Journal of Engineering and Applied Sciences 12 (13): 3338-3344, 2017
ISSN: 1816-949X

Bandwidth Optimization in Tor Nodes through Frame Configuration in Hidden Services

P. Salcedo, Octavio J., Forero R. Julian A. and E. Vera-Parra Nelson
Distrital University, Bogota, Colambia

Abstract: Traffic hiding and internet filter evasion involves an overload in the frame headers transmitted using anonymous connections with this comes a low performance in the data transfer due to the network packages processing. This study presents a frame configuration prototype in TOR nodes through onion routing using TOR network controllers such as txcon with the purpose to optimize the available bandwidth to share on each node through the definition of anonymous circuits based on hidden services. In a case study, it was determined that the available bandwidth increased to 12% in average when it was between 583 and 590 b/s download speed and 131 and 143 b/s upload speed showing an increase both in the downloading traffic by 13% as well as in the uploading speed by 11%. Likewise, it was found that the number of outbound connections dropped by 32%.

Key words: Bandwidth, internet filter, onion routing, TOR, traffic hiding

INTRODUCTION

The emergence of networks to preserve anonymity such as TOR and its expansion under the voluntary cooperation of its users using the onion routing has allowed the interests of the users in privacy. Thanks to its versatility and the use of the TCP protocol for communicating to any application, TOR use tunneling techniques for network traffic anonymity wherewith a several amount of applications such as webservers, instant messaging and file transfer applications could have some degree of anonymity (Moore *et al*., 2011).

Although, TOR has become the most preferred option for anonymous browsing it has been affected by a low performance and increasing latency due to the ineffective network circuits displacement with continuous traffic to circuits with less traffic (Tang and Goldberg, 2010). This could be translated into latency increments and bandwidth reduction therefore a critical situation is established for anonymous browsing, less performance with more privacy.

Under this line we can establish a new defined problem regarding the establishment of anonymous connections through onion routing where by means of ciphered connections towards onion routers through proxies a whole communication route by tunneling where only is allowed to know connections to the next hop but not from the request source and its destination. This way, the network access becomes a slower process due to the headers overload established in network traffic, thanks to the high amount of ciphered connections. For this reason is necessary to implement an asynchronous protocol control through network controllers in onion networks such as TOR that allows to visualize the circuits and hidden services with the purpose to optimize the available bandwidth shared on each node improving the latency levels and anonymous browsing bandwidth.

In this study is presented a prototype for network frame configuration in TOR nodes through the onion routing using TOR network controllers such as txtcon and stem with the purpose to optimize the available bandwidth to share on each node through the definition of anonymous circuits based on hidden services.

Literature review: Onion networks has been the subject of study for several researchers who have established analysis methodologies and error classification in packages from techniques considerations of mapping techniques.

Onion networks have become subject of study for several researchers who have established methodologies for analysis and classification of errors in network frames considering mapping techniques and size reduction in the physical layer of wireless networks 802.11 as exposed by Khan and Veitch (2008). Similarly through the diffusion of P2P networks based on onion routing such as TOR,

Corresponding Author: Salcedo P. Octavio J., Distrital University, Bogota, Colambia

interesting contributions have been established based on the Quality of Service (QoS) parameters under the performance analysis of TOR nodes with techniques such as Ting, considering latency as the main variable (Cangialosi *et al*., 2015). Also, using tools like Tortoise it has been highlighted the potential benefits of configuration and limiting expansion rates in the entry nodes for TOR with the purpose to increase performance in anonymous networks not having to sacrifice the security they give (Moore *et al*., 2011). In the same way, we highlight the establishments of new algorithms to update the paths in TOR (Tang and Goldberg, 2010) as well as new protocols for key exchange such as NTOR and ACE from the one way authenticated key exchange with the consideration of cryptography based on Elliptic Curves (ECC) and cell or frame modification in TOR (Backes *et al*., 2012).

