Simulació de la xarxa anònima Tor

Pol Micolau Cos

Resum– Tor ha suposat una nova forma de navegar a través d'internet. Gràcies a la seva capacitat de mantenir l'anonimat i la privacitat de l'usuari, ha fet que sigui una xarxa cada cop més popular com també més estudiada. Per això últim, sorgeixen cada cop més eines amb les quals poder experimentar amb Tor d'una forma segura, per així conèixer el seu funcionament i les seves possibilitats de ser. En aquest paper, es fa ús del simulador Shadow, conjuntament amb les eines de modelació Tornettools i TGen, per modelar i simular tres xarxes de diferent mida i avaluar la seva precisió respecte a la xarxa Tor real. Per validar aquesta precisió, es comparen diverses mètriques de rendiment extretes del Tor públic i de les mateixes simulacions. Les comparacions acaben demostrant que el simulador aconsegueix una precisió respecte a Tor suficient per a executar experiments precisos.

Paraules clau– Xarxa anònima Tor, simulació de xarxes, Shadow, Tornettools, servidor al núvol.

Abstract– Tor has made a breakthrough in the way we surf the Internet. Thanks to its capacity of preserving user's anonymity and privacy, Tor is becoming more popular as well as more studied. Therefore, new tools have been developed in the past years to help researchers experiment with Tor in safer way in order to understand its functionality and its possibilities. In this paper, the Shadow simulator has been used alongside modeling tools, such as Tornettools and TGen, to model and simulate different size networks and evaluate its accuracy. To validate this accuracy, several performance metrics extracted from the public Tor and the simulations are compared. The comparisons prove that the simulator achieves an accuracy regarding Tor to run precise experiments.

Keywords– Tor anonymous network, network simulation, Shadow, Tornettools, cloud server.

1 Introducció

A CONSEGUIR la privacitat i anonimat a Internet és una voluntat cada cop més freqüent entre els usuaris que naveguen a través d'ella. Tecnologies com les VPN solen ser una bona solució per afegir una capa extra de privacitat i aconseguir emmascarar l'adreça IP de l'usuari, tot i això, no ofereixen un anonimat complet a la connexió.

Tor sorgeix com una solució per poder navegar a través de la xarxa de forma anònima, extremant la dificultat que les dades de l'usuari es vegin compromeses. La importància d'aquesta tecnologia ha crescut entre els usuaris, com també l'interès d'investigadors per dur a terme experiments en ella amb el fi de millorar-la. Neix, doncs, la necessitat de trobar formes on poder executar aquests experiments en entorns controlats, de forma pràctica i sense posar en risc la privacitat dels seus usuaris.

• E-mail de contacte: pol.micolau@e-campus.uab.cat
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Victor Garcia Font (dEIC)
• Curs 2020/21

En aquest projecte s'utilitzen eines ja desenvolupades per modelar i simular xarxes proporcionalment més petites perquè recreïn el comportament de Tor, obtenint així un entorn controlat i extern de la xarxa real. La finalitat serà la d'avaluar la seva similitud i exactitud, extraient mètriques de rendiment de les simulacions, per finalment comparar-les amb les extretes de la xarxa Tor real.

El paper queda organitzat de la següent forma: La Secció 2 detalla els objectius proposats del projecte. La Secció 3 introdueix la metodologia aplicada. La Secció 4 estudia el funcionament de Tor. La Secció 5 introdueix l'estat de l'art i l'eina a utilitzar. La Secció 6 detalla la disposició i els passos seguits a l'experiment realitzat. Les seccions 7 i 8 exposen i discuteixen els resultats obtinguts. I la Secció 9 tanca amb unes conclusions.

2 Objectius

L'objectiu principal d'aquest projecte és el d'avaluar la precisió amb la qual s'ha aconseguit simular la xarxa anònima Tor. Per aconseguir això, serà necessari el compliment dels següents subobjectius (O: Objectiu, P: Prioritat):

- O1-P4: Construir una base teòrica de Tor, a força de saber diferenciar els components que la componen i les metodologies utilitzades per comunicar-se.
- O2-P3: Trobar desenvolupaments amb els quals poder executar Tor de forma controlada i manipulable.
- O3-P1: Modelar diferents tipologies de xarxa i poder simular-les.
- O4-P2: Comparar la simulació amb el Tor real estudiant diverses mètriques de rendiment, i així demostrar la seva precisió.

3 METODOLOGIA

Pel compliment dels objectius s'ha aplicat la metodologia "***Design Science Research***"(DSR) [1], dissenyada per la creació i ús d'artefactes per solucionar problemes o millorar el coneixement sobre investigacions existents. Es compon principalment per 5 fases: conscienciació del problema a tractar, suggeriment d'objectius i solucions, desenvolupament, avaluació i conclusions.

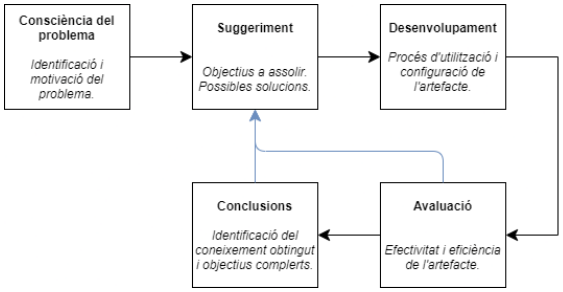


Fig. 1: Metodologia *Design Science Research*

El procés d'enteniment de les eines i la posterior avaluació de resultats pot suposar l'aparició de més objectius i noves formes de desenvolupament. Per aquest motiu, poder retroalimentar la fase de suggeriments amb informació recopilada a fases finals fa que aquesta metodologia aporti un gran valor al projecte.

4 XARXA TOR

La xarxa Tor és la implementació més gran d'una xarxa de comunicació anònima de baixa latència. Més precisament, Tor és una implementació del conegut sistema Onion Routing, en la qual s'introdueixen millores a les limitacions que presentava aquest disseny inicial [2]. Amb motiu d'aquestes millores, s'anomena a Tor com la versió millorada o segona generació de l'Onion Routing.

La xarxa està composta per milers de nodes que operen de forma voluntària amb la finalitat de preservar la privacitat i l'anonimat de l'usuari a Internet. Amb aquesta finalitat, l'usuari de Tor es connecta a través de diferents túnels de xifrat abans de poder-se connectar al servidor que ofereix el servei, en comptes d'executar una connexió directa a aquest.

Abans d'introduir a lector tots els aspectes que tenen a veure amb el funcionament de la xarxa Tor, s'explicarà amb més detall el disseny en el qual es va basar per la seva implementació, és a dir, l'Onion Routing. Un cop entesa la base sobre la qual es fonamenta, s'especifiquen els components principals que componen la xarxa, com també els canvis que Tor implementa per tal de millorar el disseny original.

4.1 Onion Routing

Per tal de preservar l'anonimat de l'usuari, Tor realitza la seva pròpia implementació de l'anomenat Onion Routing. Desenvolupat a mitjans dels anys 90 al U.S. Naval Research Laboratory per Paul Syverson, Michael G. Reed i David Goldschlag, l'Onion Routing és un disseny per a xarxes de comunicació anònimes de baixa latència, el qual proporciona connexions anònimes altament resistents a l'anàlisi de tràfic [3]. Com assenyalen els mateixos desenvolupadors, les connexions per Onion Routing permeten una comunicació bidireccional entre un iniciador i un receptor mantenint-se ambdós anònims, l'un de l'altre. La principal diferencia que es pot trobar en una connexió per socket tradicional i una connexió per Onion Routing és que mentre que la connexió tradicional es fa directament cap al receptor, la connexió Onion està establerta a través d'una sèrie d'Onion Routers (OR) o nodes. Aquests ORs es troben connectats entre ells mitjançant connexions per socket permanents, però amb una peculiaritat, l'única informació que tenen sobre la connexió global és quin és l'OR anterior i següent a ells.

