LASTor: A Low-Latency AS-Aware Tor Client
“LASTor:自律システムを考慮した低遅延Torクライアント”

Masoud Akhoondi, Curtis Yu, and Harsha V. Madhyastha
Department of Computer Science and Engineering
University of California, Riverside
fmakho001,cyu,harshag@cs.ucr.edu

原文： http://lastor.cs.ucr.edu/oakland12.pdf

抜粋ー幅広く使われているTorの匿名ネットワークは応答速度の速い匿名通信を可能にすべく設計された。しかし、実際にはTorでのやりとりの9割を占める双方向通信（TCP）が、Torを使わない時と比べて、5倍以上の応答時間[1]まで遅くなっている。それに加え、Torにおける匿名通信経路の選択アルゴリズムは物理的なインターネット回線のルーティングに依存しており、その結果としてTorの経路上の入口側と出口側の両方に配置された自律システムは通信トラフィックを相互に関連付けすることができ、Torの匿名性は破たんする可能性もある。この論文では、Torクライアントソフトだけの改造で、既存のTorネットワークに直接手を加える事なく、以上に挙げた2つのTorの欠点に対処できる事を示す。また、その成果として、新たなTorクライアントソフト「LASTor」を開発・実装する。そのはじめとして、LASTorはTorの匿名通信経路選択中にTorリレーノードの位置を簡単に推論することで、既存のTorクライアントソフトより大幅に応答速度の向上ができる事を示す。また、Torリレーノードの選択アルゴリズムで、応答速度の速いノードを好んで選択する事は、経路選択の幅を狭めてしまうので、LASTorの経路選択アルゴリズムを自由にチューニングできるように改造する。これにより、LASTorユーザはたったひとつの0～1の値を指定することで、匿名性をとるか、応答速度をとるかの適当な具合を決められる。最後に、自律システムがTorの匿名通信経路の入り口と出口の両サイドで相関関係を抽出できてしまう経路であるかどうかを判別する、効率的かつ確実なアルゴリズムを開発する。このアルゴリズムはLASTorにそのような経路を避けさせユーザ匿名性を上げつつ、実行時間を短くして全体的な実行速度を上げる。これらの技術を使って、現在のインターネットを使って、LASTorで地理的に分散した200の有名Webサイトにアクセスした測定結果から、従来のTorクライアントソフトに比べて中央値で応答速度を25%向上し、偽陰性率（y/(x+y)）で潜在的な自律システムの未検出率を57%から11%まで引き下げた事を示す。

1. はじめに

Tor[2]はインターネットでの匿名通信のために構築された広く使われている匿名ネットワークである。匿名による通信を促進するその他のシステム[3][4]とは異なり、Torは高速通信を自負している。それもあってか、大半のTorユーザはTorをTCPに使っている。しかし、このTorの匿名性を上げるいくつかの手段は、通信応答速度を著しく低下させる。例えば、普通のTorクライアントソフトはターゲットサーバとの間に、設定されたアクセス帯域と信頼性をもつリレーノードをランダムに3つ選択し、トンネルを構築する。このリレーノードの選択手法は、わざわざ遠く離れた世界中のノードを経由する可能性があり、その結果通信速度の低下を招く。これまでにも、同じくTorのスループットを上げるのに焦点をおいた[5]や、既存のTorネットワークの改造に焦点を置いた技術（例えば、すべてのノードにネット

---

[1] McCoy, Damon et al. "Shining light in dark places: Understanding the Tor network." *Privacy Enhancing Technologies* 2008: 63-76.
[2] Dingledine, Roger, Nick Mathewson, and Paul Syverson. "Tor: The second-generation onion router." 2004.
[3] Danezis, George, Roger Dingledine, and Nick Mathewson. "Mixminion: Design of a type III anonymous remailer protocol." *Security and Privacy, 2003. Proceedings. 2003 Symposium on* 11 May. 2003: 2-15.
[4] U. Moeller, L. Cottrell, P. Palfrader, and L. Sassaman, “IETF draft: Mixmaster protocol version 2,” http://www.ietf.org/internet-drafts/draft-sassaman-mixmaster-03.txt, 2005.
[5] Snader, Robin, and Nikita Borisov. "A tune-up for Tor: Improving security and performance in the Tor

ワーク協調システムに参加させる[6][7]、Torノードの通信制御を改変する[8]など）はあった。しかし、これらの実装には大変な開発努力が必要であるので、これらの手法は未だに普及していない。加えて、Torの匿名性は、Torの経路選択アルゴリズムがインターネットの実際のルーティングに依存するために、ある条件下では破たんする事が証明されている。例えば、ある経路では、自律システムがTorリレーの入り口と出口を挟むように存在しているかもしれない。その自律システムは入口側と出口側のトラフィックから、Torを介して通信しているサーバとクライアントを統計的に推論して対応付けできる可能性がある。この問題については、[9] [10]でも取り上げられ、普通のTorソフトもこの問題に対抗して、1つのリレーの3つのノードは同じ値の先頭16bitのIPv4アドレスをもたないようにノードを選ぶようになったが、それでもこの手法では大半の自律システムによる詮索を検知するのにあまり十分とは言えない。この論文ではクライアントサイドのTorソフトの改造だけで、今日のTorの上記の2つの欠点に対処する方法を導く。このアプローチはTorネットワークそのものが改良されることを待たずに自身のTorソフトのアップデートだけで匿名性と応答速度の向上が結果として得られる事を示す。それにしたがって、次の問題の解を探す。