On the other hand, in the security study within TOR networks, researchers like Backes have given valuable tools such as MAToR a framework to analyse the algorithm influence for path selection on the anonymity of TOR clients (Backes *et al*., 2014). Similarly, strategies such as fingerprinting techniques under methodologies of data collection (Panchenko and Niessen, 2011; Wang and Goldberg, 2013) the denial-of-service attacks detection using analytical models and simulations (Danner *et al*., 2012) and the application of attacks based on the TOR cells counting (Ling *et al*., 2012) have stood out in the security field of Onion networks.

However, it is important to highlight the great advances that have been made in routing in onion networks, among all we could highlight the route search from Honr's clauses, exposed by Schrijvers from the consideration of semantics which cover the expansion and execution of client nodes (Tahmassebpour and Otaghvari, 2016). Following the line, under onion environments in P2P networks, works like McLachlan's expose valuable implementations of the Torks protocol, for anonymous connection structuring with low latency (Lachlan *et al*., 2009) as well as public key management based on filter distribution.

Nevertheless, studies made by Thomas and Mohaisen determine some problems associated to requests resolution of hidden services from TOR networks, based on the namespaces determined by the high level domains such as Onion (Thomas and Mohaisen, 2014). The same way researchers like Chen *et al*. (2015) have established high speed onion routing systems operable in the network layer, based on the use of symmetric cryptography for sending data in intermediate routers although, studies such as the presented by Li *et al*. (2011) have determined that higher TOR nodes despite of giving assistance within the bandwidth allocation they are part of the life cycle that can be discovered through collection of hidden data, questioning the anonymity on those networks. In the same way, Manabe proposes encryption schemas of routes based on mailing systems and mixed networks (Atafar *et al*., 2013) which supports Wang's work setting onion routing circuits through cryptography systems of anonymous identity (Wang *et al*., 2015).

Among the interesting applications of TOR networks is the application for payment systems for network services such as P2P or cloud based services (Tahmassebpour, 2017) as well as the establishment of centralization protocols based on genetic algorithms for reduction of energy consumption in networks based on wireless sensor (Hatamian *et al*., 2016). Similarly, researchers like Sen analyze the anonymity security and privacy of connections in Mesh network. Under this line, Sochor has proposed a study about the measurement of maintenance and processing parameters over anonymous traffic in onion networks. Otherwise, Jaisooraj deepens in the anonymity for data traffic in Ad Hoc mobile networks, considering the Link Stability Metric (LSM).

MATERIALS AND METHODS

For the realization and subsequent assessment of the project, three main phases are define, first the hidden service will be running as the second instance the connection is established and the hidden service is activated through the Twisted web service. Later on, the configuration parameters are set to manage the TOR node's bandwidth using utilities as reactor and ireactortime. For this reason, the project is descriptive.

Research method: During the project's process a generic methodology will be placed as a base guide under four steps or fundamental processes.

Service analysis: In this initial stage, the amount of hidden services to manage is set using TOR controllers, at the same time is determined how many slave nodes will be taken into account for the connection with this the Twisted server start where the hidden service will be activated.

Service activation: In this stage, we proceed to activate the services using the Twisted server and with this its access from an URL with the onion ending which will show the commissioning of the hidden service.

Events listening: In this stage, the configuration is set and the monitors will be implemented those will record all the used bandwidth in terms of upload and download as well as all the alerts traffic created generated by connection errors, network drops among others.

Bandwidth change: In this last phase all the setup will be made related to the bandwidth with the purpose to establish higher transfer limits within certain time ranges.

Design: As TOR is an anonymous distributed network that uses the onion routing, it takes the available bandwidth provided by the relay or Onion Routers (OR) of community volunteers around the world who publish their bandwidth and exit policies to a set of centralized servers called Director Authorities (DA). This way to access the TOR network is necessary for an end user to run an Onion Proxy (OP) that will establish application requests through TOR and will download a (OR) given by the DA determined by the amount of available bandwidth to build circuits over them and this way address all the traffic using encoding layer to layer (Fig. 1) (Tang and Goldberg, 2010).