A aquest disseny se li afegeix un software anomenat Onion Proxy (OP), el qual es troba instal·lat i executant-se de forma local a la màquina del client. La tasca principal d'aquest software és la de construir i gestionar les connexions anònimes [4], més concretament, establir els circuits, dur a terme el xifratge del missatge a transmetre, i gestionar les connexions provinents d'aplicacions d'usuari. Quan l'OP ja ha establit un circuit amb diferents nodes, s'utilitza criptografia simètrica per xifrar el missatge que l'usuari vol enviar [5]. A mode de recordatori, en la criptografia simètrica l'emissor i el receptor utilitzen tots dos una mateixa clau, tant per xifrar com per desxifrar el missatge. Per negociar i distribuir la clau entre les dues parts (OP i node), s'utilitza el protocol Diffie-Hellman amb cada un dels nodes que conformen el circuit [12].

Com es detallarà més endavant, per defecte, el disseny de Tor implementa circuits de tres nodes, per tant, el nombre de claus a negociar seria de tres. Havent negociat les claus, l'OP procedeix a xifrar el missatge. De forma anàloga a les capes d'una ceba (en anglès onion), s'apliquen múltiples capes de xifratge sobre el missatge en clar, figuradament, una sobre l'altre. Partint d'aquesta analogia, la capa més externa sempre correspon a aquella que s'ha xifrat amb la clau negociada amb el primer node del circuit. Mentre que la capa que es troba més a l'interior de la ceba sempre correspon a la xifrada amb la clau negociada amb l'últim node del circuit.

A la Fig.2 es pot apreciar una il·lustració de com quedaria el missatge final a transmetre un cop el procés de xifratge s'hagués dut a terme.

Xifrat el missatge, aquest és enviat al primer node, el qual "elimina" la capa de xifratge utilitzant la clau negociada i retransmet el missatge amb les capes restants al següent node. Aquest procés es duu a terme a cada integrant del circuit fins a arribar l'últim, el qual desfà l'última capa del missatge, i retransmet el missatge en clar al servidor destinatari.

Aquesta metodologia de xifratge ofereix diversos avan-

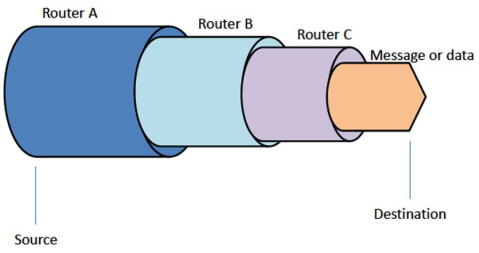


Fig. 2: Representació del missatge xifrat.

tatges per mantenir l'anonimat de l'usuari respecte al tràfic que ha generat. Una d'elles és que la identitat de l'usuari solament queda registrada al primer node del circuit, ja que rep el missatge directament d'ell, però li és incapaç de saber el contingut final d'aquest. Pel que fa als nodes intermediaris, aquests ni són conscients del contingut del missatge ni de la destinació d'aquest. Mentrestant, l'últim node sí que és coneixedor de la destinació, ja que acaba veient el contingut del missatge en clar, però sense saber qui és l'emissor. Amb aquest procés s'aconsegueix l'objectiu inicial, que no existeixi cap node amb coneixença de l'emissor i receptor alhora [12], mantenint així l'anonimat.

4.2 Components de la xarxa Tor

4.2.1 Encaminadors

Els encaminadors onion o Onion Routers (OR) en anglès, indistintament coneguts com a relays o nodes, són els encarregats de gestionar tot el tràfic que es genera a la xarxa i retransmetre els paquets que reben al següent OR o al servidor final [5]. Tots els OR que conformen Tor són executats i posats en marxa per usuaris voluntaris d'arreu del món que decideixen cedir part de la seva amplada de banda. Per tant, com qualsevol altra xarxa distribuïda, contra més voluntaris participen més relays s'executen a la xarxa i, per tant, millor és el seu rendiment.

En la implementació que fa Tor de l'Onion Routing, el nombre de relays per defecte utilitzats per establir una connexió són tres (anomenats guard relay, middle relay i exit relay), formant així el que es coneix com un circuit. Aquests tres ORs estabiliran una connexió TLS entre ells (per evitar la suplantació o modificació de les dades), mantenint cadascun d'ells: una clau d'identitat per signar certificats TLS, la qual pot ser utilitzada en més d'una connexió (long-term), i una clau única per cada sessió (short-term) anomenada onion, per desxifrar peticions d'usuari i negociar altres claus de sessió [12].

Si bé els diferents tipus d'OR duen a terme la mateixa funció d'encaminador, cadascun es troba posicionat en diferents punts del circuit, per tant, com ja s'ha detallat a l'explicació de l'Onion Routing, la informació que aquests tenen sobre les connexions de les quals són partícips és diferent.

- **Guard Relay**: Es tracta del primer dels tres OR que acostumen a conformar el circuit, per tant, és el primer relay amb el qual l'OP negocia la clau de sessió. Quan el circuit ja s'ha establert, l'única informació que té sobre la connexió és l'adreça de l'OP que ha contactat amb ell, és a dir, la identitat de l'usuari i també l'adreça del següent relay que conforma la cadena.
- **Middle Relay**: Seguidament a la cadena es troba el middle relay, sent aquest el segon dels tres OR que la conformen i, per tant, el segon amb qui es negocia la clau de sessió. Tractant-se d'un node intermediari entre el d'entrada i el de sortida, la informació que posseeix no és més que l'adreça del node que l'antecedeix i la del que el precedeix.
- **Exit Relay**: Finalment, en l'últim salt del circuit es troba l'últim relay, l'exit relay, el qual enviarà la petició inicial de l'usuari al servidor destinatari. En aquest cas, la informació que manté de la connexió és: per una banda l'adreça del node que l'antecedeix, i per altra banda el missatge en clar que s'està transportant, en el qual s'inclou l'adreça IP del servidor a qui va dirigit. L'usuari, per tant, acabarà navegant amb la IP d'aquest exit relay en comptes d'amb la seva adreça original, a diferència que aquest intermediari no tindrà cap informació sobre l'usuari que està emmascarant.

4.2.2 Serveis de directoris

Els serveis de directori són nodes de confiança que actuen a la xarxa Tor. En ells, els nodes que es troben operatius publiquen el que s'anomena com un server descriptor, que és un petit resum amb informació referent al seu rendiment i la forma en la qual altres usuaris poden connectar a ells [2]. D'entre aquesta informació que es recopila en els serveis de directori, es troba l'adreça IP del node, el port on s'està executant i la seva clau pública, entre altres. Per assegurar la confiança d'aquests descriptors, la informació que es publica es troba signada pel mateix node amb la clau d'identitat que posseeix, d'aquesta forma la veracitat de la informació podrà ser certificada per l'usuari.

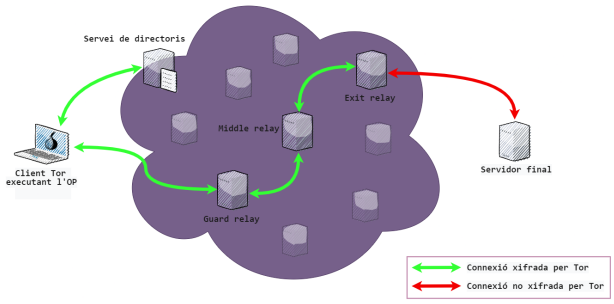


Fig. 3: Topologia bàsica de la xarxa Tor.

