Secure Bandwidth Efficient Multicasting for Wide Area Networks

S. Koneru¹, B. Gupta¹
¹ Computer Science Department Southern Illinois University
sindoora@siu.edu

Abstract: Recently an efficient multicasting protocol has been reported which uses a concept known as pseudo diameter to reduce the number of duplicate packets generated during multicasting in wide area networks (WANs). This work is superior to the Distance vector Multicast Routing protocols (DVMRP) from the viewpoint of better bandwidth utilization. In this paper security aspect of the above mentioned multicast protocol has been considered. Asymmetric key cryptography concept has been used effectively to design the protocol.

1. INTRODUCTION

The existing approaches for message propagation in WANs use the concepts such as (a) multi destination addressing, (b) spanning tree based forwarding, (c) Reverse path forwarding, (d) transmission of separately addressed packets, (e) core based tree etc. [1]-[4]. However, none of them has considered secured multicasting. A multicast message is one that is transmitted to a selected multiple recipients who have joined the appropriate multicast group. Efficient multicasting techniques [1], [2] have been developed for WANs. One such method is the Distance Vector Multicast Routing Protocol (DVMRP) [3], [11]. The DVMRP uses the distance vector routing (DVR) scheme [10] for multicasting. In this method there is no duplicate packet generation unlike in Reverse Path Forwarding method (RPF) [12]. However, to get rid of duplicate packets DVMRP employs a mechanism that defines parent-child relation among different nodes which needs very large number of control packets. Another approach called Multicasting using modified reverse path forwarding (MMRPF) has been proposed in which number of duplicate packets generated is much lower than the control packets generated by DVMRP [11]. In WAN security is needed because the communicating parties are prone to attacks. Secure multicasting [4], [5], [7] allows a node to propagate a message to a group of recipients in such a way that despite the malicious efforts of few nodes, all destinations which are honest receive the same message that is propagated by the source.
**Problem formulation:** The objective of the proposed work is to incorporate the security aspect in MMRPF with the help of asymmetric key cryptography [9] to design a bandwidth efficient secured multicasting protocol for WANs.

2. PSEUDO DIAMETER AND RELEVANT DATA STRUCTURES

In this section we first state the importance of pseudo diameter for message propagation, followed by a description of the data structures used in our proposed algorithm.

2.1 Importance of Pseudo Diameter

Pseudo diameter (τ) of a node n is defined as the maximum of all minimum delays to reach all other nodes according to its distance vector routing table. It is actually the lower bound on cost (here it is delay) to reach any node from the source node n. The following two examples of broadcasting related respectively to reverse path forwarding (RPF) and modified reverse path forwarding (MRPF) highlights the importance of using pseudo diameter for message propagation.

**Example 1:** Consider the subnet as shown in Fig. 1 in which a source node r_s is broadcasting a packet. Assume that the best path for nodes r_a, r_b, and r_c from node r_s are via r_i. Node r_s sends copies of the packet to nodes r_a and r_i. Node r_i accepts the packet because it has come along the best path. Node r_a discards the packet because it has not arrived along the best path. Node r_i now forwards copies to its neighbors. These neighboring nodes r_a, r_b, and r_c accept the packets from node r_i because they have come along the best path. Thus we observe that whenever a packet arrives at a node along the best path it forwards copies of the packet to its neighbors unconditionally.

Thus total number of packets generated in the above example is 5, even though it needs only 4 packets

![Fig. 1 Example for broadcasting with RPF](image)

**Example 2:** Consider the subnet as shown in Fig. 2 in which source node r_s is
Broadcasting a packet. Assume that the best paths for nodes \( r_a, r_b, \) and \( r_c \) from node \( r_s \) are via \( r_i \).

\[
\begin{align*}
\tau_i & \quad r_i \\
\tau_a & \quad r_a \\
\tau_b & \quad r_b \\
\tau_c & \quad r_c
\end{align*}
\]

Fig. 2 Example for broadcasting with MRPF

Node \( r_s \) initially finds the pseudo diameter \( \tau \) (which is the maximum delay) from its DVR table. Observe that the total delay of each of the three best paths from \( r_s \) to \( r_a, r_b, \) and \( r_c \) via \( r_i \) is less than or equal to the maximum delay \( \tau \). That is, \( \tau > \tau_i + \tau_a, \tau > \tau_i + \tau_b, \) and \( \tau > \tau_i + \tau_c \). We assume that \( \tau_a > \tau \). While forwarding copies of the packet to its neighbors \( r_i \) and \( r_a \), the source node \( r_s \) checks whether it has enough pseudo diameter to reach them. Since \( \tau > \tau_i \) it updates the pseudo diameter to \( \tau - \tau_i \) and forwards a copy to node \( r_i \). Node \( r_s \) does not have enough pseudo diameter to reach node \( r_a \) since \( \tau_a > \tau \). So it will not forward a copy of packet to node \( r_a \). Node \( r_i \) accepts the packet and checks whether it has enough pseudo diameter to reach its neighbors. Node \( r_i \) will forward the copies to node \( r_a \) as \( \tau - \tau_i \geq \tau_a \), to node \( r_b \) as \( \tau - \tau_i \geq \tau_b \), and to node \( r_c \) as \( \tau - \tau_i \geq \tau_c \). Observe that the total number of packets generated in this example is 4 where as it is 5 in case of RPF (example1). The reason is that node \( r_s \) did not send a duplicate copy to \( r_a \), whereas in RPF, node \( r_s \) forwards copies of the packet unconditionally to all its neighbors. So, \( r_a \) also receives a copy directly from \( r_s \), unlike in our proposed modified RPF.