Cells in TOR: Communication between RO and PO of the users is made through TLS connections using a couple of asymmetric keys that are used to establish one or more encryption keys. Thanks to this, the use of TLS hide the connection data with perfect anonymity which avoids modification of the data or RO replacement (Dingledine *et al*., 2004).

Hence, TOR traffic is placed on cells with a fixed size of 512 bytes with a header field and a payload (Fig. 2). The header includes a circuit identifier that specifies a reference to the circuit which the cell is passing through as well as a command that allows to define what will be the use for the cell (transmission, creation or destruction), interpreted by the relay nodes similarly the encrypted data which with the processing is made in the output buffer for the routing to other RO.

Based on the above, a test scenario is established with three clients with a PO assigned to each one with a DSL link with 5 Mbps asymmetric with 500 Kbps for download and 120 Kbps for upload in average, connected using three laptops that will be directly connected to three RO (input, middle and exit) to get to a web server on HTTP request will be made to access the hidden services hosted on it. The same way, in the middle RO and the exit will be the AD (Fig. 3).

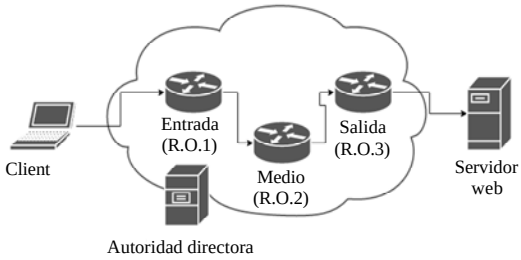

Fig. 1: Basic TOR network architecture (researcher)

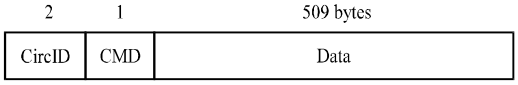

Fig. 2: TOR cell structure

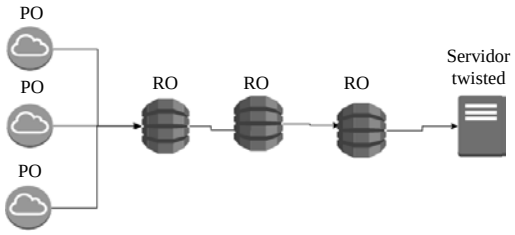

Fig. 3: Test scenario (researcher)

Table 1: Clients specifications (researcher)

Computer	Link speed	ID proxy onion
Dell Inspiron N5010	5 Mbps assymetric	PO 1
Asus K43E	5 Mbps asymmetric	PO 2
Sony Vaio VPCY2	5 Mbps asymmetric	PO 3

RESULTS AND DISCUSSION

Based on the design proposal previously exposed, two hidden services will be established through a TOR txtcon controller. To achieve this, initially the RO configuration will be made with the purpose to make the nodes behave as onion repeaters, having this, the Twisted web server will run using its own libraries, giving as a result an address with the onion extension on the 80 port where the hidden services will be hosted (Fig. 4).

This way by having a hidden service fully working and reachable from any PO (Fig. 5) a behavior monitor must be established using a process or a daemon that will be responsible for collect data about the processing capacity, the memory used by the TOR process, the bandwidth for upload and download limits as well as the total amount of traffic. Indeed, this variables must be collected for every RO with the purpose to optimize the bandwidth on each node.

```
import tempfile
import functools
import txtorcon

from twisted.internet import reactor
from twisted.internet.endpoints import TCP4ServerEndpoint
from twisted.web import server, resource

class Simple(resource.Resource):
    isLeaf = True

    def render_GET(self, request):
        return "<html><h1>SERVICIO OCULTO DISPONIBLE Y FUNCIONANDO</h1></html>"

def updates(prog, tag, summary):
    print "%d%%: %s" % (prog, summary)

def setup_complete(config, proto):
    print "Protocolo completado"
    onion_address = config.HiddenServices[0].hostname
    print "El servicio web tiene URL:"
    print "http://%s (port %d)" % (onion_address, hs_public_port)
    print "La URL Temporal es:", config.HiddenServices[0].dir
    print " http://%s" % onion_address

def setup_failed(arg):
    print "Fallo", arg
    reactor.stop()
hs_port = 9876
hs_public_port = 80
hs_temp = tempfile.mkdtemp(prefix='torhiddenservice')
reactor.addSystemEventTrigger(
    'before', 'shutdown',
    functools.partial(
        txtorcon.util.delete_file_or_tree,
        hs_temp
    )
)
```