5 Simulació

Considerant que l'objectiu principal de Tor és oferir als usuaris l'anonimat dins la xarxa, és necessari l'estudi i investigació tant dels atacs que vulneren aquest anonimat, com també d'aspectes que tenen a veure amb el disseny del mateix Tor, el seu rendiment o l'escalabilitat de la xarxa. Aquestes investigacions que es duen a terme poden comportar la implementació de nous dissenys que ajudin a reduir les vulnerabilitats que el disseny actual pateix, com també l'estudi directe d'aquelles tècniques en forma d'atacs que s'aprofiten de les vulnerabilitats, aconseguint així un major enteniment del "com" per aconseguir donar amb contramesures.

Sigui quin sigui l'objectiu i resultat de les investigacions, aquestes requereixen, com és lògic, de l'accés a la xarxa la qual volen estudiar, Tor, i a les dades que aquesta genera. No obstant això, fer ús directe de la xarxa per part dels investigadors perquè aquests puguin aplicar els seus nous dissenys o replicar atacs no és, certament, una bona idea. Haver de propagar per milers de nodes un nou disseny que tingui a veure amb la metodologia de xifratge dels paquets, o dur a terme replicacions d'atacs sobre tràfic d'usuaris reals, repercutiria de forma directa tant en l'estat i funcionament de la xarxa, com en la privacitat de les dades dels usuaris.

Aquests motius fan necessària la cerca d'alternatives on poder dur a terme la realització d'aquests experiments. Amb aquest fi, el grup d'investigadors Tor Research Safety Board [6] estableixen unes pautes que aquestes alternatives haurien de complir per tal de minimitzar els riscos de privacitat a l'hora de dur a terme estudis sobre la xarxa Tor. Es destaca: utilitzar una xarxa Tor de testatge sempre que sigui possible, això és, una xarxa sense usuaris reals; recopilar les dades necessàries i que aquestes siguin de caràcter públic; dur a terme atacs contra tu mateix i sobre el teu propi tràfic.

Per adreçar aquestes pautes, específicament la de crear una xarxa Tor sense usuaris reals, es pot optar per la **simulació**, de forma que aquesta simulació recrees o idealitzés el comportament real de la xarxa mantenint sempre l'execució completament aïllada i independent de la xarxa i els seus usuaris. Existeixen múltiples desenvolupaments [7] que mitjançant la simulació o bé l'emulació, aconsegueixen formar recreacions més o menys acurades on poder desenvolupar les recerques i investigacions, de forma que els seus resultats puguin ser extrapolats a la xarxa real. Alguns d'ells són Tor Path Simulator, per simular el procés de construcció de circuits, fet pel qual no és apte per la simulació de xarxes, o ExperimenTor, per emular xarxes, descartat per requerir més d'una màquina.

El següent apartat, però, se centrarà més precisament en el simulador Shadow [8] i en com aquest, amb l'aplicació d'eines de modelatge, aconsegueix recrear de forma acurada el comportament de Tor en un entorn controlat. S'analitzarà el seu funcionament, les diferències que manté amb la xarxa real, executant simulacions on poder validar les limitacions que puguin existir.

5.1 Shadow

Shadow és una eina d'experimentació híbrida, això és, que utilitza la simulació i l'emulació per dur a terme els seus experiments. Generalment, se'l descriu com un simulador d'esdeveniments discrets[1] capaç d'executar aplicacions reals, com Tor, en entorns simulats i en una sola màquina [8]. Principalment, Shadow es concep com una necessitat de tenir una eina que fos precisa, eficient i escalable amb la qual poder dur a terme experiments de la xarxa Tor en una única màquina, i alhora poder controlar tots els aspectes que tinguin a veure amb l'experiment [9].

Tenint present les capes que componen el clàssic model TCP/IP, Shadow simula la capa de xarxa, executant alhora el software real de Tor. Per aconseguir aquesta interacció entre el simulador i l'aplicació real, Shadow treballa conjuntament amb plug-ins, els quals no deixen de ser llibreries que es troben vinculades amb l'aplicació real i que encapsulen totes les funcions necessàries per a la interacció. Utilitzar plug-ins, doncs, permet a Shadow carregar el codi font de l'aplicació de forma dinàmica, mentre aquest executa la simulació en qüestió. Així mateix, gràcies al fet que Shadow és capaç d'emular un entorn Linux, quan l'aplicació s'executa en el simulador ho fa com si ho fes en un entorn Linux real [9].

Tal com s'assenyala en el seu disseny [9], les funcionalitats que Shadow acaba oferint en relació amb Tor són: la creació d'entorns simulats i aïllats, on hosts o usuaris virtuals poden comunicar-se entre ells sense l'ús d'Internet; la simulació múltiple d'aquests hosts virtuals en un temps virtual; l'execució d'aplicacions reals de forma nativa; i l'execució de xarxes Tor privades amb models d'usuari i tràfic creats a partir de mètriques reals, extretes també de la xarxa Tor.

Per executar una simulació a Shadow, es requereix del que s'anomena com un blueprint o un plànol de la simulació en qüestió. En aquest plànol, s'inclouen totes aquelles dades necessàries que Shadow necessita saber, tant per crear la xarxa virtual com per generar el tràfic dels nodes virtuals. Per exemple, el nombre de hosts que comprendran la xarxa, el software que executaran cadascun d'ells o també dades més específiques sobre l'estructura de la topologia de la xarxa, tals com la latència, el jitter o la taxa de pèrdua de paquets [9]. El simulador no disposa de les funcionalitats necessàries per crear el plànol en qüestió, per tant, és necessari l'ús d'altres programaris (explicats a continuació) que siguin capaços de modelar tots aquests paràmetres.

6 EXPERIMENT: SIMULACIÓ VS. TOR REAL

Amb la finalitat d'entendre com Shadow aconsegueix dur a terme una simulació, en aquest cas de Tor, i determinar si efectivament ofereix uns resultats que l'aproximen al màxim al seu model de referència, en aquest apartat es du a terme un experiment en el qual es modelaran xarxes compostes per un nombre de nodes diferent, tot utilitzant eines complementàries, per posteriorment simular-les amb l'eina descrita.

El que es pretén simular és un comportament HTTP (això és, la transferència de fitxers entre dues bandes) a través d'una xarxa Tor privada. Cada client configurat establirà múltiples connexions amb els diferents servidors, creant prèviament els circuits basats en tres relays. A través d'aquests circuits, clients i servidors duran a terme enviaments i descàrregues constants de fitxers de diferent mida, tot utilitzant la metodologia de xifratge descrita anteriorment.

La composició de les xarxes a simular es troben descrites a la Taula 1, oferint l'escala respecte a la xarxa total de Tor [10], sent per exemple 0.01 un 1% dels relays totals i un 1% de la càrrega de tràfic del Tor públic. Aquest experiment se centra solament en la manipulació del nombre de relays o OR, fet pel qual el nombre de servidors i clients es mantindrà constant per les tres simulacions. Executant xarxes de diferent mida i per tant, obtenint un major conjunt de dades, es pretén comprovar quina és la influència

[1] S'entén com a simulador d'esdeveniments discrets al mètode de modelar sistemes del món real, en aquest cas Tor, i que aquests es puguin descompondre en un conjunt de processos que progressen de forma autònoma a través del temps.

TAULA 1: COMPOSICIÓ DE LA XARXA SEGONS LA SEVA ESCALA

Mida	Relays Totals	Guard Relays	Middle Relays	Exit Relays	Exit-Guard Relays	Clients	Clients de Rendiment	Servidors	Servei de Directoris
1.2%	82	30	36	5	11	100	10	10	3
1%	67	25	29	4	9	100	8	10	3
0.7%	48	18	21	3	6	100	6	10	3

que té augmentar progressivament el nombre de relays per aproximar-se d'una forma més precisa al Tor públic.

