capable of providing the service to the client. Here, it is clearly advantageous to have multicast messages from the client delivered to only a subset of servers, rather than to all of them. Firstly, the amount of reverse communication increases with group size; the number of end-to-end acknowledgements increases and, depending on the application, the number of simultaneous replies sent by the multicast recipients to the source increases, This last problem leads to the so-called implosion effect, thereby a source is overwhelmed by simultaneous responses from all the destinations (Caesar and Rexford, 2005).

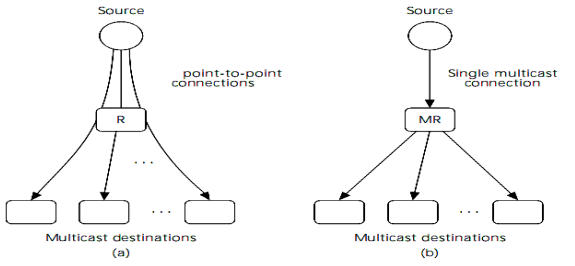

Fig. 1: An illustration of the amount of network resources employed: (R: Standard Router, MR: Multicast Router): (a): unicasting service for different users; (b): multicast service to all members

The key to supporting reliable and large scale group communication efficiently is to reduce the total message overhead at the network level. This approach has already been successfully demonstrated by the development of an efficient reliable network level multicast scheme for wide area networks. The proposal of our work plans to reveal the use of relatively simple network level protocols and more advanced services can be provided to support the sophisticated needs of distributed social networking applications. We describe robust network level multicast mechanisms for the social networking domain which is proposed in terms of:

- Reliable multicasting: Lossless and sequenced delivery of multicast messages from a single source
- Partial multicasting: Delivering multicast messages to any k members of a multicast group
- Gather: Controlled delivery of messages from several sources to a single destination
- Scalable multicasting: With growing size of users in multiple domains of the internet multicast communication need to be handled efficiently

The network model we plan to use in our research work is the internet, a collection of subnets interconnected by gateways. Internets are susceptible to link and gateway failures that affect their connectivity. Furthermore, messages can be lost in transit due to transmission errors or buffer overflow. The internet is said to be dynamic, if topological and/or traffic load changes occur frequently (Canetti *et al*., 1999a; 1999b). Thus, in a dynamic internet, the shortest-path routes between a pair of gateways change frequently, resulting in frequent updates to gateway routing tables. An internet multicast scheme is said to be reliable, if it has the following properties:

- Completeness: Multicast messages are delivered to each destination in the same order as sent by the source, without message duplication or loss
- Finiteness: Each multicast message is accepted by all the destinations in a finite amount of time after it releases from source node. The reliable multicast scheme preserves these properties in the presence of topological changes in the network. It should be noted, though it may not always be possible to achieve completeness and finiteness in a dynamic internet. Specifically, when a network partition occurs, multicasts in progress may not complete at all

MATERIALS AND METHODS

The performance parameters used to show the effectiveness of reliability of the multicast communication model and its scalability in terms of large number of users joining and leaving the communication group are investigated thoroughly below (Canetti *et al*., 1999b).

Assume that on an average there are n gateways in the multicast tree of a source. The value of n depends on the size of the internet and the distribution of group members. Under normal operating conditions, the control overhead consists of messages exchanged between neighboring gateways to ensure reliable transmission. If a multicast message usually reaches each destination along the least cost path, every multicast completes m O (D) time, where D is the diameter (i.e., the longest shortest-path) of the internet. The paths may be slightly longer when the shortest-path tree is different from the multicast tree, but the situation is corrected after the next tree construction cycle.

Some communication overhead is incurred by all internet multicast protocols, unreliable or reliable; in mapping group-id's to host locations. We therefore do not attempt to compute it here. Under reliable multicasting, this overhead depends mainly on the group dynamics, (i.e.,), the rate at which a multicast group appears on and leaves from subnets. Assuming relatively long-lived multicast groups, this overhead may not be significant. For the RPF-based internet multicast protocol, the overhead depends on two additional factors the number of host multicasting to a group and the period with which hosts advertise their group associations along the branches of the multicast trees.

