DESIGN AND DEVELOP MEASUREMENT METHODS TO PREVENT BANDWIDTH ANONYMITY SERVICE ATTACKING IN HETEROGENEOUS WIRELESS NETWORKS

Dr. P. Chitti Babu[1], D. J. Samatha Naidu[2], P. Sai Krishna Babu[3]
[1]*Dr. P. Chitti Babu, Principal MCA, APGCCS, Andhra Pradesh, India*
[2] *D. J. Samatha Naidu, Assistant Professor MCA, APGCCS, Andhra Pradesh, India*
[3] *P.Sai Krishna Babu ,Student MCA, APGCCS, Andhra Pradesh, India*

ABSTRACT

The state of the secure bandwidth measurement purpose needed to design, analysis, implementation and evaluation a new bandwidth measurements algorithm, we called it monitor trainer, for securing in cooperative bandwidth measurement in adversarial networking environment. ***In Existing Work,*** *the bandwidth inflation by a consensus view i.e opportunistic bandwidth measurements or detect bandwidth frauds via existing methods and techniques. The bandwidth inflation attacks are very insidious. From a user perspective, maintaining a good quality of the transferred video is critical. The model is, however, only valid for single-hop communication. Users Cannot Control Bandwidth Measurement. Opportunistic bandwidth measurement method has been designed to secure P2P bandwidth evaluation systems only. Existing system does not provides bandwidth Controlling Which Leads To Inflation Attacks.* ***The Proposed Work,*** *designing to use an unpredictable measurement for training set, to defeat even the smart adversaries. The proposed new measurements delay algorithm to secure capacity measurements for the first time itself.*

Keyword :*Introduction, Existing Work, Proposed Work,*

1. Introduction

The state of the secure bandwidth measurement purpose needed to design, analysis, implementation and evaluation a new bandwidth measurements algorithm, we called it monitor trainer, for securing in cooperative bandwidth measurement in adversarial networking environment. In Existing Work the bandwidth inflation by a consensus view i.e opportunistic bandwidth measurements or detect bandwidth frauds via existing methods and techniques. The bandwidth inflation attacks are very insidious. Adversarial hosts can simply delay the leading packet or rush the trailing one that the measurements packet’s a priori information. The existing measurements cannot differential whether the observed rand trip delay has been maliciously. The model is, however, only valid for single-hop communication. Users Cannot Control Bandwidth Measurement. Opportunistic bandwidth measurement method has been designed to secure P2P bandwidth evaluation systems Only. Existing system does not provide bandwidth controlling which leads to inflation attacks.

2. Existing Work

Since many networking systems highly depend on bandwidth measurement for their services optimization and load balancing false bandwidth reports could render these systems unreliable and vulnerable. A typical attack falling into this attacking category is bandwidth inflation attack, by which adversarial hosts can falsely report a larger bandwidth to others. With such attack, adversarial peers in peer-to-peer file sharing networks can induce more traffic from other peers by reporting to others an inflated bandwidth. Moreover, the bandwidth inflation attack is very insidious. In many cases, it does not do direct harm to its victims. Instead, it is often used to increase the efficiency of other attacks. For example, adversarial proxies in video caching systems can pretend to have a larger bandwidth to receive more videos for caching and later embed advertisements into these videos for profits or redirect more victim users to malicious videos. In onion routing systems, adversarial routers usually exploit bandwidth inflation to enlarge the victim population under their correlation-like attacks. Recent research has demonstrated a successful bandwidth inflation from 4 Mbps to 50 Mbps in the Tor network, therefore exposing the hidden services through correlation-like attacks more efficiently and with lower cost.

2.1. Existing Work with limitations

In Existing Work, the bandwidth inflation by a consensus view i.e opportunistic bandwith measurements or detect bandwidth frauds via existing methods and techniques. The bandwidth inflation attacks are very insidious.

