

MALICIOUS PACKET DROPPING WITH BOTTLENECK CONSIDERATION IN AD HOC NETWORKS

Yacine Rebahi, Dorgham Sisalem

Fraunhofer Institut Fokus, Kaiserin Augusta Allee 31, 10553 Berlin, Germany

Email: rebahi@fokus.fraunhofer.de, sisalem@fokus.fraunhofer.de

Keywords: malicious packet dropping, bottleneck nodes, ad hoc networks

Abstract: This paper describes a mechanism for detecting malicious packet dropping carried out by bottleneck nodes in ad hoc networks. This technique allows the packets sender, with the collaboration of the destination node, to distinguish between packet loss due to malicious intentions and the one caused by network problems such as congestion.

1 INTRODUCTION

Ad hoc networks are groups of devices that communicate between each other without the support of centralized infrastructure. Although, this technology is promising, some challenges are slowing its development and deployment. Devices in ad hoc networks are in general limited in battery power, CPU and capacity. As a consequence, these devices transmission ranges are limited which make them relying on each other to forward their packets to destination. With devices with limitations in memory and CPU, one can also image limitations in services and security. The factors mentioned earlier, namely the absence of central infrastructure and the devices limitations, have made ad hoc networks more vulnerable to frauds and attacks ranging from passive eavesdropping to active interfering. In fact, an intruder can compromise a node in the network and may eavesdrop on the communications or drop for instance packets, which is supposed to forward them further, to carry out a denial of service attack. If a packets dropping attack occurs, the node sending the packets may misunderstand the reason behind this problem as it might be caused by a broken connection, a hardware limitation such as buffer congestion or a malicious intention. In this case, the packets sender cannot take the right decision and thus will expose himself to a real menace.

In ad hoc networks, every node is a potential victim in the sense that it might be compromised and

used to disrupt the network. However, nodes with a bottleneck character attract more interest in being compromised because of the services they can provide. Securing the network in this situation is a critical task because a compromised bottleneck node can eavesdrop on a significant number of communications or even carry out simultaneous denial of service attacks.

The aim of this paper is to develop a defense mechanism to face denial of service caused by packet dropping. To be more precise, we suggest a technique that allows a source node to distinguish between what is malicious and spurious regarding a packet dropping performed by its next hop node that is a bottleneck node and which the source node is relying on to forward the packets to their destination. This mechanism requires the support of the destination node. In fact, the destination node is observing the packets that the intermediate bottleneck node is forwarding and when it detects a misbehavior, it notifies the source node. The latter, in turn and based on some information that he has about this intermediate node, estimates the natural congestion level at this node and compare it with the received information. If they mismatch, the intermediate node is declared to be an intruder. Our mechanism can be applied as a monitoring technique for ad hoc network models similar to the ones investigated herewith and which build trust between

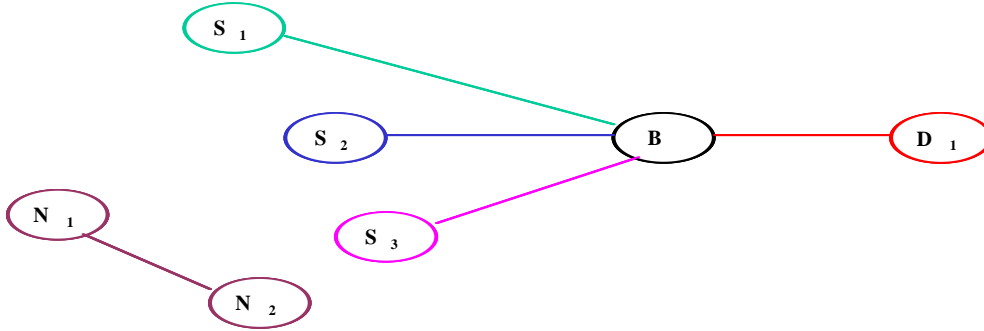


Figure 1: Network Model

- their nodes based on the reputation concept. Among these reputation-based trust techniques, one can mention CONFIDANT (Buchegger, 2002).

Malicious packet dropping was investigated in (Zhang, 2000) and (Rao, 2001). In (Zhang, 2000), the impact of a set of TCP packet dropping attack patterns on FTP file transfer was investigated. In addition, a statistic-based approach to detect malicious attacks was proposed. In (Rao, 2001), a statistic approach is also suggested to detect whether bottleneck nodes are maliciously dropping packets. Even the objectives and the models studied in this paper and in (Rao, 2001) are similar, the used techniques differ in two major points. First of all and contrary to (Rao, 2001) in which, the destination is in charge of determining whether the bottleneck is an intruder, our model assumes that this task is achieved by the source nodes. This makes, in particular, our technique a means for neighborhood monitoring for reputation-based protocols such as CONFIDANT as mentioned earlier. Secondly, our approach is not statistic-based as in (Rao, 2001), but uses a deterministic model to detect malicious packet dropping.

