International Journal of Advanced Science and Technology
Vol. 29, No. 8, (2020), pp. 64-71

Intrusion Detection and Prevention System for DDoS Attacks in 5G Network using SDN

R.Jeya[1], B.Amutha[2], K.Shreya[3], Kavita Teltia[4]

[1,2]*Departement of Computer Science and Engineering, SRM Institute of Science and Technology, India*
[3,4]*Undergraduate Student, Branch of Computer Science and Engineering,*
SRM institute of Science and Technology, India
[1]*jeyar@srmist.edu.in,* [2]*amuthab@srmist.edu.in,* [3]*shreya_kalyan@srmuniv.edu.in,*
[4]*kavitateltia_suresh@srmuniv.edu.in*

Abstract

This paper provides the implementation of an intrusion detection and prevention system to research the security capacities of Software Defined Networking (SDN) in a 5G-like environment under Distributed Denial of-Service (DDoS) attacks. OMNeT++ along with two extension libraries, SimuLTE and OpenFlow OMNeT++ Suite is used as the simulator. The distributed synchronize (SYN) flood attack performed by malicious hosts is analyzed and reported. The algorithm implemented provides an improvement in detection and prevention of DDoS attack in 5G network using SDN from the base paper taken as reference.

Keywords: *5G Network, Intrusion Detection System (IDS), Intrusion Prevention System (IPS), Software Defined Networking (SDN), DDoS Attack and OMNET++*

1. Introduction

Two primary changes lead the progress towards the 5G mobile network: (i) developmental change, where higher data rates, capacity and connectivity will be offered by incorporating existing and new radio access technologies, and (ii) progressive change, where heterogeneous administrations with various prerequisites will be given over a similar foundation. Software Defined Networking (SDN) and Network Function Virtualization (NFV) are the key empowering advancements, specifically, for the subsequent change. SDN divides the control plane from the data plane, in this manner, it empowers a centralized control over the system by means of a SDN controller. NFV is exceptionally complementary to SDN, and it actualizes network administrations, that are generally run on exclusive dedicated hardware, as system works on off-the-rack hardware.

While SDN as of now used to build interface usage [5], utilizing SDN and NFV for security objects is still in a beginning period. In a security-setting, SDN has some helpful qualities. The SDN controller has a worldwide perspective on the network, centralized control and gives programmability of the network components. It screens and accumulates network statistics through the southbound API. The controller can utilize its worldwide perspective on the system to distinguish malicious patterns, i.e. intrusion detection system (IDS), and its centralized control to react to occasions by refreshing stream tables in the network components to filter through the traffic or divert it to an intrusion prevention system (IPS). Each early threat detection at any network area and quick response at run-time is particularly significant in 5G networks where many use-cases and clients will be served at the same time over a similar foundation.

2. State of the art (Literature Survey)

The authors in [1] recognized eight security challenges in 5G. Two difficulties are about the threat of Denial of Service (DoS) attacks, where both the infrastructure and end-user devices are vulnerable.

ISSN: 2005-4238 IJAST

Applying SDN for detection and mitigation of various forms of TCP SYN Scanning attack was proposed in [3]. The authors of [10] applied four kinds of security capacities with SDN: in-line mode (e.g. firewalls and IPS), passive mode (e.g. IDS), network anomaly detection (e.g. scan and DDoS detector), and advanced security functions (e.g. stateful firewall and reflector networks). Reference [6] proposed running intrusion detection as a service (DaaS) with SDN for anomaly detection system in mobile network operators. The proposed architecture contains various DaaS nodes which analyses each packet received and notify the SDN application when a malicious packet is detected, yet it doesn't indicate the detection algorithm operated by the DaaS nodes. Reference [] has taken the past references as a base and a system was implemented to detect and prevent DDoS attack on the 5G network.

This paper provides the implementation of an intrusion detection and prevention system to research the security capacities of Software Defined Networking (SDN) in a 5G-like environment under Distributed Denial of-Service (DDoS) attacks. OMNeT++ along with two extension libraries, SimuLTE and OpenFlow OMNeT++ Suite is used as the simulator. The distributed synchronize (SYN) flood attack performed by malicious hosts is analyzed and reported. The algorithm implemented in the proposed system gives an improvement in the detection and mitigation of DDoS attack in 5G network using SDN from the base paper taken as reference.