A fi de determinar la precisió que mantenen amb el seu model de referència, s'extrauran mètriques de rendiment a partir de les tres simulacions i de la xarxa Tor real. Les mètriques a comparar són tres: el temps de construcció de circuits individuals, el goodput que mantenen els relays i clients, i el temps de transferència de fitxers de mida de 50 KiB, 1 MiB i 5 MiB. Per això, tant per simular xarxes el màxim representatives del Tor real, com per la posterior comparació dels resultats, serà necessària l'extracció de dades reals de Tor, detallades a apartats següents.

A continuació, es passa a detallar tota la configuració tècnica de l'experiment, conjuntament amb les diferents fases que s'han dut a terme fins a extreure els resultats.

6.1 Programari

Com bé s'ha descrit en apartats anteriors, prèviament a simular una xarxa és necessari modelar la seva topologia i el comportament que prendran els nodes que la componen. A la Taula 2 s'introdueixen totes les eines que s'utilitzaran per al modelatge i la simulació[2]. Pel cas de Tornettools [11], aquesta es tracta d'una eina creada per guiar a l'investigador a través de tot el procés d'experimentació, incloent-hi les etapes referents a la modelació de la xarxa, la simulació d'aquesta i la posterior visualització dels resultats [11], conjuntament amb OnionTrace. Pel que fa a TGen, aquest s'executa pel mateix Tornettools per tal de modelar i generar tot el tràfic que circularà entre els nodes de la xarxa a simular, tant l'originat per usuaris com l'originat pels servidors.

TAULA 2: PROGRAMARI UTILITZAT.

Programari	**Utilitat a la simulació**
Shadow	Simula la xarxa generada.
Tornettools	Modela i genera la topologia de la xarxa.
TGen	Modela i genera el tràfic entre els nodes.
OnionTrace	Registra paràmetres rellevants.

6.2 Entorn

D'acord amb les estimacions oficials [11] i les limitacions amb les quals compta Shadow, executar una xarxa equivalent a l'1% de la xarxa Tor pública requeriria uns requisits hardware mínims, com 4 CPU-cores i almenys una memòria RAM de 35 GB.

Per complir amb els anteriors requisits, s'ha optat per la utilització d'un servidor al núvol (cloud server) que posseeixi els recursos suficients per a executar les simulacions previstes. La plataforma cloud escollida ha estat 'IONOS by 1&1' [12], la qual ofereix diferents paquets de servidors amb diferents recursos ja definits i escalables. El servidor amb el qual s'ha comptat per executar les simulacions ofereix les característiques descrites a la Taula 3, més que suficients per abastir els requeriments de les tres simulacions.

TAULA 3: CARACTERÍSTIQUES SERVIDOR CLOUD.

CPU	vCores	RAM (GB)	SO
Intel® Xeon®	24	48	Ubuntu 18.04

6.3 Fases de l'experiment

En aquest apartat es detalla pas a pas quins són els passos seguits per aconseguir simular les tres xarxes especificades. De la mateixa forma, s'especifiquen les funcions de Tornettools que s'han utilitzat en aquest procés. A la Fig. 4 s'observa el circuit de passos seguits fins a obtenir els resultats.

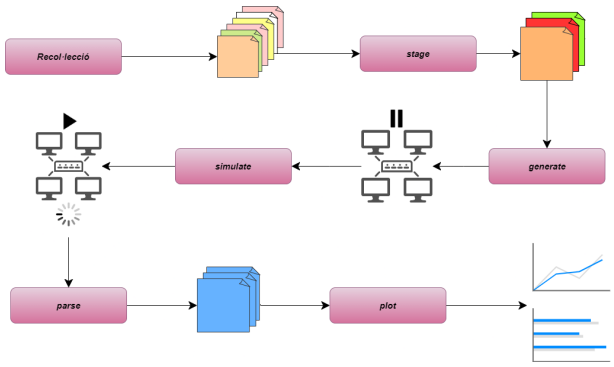


Fig. 4: Passos de l'experiment.

6.3.1 Fase 1: Recol·lecció de dades

Per la realització d'aquest experiment, s'han extret dades i mètriques que es van registrar a Tor durant el mes de març del 2021, en altres paraules, les xarxes a simular i les comparacions de resultats tindran com a model la xarxa Tor d'aquest mes esmentat.

A continuació, es llisten les dades recol·lectades d'un servei de col·lecció de dades anomenat CollecTor, on s'extreuen dades mensualment de diferents nodes i serveis del Tor real [13].

- **Consensos de l'estat de la xarxa**: col·lecció de dades generada pels serveis de directoris (cada hora) on es manté informació sobre la xarxa, és a dir, informació sobre els relays que la componen.
- **Descriptors dels serveis**: informació que els ORs publiquen sobre ells als serveis de directoris per fer-se visibles al públic.
- **Dades d'OnionPerf**: conjunt de dades recol·lectades per OnionPerf, una eina que mesura el rendiment a llarg termini de la xarxa Tor [14].

[2]https://github.com/shadow/ {oniontrace, tgen, tornettools, shadow}

Altres dades extretes, no de CollecTor sinó del portal de mètriques oficial de Tor [15], són les següents:

- **Estadístiques dels usuaris**: dades referents a les estimacions del nombre d'usuaris per països existents a Tor.
- **Amplada de banda**: mesures del total d'amplada de banda disponible i la capacitat de la xarxa.

Complementàriament a totes aquestes dades, és necessari descarregar el **codi font de Tor** per fer ús d'algunes de les seves funcions, tals com generar les claus que cada OR posseeix. Igualment, es necessiten mesuraments de tràfic fets amb **PrivCount**. Aquest és un sistema de mesuració que permet extreure moltes més mètriques de Tor d'una forma més segura, en la qual no existeixin possibilitats de filtracions d'usuaris reals [16].

6.3.2 Fase 2: Configuració de la simulació

Tornettools-*stage*. Seguidament, s'executa la funció *stage* passant-li per paràmetres totes aquestes dades esmentades. L'objectiu d'aquesta funció és la de processar tota la informació recol·lectada sobre relays i usuaris, per posteriorment poder-la utilitzar per generar la xarxa i comparar-la.

El resultat de l'execució és, primerament, la generació d'un fitxer creat per totes les funcions de Tornettools, el qual registra errors i avisos d'aquestes funcions. En segon lloc, es generen uns altres tres fitxers: dos d'ells guarden totes les dades dels relays i usuaris processades en format JSON, i un tercer fitxer JSON per guardar les mètriques de Tor per la posterior comparació de resultats.

Tornettools-*generate*. Al següent pas s'executa *generate*, funció que genera la configuració que tindrà la xarxa Tor a simular utilitzant les dades que s'han processat a la funció anterior. Amb aquest fi, com s'observa a la Taula 4, s'**especifica** l'escala de la xarxa i també la ràtio de clients i relays, en aquest cas 1/0.01=100 clients i 1/0.1=10 servidors.

TAULA 4: CONFIGURACIONS DE LA XARXA.

Mida	Escala	Ràtio Clients	Ràtio Servidors
0.7%	0.007	1/0.01	1/0.1
1%	0.01	1/0.01	1/0.1
1.2%	0.012	1/0.01	1/0.1

A fi de poder entendre millor quins són els passos en l'execució de tot aquest procés i, per tant, quina forma prendrà la xarxa, s'analitzarà els continguts més rellevants de l'arxiu de registre o log que la funció *generate* crea per registrar allò que fa en tot moment.

Primerament, es determina el nombre de relays que conformaran la xarxa. Segons les dades recol·lectades al mes de març de 2021, se'n detecten un total de 6718 com actius, per tant, es creen d'acord amb l'escala especificada, com s'ha pogut observar a la Taula 1.