The rest of this paper is organized as follows, sect 3 describes our mechanism for detecting malicious packet dropping and the technique utilized to protect the sensitive exchanged data between the source and the destination nodes. Sect 3 concludes the paper.

2 PROPOSED ARCHITECTURE

In this section, we present a mechanism for detecting denial of service attacks caused by packet dropping and carried out by bottleneck nodes.

Let's consider the simple network model depicted in Figure 1. This model involves different

source nodes S_1, S_2 and S_3 which might belong to different transmission ranges, and a destination node denoted by D_1 . The node B is an intermediate node that is used by all the source nodes to transmit packets to the destination node D_1 . Note that the source nodes that use node B to forward their packets to some destinations other than node D_1 are not shown in this figure.

As mentioned earlier, node B may drop packets to carry out a denial of service attack. However, the source nodes, which are sending the packets may misunderstand the reason behind this. Distinguishing between what is malicious and spurious can only be achieved by determining the true congestion level at the bottleneck node, node B . With other respects, collaboration between nodes in ad hoc networks is mandatory for any service provision achievement, in particular for packet dropping detection. To be more precise, we require collaboration between the source and the destination nodes in order to face malicious packet dropping threats as it will be clarified in the sequel.

2.1 ONE FLOW MODEL

Since we seek simplicity, we will start with a network with one source node, node S_1 . The general model will be discussed in sect 2.2. Let's also make the following assumptions,

- S_1 is sending packets of a fixed size at a constant rate α p/s towards node D_1
- Node B forwards the packets that it receives from the source node at a constant rate β p/s towards their destination. A way for deriving the rate β is as follows: node D_1 simply checks the number of packets arriving from node B during a time period T and computes an estimate of the packet transmission rate of node B . The computed value will be set to the rate β . This value will be also sent in a secure way (similar to

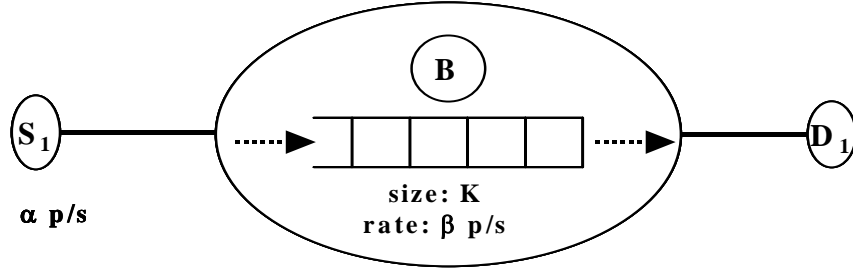


Figure 2: Leaky Bucket Model

the one described in sect 2.3) to the source node

- Node B uses one single queue buffer (see Figure 2) of size K
- The source node is aware of node B 's buffer size

Each TCP packet has a sequence number that allows the destination node to ensure that the packets being sent are arriving in a sequence. To be more precise, the sequence number simply refers to a byte position in the TCP stream 0. As we have already assumed that the source nodes sent data in fixed segments, let's then denote by l the size of these segments. In fact, there is a simple relation that links the packet number i to the sequence number SN_i of the first byte in this packet,

$$i = \frac{SN_i}{l} + 1 \quad (1)$$

Using sequence numbers, node D_1 can periodically monitoring the packet loss occurred at the bottleneck node, node B . It turns out that at time T , the number of packets that node B has dropped (maliciously or not) is expressed by the formula,

$$P_{loss,T} = \frac{SN_i - (SN_j + l - 1)}{l} \quad (2)$$

where SN_i is the sequence number of the first byte of the last packet arrived to the destination node and $(SN_j + l - 1)$ is the sequence number of the last byte acknowledged 0. If the value $P_{loss,T}$ is strictly greater than 0, this means that some packets sent by the source node are dropped by node B and the sender, i.e node S_1 has to be informed. Note that node B may alter the data exchanged between the nodes S_1 and D_1 , in particular modify the sequence

numbers of the packets that it is forwarding to node D_1 just to hide that some packets are lost. To prevent any information access by node B , the packets payloads must be encrypted. The way exchanged information is encrypted will be discussed in sect 2.3.