3. Proposed Work

In this section we have discussed about the new proposed work called the Trust Openflow Traffic Aware Routing to make sure that the 5G network is safe and secure from DDoS Attack. In this paper we compare this method with the connection rate-based detection method to eliminate the drawback of the connection rate-based method. In Trust Openflow Traffic Aware Routing, the attacks are initiated by the hosts or malicious attackers connected to the local servers. The openflow switches makes sure that the attack is detected and prevented or mitigated. Intrusion Detection System (IDS) identifies the attack that takes place in the network. It also tells us the details of the malicious host.

Intrusion Prevention System (IPS) or Mitigation is used to protect the network from the attack that was identified. Along with detection and prevention we are going to look at the key factor of importance which is known as congestion reduction. When there is an overflow of requests coming from multiple hosts in the network to the server through a switch, the switch may not be able to process and send all the data from all the hosts at the same due to overcrowding of requests coming in the form of data packets. So, to decrease the overcrowding we make sure that a part of packets is kept in buffer.
The Trust Openflow Traffic Aware Routing algorithm does detection and prevention of DDoS attack along with congestion reduction of traffic in the openflow switch.
We are taking performance based on three parameters:

1. Through put ratio - Over all network performance with DDoS attack.
2. Packet delivery ratio - the ratio of packets successfully received to the total sent.
3. Average end to end delay - time taken for a packet to be transmitted across a network from source to destination.

When a DDoS assault is recognized, mitigation procedures like dropping packets, blocking ports, or on the other hand redirection of traffic can be effortlessly performed utilizing SDN. SDN can uphold these procedures by sending ow-mod messages to its switches and put in new or alter existing ow passages.

To furnish OMNeT++ with capacities of reproducing SDN in a 5G-like condition, the third-party augmentation OpenFlow OMNeT++ Suite has been picked as a lot of libraries for SDN-based traffic steering and SimuLTE as a measured framework level test sysm for LTE-A systems. OpenFlow OMNeT++ Suite [4] is a library that incorporates OMNeT++ modules for SDN switch furthermore, SDN

controller, and it executes the OpenFlow convention for correspondence between the switches and the controller. SimuLTE [9] is a test system dependent on OMNeT++ that permits simlating LTE and LTE-A remote systems. It executes a full convention stack as indicated by LTE gauges, and it gives modules to advanced NodeB (eNB), User Equipment (UE), Packet Information Network Gateway (PGW), just as a sensible portrayal of planning and physical layer. The two libraries depend on the INET library, which gives the most well-known Internet conventions and backing for portability and remote advancements. In this manner, it permits setting up an flexible and blended situation in with LTE and SDN as parts of a more extensive correspondence arrange. The working form was introduced on Ubuntu 16.04 Operating System, made out of OMNeT++ 4.6, INET 2.5, simuLTE v0.9.1 and OpenFlow OMNeT++ Suite. Figure 5.1 shows our proposed 5G-like engineering. The topology comprises of seven hub types: ue, eNodeB, and pgw are modules imported by the SimuLTE library; open stream switch and open stream controller are modules taken from the OpenFlow OMNeT++ Suite library; and the server is a nonexclusive Internet server that answers demands from UE associated with eNB. So as to interconnect the SimuLTE and OpenFlow OMNeT++ Suite, a standard switch was presented. We have discussed about the network architecture of the proposed system. We are considering that the communication is taking place between either two different cities or countries.

As given in the Figure 1 the global server is the main server which is the center of communication to both networks. In the first network we have four local servers which is connected to switch called the openflow_switch1. This switch is in turn connected to the global server. In the second network we have three local servers connected to another switch called openflow_switch2. This switch is connected to the data center which is connected to global server. OpenFlow as the first standard communications interface defined between the control and forwarding layers of an SDN architecture. The two openflow switches are integrated with 5G network. Each of the local server is connected to several hosts. The architecture has been more deeply established in Figure 2 where we see how the packets are being transferred and the attack that is taking plcae on the nodes.