「今のTorネットワークを活かしつつ応答速度を上げ、さらに匿名性を脅かす自律システムを避ける経路とは？」

この解として、従来のTorソフトとは異なる経路選択アルゴリズムを持つ新しいTorクライアントソフト「LASTor」を実装する。LASTorの開発にあたり、3つの主要な要因を示す。ひとつは、大幅な応答速度の向上が、最新のインターネットの応答速度情報を一々取得するより、Torリレーノードの地理的な位置情報をも推測する方法の方が、より期待できることを証明する。これをWSP(重み最小化経路選択アルゴリズム)で実装したが、WSPの仕様上、Torユーザとターゲットサーバの間の地域に大量のノードを仕組む事で、敵はTorリレーの経路を掌握できる可能性が高くなってしまう。これに対処するため、LASTorに、WSPを、地理的に分散したTorノードを仮想のTorネットワーク上で隣同士に密にマッピングさせ実行させた。そうすることで、敵はリレーをその統制の中に治めるには様々な国家でノードを構築せざるを得なくなり、結果として敵の負担を増やすことができる。また、この副産物として、WSPの実行時間を大幅に減らすことが出来る。ふたつめは、LASTorに、自律システムによる選択経路の入口側と出口側での通信の対応付け攻撃に、そのような自律システムの存在する経路を避けさせる事で耐性をもたせた。そのためにLASTorにTorリレーとTorユーザ、ターゲットサーバの間にある2つのインターネットルーティングを推測する機能を追加した（リレーノードにルーティング情報を送信させる手法は、クライアントサイドだけの改造とは言えなくなるので、今回は行わない）。過去にでたインターネットの経路情報の予測手法はクライアントに一日数GBのダウンロードを要したり[11] [12]、非常に高機能な実行環境を要す[13]ので、（応答速度の速い経路を探すための準備時間があまりにも大きすぎて本末転倒であるという理由から）実用的ではない。したがって、そのかわりに詮索自律システムにひっ

---

network." *Proceedings of the Network and Distributed Security Symposium-NDSS* Feb. 2008.

[6] Sherr, Micah, Matt Blaze, and Boon Loo. "Scalable link-based relay selection for anonymous routing." *Privacy Enhancing Technologies* 2009: 73-93.

[7] Panchenko, Andriy, and Johannes Renner. "Path selection metrics for performance-improved onion routing." *Applications and the Internet, 2009. SAINT'09. Ninth Annual International Symposium on* 20 Jul. 2009: 114-120.

[8] AlSabah, Mashael et al. "DefenestraTor: Throwing out windows in Tor." *Privacy Enhancing Technologies* 2011: 134-154.

[9] Feamster, Nick, and Roger Dingledine. "Location diversity in anonymity networks." *Proceedings of the 2004 ACM workshop on Privacy in the electronic society* 28 Oct. 2004: 66-76.

[10] Edman, Matthew, and Paul Syverson. "AS-awareness in Tor path selection." *Proceedings of the 16th ACM conference on Computer and communications security* 9 Nov. 2009: 380-389.

[11] Mao, Z Morley et al. "On AS-level path inference." *ACM SIGMETRICS Performance Evaluation Review* 6 Jun. 2005: 339-349.

[12] Madhyastha, Harsha V et al. "iPlane: An information plane for distributed services." *Proceedings of the 7th symposium on Operating systems design and implementation* 6 Nov. 2006: 367-380.

[13] Madhyastha, Harsha V et al. "iPlane Nano: path prediction for peer-to-peer applications." *Proceedings of the 6th USENIX symposium on Networked systems design and implementation* 22 Apr. 2009: 137-152.