A continuació, es generen fitxers de configuració de Tor en format .torrc. En ells es registra informació relacionada amb els actors que conformen la xarxa, per exemple, l'adreça IP i port dels tres serveis de directoris, els diferents ports i timeout que tenen els clients, si la política de sortida d'un relay està activa (és a dir, és un exit relay) o no, entre altres:

[INFO] Generating Tor configuration files

Seguidament, es generen els clients. D'acord amb la ràtio especificada a la Taula 4, se'n generen 100 utilitzant TGen. Aquests 100 clients, però, emulen el comportament (extret de CollecTor) d'un nombre diferent de clients reals, com també creen una quantitat diferent de circuits cada 10 minuts, depenent de la mida de la xarxa. S'observa a la següent Taula 5.

TAULA 5: COMPORTAMENT DELS CLIENTS.

Mida	Clients	Clients emulats	Circuits creats / 10min
0.7%	100	5544	10400
1%	100	7919	14900
1.2%	100	9503	17900

Al Tor real, a fi de tenir un registre sobre el rendiment de la xarxa en tot moment (benchmark), s'hi descarreguen fitxers de diferents mides per així obtenir dades relatives a la velocitat, retard o altres paràmetres en relació amb el rendiment. Per cada simulació, es duu a terme un procés similar creant clients addicionals (clients de rendiment): 6 per la xarxa equivalent al 0.7%, 8 per l'1% i 10 per l'1.2%, els quals descarregaran fitxers de 50 KiB, 1 MiB i 5 MiB per així poder comparar de forma directa el rendiment obtingut a la simulació i a la xarxa real [11].

Com últim actor en la simulació, es generen els 10 servidors finals, utilitzant també TGen, perquè aquests responguin a totes les peticions que rebin per part dels clients.

[INFO] Generating Servers

[INFO] We will use 10 TGen servers to serve 100 TGen clients

Finalment, la funció *generate*, valgui la redundància, genera els fitxers de configuració tant de TGen com de Shadow, tots ells modificables per l'usuari. En el cas dels fitxers de configuració de TGen, se'n produeixen de diferents per tal de modelar les característiques del tràfic d'acord amb el Tor real. Per començar, es creen dos fitxers on s'especifica: el model que prendrà el paquet individual (i.e. el seu identificador, el seu tipus, el seu pes, la taxa d'enviament...) i el model que prendrà el flux de paquets (stream).

Igualment, se'n creen fitxers de configuració on es modela el comportament del tràfic de tots aquells actors que generin tràfic. Existeix un de sol per tots els servidors i un altre pels clients de rendiment, on s'especifica la mida dels paquets que descarreguen. En canvi, pels 100 clients restants, existeix un fitxer de configuració individual per cadascun d'ells per aconseguir que tinguin diferents característiques en el tràfic, com un país i ciutat d'origen diferent o diferent quantitat de temps entre creació de circuit i enviament de paquets, com també els models creats anteriorment pels paquets i fluxos.

Pel cas de Shadow es crea un sol fitxer en format XML, anomenat shadow.config.xml. Aquest fitxer serà el fitxer utilitzat per Shadow per dur a terme la simulació (el blueprint). En ell, com si es tractés d'un índex, es carrega la localització de tots aquells plug-ins utilitzats i s'indica, per tots els components o nodes de la xarxa, el seu respectiu

fitxer de configuració de TGen i Tor. Així mateix, s'especifica l'identificador de cada actor, una amplada de banda tant de pujada com de baixada, i s'assigna una adreça IP a cada host:

[INFO] Generating TGen configuration files

[INFO] Constructing Shadow config XML file

6.3.3 Fase 3: Simulació de la xarxa

Tornettools-*simulate*. Arribats a aquest punt, s'ha finalitzat el procés de modelació de la xarxa, generant tots aquells fitxers necessaris per poder dur a terme la *simulació*. Per tant, es fa ús de la funció *simulate* de Tornettools, la qual simplement fa executar **Shadow** passant-li el fitxer anteriorment definit com a shadow.config.xml, començant així la simulació.

D'igual forma que en fases anteriors, un cop la simulació ha finalitzat amb èxit, es generen diferents fitxers de sortida. En el cas de Shadow, aquest genera fitxers que tenen a veure amb el procés i resultat de la simulació. D'entre tots aquests, es destaca la creació d'un directori anomenat shadow.data/hosts, el qual conté directoris individuals per cada host o actor de la xarxa on s'emmagatzemen dades de caràcter privat i fitxers de registres. Cada relay i client emmagatzema un fitxer amb la sortida estàndard (stdout) de Tor, TGen i OnionTrace, entre altres. En ells, es pot trobar informació relacionada amb l'establiment de circuits amb altres relays, fluxos de dades completats o la quantitat de dades rebudes i enviades, entre altres. Igualment, a part de fitxers de registre, s'hi emmagatzemen les diferents claus criptogràfiques que posseeix l'OR.

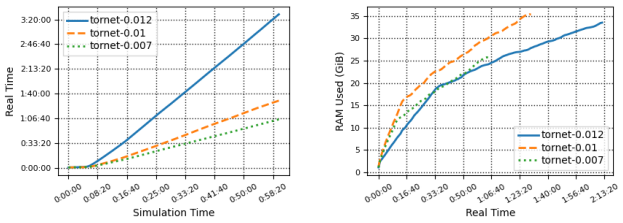
7 RESULTATS

Executades totes les simulacions satisfactòriament, es fa ús de la funció *parse*, una de les altres funcionalitats de l'eina Tornettools perquè aquesta pugui 'parsejar' o processar les dades que es consideren més rellevants i així puguin ser manipulables per ser representades. Per últim, s'opta per la funcionalitat *plot* per tal de representar les dades 'parsejades' en forma de gràfiques segons diferents mètriques de rendiment.

En els següents subapartats es començarà per analitzar les dades que tinguin més a veure amb les simulacions per seguidament passar a la comparació de mètriques de rendiment, entre les diferents simulacions i la xarxa Tor pública.

7.1 Temps d'execució i simulació

A la Fig. 5(a) s'exposa el temps que transcorre per simular les xarxes de l'1.2% (tornet-0.012), l'1% (tornet-0.01) i el 0.7% (tornet-0.007) del Tor real. Es mesuren dos tipus de temps: situat l'eix d'ordenades, el temps d'execució de la simulació ("Real Time"a la figura), i el temps de simulació, a l'eix d'abscisses. El primer fa referència al temps que triga el programa a simular la xarxa (des que s'inicia la simulació fins que finalitza). El segon, en canvi, fa referència al temps utilitzat a la simulació, el qual no transcorre al mateix ritme que el temps d'execució. Com una de les característiques dels simuladors, aquests tenen un control absolut sobre el temps, permetent que el temps pugui avançar més



(a) (b)

Fig. 5: Resultats de les tres simulacions: (a) Temps d'execució i simulació. (b) RAM consumida en temps d'execució.

o menys ràpidament segons la càrrega de treball i així ser més fidel al model de referència.

Donat que l'escala de la xarxa determina igualment la càrrega de tràfic d'aquesta, s'aprecia com el temps d'execució obtingut varia proporcionalment a la mida de la xarxa, obtenint temps de més de 3:20h per la tornet-0.12 i temps d'entre 1 i 2h per les altres tornet. Pel temps de simulació, obtenim sempre un mateix temps de menys de 60 minuts, gràcies al control sobre el temps de Shadow.

7.2 Recursos consumits

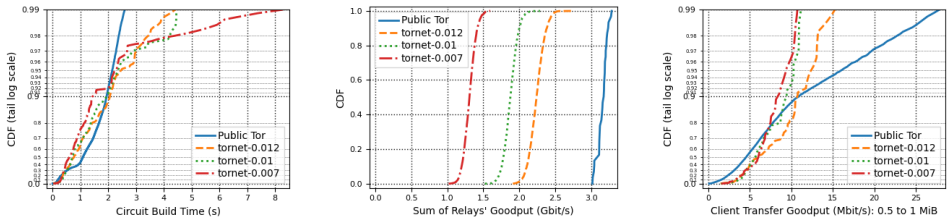
És cert que Shadow es troba directament limitat per la memòria RAM de la màquina on s'està executant i, per tant, no és capaç de simular xarxes de gran escala en màquines amb pocs recursos. Tanmateix, gràcies a la capacitat de controlar el temps, s'aconsegueix que Shadow pugui tenir l'escalabilitat necessària en tot moment, indiferentment de la càrrega del tràfic com de les capacitats de la màquina local.