This topology somewhat varies from a perfect 5G situation, where the eNB would be legitimately associated with the SDN switch, with the LTE convention handling capacity being facilitated in a remote/cloud server. However, all the traffic from UEs to the server ows through the SDN switch, permitting a total observing of the traffic by the SDN controller. This topology can be considered as a substantial simplication that doesn't influence the nature of the outcomes. The SDN switch and controller speak profoundly of IDS. Figure 3 shows the OMNeT++ modules of SDN switch and controller. The traffic from the LTE UE to the server is steered through the SDN switch. Through the SDN controller, the SDN switch permits traffic checking also, it performs alleviation activities if there should arise an occurrence of an interruption location. A SDN DDoS Defense controller application has been created. The application controls the conduct of the open stream controller by indicating the steering rationale, how to deal with parcel in messages from the open stream switch and how to develop ow mod and bundle out messages. It is actualized as an adjusted variant of the OpenFlow controller application Learning Switch, which is remembered for the OpenFlow OMNeT++ Suite bundle. The application works in two phases, identification stage and relief stage. The open stream controller keeps an outline of the association endeavor pace of all dynamic has in the system. Rather than keeping a table with the condition of all SYN endeavors for every IP address in the system, the controller isolates TCP SYN parcels from different bundles.

Our simplied DDoS discovery and relief strategies are given with method together with the C++ contents of the OpenFlow OMNeT++ Suite that have been changed. The coordinating is done on the fields for Ethernet type, approaching port, source MAC address, goal MAC address and TCP SYN ag, while extraordinary ow passages are made for any bundle that has the TCP SYN ag raised. These exceptional ow sections incorporate extra match fields of the ACK ag, IP source address and IP goal address. The ACK ag is incorporated to isolate SYN messages from SYN-ACK messages. The IP source and goal

delivers are incorporated to recognize the have that sends a strangely high number of TCP SYN messages to a specific target. For each SYN parcel, if the pair (IP source, IP destination) isn't as of now arranged as noxious, at that point the switch contrasts the limit esteem T and its counter worth. When the limit of a ow section is outperformed, the open stream switch is told to transmit bundle in that contains the activating parcel and a special incentive for the field parcel in motivation to the open stream controller. The controller application answers to the uncommon parcel in with ow mod message training the change to change the activity of the individual ow passage to drop, just as a bundle out message teaching the change to drop the activating packets. Any SYN bundle that coordinates this ow section is dropped when it enters the switch, relieving the debilitating assault on the server.