かかる可能性のある経路を陰性として偽陰性率が低くなる、計算工学的に軽量な技術を開発した。その鍵は、インターネットの経路情報をくまなく調べるのではなく、2つのIPアドレスから予測される通信経路を用いて中継されるであろう自律システムの集合を予測することにある。重要なのは、この自律システムの集合を予測するアルゴリズムでは、Torユーザは13MBの初期化データ、1週間に一度の1.5MBのデータをダウンロードするだけで済むということだ。そして、LAStorは経路選択をオプションでチューニングできるようにした。最短経路指向の経路選択アルゴリズムは選択肢の確率的エントロピーを減少させる。ユーザは応答速度の向上のために匿名性を犠牲にしたくはないだろう。したがって、LASTorはユーザに応答速度と匿名性の適切なバランスを自分で設定できるようにした。0~1の値のパラメータだけで、LASTorは経路選択を適切に行う。LASTorの高速化効果はPlanetLab(インターネットのテストサービス)の、地理的に分散している50のノードの、TOP200のWebサイトにアクセスすることで証明された。LASTorはTorネットワークに手を加えることなく、従来のTorクライアントソフトに比べて、中央値で25%の応答速度向上が見られた。また、インターネット上の20万以上の自律システムレベルでの経路を測定することで、LASTorはクライアントの匿名性を危うくする詮索自律システムを避ける能力を得た。そして、ユーザ-ターゲットサーバ間のペアで、中央値で、Lastorは詮索自律システムの内11%しか見逃さなかった。ちなみに、比較として、従来のTorクライアントソフトでは、偽陰性率で57%の詮索自律システムを見逃している。

2.動機と背景 このセクションでは、Torの様々な背景とこの論文の取り組みの動機を論ずる

A. Torの概要

ユーザにインターネットを匿名に利用させる、応答速度の速い、オープンソースのTor[14]は2002年に開発された。Torではクライアントが「ディレクトリサーバ」から「Torリレーのリスト」と「そのリレーの情報」をダウンロードする。コネクションを確立するために、クライアントは「entry」「middle」「exit」の3つのノードからなるリレーを選択し、匿名通信路を生成する。クライアントは、トラフィックを適切に暗号化した後、リレーの「entry」ノードに渡す。各ノードはその前後のノード(例えば、entryノードはクライアントとmiddleノード、middleノードはentryノードとexitノード、exitノードはmiddleノードとターゲットサーバ)のIPアドレスしか知らない。この「オニオンルーティング[15]」は、クライアントしかそのターゲットサーバが何であるか知り得ない事を利用し、クライアントの匿名性を保っている。従来からのTorクライアントソフトは、統計的な特定攻撃を避けるため、Torリレーのentryノードに、「entry guard」と呼ばれている、ランダムに選ばれた3つのノードを、固定されたリストから選択するようになっている[16]。middleノードは、可能なリレーのリストを帯域幅でソートし、その帯域幅が大きいほど高い確率で選ばれるようにした上で、ランダムに選択する。exitノードの選択においてはクライアントは、多くのリレーノードがexitノードとしてサービスしていない、という制約を受けるだろう。これはなぜならターゲットサーバはTorのexitノードを実際の通信相手としてとらえるからである(例えばもしターゲットサーバ上で悪質な行動が検知された場合、ターゲットサーバはexitノードにその責任があるとみなす)。従って、クライアントは、exitノードを選択する際には、クライアントが使いたいサービスを提供するターゲットサーバと通信する意志のあるexitノードをもつ(これもやはり、より広帯域幅の)リレーを選ばなくてはならない。

B. 動機

この論文の動機は今日のTorの経路選択アルゴリズムにおける2つの非効率的な原因に由来している。(つまりは回りくどい経路選択による大幅な遅延、インターネットルーティングに依存した経路選択による匿名性の劣化)

---

[14] “The Tor Project, Inc.” http://www.torproject.org.

[15] Reed, Micheal G, Paul F Syverson, and David M Goldschlag. "Anonymous connections and onion routing." *Selected Areas in Communications, IEEE Journal on* 16.4 (1998): 482-494.

[16] Wright, Matthew et al. "Defending anonymous communications against passive logging attacks." *Security and Privacy, 2003. Proceedings. 2003 Symposium on* 11 May. 2003: 28-41.

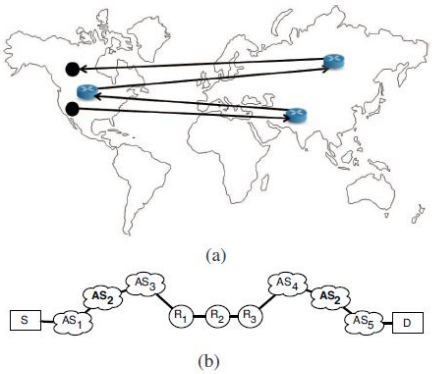

Fig. 1. (a) Random relay selection can inflate end-to-end latencies due to circuitous routing, and (b) an example in which an AS (AS2) can subvert the client's anonymity by correlating traffic across the entry and exit segments.

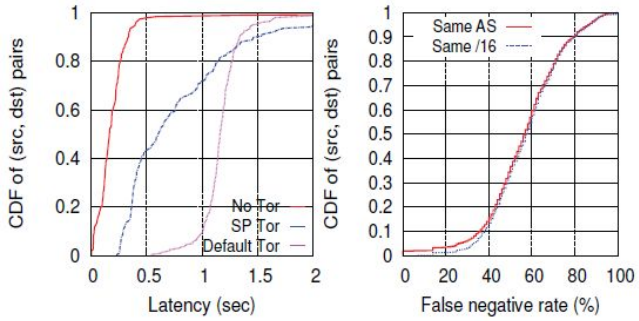

Fig. 2. (a) Comparison of latencies on the direct Internet path, with Shortest Path routing on Tor, and with the default Tor client. (b) False negatives in detecting snooping ASes with default Tor client.