Major Limitations

- Adversarial hosts can simply delay the leading packet or rush the trailing one that the measurements packet's a priori information.
- The existing measurements cannot differential whether the observed rand trip delay has been maliciously.
- The model is, however, only valid for single-hop communication.
- Users Cannot Control Bandwidth Measurement.
- Opportunistic bandwidth measurement method has been designed to secure P2P bandwidth evaluation systems Only.
- Existing system does not provide bandwidth controlling which leads to inflation attacks

2. Proposed Work

Novel measurement method to detect smart bandwidth inflation frauds. Consensus analyses or forgeable tricks, our method employs an unpredictable, yet long enough, packet train (we call it monitor trainer) instead. Monitor trainer is designed with unpredictable elements adversarial hosts cannot have a priori knowledge. smart adversaries can still have a chance to evade the above mentioned detection methods required. Novel measurement method to detect smart bandwidth inflation frauds. Consensus analyses or forgeable tricks, our method employs an unpredictable, yet long enough, packet train (we call it monitor trainer) instead. Train is designed with unpredictable elements, adversarial hosts cannot have a priori knowledge. Smart adversaries can still have a chance to evade the mentioned detection method.

Contribution Work

- Novel measurement method to detect smart bandwidth inflation frauds.
- Consensus analyses or forgeable tricks, our method employs an unpredictable, yet long enough, packet train (we call it monitor trainer) instead.
- Monitor trainer is designed with unpredictable elements adversarial hosts cannot have a priori knowledge. smart adversaries can still have a chance to evade the above mentioned detection methods required.

2.1Design Goals

To effectively detect bandwidth inflation attacks, our magic train should have the following properties.

- **Unpredictability:** The magic train should be designed with unpredictable elements to prevent adversarial proverbs from regularly delaying and/or correctly rushing measurement packets on the fly.
- **Posterior large delay capability:** The length of the magic train should be long enough, so that the resulted long delay will prevent the adversarial provers from receiving all the measurement packets before responding.
- **Round-trip link ability:** The magic train's packets should be round-trip linkable (i.e., each request packet can be linked to its corresponding response packet), so that the verifier can detect packet loss and reordering events.
- **Stability:**A magic delay algorithm should mitigate the cross traffic's impacts on the magic train. As a result, the packet train is still stable in the midst of heavy cross traffic. It can also facilitate the measurement of bottleneck bandwidth (i.e., capacity). The four goals are used to make the magic train effective in detecting bandwidth/capacity inflation.

3.Theorem

To successfully inflate bandwidth result through response dropping attack, even smart adversaries must drop successive response packets at the end of the train. Proof. When qN has not been dropped, the actual _t1N can be successfully recognized even if all the qG1+1; : : : ; qN>1 have been dropped. Hence, the bandwidth result remains the same. We therefore prove theorem 4 by contradiction. To handle this response packet dropping attack, we make our magic train round-trip linkable by exploiting TCP's inherent features. In our design, we first consider the use of TCP's sequence and acknowledgment numbers [41] for linking. By exploiting this feature, we use in-flow TCP data packets (i.e., the packets belong to a TCP flow whose three-way handshake has been completed) as pi's to trigger TCP ACKs as qi's. We then can link each pi with its corresponding response qi using-

$$SEQ(p_i) + |p_i| = ACK(q_i),$$

and ACK(qi) is the acknowledgement number of ACK packet qi. We call this kind of packet train as in-flow data packet train (IF-Data-Train in short). However, the IF-Data-Train's linkability is not reliable, since TCP acknowledges packets in a cumulative manner [41]. That is, if some pi is lost during the measurement, all the following qi+1; : : : ; qN will acknowledge. To address this weakness and obtain reliable linkability, we propose three new types of trains for linking. The first is to use out-of-flow TCP data packets (i.e., the packets do not belong to any existing TCP flows) as pi's to trigger TCP pure RSTs as qi's. We call this type of train out-of-flow data packet train (OF-Data-Train for short). Since RSTs cannot acknowledge any received data, we link pi with qi in the OF-Data-Train using TCP port numbers. The linking equation is as follows.

SrcP ort(pi) = DstP ort(qi); (8)

where SrcP ort(pi) returns the source port of packet pi and DstP ort(qi) returns destination port number of qi. As can be seen, OF-Data-Train employs packets from different TCP flows for measurement. This method may cause different request packets being routed through different network paths if per-flow load balancing is enabled by some routers [42]. However, we can also use the OF-Data-Train to implement our magic train, because the train can work well when the bottleneck device locates at a converged point of all the load balanced paths or there are no routers in the measurement path doing per-flow load balance.