For some applications, it is adequate to have a multicast message be delivered to a subset of the group members. The partial multicasting facility allows a source to specify the number of destinations for each message. This feature is particularly helpful when the target group is large. The network guarantees the delivery of messages to exactly k destinations, if the group size remains greater than or equal to k until the completion of the multicast. If group membership changes during a multicast, the message may be delivered to less than k destinations. Partial multicast is useful in two ways: first, the message overhead in delivering the multicast is decreased. Second, a lot of redundant computation and possibly reverse communication are eliminated by not delivering messages to all the hosts (Canetti *et al*., 1999a).

Partial multicast can be implemented using the protocols of reliable multicast First; we note that the group association protocol lets each end gateway know the host addresses of all group members on directly connected subnets. But the internal gateways know only the subnet addresses of group members and not the number of members on each such subnet. Therefore, the group association protocol (Canetti *et al*., 1999a), it must be modified such that the end gateways include this information while propagating association messages. Instruct intermediate gateways to perform computations that affect further propagation of these messages.

It reduces the communication overheads of multicast transport protocols as well as application level group communication protocols. Gather is a controlled inverted multicast message flow from a set of source nodes to a single destination node along the multicast tree to the destination, usually in response to a multicast from the destination. But unlike multicast messages, gather messages contain information that instructs intermediate gateways to perform computations that affect further propagation of these messages. Id denotes the unique identifier for the set of related gather messages. Since gather messages usually rise in response to multicasts, the uniqueness of the id can be guaranteed by letting the respondents derive the id from the multicast message itself. The Group-id field indicates the multicast group-id. Traditionally, the key distribution function has been assigned to a central network entity, or Key Distribution Centre (KDC) (Ballardie, 1996), but this method does not scale for wide-area multicasting, where group members may be widely-distributed across the inter network and a wide-area group may be densely populated. Even more problematic is the scalable distribution of sender-specific keys. Sender-specific keys are required if data traffic is to be authenticated on a per-sender basis. This memo provides a scalable solution to the multicast key distribution problem (Burmenster and Desmedt, 1994).The essential problem of distributing a session (or group) key to a group of multicast receivers lies in the fact that some central key management entity, such as a Key Distribution Centre (KDC). A Key Distribution Centre (KDC) (Ballardie, 1996) is a network entity, usually residing at a well-known address. It is a third party entity whose responsibility is to generate and distribute symmetric key(s) to peers, or group receivers in the case of multicast, wishing to engage in a "secure" communication. It must therefore be able to identify and reliably authenticate requestors of symmetric keys to authenticate each of the group's receivers, as well as securely distribute a session key to each of them.

In short, existing multicast key distribution methods do not scale (Aslan, 2004; Burmenster and

Desmedt, 1994). Reliance on two separate hosts to create group keys maximizes the probability that the resulting key will have the appropriate cryptographic properties. A single host could create the key if the randomization function were robust and trusted. Unfortunately this usually requires specialized hardware not available at most host sites. The intent of this protocol was to utilize generic hardware to enhance the extendibility of the Group Key Management Protocol. Hence, cooperative key generation mechanisms are used (Harney and Muckenhirn, 1997).

RESULTS

The performance improvement is made in the direction of minimizing the rekeying communication complexity. Thus the Maximum Distance Separable Codes (MDS Codes) needed are (L, 2) and (L, 3) codes and their encoding and decoding are significantly simpler than general (L, n) MDS codes, where n = 3. Thus the MDS-Code based scheme is particularly suitable and practical for rekeying operations in 3-ary balanced key trees. In most of the current applications, a session key needs to be at least 128 bits to be considered reasonably secure. Thus a multicast key distribution scheme is implemented for distributing 128-bit session keys.