遅延

これまでにも説明した通り、Torクライアントはentry,middle,exitノードを多かれ少なかれランダムに、ひとつの経路として選択する。その結果、クライアントとターゲットサーバとの実際の通信経路はしばしば回りくどくなり、Torを使わなかった場合の通信経路と比べて大きく遅延してしまっている。Torは主に、双方向通信（TCP）で使われている（例：Webサイトを見る）ので、遅延はユーザエクスペリエンスを劣化させることに直結してしまう。そのような例を図1の(a)に示す。これはアメリカのクライアントとカナダのサーバとの通信である。このクライアントは、送信したすべてのパケットがサーバに到着するまでに2度も世界中を飛び回るような非効率的なリレーの選択により、大幅な遅延を被るはめになっている。この遅延の度合を定量化するために、50の世界中のPlanetLabノード上の200のWebサイト[17]にアクセスし、遅延を調査した。それには、すべてのノードのWebサイトに対して5つのHTTP HEADリクエストを送信したときの遅延を観測した。まずはじめにPlanetLabのノード[18]に、Webサイトへ Torを介さず直接アクセスさせた。次に、同じ事を今度は、普通のTorで生成した接続を用いて行った。最後に、Torの3つのノードとクライアント、ターゲットサーバの地理的情報（MaxMind's IP geolocation database[19]を使って推論した）をベースにして定めた地理的な最短経路を普通のTorに指定してから同じ事を行った。すると、普通のTorでの通信は、直接の通信に比べて、中央値で5倍以上遅延したことを観測した。また、地理的な最短経路を指定させたTorは何もしていないTorよりも中央値で50%の遅延しか発生しなかったことを観測した。それでも、Torの経路構築を単純にクライアントとサーバの最短経路へと改変することはできない。なぜなら、それをすると選択経路が決まりがちになり、敵は戦略的にクライアントの匿名性を覆すリレーを組むことができるからだ。一つの案として、地理的な最短経路選択により遅延解消が見込めることから、本論文の最終目的は匿名性をないがしろにせずに、これらの遅延を解消できる確率的経路選択を可能にすることにある。

自律システム対策の欠如

Torのオニオンルーティングはクライアント以外がクライアントの通信しているサーバを知り得ないことを確保しようとしているにもかかわらず、そのような情報を推測しうる様々な攻撃手法が存在する[20] [21]。そのような攻撃がまかり通る理由の一つはTorの経路選択がインターネットのルーティングに依存しているからである。自律システムがクライアントとentryノード、exitノードとターゲットサーバの両方の間に存在している場合、その自律システムはその2つのト

---

[17] "Quantcast," http://www.quantcast.com/top-sites-1.

[18] "PlanetLab," http://www.planet-lab.org.

[19] "MaxMind - GeoLite City," http://www.maxmind.com/app/geolitecity.

[20] Edman, Matthew, and Bülent Yener. "On anonymity in an electronic society: A survey of anonymous communication systems." *ACM Computing Surveys (CSUR)* 42.1 (2009): 5.

[21] Hopper, Nicholas, Eugene Y Vasserman, and Eric Chan-Tin. "How much anonymity does network latency leak?." *ACM Transactions on Information and System Security (TISSEC)* 13.2 (2010): 13.