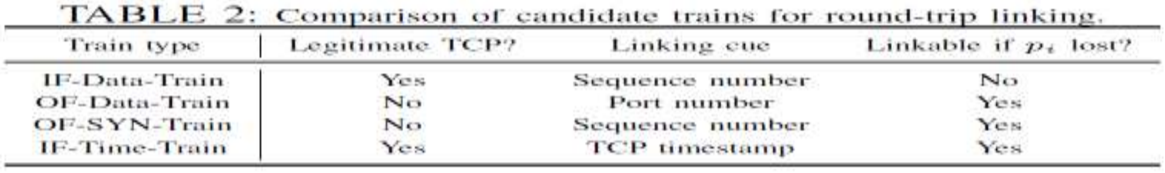
The second reliably linkable train we propose here is out-offlow SYN data packet train (OF-SYN-Train in short). This train employs TCP SYN packets with payload as the request packets to trigger TCP SYNjACK or RSTjACK packets as response packets. Both SYNjACK and RSTjACK can acknowledge the payload of TCP SYN packets. Therefore, the sequence number can be used for linking (see Eqn. (7)). The OF-SYNTrain cannot be affected by the cumulative acknowledgement scheme as TCP flows have not been completely established.

However, some enterprise firewall may treat the OF-SYNTrain as SYN flooding attacks [43]. Although the two aforementioned trains achieve reliable linkability, both of them do not follow legitimate TCP protocol, hence being potentially refused by TCP stack. For this reason, we propose a legitimate TCP packet train as the third solution. This new train also uses in-flow TCP data packets as pi's to trigger TCP ACKs as qi's, but do not rely on the sequence number for linking. Instead, we exploit a TCP timestamp option to avoid the cumulative acknowledgement scheme. Any TCP acknowledgement packet (i.e., ACK bit is set) can echo a timestamp value sent by the most recent TCP packet from remote hosts [44]. We therefore embed a TCPtimestamp value (i.e., TSval) to each request packet and readTCP timestamp echo reply (i.s., TSecr) field in each responsepacket for linking using Eqn. (9).

TSval(pi) = TSecr(qi): (9)

We call this train in-flow data packet timing train (IF-TIMETrain for short). This train is subject to a practical issue due to a so-called TCP delayed acknowledgment scheme [45]. That is, TCP stack may not respond with an ACK packet (i.e., qi) for each received data packet (i.e., pi). To address this issue, we either reorder each two successive request packets or assign mismatched sequence number to them. Our experiments have confirmed that both methods can work around the delayed acknowledgment scheme for the Linux systems. We compare the four round-trip linkable trains in Table. Except IF-Data-Train, the other three trains can achieve reliable round-trip linkability even if some request packets are lost during transmission.

TABLE 2: Comparison of candidate trains for round-trip linking.



Train type	Legitimate TCP?	Linking cue	Linkable if p_i lost?
IF-Data-Train	Yes	Sequence number	No
OF-Data-Train	No	Port number	Yes
OF-SYN-Train	No	Sequence number	Yes
IF-Time-Train	Yes	TCP timestamp	Yes

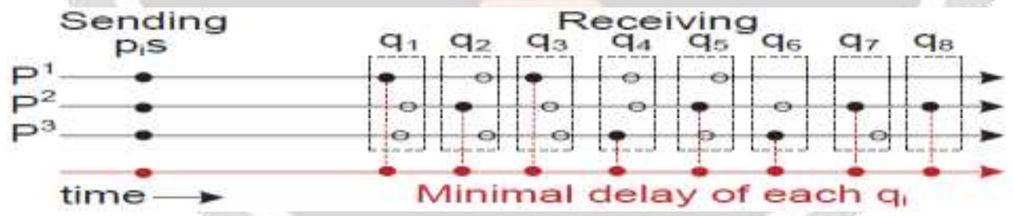
Detection of response dropping attack

By exploiting reliably round-trip linkable trains, we are able to detect response dropping attacks using Theorem 4. The basic idea is to check the number of unacknowledged successive request packets to the end of the train (i.e., the corresponding response packets are considered lost by the verifier). Although packet loss is a normal phenomenon, a sudden lost of a bulk of successive packets at the end of multiple trains is rare and thus anomalous. Given a set of magic trains P = fPmj1 _ m _ Mg, let lm be the number of unacknowledged successive request packets to the end of the train Pm (i.e., the verifier cannot receive qN□lm+1; qN□lm+2; : : : ; qN in Pm). We then regard a possible response dropping attack if 1 is a threshold for the detection. We should choose an appropriate Hd based on our experiments to balance the detection rates and false positives. A smaller Hd generally achieves a better detection capability but leads to a higher false positive as well.