A la Fig.5(b), es representa la memòria RAM que s'ha requerit per la simulació de les tres xarxes respecte al seu temps d'execució. D'acord les estimacions dels desenvolupadors mencionades a anteriors apartats, es confirma que el consum augmenta progressivament en el temps fins a requerir un total de 35 GB en el seu punt màxim, que correspon amb la part final de les simulacions tornet-0.012 i 0.01. Pel cas de la tornet-0.007, s'observa com la seva menor escala no acaba requerint més de 26 GB en el seu punt màxim.

7.3 Construcció de circuits

Com s'ha comentat a la primera part d'aquest projecte, de la mateixa forma que qualsevol altra xarxa distribuïda, contra més relays s'executin millor és el rendiment global de la xarxa. Igualment, una de les parts en les quals és necessari consumir més temps que en les comunicacions tradicionals és en la construcció d'un circuit, els quals es componen de forma general per tres relays.

A la gràfica 6(a), es mesura la quantitat de temps necessari per construir un circuit dins la xarxa, tant pel Tor públic, com per les tres simulacions de diferents mides executades. El temps es mesura respecte a la funció de distribució acumulada (CDF en anglès), això és, la probabilitat que la variable X, en aquest cas el temps de construcció, prengui un valor inferior o igual a un determinat valor x, sent x el nombre de segons. En altres paraules, pel cas de la següent gràfica, ens ajuda a determinar la probabilitat que la construcció del circuit s'hagi completat en una quantitat igual o inferior de segons. Com s'aprecia, arribats al 90è percentil



(a) (b) (c)

Fig. 6: Resultats de les tres simulacions i el Tor real: (a) Temps de creació de circuits. (b) Goodput dels relays. (c) Goodput dels clients.

es manté una igualtat gairebé exacta amb la Tor pública, sobretot les tornet-0.01 i 0.012, volent dir que amb un 90% de probabilitat, passats gairebé 2 segons, el circuit ja s'hauria construït. En aquest percentil s'observa igualment una petita desviació de la xarxa més petita tornet-0.007, que amb un 90% de probabilitats el circuit s'hauria pogut construir unes dècimes de segon més ràpid. Finalment, amb aquestes dades s'extreu la conclusió que amb un 92% de probabilitats, la construcció del circuit no durarà més de 2 segons aproximadament, tant en les tres simulacions com en la xarxa pública.

Si bé es considera que una probabilitat del 92% ja és concloent per demostrar l'efectivitat de Shadow, el gràfic també mostra que per alguns casos aïllats el circuit s'acabaria per construir en un temps superior als esmentats. Hipotetitzant, el motiu podria tenir a veure amb la relació entre el nombre de clients-relays, tenint en compte que és a la xarxa amb menys número de relays (tornet-0.007) on s'obtenen temps més grans en aquests casos aïllats. Aquesta relació clients-relays fa referència al fet que si els 100 clients tenen al seu abast més relays, per molt que la càrrega de tràfic augmenti, segueixen havent-hi més relays disponibles amb els quals conformar circuits, aconseguint una elecció més ràpida.

7.4 Goodput

Es coneix com a goodput a la taxa o velocitat amb la qual s'envien dades útils en una unitat de temps. A diferència del throughput que mesura el rendiment de totes les dades transmeses, incloent-hi les no desitjades com paquets d'error o les capçaleres dels paquets, el goodput sol mesura el rendiment de les dades originals, anomenades útils.

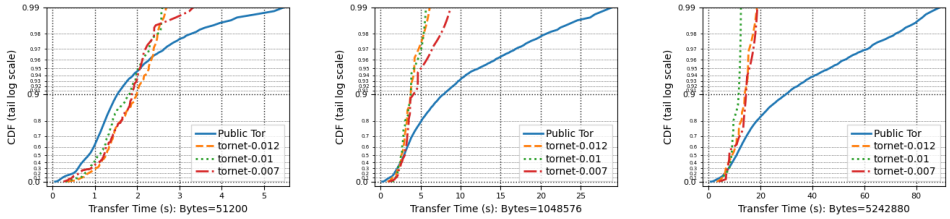
Entenent aquesta mètrica, a la gràfica 6(b) es representa la suma del goodput de les dades retransmeses pels relays, per cadascuna de les simulacions executades, conjuntament amb l'extreta de la xarxa Tor. Per la xarxa equivalent al 1.2%, amb un 100% de probabilitats s'obtindria un goodput d'aproximadament 2.6 Gbit/s, inferior als relays del Tor públic que poden arribar a aconseguir una taxa d'aproximadament 3.4 Gbit/s. Aquesta diferència de menys d'1Gbit/s augmenta a mesura que es redueix l'escala de la xarxa, obtenint aproximadament 2.3 Gbit/s amb l'1%, o pel cas del 0.7%, una diferència molt major de poc més d'1 Gbit/s respecte a l'1.2%. Considerant que el goodput s'obté a partir de la divisió de la mida de les dades transmeses entre el temps de la transferència, com major sigui aquest temps menor és la velocitat amb la qual s'envien aquestes dades. Si a més es considera que la mida de les dades és la mateixa per les tres xarxes, es pot arribar a la conclusió que en relays de xarxes de major escala s'obtindrien menors temps de transferència, aproximant-se més al Tor públic.

S'obtenen resultats similars a la gràfica 6(c), on es representa el goodput dels mateixos clients en transferències de 0.5 a 1 MiB. S'observa com les xarxes tornet-0.01 i 0.007 aconsegueix la màxima igualtat amb el Tor públic passat el 75è percentil, volent dir que amb una probabilitat aproximada del 75% els clients d'aquestes xarxes aconseguirien goodputs de transferència menors o iguals a 7 Mbit/s. Un cop se supera el 80è percentil, les tornet-0.01 i 0.07 començarien a patir un desviament del seu goodput respecte al Tor real. S'obtenen uns millors resultats pel cas de la tornet-0.012, la qual aconseguiria una igualtat amb Tor al 90è percentil, amb un goodput d'aproximadament 11 Gbit/s. Els motius pels quals el goodput es pot veure afectat en una xarxa poden ser diversos, des d'un augment de la latència en la transferència dels paquets, la pèrdua de paquets o un major grau de congestió de la xarxa.

7.5 Temps de transferència

Com s'ha especificat a la Fase 2, la funció *generate* crea els fitxers de configuració de 8 clients addicionals anomenats performance clients o perfclients, encarregats de mesurar diferents mètriques de la xarxa simulada i així tenir un índex de referència del rendiment de la simulació. Per fer-ho, descarregaven diferents fitxers de mida 50 KiB (51200 bytes), 1 MiB (1048576 bytes) i 5 MiB (5242880 bytes), i calculaven el temps de transferència obtingut. A les gràfiques següents, es proporciona el temps de transferència per descarregar els fitxers de diferent mida, obtinguts tant pels 8 perfclients a les tres simulacions, com pels perfclients del Tor públic.