Let's assume now that node B is dropping packets destined to node D_1 . Let's also assume that the latter has detected this misbehaviour and sent the corresponding information to node S_1 . In order to check whether a malicious intention is behind this packet loss, node S_1 has to compute an approximate value of the natural congestion level at node B and consequently the "natural" resulting packet loss and compare it with the information received from node D_1 . From Figure 2, one can see that this situation is very similar to the leaky bucket problem (Le Boudec, 2002).

For simplicity, we assume that the buffer of node B was empty at time 0. In this case, the content of the buffer at time T is given by,

$$cont_T = (\alpha - \beta)T \quad (3)$$

If there is an overflow of the buffer at time T , this will be reflected by the value $(\alpha - \beta)T - K$. Note that the value $(\alpha - \beta)T - K$ might be negative if there is no overflow of the node B ' buffer. As a result, the packet loss can be described in a general way by the following formula,

$$P_{loss-approx,T} = Max[0, (\alpha - \beta)T - K] \quad (4)$$

where $Max(X, Y)$ is the maximum of X and Y . Here, the value $P_{loss-approx,T}$ provides an estimate of the packet loss at node B that can be caused by a natural congestion problem at time T . Node S_1 has to compare the value $P_{loss-approx,T}$ with the value $P_{loss,T}$ provided by the formula (2), in order to determine whether node B is dropping packets maliciously. As $P_{loss-approx,T}$ and $P_{loss,T}$ are not exact values, one can consider a threshold

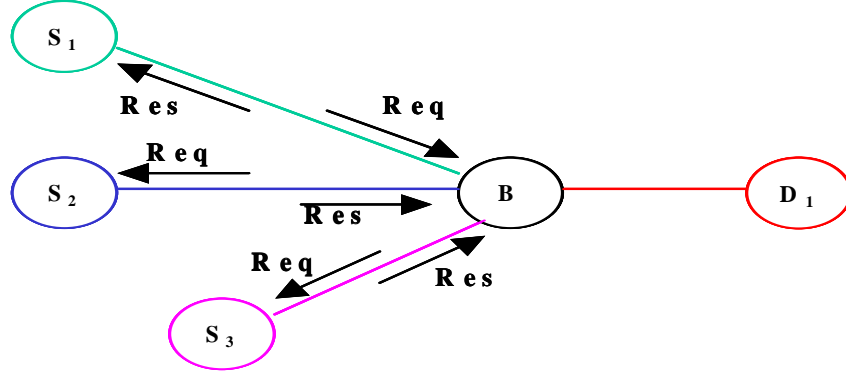


Figure 3: Nodes Throughput Rates Information Exchange

value, that is when exceeded, node B is considered as a malicious node. In other words, if

$$|P_{loss,T} - P_{loss-approx,T}| > P_{threshold} \quad (5)$$

node B is presumed to be an intruder. The notation $|x|$ simply refers to the absolute value of x .

2.2 MULTI-FLOW MODEL

Let's consider now the general model and assume that both nodes S_1 , S_2 and S_3 are emitting packets towards node D_1 . If the latter notes that node B is dropping packets, the source nodes need to be informed. Node D_1 can use the IP addresses of the source nodes to correctly bind the packets loss information to the sender of the corresponding packets.

Node B is forwarding the packets issued by node S_1 towards node D_1 but also the packets issued by some other nodes, in particular the nodes S_2 and S_3 . Thus, if node S_1 computes an estimate of the packet loss at node B based only on its throughput, formula (4), this will lead to an incorrect evaluation of node B . However, the only way to have a reasonable and fair estimate of the congestion level at node B is to know the throughput of all the nodes that are one hop away from node B and which rely on it to forward their packets. To get the throughput rates of the other nodes, node S_1 must contact them. Unfortunately, this can be achieved only through node B , which is considered so far by node S_1 as a suspicious node. The way the throughput rates information is gathered is as follows (see Figure 3): node S_1 sends a request to node B asking it to contact its one hop neighboring nodes that are relying on it to transmit their packets, to send to node S_1 their throughput rates. Node B has to cooperate and forward this request to these

nodes, otherwise, it will be under-estimated by node S_1 that consequently might choose another path. Here again, the information exchanged between node S_1 and the neighboring nodes of node B , in particular the nodes S_2 and S_3 , is secured as it will be clarified in sect 2.3. This will prevent node B to alter this data while forwarding the packets.