Monitor Trainer Algorithm

To measure the capacity, our magic train must filter out the queuing delay induced by cross traffic and thus being unable to accurately report bottleneck bandwidth (i.e., capacity). To this end, we propose a new magic delay algorithm, which is designed by extending the minimal delay difference algorithm for packet pairs [35] to our magic train. The basic idea is to use minimal RTT of each packet from different magic trains to mitigate cross traffic's impact as much as possible. In particular, given a set of magic trains P = fPmj1 _ m _ Mg, our magic delay algorithm calculates ti, the RTT between the time of sending pi and the time of receiving qi, as

$$t_i = \min_{1 \leq m \leq M} (t_i^m),$$



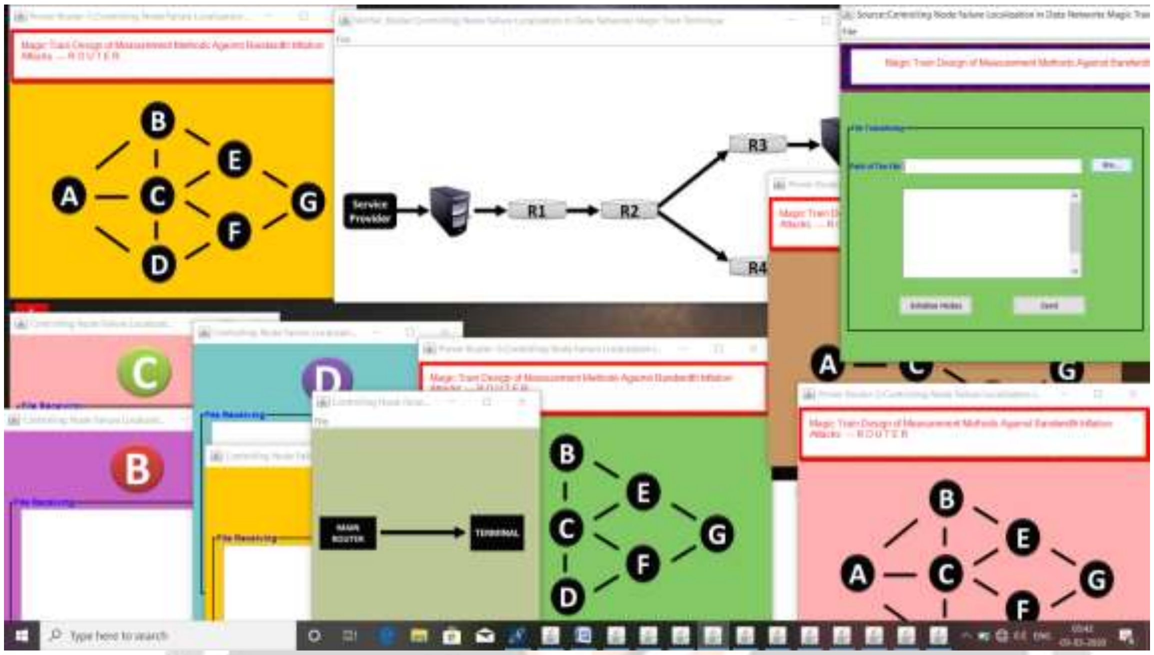
Detection of capacity inflation attack

Our magic delay algorithm can estimate the minimal RTT of linkable packets from a magic train set P and report capacity as

$$c_i = \frac{|p|}{\Delta t_{(i-1)i}} = \frac{|p|}{t_i - t_{i-1}}, \ \forall i \in [2, \max_{1 \leq m \leq M} (N^m)].$$