ラフィックペアを観察することで相関を導くことが出来る[22] [23]。図1(b)はそのようなAS2（クライアントSとentryノードR1,exitノードR2とターゲットサーバDの間の両方にAS2が存在する）を示す。また以後、そのような潜在的にトラフィック相関攻撃が可能な自律システムの事をSnoopingASと呼ぶ。ここで、注意してほしいことは、例えクライアントとentryノードの通信が暗号化されていたとしても、自律システムは暗号パケットのヘッダ部からクライアントのIPアドレスを見ることが出来るということだ。FreamsterとDingledine[9]は、、SnoopingASの存在確率は10~30%であると示した。この観測はEdmanとSyveson[10]によって再評価された。彼らは（当時に比べて）Torリレーのかずが多くなったので再度観測を行ったが、このTorネットワークの拡大は、SnoopingASの攻撃をほんのわずかしか緩和しなかったことを確認した。これは、なぜなら、Torリレーは自律システムにまんべんなく広がるのではないので、Torネットワークの成長は経路の地理的多様性を保証しないからである。さらにSnoopingASの出現は、クライアントとターゲットサーバが同じ中継自律システムを使う可能性から、クライアントとターゲットサーバが同じ場所に同居している場合の場所に特に多い。したがって、匿名性を守るために、Torクライアントは Torの匿名経路の両サイドに共通のをもつことを防ぐか、もしくは最悪でもそれを最小に留めることを保証しなければならない。自律システムによる攻撃を予防し匿名性を確保するために、Torは、entryノードとexitノードは同じ先頭16bitのIPアドレスであってはならないようにしている[24]。しかし、このヒューリスティックはSnoopingASを避けるには実際には効果的ではない。その理由の一つは、2011年6月にTorが開発されたとき、60%もの自律システムがその管理下に、複数の異なる16bitマスクのIPをもつTorリレーを持っていたことがわかったことだ。加えて、この「16bitプレフィックスヒューリスティック」を、AS経路DB（第3章で述べるPLBGP-Randデータセット）で評価した。サンプルデータのすべてのクライアント・サーバペアに関して、同じ自律システムを経路に持つ異なる16bitマスクのペアの数を、安全なペアの数と比較した。図2（b）はこれの偽陰性率を図示したものである。この「16bitプレフィックスヒューリスティック」は80%以上のペアで40%以上のSnoopingASを避けることに失敗している。これらのヒューリスティックの欠点を導くために、Torクライアントは、クライアントとentryノード間、exitノードとターゲットサーバの間の自律システムの存在を、実際のインターネットルーティングから決定しなければならない。この論文がクライアントサイドだけの改造を目標としている以上、Torリレーにそれを教えてもらうように、Torソフトを改造することはできない。また、iPlane[12]のような経路推測システムに問い合わせることはそもそもそのサービスに自分の通信相手を明かすことになり、それも論外である。一方で、クライアントに、事前に用意したすべてのクライアント・entryノード、サーバ・exitノード間にあるすべてのAS経路を知らせるには、法外な量のデータをクライアントがダウンロードしなくてはならない。例えば、もしインターネット上のすべてのリレーノードとエンドユーザを{平均ASパス＝4}としてBGPの集合[25]として数えたとしても、クライアントは500MB以上のデータをダウンロードしなくてはならない。しかも、そのデータはインターネットルーティングが動的な変動をすることから、常にアップデートしなければならない。それでも、クライアントに経路予測をローカルで行わせるべく、インターネットトポロジのスナップショットや経路情報をダウンロードさせることは不可欠である。しかしながら、現代の技術ではそのようなローカルの経路予測を可能にするには2つの問題が存在する。一つは「iPlaneのインターネット地図」のような数GBのデータを、経路予測のためだけに、クライアントにダウンロードさせること非現実的であるということである。もう一つはより軽量な「インターネット地図[11][13]」を使用するAS経路予測技術は膨大なCPU資源を、IPペア間のAS経路を予測するのに必要とすることである。Torクライアントは約1,000個のexitリレーを、経路構築の時に選ばなければならないため、このようなCPU的に「思い」方法をAS経路探索にとると経路選択に大きなオーバヘッドを課し、経路選択で得られる応答性の改善利益を議論の余地があるものにしてしまう。

---

[22] Mathewson, Nick, and Roger Dingledine. "Practical traffic analysis: Extending and resisting statistical disclosure." *Privacy Enhancing Technologies* 2005: 784-786.

[23] Murdoch, Steven, and Piotr Zieliński. "Sampled traffic analysis by internet-exchange-level adversaries." *Privacy Enhancing Technologies* 2007: 167-183.

[24] Dingledine, Roger, and Nick Mathewson. "Tor path specification." *2009-06-12]. http://www. freehaven. net/tor/cvs/doc/path-spec.txt* (2008).

[25] Broido, Andre. "Analysis of RouteViews BGP data: Policy atoms." (2001).

3. 概要　このセクションでは本論文がこれまでの実験で示し、焦点を当てた詳しい問題点について述べる。また、新たに開発するぎじゅつを検証するためのデータセットについても触れる。

A. 問題提議

Goal	Technique
Reduce latency of communication on Tor	Weighted Shortest Path (WSP) algorithm for probabilistic selection of paths with preference for low-latency paths
Defend against strategic establishment of relays to increase probability of compromised relays on chosen path	Clustering of relays in nearby locations
Enable user to choose trade-off between latency and anonymity	Augment WSP with parameter α that can be varied between 0 (lowest latency) and 1 (highest anonymity)
Account for distributed destinations	DNS lookup service on PlanetLab nodes
Preempt traffic correlation attacks by ASes	Lightweight algorithm to determine set of ASes through which Internet may route traffic between a pair of IP addresses

TABLE I
OVERVIEW OF TECHNIQUES DEVELOPED TO BUILD *LASTor*.