2.2 Relevant Data Structures

Since our work is a further modification of the MMRPF, so as in MMRPF every node maintains its DVR table that contains the best delay (minimum cost) information to reach every other node. Every node in the network has a public key (p) and a private key (d). The encrypted text is called as cipher text and is represented by C. Suppose a node \( n_1 \) should send data to node \( n_2 \), securely, then \( n_1 \) encrypts the data using the public key (p) of \( n_2 \). This is represented by \( C = E_p n_2 (M) \). This explains encrypting (E) message (M) with \( n_2 \)'s public key (p). Decrypting the cipher text gives us the original message and it is represented by M.

Now let us assume that there is a multicast group (g). In addition to the individual public and private keys for the nodes in the group g the multicast group also has a unique group public (\( G_p \)) and group private keys (\( G_d \)). A node which first joins the multicast group (say, g) is
considered as the Group Head (GH). Each multicast group can have only one GH. This group head maintains a data structure called GH.table which contains public keys (p) of all the group nodes along with its own public key, private key, and also group public key, group private key. The number of entries in GH.table is equal to the number of members in the group. Each node (n_i) in a multicast group maintains a table called GM.table that contains its own private key (p), group private key (G_d) and group public key (G_p). Therefore, the number of entries in GM.table is only two.

3. ALGORITHM FOR SECURE MULTICASTING IN WANS

Before we present our proposed algorithm we first state briefly how multicasting using modified reverse path forwarding (MMRPF) works.

3.1 Multicasting Using Modified Reverse Path Forwarding (MMRPF) [6]

In this work, distance vector routing is used to decide whether a packet has come along the shortest path. This work is based on the modification of the MMRPF to make it suitable for multicasting. In this approach each packet has three fields: information, source address and a pseudo diameter field.

Example 3:

Table 1 DVR table of A

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Next</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>40</td>
</tr>
<tr>
<td>C</td>
<td>B</td>
<td>60</td>
</tr>
<tr>
<td>D</td>
<td>B</td>
<td>90</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>100</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 2 DVR table of B

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Next</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>40</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>20</td>
</tr>
<tr>
<td>D</td>
<td>C</td>
<td>50</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>60</td>
</tr>
<tr>
<td>F</td>
<td>C</td>
<td>30</td>
</tr>
</tbody>
</table>
In example 3, nodes B, C, E belong to a particular group denoted as g. Node B is the multicasting source. From the DVR table of node B, it is seen that the pseudo-diameter is 60. In Fig. 4 node B finds that its pseudo-diameter is greater than its outgoing link delays. So node B reduces the pseudo-diameter by its respective outgoing link delay and forwards a copy of the packet to each of its links, viz., BA, BC, and BE which are indicated by solid lines. The link BA is shown by dash and dot line implying that A also receives the packet and later sends a prune message to node B (explained later). E finds in its received packet the updated pseudo-diameter as 0 and so it does not forward to anybody. However, since it belongs to the group, so it accepts the packet for itself. A does not belong to any group and in its received packet the updated pseudo-diameter is 20 (= 60-40) wherein the pseudo-diameter is not covering its outgoing link’s delay which is 50. So it cannot forward to node F. Therefore, it sends a prune message back to ‘B’ which is indicated by dash and dot line.

Now after receiving the prune packet, node B will not send any multicast packet to node A until next ‘pruneTimer’ seconds. Similarly nodes D and F send prune messages back to node C which are indicated by dash and dot lines meaning thereby that C will not send multicast packet to them until the next ‘pruneTimer’ seconds. However Node C accepts the packet for itself since it is a group member. Later if D and F want to join the group they can send ‘graft’ packet to node C which is followed by graft acknowledgement from C and then they will again start receiving the multicast packets from C.