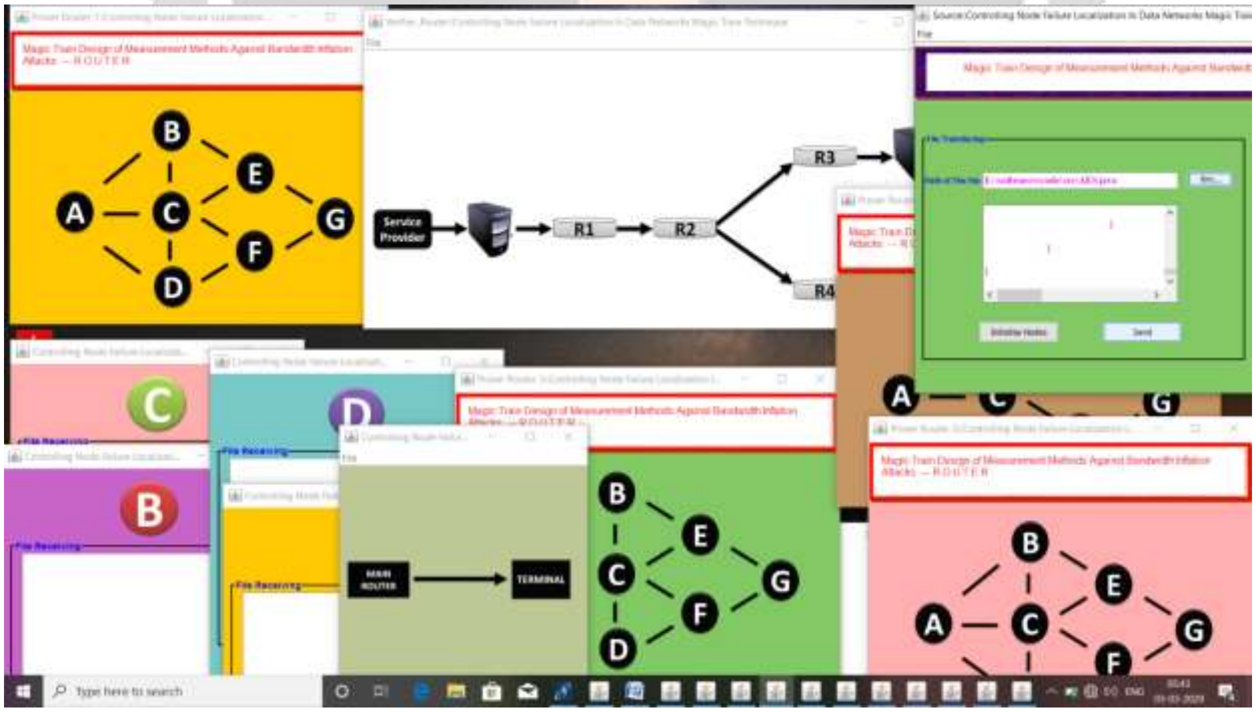
We then propose a new algorithm to detect capacity inflation attacks by calculating the standard deviation of cis over the minimal RTT of all linkable packets. We denote this deviation as V (c). Without being attacked, V (c) should be equal to 0 since any two different linkable packets should report the same capacity estimation (if they are all converged to minimal values). While capacity inflation attacks (through a priori/posterior delay/rush, or response dropping) are more likely to cause an irregular distribution of minimal delays and hence resulting in unequal cs. We therefore detect a possible capacity inflation attack if V (c) > Hc. We choose Hc a value slightly larger than 0, since a finite set of magic trains may not ensure all the RTTs to converge to their minimal values. We note that the more magic train samples in the set P, the higher probability that all the delays can be converged to minimal values and thus a better detection capability can be attained.