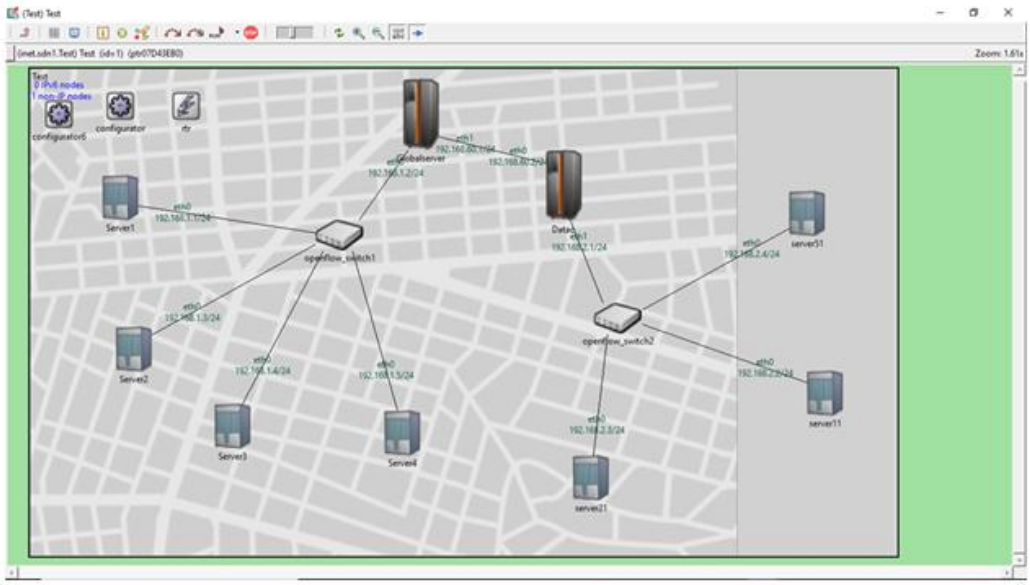

FIGURE 1. Basic Topology of Network

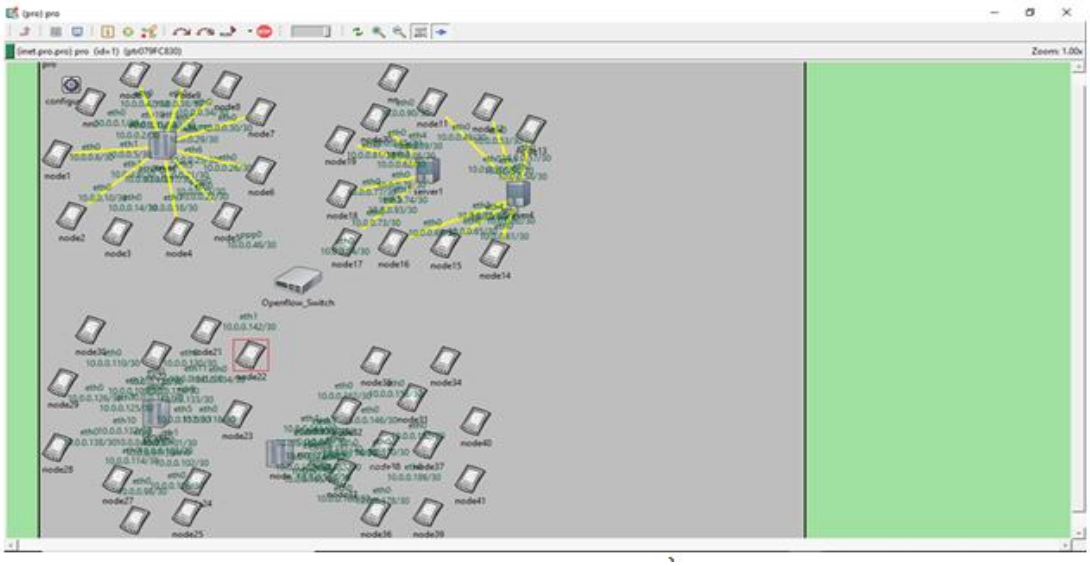

FIGURE 2. Working of the Network

4. Implementation

Trust Openflow Traffic Aware Routing Algorithm

```
//**********Trust_Function**********
void TOTAR:trust_lookup(nsaddr_t node_id)
{
trust_entry *rp;
 //assert(tr_lookup(dst_seq_no) == 0);
//routing protocol
 rp = new trust_entry;
    assert(rp);
    rp->node_id = node_id;
    rp->prev_node = prev_node;
    rp->next_node = next_node;
    rp->trust_value = trust_value;
    LIST_INSERT_HEAD(&trusthead, rp, trust_link);
    return rp;
 trust_entry *rp = trusthead.lh_first;
     for ( rp; rp = rp->trust_link.le_next) {
           if (rp->node_id == node_id)
              break;
}
    return rp;
}
void TOTAR::rt_purge()
{
TOTAR_rt_entry *rt, *rtn;
double now = CURRENT_TIME;
double delay = 0.0;
Packet *p;
     for(rt = rtable.head(); rt; rt = rtn) {  // for each rt entry
   rtn = rt->rt_link.le_next;
   if ((rt->rt_flags == RTF_UP) && (rt->rt_expire < now)) {
   // if a valid route has expired, purge all packets from
   // send buffer and invalidate the route.
     assert(rt->rt_hops != INFINITY2);
     while((p = rqueue.deque(rt->rt_dst))) {
#ifdef DEBUG
       fprintf(stderr, "%s: calling drop()\n",
                    __FUNCTION__);
#endif // DEBUG
       drop(p, DROP_RTR_NO_ROUTE);
     }
     rt->rt_seqno++;
     assert (rt->rt_seqno%2);
     rt_down(rt);
   }
   else if (rt->rt_flags == RTF_UP) {
   // If the route is not expired, and there are packets in the //sendbuffer waiting, forward them. This should
not be //needed, but this extra check does no harm.
     assert(rt->rt_hops != INFINITY2);
```

```
    while((p = rqueue.deque(rt->rt_dst))) {
      forward (rt, p, delay);
      delay += ARP_DELAY;
    }
  }
  else if (rqueue.find(rt->rt_dst))
  // If the route is down and
  // if there is a packet for this destination waiting in
  // the sendbuffer, then send out route request. sendRequest
  // will check whether it is time to really send out request
  // or not.
  // This may not be crucial to do it here, as each generated
  // packet will do a sendRequest anyway.

    sendRequest(rt->rt_dst);
  }

}
```

5. Results Discussion

The three parameters i.e. throughput ratio, average end to end delay and packet delivery ratio are plotted and represented graphically. The green line in the graphs represent the prosed system and the redline represents the existing

The first graph compares the throughput ratio of existing system with proposed system. Here we can see that the proposed system is more efficient and effective as compared to the existing system. The overall network performance under DDoS attack simulated is better with proposed system.