Fig. 4 MMRPF Execution
3.2 Security Consideration on MMRPF

To achieve security key distribution must take place among the group members before any multicasting starts. To understand the key distribution clearly we consider the example of Fig.5. Let us assume that node B is the group head (GH) and the masticating group is denoted as ‘g’. Also assume that this group consists of nodes B, C, E. First, group members C and E send their respective private keys \( P_c \) and \( P_e \) to the group head B. It is shown in Fig. 5a. Next, the group head sends the group public key \( (G_p) \) and group private key \( (G_d) \) to C and E respectively in encrypted form. These are: \( [E^{P_c} (G_p, G_d)] \), \( [E^{P_e} (G_p, G_d)] \). It is shown in Fig. 5b. Here \( E^{P_c} \) and \( E^{P_e} \) mean encrypting using public key of C and E respectively. The final data structures containing the key information for nodes B, C, and E are shown in Fig. 5c. It may be observed that for secure multicast a member (not the GH) does not need its public key. Therefore, in Fig. 5b and Fig. 5c for nodes C and E we have not included their public keys in the respective data structures. Note that for key generation the RSA method [9] can be adapted. We shall follow the MMRPF method for multicasting and each multicast packet will be encrypted before being forwarded.

![Key Distribution Diagram](image)

(a) Group members sending public keys (b) Encrypting Group keys (c) Decrypting group keys

Fig. 5 Key distribution

3.3 Algorithm for Secure Multicasting Using Modified RPF (MMRPF)

Below we first state the proposed key distribution algorithm among the group members of a multicast group. After that we will present the secured multicast algorithm.

3.3.1 Algorithm Key Distribution

We assume that each node in the group, \( g \), has its public key and private key
The responsibility of each node $n_i$ (other than GH) is stated below.

At each node $n_i$
1. Step 1: sends its public key $p_i$ to GH;
2. Step 2: receives $[^\text{E}_{p_i}(G_p,G_d)]$ from GH;
3. Step 3: decrypts $[^\text{E}_{p_i}(G_p,G_d)]$ to get $G_p$ and $G_d$;
4. Step 4: it forms its GM.table;
   /* GM.table contains $<G_p, G_d>, <n_i’s p_i, d_i>*/

At group head (GH)
1. Step 1: receives $p_i$ from every member $n_i \in g$;
2. Step 2: it sends $[^\text{E}_{p_i}(G_p,G_d)]$ to the corresponding node $n_i$;
3. Step 3: it forms its GH.table;
   /* GH.table consists of $<G_p and G_d>, <GH’s p_i, d_i>, <p_i , i \in g>*/

3.3.2 Algorithm Secured MMRPF

Input: Each node knows from its DVR table the delay on its links connected to its neighbors and knows its group membership.
Output: Each member of a group will receive a copy of the multicast packet.
At source node (s)
Links = links to neighbors of source s;
pd = pseudo-diameter;
   /*pd is maximum delay value from DVR table*/
for each link in Links loop
   if link is pruned
      continue;
   end-if
   if pd > link-delay on the link
      pd = pd – link-delay;
      /*reduce pseudo-diameter by link delay*/
send $<E_p_g\text{ (mpacket)}, pd>$ to link;
/* send encrypted multicast packet with
the new pseudo diameter*/
end-if
End loop
At intermediate node ($n_i$)
if $<E_p_g\text{ (mpacket)}, pd>$ did not come on best path
    /* RPF check*/
        ignore the packet;
        /* do not forward the packet*/
    end-if
if $n_i \in g$
    accept $<E_p_g\text{ (mpacket)}, pd>$;
    Links = links that are not pruned;
    for each link in Links
        loop
            if this link is the link on which packet
                is received
                ignore the packet;
                /* do not forward the packet*/
            end-if
            if pd > link-delay on the link
                pd = pd – link-delay;
                /* reduce pseudo-diameter by link
delay*/
                send $<E_p_g\text{ (mpacket)}, pd>$ to link;
                /* send encrypted multicast packet
with the new pseudo diameter*/
            end-if
        end loop
    end loop
$D_G_d[ E_p_g \text{ (mpacket) } ];$
    /* Decrypt the packet*/
end-if
4. PERFORMANCE

In [4] every node in a multicast group obtains the public keys of all other nodes in the group. So each node may need a large amount of space (node’s memory) to save every other node’s public key. This is a reasonably high overhead. In our proposed key distribution algorithm we have used the concept of group public and group private keys. Only GH needs to store all the public keys of group members. Every other node needs to save only its own private key, group’s private key and group’s public key. So there is much less space needed per node when compared to the work in [4]. Hence the space overhead in our work is quite low compared to that of [4] it is shown in Fig. 6. In Fig. 6 we have assumed that space taken by each key is 1 byte. It is also shown in Fig. 7 that our work is more bandwidth efficient as the number of control messages generated (i.e., messages needed for key distribution) initially for security purpose is much low compared to that in [4]. Hence Fig. 7 represents our superior bandwidth utilization as well.

![Fig. 6 Comparison with [4] in terms of space](image1)

![Fig. 7 Comparison with [4] in terms of bandwidth](image2)

At present we are in the process of writing simulation software to compare the time our algorithm will take with that of [6] which does not consider security in it.

5. CONCLUSION

In our proposed work we have incorporated security aspect in MMRPF algorithm which has been shown to be more bandwidth efficient compared to DVMRP. We have also shown that our work is superior to the one in [4] from the viewpoint of space overhead and bandwidth utilization.
6. REFERENCES