4.COMPARATIVE RESULT

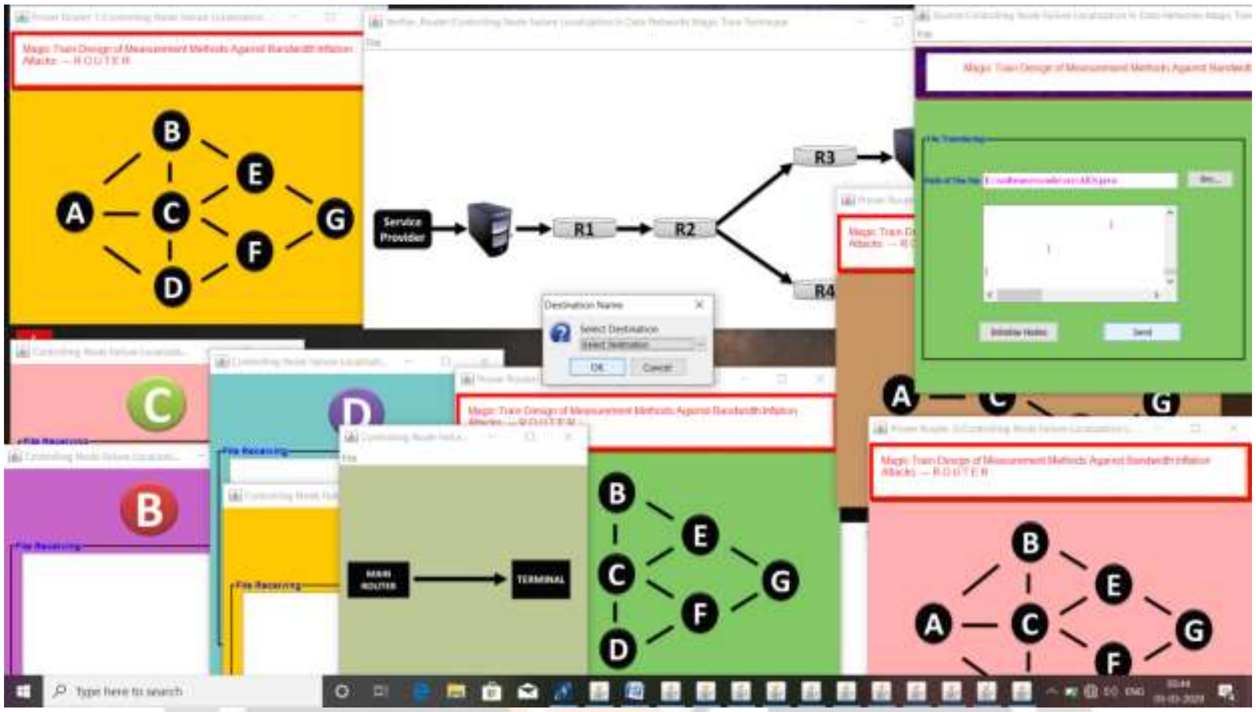

Magic Train Design of Measurement Methods Against Bandwidth Inflation Attacks -- ROUTER
A
B
C
D
E
F
G
Service Provider
R1
R2
R3
R4
MAIN ROUTER
TERMINAL
File Receiving
Initialize Nodes
Send

Screen 1 : Home page

Description: Select packets from memory for the purpose of sending packets to provers