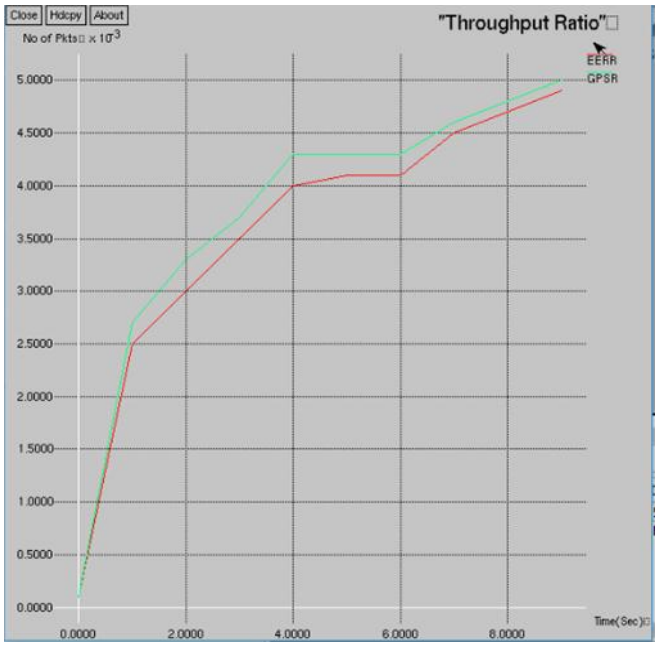

FIGURE 3. Comparing Throughput Ratio with Proposed and Existing System

Now, the second graph compares average end to end delay between the existing system and proposed system. Here we can see that the proposed system is more efficient and effective as compared to the existing system. Unlike the existing system, the delay of transfer of data with DDoS Attack simulated is of less time in the proposed system.

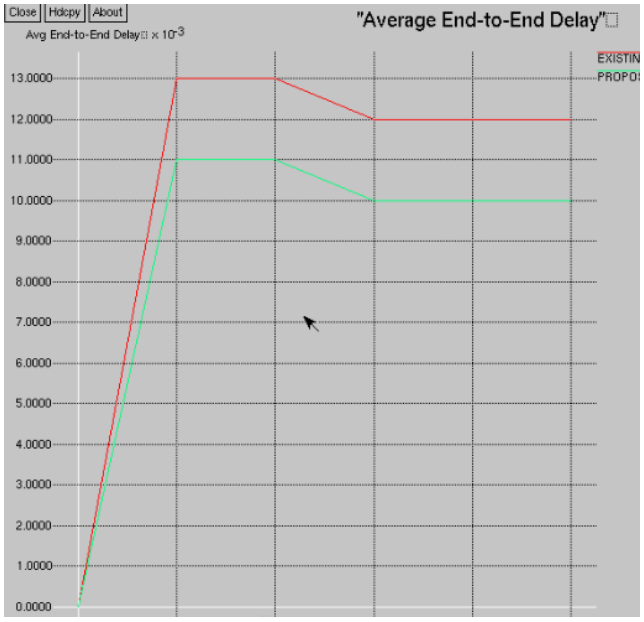

FIGURE 4. Comparing Average End to End Delay with Proposed and Existing System

The last graph compares packet delivery ratio between the existing system and proposed system. Here we can see that the proposed system is more efficient and effective as compared to the existing system. Unlike the existing system, the delay of transfer of data with DDoS Attack simulated is of less time in the proposed system