2.3 SECURE INFORMATION EXCHANGE

The communications between the source and the destination nodes in the network have to be secured especially that,

- If the sequence numbers of the packets that node B is forwarding, are not encrypted, node B can access this information and modify it
- If sensitive information such as misbehaving node suspicion is treated by the source node without paying attention to the identity of the sender and the integrity of the data, the source node will face a real threat because node B might have accessed this information and changed it
- If when requesting the throughput rates from the neighbours of node B as describe in sect 2.2, this information is transmitted without any integrity detection mechanism, node B can also in this case access the corresponding data and alter it

The just mentioned issues can be addressed using secure associations between the source and the destination nodes. A secure association can be achieved either through a secret key that the source and the destination nodes share or through a public key system. IPSec (Kent, 2004) is an example of the

first category, however, TLS (Dirks, 1999) reflects the second one and both are robust enough to be used for establishing the secure associations.

The keys used to build the secure association, will be used to encrypt the sequence numbers of the exchanged packets. In this case, node B cannot change the involved information.

On the other side, both the misbehaving suspicion information and throughput rates data will be signed by the keys used for establishing the secure association. This will allow the source and the destination nodes to authenticate the sender and check the integrity of the transmitted data.

3 CONCLUSION

This paper discusses first security problems in ad hoc networks and the ways currently used to face them. These security mechanisms are not enough and cannot prevent malicious nodes, which got already access to the network resources, to carry out denial of services attacks such as packet dropping. This situation is more critical if these malicious nodes are some bottleneck nodes that are used to forward packets to their destinations. This problem is discussed in details here and a technique that allows the packets sender to determine whether the intermediate bottleneck node is dropping packets maliciously or spuriously, is provided. As this technique requires collaboration between nodes in the network, a way of securing sensitive exchanged data is also described.

REFERENCES

- Buchegger, S., et Al, 2002. Performance Analysis of the COFIDANT Protocol: Cooperation Of Nodes- Fairness In Dynamic Ad hoc NeTworks. In *IC/2002/01*.
- Zhang, X., et Al, 2000. Malicious Packet Dropping: How it Might Impact the TCP Performance and How we can Detect it. In *IEEE ICNP 2000*.
- Rao, R. and Kesidis, G., 2001. Detecting Malicious Packet Dropping Using Statistically Regular Traffic Patterns in Multihop Wireless Networks that are not Bandwidth Limited. In *Proc. Of IEEE Globecom*. San Francisco, CA, USA.
- Stallings, W., 1999. *Cryptography and Network Security*, Prentice Hall, N.J.
- Luo, H., et Al, 2000. Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks. *UCLA-CSD-TR-200030*.
- Hubaux, J. P., et Al, 1979. Self-Organized Public key Management for Mobile Ad Hoc Networks.
- Shamir, A., 1979. How to share a secret. In *Communications of ACM*.
- Comer, D. E., 1995. *Internetworking with TCP/IP: Principles, Protocols and Architectures*. Prentice Hall, Fourth Edition, Vol 1.
- Le Boudec, J. Y., et Al, 2002. *Network Calculus, A Theory of Deterministic Queuing Systems for the Internet*. Springer Verlag- LNCS 2050, Online version of the book.
- Dahill, B., et Al, 2001. A Secure Routing Protocol for Ad Hoc Networks. *Tech. Rep. 01-37*, Department of Computer Science, University of Massachusetts.
- Zapata, M., et Al, 2002. Securing Ad Hoc Routing Protocols. *ACM WiSe 2002*.
- Yi, S., et Al, 2001. Security-Aware Ad hoc Routing for Wireless Networks. *MobiHOC 2001*, Long Beach, CA, USA.
- Carter, H., et Al, 2001. Secure Position Aided Ad hoc Routing Protocol. *Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02)*.
- Papadimitratos, P., et AL, 2003. Secure Link State Routing for Mobile Ad Hoc Networks. *IEEE Workshop on Security and Assurance in Ad hoc Networks, in conjunction with the 2003 International Symposium on Applications and the Internet*. Orlando, FL.
- Zhou, L., et Al, 1999. Securing Ad Hoc Networks. *IEEE Network Magazine*. Vol. 13, no.6, November-December 1999.
- Diffie, W., et Al, 1976. New Direction in Cryptography. *IEEE Tran. On Info. Theory*, Vol. IT-22, pp. 644-654.
- Dirks, T., et AL, 1999. The TLS Protocol, version 1.0. *RFC 2246*, January 1999.
- Kent, S., et Al, 2004. Security Architecture for the Internet Protocol. *draft-ietf-ipsec-rfc2401bis-05.txt*, Dec2004.