本論文の目的は既存のTorネットワークの設計に改造を加えずに上記に述べた応答速度と匿名性についての欠点に対処することにある。クライアント側でTorを活用するため、Torリレーのアップデートの開発・普及を待たずに、クライアントが今日のTorで利益を得るために、クライアントサイドの経路選択アルゴリズムの改造だけを探求する。その中で、伝統的Torのクライアントの匿名性を確保する仕組み（たとえば、Torリレー選択の偏化を防ぎ、Torリレーにある程度のランダム性を持たせて統計的特定攻撃を防ぐために3つの「entryGuards」を使う仕組み）も取り入れる。テーブルⅠはLASTorの開発にあたり、挙げられた問題点とその対処法についての表である。

B. 測定データセット

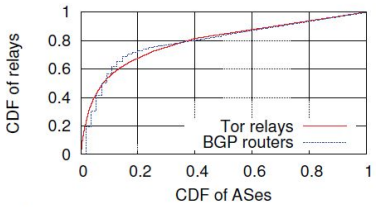

Fig. 3. Distribution of relays across ASes in *PL-BGP-Rand* and *PL-Tor-Web* datasets.

Dataset	Clients	Relays	Destinations
PL-Tor-Web	50	2423	200
PL-BGP-Rand	50	378	500
PL-PL-Web	50	50	500

TABLE II
SUMMARY OF DATASETS.

LASTorのアルゴリズムを評価するために、我々はTorクライアント[26]が世界中に散らばるよう分布されるように、50のPlanetLabノードをクライアントとして使い、テーブルⅡに示す3つすべてのデータセットに利用した。我々は最初のデータセットであるPlanetLabNode<=>Tor<=>Webサイトについては、Webサイトを200個[17]、実際に中継するTorリレーネットワークの宛先サーバとした。このデータセットでは、我々は遅延と自律システムレベルの経路情報を、Torリレーと宛先サーバの間のそれらの情報を取得する手段がなくとも、計算することができる。我々は、次のデータセットであるPlanetLabNode<=>BGP<=>Randomサイトについては、さまざまなBGPフィードサイト(BGPルータの情報を掲載しているサイト)[27] [28]にみられるBGPルータをリレーに設定し、互いに異なる24bitのネットマスクからランダムに抽出した、IPアドレスの末尾8bitが1である500個のサーバを宛先サーバに設定した。ここで再び、我々は`PlanetLabとBGPルータの間の遅延と自律システムレベルの経路情報を、直接計算することができる。それに加えて、我々はそれらの経路をたどった際の遅延を知らずとも、さまざまなBGPフィードからBGPルータと宛先サーバの間の自律システムレベルの経路情報を取得することができる。このデータセットは[9][10]のように推測された自律システムレベル経路を使うものとは異なり、我々の自律システム回避経路選択アルゴリズムを、厳密にインターネットで測定された自律システムレベル経路に基づいて評価することができるようにさせた。この目的ではBGPルータをTorリレーの代替として使用しなければならないため、図3に、PlanetLabNode<=>BGP<=>Randomサイトデータセットにおける、自律システムをまたがるリレーの分布が、実際のTorリレーを使用した時のものと似て分散していることを示す。最後のデータセットでるPlanetLabNode<=>PlanetLabNode<=>Webサイト では、我々はクライアントとリレーの両方にPlanetLabNodeを設定し、宛先にトップ200のWebサイトを設定した。このデータセットでは、我々はあらゆるクライアントとリレー、あらゆるリレーと宛先サーバの間の遅延と自律システムレベル経路を計測することができる。今回は典型的なTorクライアントたちを想定しているため、我々は全体を通して我々の評価行動に際しクライアントとして用いた50個のPlanetLabNodeで得たいかなるインターネットトポロジデータをiPlane[12]プロジェクトに対して提供していない。のちのセクション5でも述べるが、我々は自律システムの集合の推測にiPlaneのAS経路長を使用しているからである。

4. 経路選択　インターネットの経路は次の3つの要因が重なったものである。一つは伝搬遅延(パケットが回線を伝達する速度)、一つは待ちパケット行列遅延(パケットがエンドユーザもしくは中継するルータの送信キューに装填され、それが実際に伝搬されるまでキューに残る時間)、伝送遅延(パケットがキューから出て回線に移動するまでの時間)である。クライアントとTorリレーの通信帯域幅は我々の手が届かないので、伝送遅延については我々が削減できるものではない。のちにセクション7でも示すように、ある意味では、待ちパケット行列遅延はTorリレーの改造が不可欠である。したがって、我々はここでは伝搬遅延を削減することに焦点を当てる。

[26] Loesing, Karsten. "Tor metrics portal." Dec. 2010.
[27] Meyer, David. "Routeviews project." 2002.
[28] RIPE, NCC. "Routing Information Service." 2006.