The scheme uses a 3-ary balanced key tree. The well-known Reed-Solomon (RS) code is employed as a MDS code. As discussed in the basic scheme, where m = lr = t = 128 bits. Thus the RS code is constructed in a finite field. However, The Finite Field is too large to be computationally feasible to implement. Thus Finite field is chosen instead. Since an RS code in Finite field can only process 16 bit symbols, 8 erasure decoding operations in Finite field are needed to produce a 128-bit session key. The RS code is used to distribute the immediate subgroup key to the remaining 2 members of the lowest level subgroup tree affected by the old member. A (216, 3) RS code is used to distribute all other affected subgroup keys. When member 9 leaves, the (216, 2) RS code is used for distributing a new key K7-8 to the members 7 and 8 and the (216, 3) RS code is then used to distribute the remaining keys, including the new session key. Finally, the MD5 algorithm is used as a one-way hash function to produce 128-bit hash outputs. A more computation efficient hash function can certainly be used, as long as the one-way property is guaranteed. This implementation simply uses the widely used existing components which are readily available and whose properties have been studied for a long time.

Table 1: Computation Time of a 128-bit symbol

Algorithms	Encryption time (μ sec)	Decryption time (μ sec)
RC4	43	43
AES-128	8	12
RS(2)	6	6
RS(3)	12	8

Note: Computing time of Encryption and Decryption time of selected algorithms in 128 bit symbol

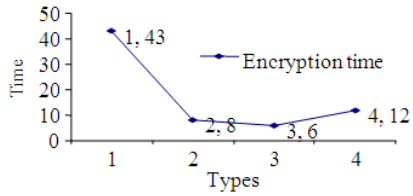

Fig. 2: Encryption time chart. **Note:** The encryption time taken of several algorithms

Experiments have been conducted to compare the computation time using this scheme (which are called RS(2) and RS(3) in the Table 1, since (216, 2) and (216, 3) RS codes are used) and the conventional schemes using encryptions. The encryption algorithms compared are AES and RC4. The encryption keys for these algorithms are all of 128 bits and an optimal C implementation of AES is used. DES and triple-DES are experimented as well, since their encryption keys are not 128 bit and they are much slower to compute than the above algorithms.

Table 1 lists the experimental computation time of one encryption/decryption of a 128 bit data symbol on a 2.5 GHz Pentium Dual Core PC running on Windows XP. In the scheme using RS codes, encryption and decryption of a 128 bit data symbol correspond to 8 erasure decoding of RS code (RS(2)) or RS code (RS(3)). The erasure decoding of the RS codes are implemented by solving linear equations using standard Gaussian Elimination. Additions and multiplications are based on table look-ups. In addition, it takes about 2 μ sec to produce a 128 bit hash output using the MD5 on the same PC.

The Fig. 2 illustrates the encryption time taken for various algorithms specified in Table 1. For Example the encryption time which is taken for RS(3) is 12 μ sec.

Notice that in distributing a key, i.e., in the encryption process, RS(3) needs to decode two 128bit symbols, while in recovering a key, i.e., in the decryption process, RS(3) needs to decode only one 128 bit symbol. On the other hand, RS (2) only needs to decode one 128 bit symbol in both processes. The AES-128 uses an optimal implementation which avoids expensive additions and multiplications in finite field (28) by using only binary exclusive ORs.

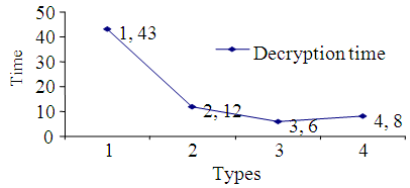

Fig. 3: Decryption time chart. **Note:** The decryption time taken of several algorithms

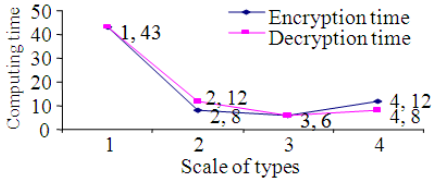

Fig. 4: Computation time chart for encryption and decryption in 128 bit symbols. **Note:** Computation chart for encrypting and decrypting process in 128 symbols