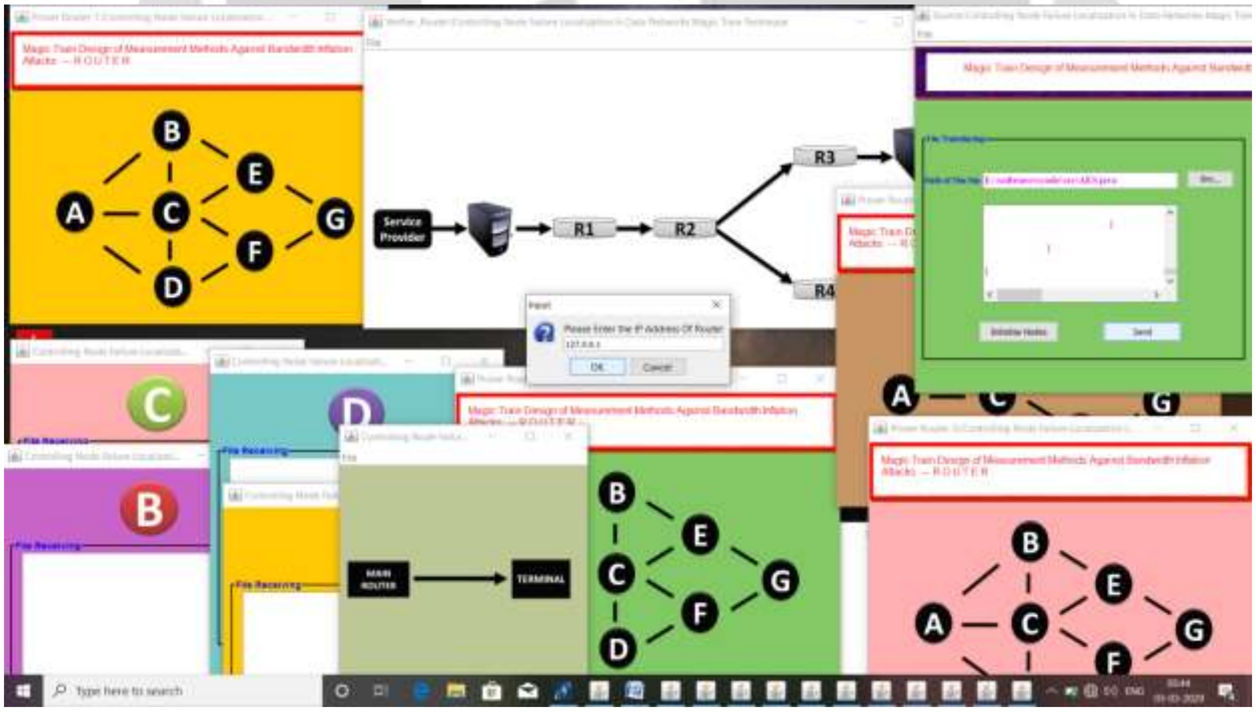

Magic Train Design of Measurement Methods Against Bandwidth Inflation Attacks -- ROUTER
A
B
C
D
E
F
G
Service Provider
R1
R2
R3
R4
MAIN ROUTER
TERMINAL
File Receiving
Initialize Nodes
Send

Screen 2: Selecting packets from memory

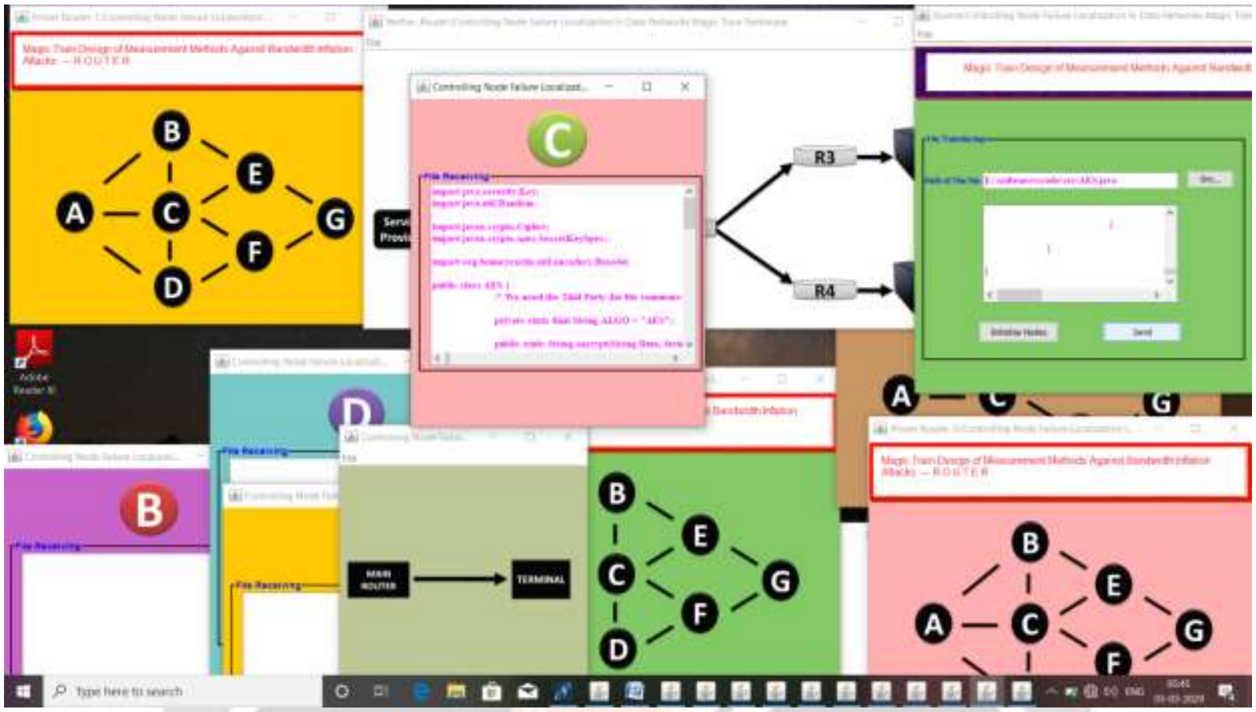


Screen 3 : Diagram for selection destination

Description: Select the destination node this process is done by after selection of packets**.**



Screen 4: diagram for IP address of sourec and destination nodes



Screen 5 : Successfully reach the destination

5. CONCLUSIONS

Width measurement. We designed, analyzed, implemented and evaluated a new bandwidth measurement algorithm, we call it monitor trainer, for securing uncooperative bandwidth measurement in adversarial networking environment. Our magic train is carefully designed to use an unpredictable measurement train to defeat even the smart adversaries. We also proposed a new magic delay algorithm to secure capacity measurement for the first time. We have shown through a rich set of testbed and Internet experiments that, our design achieves a good detection capability against even smart bandwidth inflation attacks and can adapt well in today's Internet.