Fig. 4: Hidden service opening using Twisted (researcher)

Fig. 5: Working hidden service (researcher)

```
import datetime
from twisted.internet import reactor
from twisted.internet.interfaces import IReactorTime
from txtorcon import build_local_tor_connection, TorConfig

class anchoBandaUpdater:
    def __init__(self, config, planificador):
        self.anchoBanda = 0
        self.config = config
        self.planificador = IReactorTime(planificador)
        self.generador = self.siguiente_Actualizacion()

    def siguiente_Actualizacion(self):
        while True:
            if self.anchoBanda:
                self.anchoBanda = 0
                self.trama = 0
            else:
                self.anchoBanda = 20 * 1024 * 1024
                self.trama = self.anchoBanda
            yield (datetime.datetime.now() + datetime.timedelta(minutes=20),
                   self.anchoBanda, self.trama)

    def actualizarTiempo(self):
        x = self.generador.next()
        future = x[0]
        self.new_anchoBanda = x[1]
        self.new_trama = x[2]

        tm = (future - datetime.datetime.now()).seconds
        self.planificador.callLater(tm, self.actualizarAnchoBanda)
        print "esperando", tm, "segundos para ajustar ancho de Banda"

    def actualizarAnchoBanda(self):
        print "estableciendo ancho de Banda + trama a", self.new_anchoBanda, self.new_trama
        self.config.set_config('anchoBandatrama', self.new_trama,
                               'anchoBandaRate', self.new_anchoBanda)
        self.doUpdate()
def setup_completo(conf):
    print "Conectado."
    bwup = anchoBandaUpdater(conf, reactor)
    bwup.actualizarTiempo()

def setup_error(arg):
    print "Error", arg
    reactor.stop()

def bootstrap(proto):
    config = TorConfig(proto)
    config.post_bootstrap.addCallback(setup_completo).addErrback(setup_error)
    print "La conexion esta arriba ..."

d = build_local_tor_connection(reactor, build_state=False,
                               wait_for_proto=False)
d.addCallback(bootstrap).addErrback(setup_error)

reactor.run()
```

Fig. 6: Bandwidth optimization using Txtorcon (researcher)

Bandwidth in TOR: It should be noted that the variables such as the bandwidth have to be determined by the values reported in the TOR's control ports, by the controllers based on pipes which determine the total bandwidth measured in bytes per second both in read operations (download) as in write operations (upload) by the repeater nodes (relays). This way this bandwidth will correspond to the used bandwidth in a repeater node (relay). To measure the variables previously mentioned is used the command line interface arm which allows to visualize in real-time the changes made in the network. In the same way, it will allow to see a list of repeater's connections using an interpreter that will alert for possible failures in the connection.

According to the collected data from arm, it could be determined that for each relay OR the average available bandwidth is between 520 and 540 b/s download and 120 and 132 b/s upload showing a low performance both in the download traffic and in the upload traffic. Likewise, it was determined that the number of outbound connections changed according to the traffic that was addressed on each RO while this number of connections didn't have significant variations in regard to the time. Based on the above is necessary to establish and control the amount of bandwidth available for each of the RO according to the amount of traffic making use of the functionalities of the txtorcon controller, it is determined a period of time X on which the values of the limit values for the bandwidth will range. This way the update for the band width will happen every X amount of time according to the amount of traffic processed by the RO (Fig. 6).