The Fig. 3 illustrates the decryption time taken for various algorithms specified in Table 1. For Example the Decryption time which is taken for RS(3) is 8 μ sec. The erasure decoding time of RS codes can certainly be further reduced by carefully employing binary exclusive ORs rather than additions and multiplications in finite field. Also observe that for most encryption algorithms, before encrypting user data, encryption keys need to be preprocessed. This key scheduling process takes relatively little time when the user data to be encrypted is large. In key distribution schemes, however, the user data has only 128 bits while the encryption key has 128 bits too. Thus the key scheduling time is usually much longer than that on encrypting the 128 bit user data symbol.

Unfortunately such key scheduling operations can neither be avoided nor be amortized to large user data, since data to different members have to be encrypted using different keys. The erasure decoding of RS codes (as well as all other practical MDS codes), on the other hand, does not require such a key scheduling process. Thus key distribution using MDS codes is more efficient than using conventional encryption algorithms (Aslan, 2004). Based on encryption/decryption time listed in Table 1, it is easy to calculate total rekeying time of changing a new session key to a group of n members using a 3-ary balanced key tree. The rekeying process consists of two phases: key dissemination (during which the GC's computes and sends necessary data to group members) and key recovery (during which group members receive and compute necessary data to obtain a new session key).

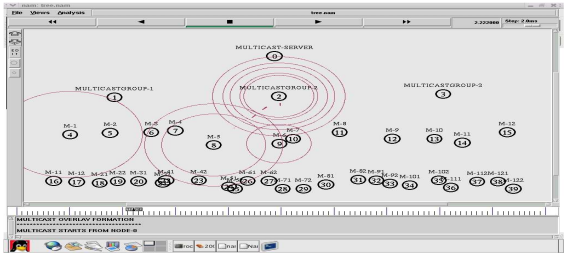

Fig. 5: The minimal of control overhead message. **Note:** Simulation results shown the scalability issues of minimizing control overhead

The experiments begin with a delivery tree established by a heuristic for solving the degree-constrained minimum spanning tree problem. Then end hosts join or leave the tree dynamically. The join and leave are both modeled as a Poisson process with rate minute, which means that on the average, there is a node joining and a node leaving the tree every 15 sec. Each experiment lasts for two hours.

Responsiveness indicates how fast each scheme can restore the delivery tree after a node fails or leaves the tree. The first metric used for measuring the responsiveness is the average recovery time, which is the average time for an affected node to find a new parent. For each node, we calculate the sum of latencies of the links the control message travels for finding the new parent as its recovery time. The Fig. 5 shows the responsive measure of the node joining or leaving time to that of the bandwidth overhead generated by the proposed multicast scheme Fig. 5. As a node is willing to join and leave from the multicast group the time increases and increases the bandwidth overhead of the group communication also.

Overlay multicast differs from traditional IP multicast in that the problem of degree constraints is more prominent and non-leaf nodes in the multicast tree are unstable. This makes the problem of restoring multicast tree after node failures or leaves quite different. The proactive approach has been used in recovering link or node failures in multicast tree in the context of the traditional network-layer multicast. The fault-tolerant multicast routing was proposed to use backup paths from the grandparent to deal with the link or path failures. These schemes dealt with the network-layer fault-tolerant multicast routing problem and did not consider the important degree constraints on the nodes in overlay multicast. In the reactive approach for dealing with node failures in overlay multicast, the time to find an appropriate place was long and those affected

nodes competed with each other and it also did not mention the degree limit of the nodes.