Començant per la Fig.8(a), s'aprecia com a la descàrrega del fitxer de 50 KiB les simulacions obtenen uns temps de transferència lleugerament menors a Tor fins a arribar al 95è percentil. Arribats a aquest percentil, es veu com el Tor real conjuntament amb tornet-0.01 i tornet-0.07 aconsegueixen un temps de transferència menor o igual a 2 segons, amb una probabilitat del 95%. Pel cas de la xarxa tornet-0.012, la igualtat més gran amb Tor es veu al percentil següent, amb un temps màxim aproximat de 2.4 segons amb un 96% dels casos. A mesura que la mida del fitxer augmenta, s'observa a les figures 8(b) i 8(c) que es comencen a obtenir temps de transferència més petits per les simulacions. Amb un 90% de probabilitats, Tor aconsegueix descarregar fitxers d'1 MiB en aproximadament 7 segons, mentre que amb la mateixa probabilitat i mida Shadow ho aconsegueix entre 4 i 5 segons. En el cas d'augmentar la mida per 5, la



(a) Fitxer de 50 KiB (b) Fitxer d'1 MiB (c) Fitxer de 5 MiB

Fig. 7: Temps de transferència de fitxers a les tres simulacions i a la xarxa Tor real.

diferència de temps segueix augmentant, obtenint pel cas de les simulacions descàrregues en un màxim d'entre 10 i 15 segons amb una probabilitat del 90%, depenent la xarxa, mentre que Tor ho aconsegueix en un màxim de 30 segons amb una mateixa probabilitat.

S'extreu, doncs, que la descàrrega de fitxers de major mida provocarà un distanciament cada cop més gran de les simulacions respecte al Tor real. Tanmateix, centrant-nos en la mida dels fitxers més grans, es tractaria d'uns 2 segons en el cas d'1 MiB o de 15 segons pel cas de 5 MiB, un retard que pot ser important per l'execució de certs experiments a Shadow o menyspreable en altres.

8 Discussió dels Resultats

Havent estudiat tres mètriques que demostren el rendiment de les simulacions i per tant, el rendiment de les xarxes, es pot concloure amb què Shadow manté una precisió respecte al Tor real suficient per a poder dur a terme experiments relacionats amb la xarxa anònima i mantenir uns resultats acurats i extrapolables amb la realitat. Les simulacions han demostrat mantenir una igualtat amb Tor amb probabilitats major o iguals al 90% tant per mètriques com el temps de construcció de circuits, com en el goodput que mantenen tant els relays com els clients que la componen, una probabilitat suficient per a aquest cas determinar la precisió del simulador simulant xarxes tan petites en comparació al total de la xarxa Tor.

Com s'ha esmentat anteriorment, la precisió requerida pel simulador pot ser diferent segons l'usuari i l'experiment que estigui duent a terme. És possible que segons la temàtica de l'experiment a efectuar, la precisió que es requereixi el simulador sigui molt més elevada a la qual s'ha requerit en aquest projecte. En aquest cas, la solució que s'hauria de seguir és la d'augmentar progressivament la mida de la xarxa per observar millors resultats. Al cap i a la fi, les diferents xarxes que s'han simulat en aquest projecte amb Shadow representen un percentatge molt petit de la xarxa global de Tor, per això, mètriques com el temps de transferència de fitxers poden no adequar-se amb el temps real segons la mida del fitxer. Tenint en compte aquesta situació, encara simulant xarxes que representen un 1% aproximat de Tor, el simulador ha acabat retornant resultats prou optimistes per a tenir una seguretat suficient a l'hora de simular Tor i extreure conclusions.

Es remarca igualment els relativament pocs recursos necessaris per executar aquestes simulacions. Sense la necessitat d'executar més d'una màquina alhora, com és el cas d'altres desenvolupaments, i amb poc més de 35 GB de memòria RAM, com s'ha pogut observar a la Fig. 5(b), s'han pogut executar xarxes compostes fins a 200 hosts els quals feien córrer múltiples processos alhora.

Finalment, es considera que la mida de les tres simulacions executades no ha estat prou variada per a obtenir resultats més significatius i per tant, haver pogut comprovar més eficaçment els efectes positius que la simulació d'una xarxa més gran (p. ex. un 10%) pogués tenir respecte a una de menys relays. No obstant això, tenint en compte aquesta limitació, s'han pogut estudiar mètriques com el goodput on sí que s'han pogut trobar distincions segons la mida de la xarxa, obtenint un millor goodput en aquelles xarxes amb més relays i, per tant, una major precisió respecte al Tor públic.

9 Conclusions

9.1 Avaluació dels objectius

En aquest treball s'ha efectuat un estudi de la xarxa anònima de baixa latència Tor. S'han estudiat els diferents tipus de sistema de comunicació anònima que existeixen, com també el model Onion Routing, sobre el qual el projecte Tor es va implementar. En ell, s'ha pogut entendre el procés de xifratge i com una comunicació formada per tres relays aconsegueix que cap dels relay del circuit sigui capaç de conèixer la identitat de l'usuari i alhora tenir el contingut del missatge que s'està transmetent. S'ha introduït igualment al lector els diferents components que componen la xarxa Tor, explicant el seu funcionament, els diferents tipus que poden existir i la importància que tenen per garantir la seguretat de l'usuari. Per tant, es dona com complert l'objectiu **O1** on es pretenia assolir tot aquest coneixement previ.

Havent comprès una base teòrica de Tor, s'ha introduït la importància de trobar metodologies per ajudar al fet que investigacions o millores puguin ser efectuades sobre una xarxa externa a la real. Si bé s'ha comentat breument sobre alguns desenvolupaments per assolir aquest objectiu, assignat com **O2**, la metodologia en la qual s'ha centrat finalment el treball ha estat la simulació, utilitzada amb l'eina Shadow. La simulació ha permès tenir l'oportunitat de poder simular xarxes proporcionalment menors a la Tor real, on els hosts que les componen poguessin realitzar les mateixes funcions detallades a la base teòrica del projecte. Utilitzant Shadow i eines de modelatge com Tornettools o TGen, s'ha complert l'objectiu proposat **O3**, en el qual es pretenia com a màxima prioritat poder executar simulacions de forma satisfactòria. En aquest procés, s'ha volgut centrar la lectura en el funcionament i passos que Tornettools duia a terme per generar els fitxers de configuració necessaris per-

què el simulador fos capaç d'executar la xarxa. D'aquesta manera, s'ha demostrat l'efectivitat amb la qual aquestes eines ofereixen a l'investigador una plataforma on poder dur a terme una experimentació precisa de Tor, com també de les múltiples configuracions amb les quals l'usuari compta a l'hora de configurar la xarxa.

Finalment, s'han extret totes aquelles dades resultants de la simulació que podrien ser representatives del comportament i rendiment del simulador, per posteriorment exposarles de forma gràfica. Aquestes gràfiques han permès determinar l'eficàcia amb la qual les simulacions s'han executat respecte al Tor original, amb les respectives limitacions ja comentades en certes mètriques, podent així confirmar el bon rendiment del simulador Shadow com la precisió de les eines per modelar les xarxes i el tràfic. Havent obtingut aquests resultats, es donaria per complert igualment l'últim dels objectius, l'**O4**.

9.2 Problemàtiques trobades

Els contratemps que més han influït en l'execució de les simulacions han tingut a veure amb Shadow. Si bé molts han estat relacionats amb el control de versions del programa (existint versions on certes funcions no funcionaven a causa d'una nova actualització), es pot destacar la falta de recursos com una de les troballes més importants. La falta de capacitat de memòria RAM en algunes de les simulacions i la poca informació que Shadow retornava sobre aquesta situació, suposava l'aparició d'errors dels quals no tenien una explicació lògica.