In this manner when the functionality was applied it was possible to determine that the available bandwidth increased by 12% being between 583 and 590 Kb/s

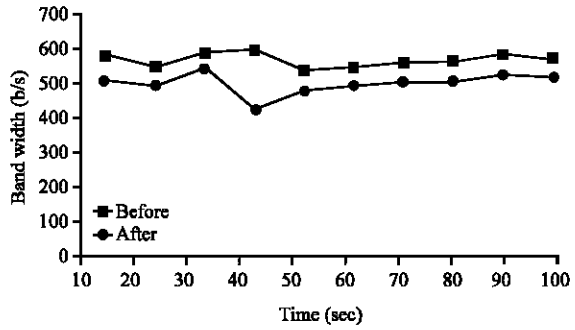

Fig. 7: Donwload bandwidth (researcher)

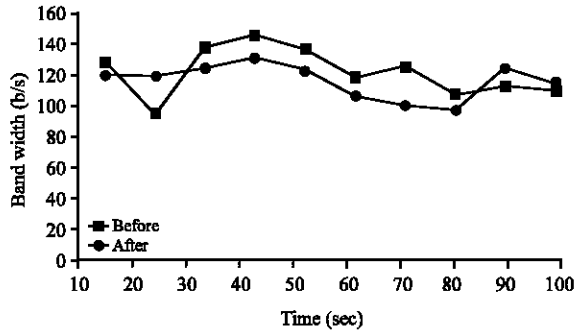

Fig. 8: Upload bandwidth (researcher)

download and 131 and 143 b/s upload which shows an increase on the performance both in the download traffic by 13 and 11% in the upload traffic. Likewise, it was established that the number of outbound connections decreased by 32%.

According to the results exposed above is possible to establish an analysis between bandwidth and connections according to the time and the TOR default configuration, showing an increase on the web traffic performance and the access to the hidden services using anonymous browsing (Fig. 7 and 8). In comparison with some studies presented as the results from Tang and Goldberg (2010) is possible to note how the access to hidden services show a lower performance in the normal web traffic although, the implementation of priorization algorithms for circuits can affect drastically to the duration for their creation and with this a reduction of time loading web sites. However unlike what Tang and Goldberg presented, in this study it was evident how the modification of the bandwidth can be made on any moment which avoids the creation of a new circuit and with it the search of the AD for a TOR connection establishment.

In the same way, Moore *et al.* (2011) point out that the modification of the upload or download limits within a RO must be a critical issue in the use of a community network resources as TOR is since with high levels of bandwidth provided in a RO is possible that they could be consumed for several users with higher access speeds. For this reason, they propose the use and the spread of RO among users with more bandwidth in their access to internet with the purpose to have a little more equitable communication channels. Besides of that in contraposition to the results presented above that study establish that the capacity of the anonymous network will be determined by the placement of clients or proxies in the same RO where with the shared bandwidth must be limited in turn by variables like connection speed, the amount of clients, the available bandwidth among others.

CONCLUSION

By establishing and controlling the amount of bandwidth available for each RO according to the amount of traffic, during a determined period of time X which the limit bandwidth values will range, it was possible to find that the available bandwidth increased by 12% as it was between 583 and 590 Kb/s download and 131 and 143 b/s upload which shows an increase in the performance both in the download traffic by 13% and upload traffic by 11%. Likewise, it was found that the number of outbound connections decreased by 32%.

REFERENCES

Atafar, A., M. Shahrabi and M. Esfahani, 2013. Evaluation of university performance using BSC and ANP. Decis. Sci. Lett., 2: 305-311.

Backes, M., A. Kate and E. Mohammadi, 2012. Ace: An efficient key-exchange protocol for onion routing. Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society, October 15-15, 2012, ACM, Raleigh, North Carolina, USA., ISBN:978-1-4503-1663-7, pp: 55-64.

Backes, M., A. Kate, S. Meiser and E. Mohammadi, 2014. Monitoring the anonymity of tor's path selection. Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, November 03-07, 2014, ACM, Scottsdale, Arizona, USA., ISBN:978-1-4503-2957-6, pp: 513-524.