DISCUSSION

The quality of the restored tree can be measured in two aspects. One is the tree cost, which measures the resource usage of the tree. The other is the maximum delay of the nodes in the tree from the root. The bandwidth overhead provides the measure of quality of multicast group communication on tree restoration. In the end host (nodes) of the multicast tree is taken as the implication for quality point of tree restoration. The bandwidth overhead is high when the end host is minimal and the overhead decrease for increased end host shows the quality of the multicast tree constructed for message communication. Since the recipients may have different delay tolerance and are typically connected to the source via paths of different delay, bandwidth and loss characteristics, traditional approaches to flow controls based on source adaptations need to be improved. The Multicast communication required additional network support such as Congestion and dynamically changing overall bandwidth utilization also. The main issue focused in this paper is to minimize control overhead messages in all the level of multicast communication by generating re keying mechanism of Encryption and Decryption process.

CONCLUSION

Network level mechanisms can efficiently support the sophisticated communication needs of distributed applications. In our work, we have presented schemes for supporting reliable and large scale one-to-many and many-to-one communication which provide lossless and sequenced delivery of messages to all the destinations. The performance studies have shown that our proposed multicast reliable and scalable framework is more efficient than both multiple, reliable uni-casts and unreliable multicast with multiple end-to-end acknowledgements, for supporting application level fault-tolerant multicast schemes, even with small group sizes. Using the protocols developed, a partial multicast scheme was outlined which reliably delivers messages to a subset of the multicast group members, thus allowing large group sizes for many applications. Finally, we have presented a mechanism for many-to-one communication, which allows the network itself to filter redundant messages. Further enhancements can be taken in the direction of issues related to selecting the proper window size for our multicasts and performance aspects in application versatility.

REFERENCES

Aslan, H., 2004. A scalable and distributed multicast security protocol using a subgroup-key hierarchy. Comput. Security, 23: 320-329.

Ballardie, A., 1996. RFC 1949: Scalable multicast key distribution. http://www.faqs.org/rfcs/rfc1949.html

Burmenster, M. and Y. Desmedt, 1994. A secure and efficient conference key distribution system. Lecture Notes Comput. Sci., 950: 275-286. DOI: 10.1007/BFb0053443

Caesar, M. and J. Rexford, 2005. BGP policies in ISP networks. IEEE Network, 19: 5-11.

Canetti, R., J. Garay, G. Itkis, D. Micciancio, M. Naor and B. Pinkas, 1999a. Multicast security: A taxonomy and some efficient constructions. Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE Computer Society, New York, pp: 708-716. DOI: 10.1109/INFCOM.1999.751457

Canetti, R., T. Malkin and K. Nissim, 1999b. Efficient communication-storage tradeoffs for multicast encryption. Lectures Notes Comput. Sci., 1599: 459-474. http://portal.acm.org/citation.cfm?id=937506

Hardjono, T. and B. Cain, 2000. Key establishment for IGMP authentication in IP multicast. Proceeding of the 1st European Conference on Universal Multiservice Networks, (UMN'00), IEEE Computer Society, Colmar, pp: 247-252. DOI: 10.1109/ECUMN.2000.880748

Harney, H. and C. Muckenhirn, 1997. Group Key Management Protocol (GKMP) Specification. RFC 2093. http://portal.acm.org/citation.cfm?id=RFC2093

Judge, P. and M. Ammar, 2002. Gothic: Group access control architecture for secure multicast and anycast. IEEE INFOCOM, 3: 1547-1556. http://direct.bl.uk/bld/PlaceOrder.do?UIN=118582688&ETOC=RN&from=searchengine

Kumar, A., J. Xu, L. Li and J. Wang, 2004. Space-code bloom filter for efficient per-flow traffic measurement. Proceeding of the 3rd ACM SIGCOMM Conference on Internet Measurement, Oct. 27-29, ACM Press, Miami Beach, FL., USA., pp: 167-172.

Raghavan, B., K. Vishwanath, S. Ramabhadran, K. Yocum and A.C. Snoeren, 2007. Cloud control with distributed rate limiting. Proceeding of the 2007 conference on Applications, Technologies, Architectures and Protocols for Computer Communications, Aug. 31-37, ACM Press, Kyoto, Japan, pp: 337-348. DOI: 10.1109/ICN.2009.71