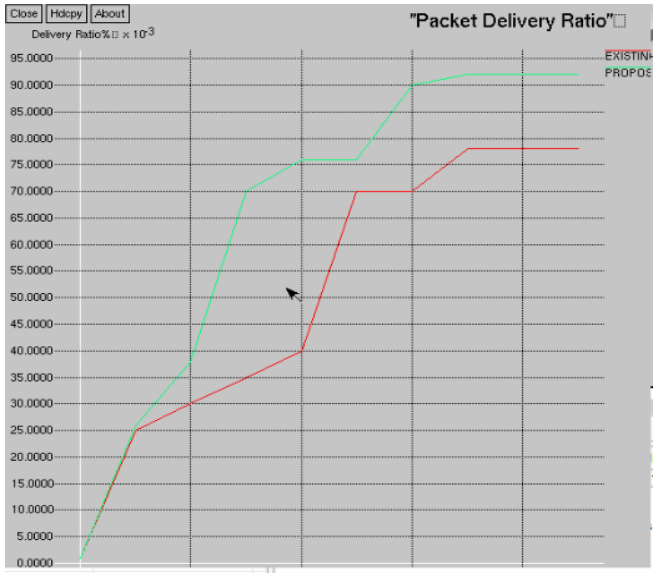

FIGURE 5. Comparing Packet Delivery Ratio with Proposed and Existing System

6. Conclusion

In this paper we have made an IDS/IPS for DDoS attacks in 5G network using SDN controller in OMNET++ simulation tool. The proposed algorithm in this paper is the Trust Openflow Traffic Aware Routing algorithm which detects and prevents or mitigates the DDoS attack along with congestion reduction of traffic.

References

[1] Hai-xiang Guo, Ke-jun Zhu, Si-wei Gao, Ting Liu, ”An Improved Genetic k-means Algorithm for Optimal Clustering”, Sixth IEEE International Conference on Data Mining (2006).
[2] Tou J. T, Gonzalez R. C, ”Pattern recognition principle.”, Addison Wesley, Reading, MA, 1974.
[3] Adetunmbi A.Olusola., Adeola S.Oladele. and Daramola O.Abosede, ”Analysis of KDD’99 Intrusion Detection Dataset for Selection of Relevance Features.”, Proceedings of the World Congress on Engineering and Computer Science 2010 Vol I, WECECS 2010, October 20-22, 2010.
[4] Dr.S.Vijayarani and Ms. Maria Sylviaa.S, INTRUSION DETECTION SYSTEM-A STUDY.”,International Journal of Security, Privacy and Trust Management (IJSPTM) Vol4, No 1, February 2015
[5] K. Jayan and A. K. Rajan, ”Sys-log classifier for complex event processing system in network security,” in Advances in Computing, Communications and Informatics (ICACCI, 2016 International Conference on IEEE), 2016, pp. 2031-2035.
[6] K. Jayan and A. K. Rajan,”Preprocessor for complex event processing system in network security,” in Advances in Computing, Communications (ICACC), 2014 Fourth International Conference on. IEEE , 2014, pp.187-189.
[7] P. M S, V. Hedge, H.Anushadevi, and V. Ambika,”Automated spam detection in email using svm,” vol. 10, pp. 25219-25228, 01 2015
[8] S. Patil, P. Rane, Dr. B. B. Meshram, “IDS vs IPS”, International Journal of Computer Networks and Wireless Communications, Vol. 2, No. 1, 2012.
[9] K. Scarfone, P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS)” NIST Publication, 2007.
[10] J. Aldwairi, M. Monther, and D. Alansari. "Exscind: Fast pattern matching for intrusion detection using exclusion and inclusion filters." Next Generation Web Services Practices (NWeSP), 7th International Conference on. IEEE, 2011.
[11] B. Jakub, P. Buciak, and P. Sapiecha. "Building dependable intrusion prevention systems." Dependability of Computer Systems, International Conference on. IEEE, 2006.
[12] T.H. Corman, C.E. Leiserson, R.L. Rivest and C. Stein, “Introduction to Algorithms”, The MIT Press, 2009, Cambridge, Massachusetts London, England.
[13] M. R. Karp and M. O. Rabin. Efficient randomized pattern-matching algorithms.” IBM Journal of Research and Development, Vol. 31, No. 2, 1987.
[14] D.E. Knuth, J.H. Morris, and V.R. Pratt. "Fast pattern matching in strings." SIAM journal on computing 6.2, pp. 323-350,1977.
[15] R. S. Boyer and J. S. Moore. “A fast string searching algorithm.” In Communications of the ACM, Vol. 20, pp. 762–772, October 1977.54