A. 遅延が少ない経路を優先的に選択する手法

待ちパケット行列遅延を削減するために、我々は遠回りの経路を選択する可能性を減らさなくればならない。しかし、我々は安易に、クライアントと宛先サーバの間にある3つのリレーノードを、考えうる最短の経路として選択することはできない。これは、確定的な経路を選択することになり、意図的に敵によって配置されたTorリレーの影響を受けやすくなるからだ。したがって、我々は最小重み探索アルゴリズム(WSP)を実装する。WSPではその重みとして、クライアントと宛先サーバ間にある、推測されたすべての経路の遅延情報を要求する。その、経路による遅延とは、次の4つのセグメントにおける遅延の合計のことを指す。

Cllient<=Seg1=>EntryNode<=Seg2=>MiddleNode<=Seg3=>ExitNode<=Seg4=>destServer

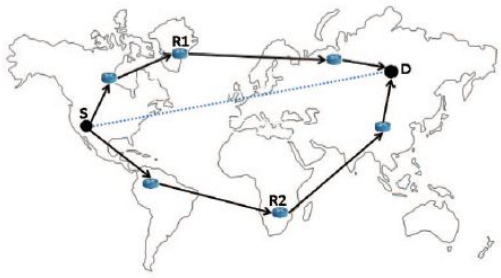

Fig. 5. WSP results in greater preference for paths through relays located close to the direct line between the client and the destination.

つまり、ある経路が選択されるという確率は、その推測された重み（遅延）に対して反比例することになる。しかしながら、クライアントと宛先サーバ、そして3つのノードの間にある考えうるすべての経路について遅延の推定を行うには、我々はこれらのリレーのペアについて、候補となるすべての経路について遅延情報を知る必要があるだろう。過去に提案されたTorの遅延を削減する手法[6][7]でもあるように、これらの遅延情報を掌握するにはTorリレーの改造を必要とするだろう。このインターネットの広大なスケールを考ええると、そのような調査活動は簡単なものではない。その結果として、これらの過去の提案はいまだに実行に移されたことがない。それに代わって我々の追及するのはTorクライアントの改造だけで行うWSPの実践的な実装である。ゆえに、我々は経路に沿ったエンドツーエンドの地理的な距離を、その経路に対する遅延情報の代わりとして使用した。これはつまり我々はその間にある遅延情報を追跡できるようにTorリレーを改造する必要をなくす代わりに、推測されたクライアント・リレーノード・宛先サーバの地理的情報に頼るということを意味する。我々は経路にかかわる、それぞれのセグメントの両端の間の地理的な距離(緯度、経度)をもとに順番に計算し、その距離を合計した。我々はエンドホスト(クライアントと宛先サーバ)とリレーの地理的な位置情報を（MaxMaind[19]のような）IP地理情報サービスくを使って知ることができる。これによりWSPはすべての候補経路の内、ひとつの経路を、確率的に重みの割合に基づいて、選択することができる（経路の重みはあらゆる通信経路の差異の最大値であり、それはすなわち、ある経路の距離としている）。この地理的な距離の使用は、既存のインターネットルーティングがもたらした（インターネットでもパケットを回りくどい経路で中継するかもしれないが[29]）経路情報を無視しているものであるけれども、我々はここで実験結果から、このWSPを実行する際のグラフで重みとして使われている地理的な距離に関するすべてのエッジ(グラフ理論でいう枝)が、遅延に関するすべてのエッジに対して合理的に代替品となっていることを強調する。我々はPlanetLabNode<=>PlanetLabNode<=>Webサイト のデータセットの経路しか、エンドツーエンドの遅延を計算することができないので、我々はこのデータについて解析を実行する。我々はこの解析を、初めにWSPを動かす時の重みとして、枝の重みを用い、次からそれと同じことを、枝の重みとして地理的な情報を使用して繰り返した。図4. はエンドツーエンドの遅延が、WSPが遅延をエッジの重みに使用している（PL-PL-WEBデータセットより。）場合と、地理的な距離をエッジの重みに使用している場合と、どちらかの場合にもかかわらず、似た遅延を持つ経路を選択したことを示している。したがって、我々はこの地理的距離を使用した手法は、骨の折れる仕事である、あらゆる地域に分散したインフラの遅延を計測する必要を確実にすることなく、多くの伝搬遅延を減らすことができると考える。

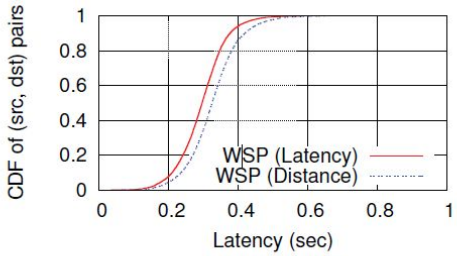

Fig. 4. Comparison of end-to-end latency when using geographical distance versus the use of path latency in the Weighted Shortest Path algorithm.

[29] Krishnan, Rupa et al. "Moving beyond end-to-end path information to optimize CDN performance." *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference* 4 Nov. 2009: 190-201.