6.ACKNOWLEDGEMENT

We thank all the authors and their values contribution work helped me to complete my final year project.

7.REFERENCES

[1] Xiapu Luo, Edmond W. W. Chan, and Rocky K. C. Chang. Design and implementation of TCP data probes for reliable and metric-rich network path monitoring. In Proc. USENIX Annual Technical Conference, 2009.
[2] Jain Manish and Dovrolis Constantinos. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. ACM SIGCOMM Computer Communication Review, 2002.
[3] Dovrolis Constantinos, Ramanathan Parameswaran, and Moore David.What do packet dispersion techniques measure? In Proc. Annual Joint Conference of the IEEE Computer and Communications Societies, 2001.
[4] Crovella Mark E and Carter Robert L. Dynamic server selection in the Internet. Technical report, Boston University Computer Science Department, 1995.
[5] Vern Paxson. End-to-end Internet packet dynamics. IEEE/ACM Transactions on Networking, 1999. analysis of bittorrent-like peer-to-peer networks. ACM SIGCOMM Computer Communication Review, 2004.
[7] Xiaojun Hei, Chao Liang, Jian Liang, Yong Liu, and Keith W Ross. A measurement study of a large-scale P2P IPTV system. IEEE Transactions on Multimedia, 2007.
[8] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In Proc. USENIX Security Symposium, 2004.

[9] Robin Snader and Nikita Borisov. A tune-up for Tor: Improving security and performance in the Tor network. In Proc. Network and Distributed System Security Symposium (NDSS), volume 8, page 127, 2008.
[10] Robin Snader and Nikita Borisov. Eigenspeed: secure peer-to-peer bandwidth evaluation. In Proc. International Workshop on Peer-to-Peer Systems (IPTPS), 2009.
[11] Ghassan Karame, Boris Danev, Cyrill Bannwart, and Srdjan Capkun. On the security of end-to-end measurements based on packet-pair dispersions. IEEE Transactions on Information Forensics and Security, 2013.
[12] Eddie Mitchell. Dailymotion.com redirects to fake AV threat [online]. http://www.invincea.com/2014/01/dailymotion-com-redirects-tofake-av-threat/, 2014.
[13] Lasse Overlier and Paul Syverson. Locating hidden servers. In Proc.IEEE Symposium on Security and Privacy, 2006.
[14] Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Low-resource routing attacks against tor. In Proc. ACM Workshop on Privacy in Electronic Society, 2007.
[15] Alex Biryukov, Ivan Pustogarov, and Ralf-Philipp Weinmann. Trawling for tor hidden services: Detection, measurement, deanonymization. In Proc. IEEE Symposium on Security and Privacy, 2013.
[16] Ghassan Karame and Srdjan Capkun. On the security of end-toend network measurements [online]. http://www.syssec.ethz.ch/research/Security End to End.pdf, 2009.
[17] Martin Devera. Htb traffic shaper [online]. http://luxik.cdi.cz/_devik/ qos/htb/, 2004.
[18] Netlimiter [online]. http://www.netlimiter.com/, 2013.
[19] Netequalizer [online]. http://www.netequalizer.com/, 2014.
[20] Linux programmer's manual - raw(7) [online]. http://man7.org/linux/man-pages/man7/raw.7.html, 2012.
[21] The libpcap project [online]. http://sourceforge.net/projects/libpcap/,2013.
[22] Douglas E Comer. Computer Networks and Internets. 2008.
[23] Mark Crovella. Measuring bottleneck link speed in packet-switched networks [online]. http://www.cs.bu.edu/_crovella/src/bprobe/Tools.html,
1996.
[24] Van Jacobson. Pathchar: A tool to infer characteristics of Internet paths [online]. http://www.caida.org/tools/utilities/others/pathchar/, 1997.
[25] Stefan Saroiu, P Krishna Gummadi, and Steven D Gribble. Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments. In Proc. IEEE International Conference on Computer Communications, 2002.