Un exemple és el codi d'error -9, que apareixia després de 90 minuts d'haver iniciat la simulació i provocava la detenció d'aquesta. Aquest codi sol alerta de la incorrecta execució d'un procés, i per tant no indica mai el problema concret. Arran de mantenir contacte amb els desenvolupadors, es va arribar a la conclusió que la detenció del programa passats 90 minuts podria tenir relació amb la falta de memòria RAM, ja que inicialment s'executava sobre una màquina amb 32 GB de RAM. Aquesta hipòtesi es reforça observant la Fig. 5(b), on el consum de memòria augmenta progressivament en el temps, raó per la qual triga un cert temps abans de la detenció. A causa d'aquesta incidència, es va optar pel lloguer d'un servidor al núvol i, gràcies a reportar-la, els desenvolupadors van acordar en crear, en un futur, una nova funcionalitat a Shadow per monitorar els recursos que aquest consumeix i així poder retornar missatges d'alarma, evitant aquestes situacions.

9.3 Futur Treball

Havent entès la funcionalitat de tot el programari utilitzat, com també de les múltiples opcions que s'ofereixen per modelar una xarxa Tor al gust de l'investigador, seria interessant aprofitar aquest coneixement per recrear escenaris més complexos que es poden dur a terme en el dia a dia de la xarxa Tor, com són els atacs. Donat que es té a l'abast l'opció de modelar el tràfic de cada client, així com el control de tots els relays de la xarxa (això inclou, control del tràfic d'entrada i de sortida) és possible recrear atacs on la manipulació i control del tràfic siguin fonamentals (p. ex. atacs de correlació).

AGRAÏMENTS

A la meva família, pel seu inestimable suport que m'ha donat la força per seguir endavant. Als meus amics, per l'ajuda desinteressada que m'han donat. A Rob Jansen, per ser partícip del desenvolupament de les eines utilitzades i la seva ajuda per entendre-les. I al meu tutor, Victor Garcia, per la seva guia i comentaris constants que han ajudat a donar forma a aquest treball. A tots vosaltres, gràcies.

REFERÈNCIES

[1] V. Vaishnavi, W. Kuechler and S. Petter, "Design Research in Information Systems," Desrist, Design Science Research Portal. Jan. 2004.

[2] R. Dingledine, N. Mathewson, P. Syverson, "Tor: The Second-Generation Onion Router," Naval Research Lab, Jun. 2004.

[3] M. G. Reed, P. F. Syverson and D. M. Goldschlag, "Anonymous connections and onion routing,ïn IEEE Journal on Selected Areas in Communications, vol. 16, no. 4, pp. 482-494, May 1998, doi: 10.1109/49.668972.

[4] D. M. Goldschlag, M. G. Reed and P. F. Syverson, "Onion Routing for Anonymous and Private Internet Connections," in Communications of the ACM, vol. 42, pp. 39-41, Feb. 1999, doi: 10.1145/293411.293443.

[5] R. Dingledine and N. Mathewson, "Tor protocol specifications". Available: https://github.com/torproject/torspec/blob/master/tor-spec.txt

[6] "Research Safety Board — Tor Project — Research", Research.torproject.org, 2021. [Online]. Available: https://research.torproject.org/safetyboard/. [Accessed: 22- Apr- 2021].

[7] F. Shirazi, M. Goehring and C. Diaz, "Tor Experimentation Tools,"2015 IEEE Security and Privacy Workshops, 2015, pp. 206-213, doi: 10.1109/SPW.2015.20.

[8] R. Jansen and N. Hopper, "Shadow: Running tor in a box for accurate and efficient experimentation.," in Proceedings of the Network and Distributed System Security Symposium - NDSS'12, The Internet Society, Feb. 2012. See also https://shadow.github.io

[9] R. Jansen and J. Newsome, "Shadow Documentation: Design Overview", GitHub, 2020. [Online]. Available: https://github.com/shadow/shadow/blob/main/docs/0-Design-Overview.md. [Accessed: 26- Apr- 2021].

[10] "Servers – Tor Metrics", Metrics.torproject.org, 2021. [Online]. Available: https://metrics.torproject.org/networksize.html. [Accessed: 20- May- 2021].

[11] R. Jansen, J. Tracey and I. Goldberg, "Once is Never Enough: Foundations for Sound Statistical Inference in Tor Network Experimentation," in 30th USENIX Security Symposium (Sec), 2021. See also https://neverenough-sec2021.github.io/

[12] ÏONOS by 1&1 ≫ Hosting Provider — Websites. Domains. Server.", Ionos.com. [Online]. Available: https://www.ionos.com/. [Accessed: 11- May- 2021].

[13] "Sources – Tor Metrics", Metrics.torproject.org, [Online]. Available: https://metrics.torproject.org/collector.html. [Accessed: 16- May- 2021].

[14] I. Learmonth, "OnionPerf · Wiki · The Tor Project / Metrics / Team", GitLab. [Online]. Available: https://gitlab.torproject.org/tpo/metrics/team/-/wikis/onionperf. [Accessed: 16- May- 2021].

[15] "Welcome to Tor Metrics", Metrics.torproject.org, [Online]. Available: https://metrics.torproject.org/. [Accessed: 16- May- 2021].

[16] R. Jansen and A. Johnson, "Safely Measuring Tor," Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS '16), 2016.

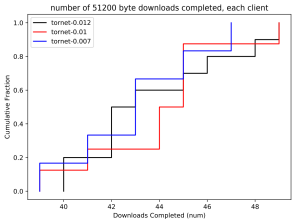
APÈNDIX

A.1 Resultats addicionals: Descàrregues per client

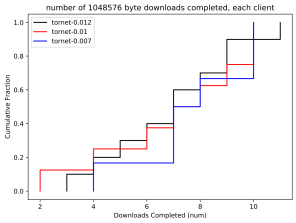
Altres resultats que aporten més informació sobre quin ha estat el comportament a les xarxes simulades són els que s'aprecien a la Fig. 8. En ella, s'especifica el nombre de descàrregues de fitxers completades per cada client de rendiment, segons la seva mida (en aquest cas 50 KiB, 1 MiB i 5 MiB).

Entre el nombre de descàrregues i la mida del fitxer s'observa una relació indirectament proporcional, produint-se un menor nombre de descàrregues en fitxers grans i un major nombre en fitxers petits. Recordant que el nombre de clients de rendiment varia segons la xarxa (vegeu la Taula 1), el nombre total de descàrregues fetes per aquests oscil·la entre 376 i 500 per fitxers de 50 KiB, 60 i 110 per 1 MiB, 30 i 70 per 5 MiB.

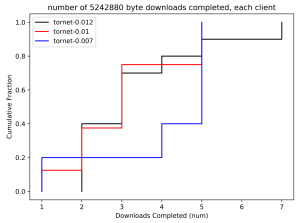
Entre les xarxes no es denota una influència destacable en el nombre de fitxers descarregats. Tanmateix, indiferentment de la mida del fitxer, el client de rendiment completa més descàrregues a la xarxa més gran, demostrant així una millor eficiència a la xarxa tornet-0.012.



(a) Fitxers de 50 KiB



(b) Fitxers d'1 MiB



(c) Fitxers de 5 MiB

Fig. 8: Nombre de descàrregues de fitxers, segons la seva mida i simulació.

A.2 1% de la Xarxa Tor

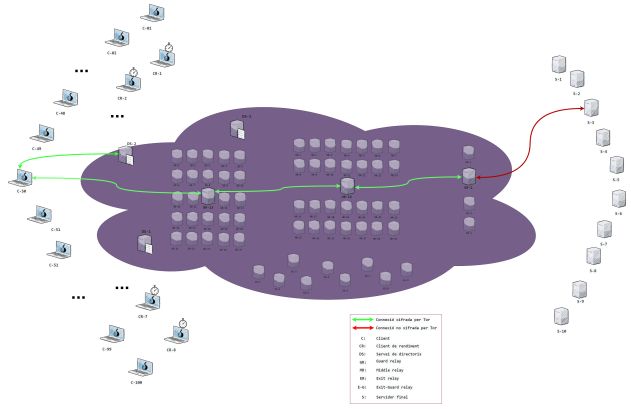


Fig. 9: Topologia de l'1% de la xarxa Tor.