Cangialosi, F., D. Levin and N. Spring, 2015. Ting: Measuring and exploiting latencies between all tor nodes. Proceedings of the 2015 ACM Conference on Internet Measurement Conference, October 28-30, 2015, ACM, Tokyo, Japan, ISBN:978-1-4503-3848-6, pp: 289-302.

Chen, C., D.E. Asoni, D. Barrera, G. Danezis and A. Perrig, 2015. HORNET: High-speed onion routing at the network layer. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, October 12-16, 2015, ACM, Denver, Colorado, USA., ISBN: 978-1-4503-3832-5, pp: 1441-1454.

Danner, N., K.S. Defabbia, D. Krizanc and M. Liberatore, 2012. Effectiveness and detection of denial-of-service attacks in Tor. ACM. Trans. Inf. Syst. Secur. (TISSEC), 15: 1-25.

Dingledine, R., N. Mathewson and P. Syverson, 2004. Tor: The second-generation onion router. U.S. Naval Research Laboratory, Washington, USA., http://www.dtic.mil/docs/citations/ADA465464.

Hatamian, M., H. Barati, A. Movaghar and A. Naghizadeh, 2016. CGC: Centralized genetic-based clustering protocol for wireless sensor networks using onion approach. Telecommunication Syst., 62: 657-674.

Khan, M.A.Y. and D. Veitch, 2008. Peeling the 802.11 onion: Separating congestion from physical per. Proceedings of the 3rd ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, September 19-19, 2008, ACM, San Francisco, California, USA., ISBN:978-1-60558-187-3, pp: 33-40.

Lachlan, M.J., A. Tran, N. Hopper and Y. Kim, 2009. Scalable onion routing with torsk. Proceedings of the 16th ACM Conference on Computer and Communications Security, November 09-13, 2009, ACM, Chicago, Illinois, USA., ISBN: 978-1-60558-894-0, pp: 590-599.

Li, C., Y. Xue, Y. Dong and D. Wang, 2011. Super nodes in tor: Existence and security implication. Proceedings of the 27th Annual Conference on Computer Security Applications, December 05-09, 2011 ACM, Orlando, Florida, USA., ISBN: 978-1-4503-0672-0, pp: 217-226.

Ling, Z., J. Luo, W. Yu, X. Fu and D. Xuan *et al.*, 2012. A new cell-counting-based attack against Tor. IEEE-ACM. Trans. Networking (ToN), 20: 1245-1261.

Moore, W.B., C. Wacek and M. Sherr, 2011. Exploring the potential benefits of expanded rate limiting in tor: Slow and steady wins the race with tortoise. Proceedings of the 27th Annual Conference on Computer Security Applications, December 05-09, 2011, ACM, New York, USA., ISBN: 978-1-4503-0672-0, pp: 207-216.

Tahmassebpour, M. and A.M. Otaghvari, 2016. Increase efficiency big data in intelligent transportation system with using IoT integration cloud. J. Fundam. Appl. Sci., 8: 2443-2461.

Tahmassebpour, M., 2017. A new method for time-series big data effective storage. IEEE Access, 2007: 10694-10699.

Tang, C. and I. Goldberg, 2010. An improved algorithm for Tor circuit scheduling. Proceedings of the 17th ACM Conference on Computer and Communications Security, October 04-08, 2010, ACM, Chicago, Illinois, USA., ISBN:978-1-4503-0245-6, pp: 329-339.

Thomas, M. and A. Mohaisen, 2014. Measuring the leakage of onion at the root: A measurement of tor's onion pseudo-TLD in the global domain name system. Proceedings of the 13th Workshop on Privacy in the Electronic Society, November 03-03, 2014, ACM, Scottsdale, Arizona, USA., ISBN:978-1-4503-3148-7, pp: 173-180.

Wang, C., D. Shi and X. Xu, 2015. AIB-OR: Improving onion routing circuit construction using anonymous identity-based cryptosystems. PloS One, Vol.10,

Wang, T. and I. Goldberg, 2013. Improved website fingerprinting on tor. Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society, November 04-04, 2013, ACM, Berlin, Germany, ISBN:978-1-4503-2485-4, pp: 201-212.