B. Torリレーの分散

しかしながら、単純にWSPを実装することは2つの問題を引き起こす。一つは、WSPが地理的に距離が短いエンドツーエンドの経路を好んで選択することで、その結果、最終的なリレーはさらにクライアントと宛先サーバの間が直線的にもっとも近くなるように選択されてしまう。例えば、図5. では、WSPはリレーノードR1を、R2よりも高い確率で経路として選択してしまうだろう。結果として、もし敵がユーザがSからDまでのリレーを自分の管理下に引き込みたいときは、敵はSからDの間に直線を引いて、その近くを選択して大量のリレーノードをたちあげるだろう。この複数のリレーを同じ場所に配置する手法は、敵にとっては比較的簡単に実行できる。たとえば、クラウドサービスから複数の仮想マシンを借りることでも実行可能である。最低でもひとつ、敵の支配するリレーノードを選択してしまう可能性が高いことは、敵に、今日のTorのクライアントと宛先サーバを対応付ける攻撃[30]の隙を与えてしまう。二つ目の問題は、叩上げのWSPの実装であり、その計算に必要な時間である。今日、Torはざっと2500以上のリレーノードを持っており、そのうち1000のノードがexitノードとしても機能するように設定されている。クライアントとサーバの間にある経路の候補の数は、それに従い数十億単位である。なので、エンドツーエンドの地理的距離をすべての候補について計算することはCPU資源的に高価で、2.5GHzのCPUであっても一つの候補につき大体6.5秒もかかる。このような膨大なランタイム(普通のインターネットを使う場合の速度は一つにつき10~100msであるのに)はこの経路選択による遅延削減の成果を無駄にしてしまう。

この二つの問題を同時に解決するために、我々はTorリレーノードを地理的に「近くの場所」に集合(クラスタ化)させた。我々はこの世界を一つの四角い格子に分けて、あらゆるリレーノードをその格子に割り当てるという簡単な集合アルゴリズムを採用した(各セルの辺の長さは変更可能なパラメータである)。我々は、すべてのノードがリレーの集合として配置された&あらゆる経路の候補は3つの集合を通るような、集合TorネットワークについてWSPを実行した。WSPはすべての集合レベルのエンドツーエンドの経路距離を算出し、これまでと同様に、より距離が短くなるように一つの経路を導きだした。我々は、この算出された集合レベルの経路の内3つの集合から、それぞれ一つずつリレーノードを抽出し、それを最終的なTorリレーとして扱う。

このWSPアルゴリズムの改良により、従来の実装で6.5秒もかかる今日のTorネットワークにおけるクライアント・宛先サーバ間の経路の選択にかかる時間が、245ミリ秒にまで短縮させることができる。ここで重要なのは、この(大量のリレーを一つの「格子」として設置する)改良WSPアルゴリズムは、WSPが「経路」をその経路のノードが所属する集合という精度でとらえる(同じ集合内の異なるノードは同一のものとみなされる)という理由から、それらリレーの経路選択において偏りを見せないことである。かくして、この改良WSPアルゴリズムは敵(Torの匿名性を脅かす機関)の負担を増大させることができる(かなりの高確率で、敵に、自分の管理するノード選択させるためにより多くの地点にリレーノードを構築せざるを得なくさせることができる)。

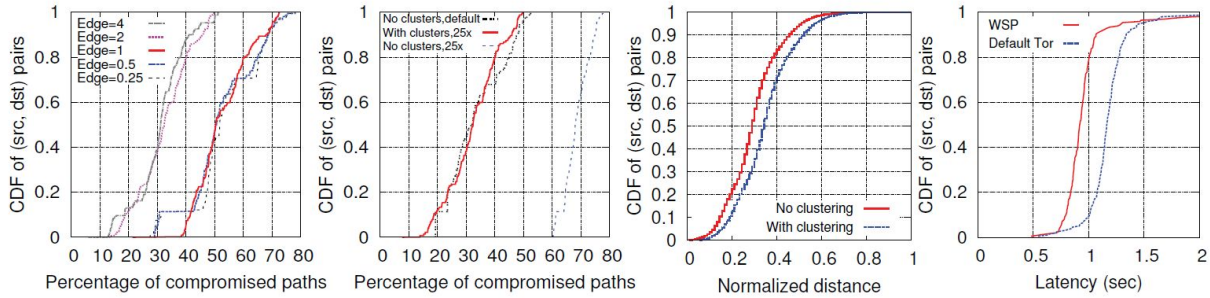

Fig. 6. (a) Clustering with higher cell sizes provides better resilience. Clustering of relays (b) reduces the probability of an adversary compromising a large fraction of paths, but (b) increases the length of the chosen path. (d) WSP yields latencies lower than those obtained with the default Tor client.

我々は実験1をリレーノードを囲う格子の大きさを決定するため行った。そして、実験2を上記のようにWSPが敵の

[30] Mittal, Prateek et al. "Stealthy traffic analysis of low-latency anonymous communication using throughput fingerprinting." *Proceedings of the 18th ACM conference on Computer and communications security* 17 Oct. 2011: 215-226.

攻撃に対する耐性を向上したことを証明するために行った。PlanetLabNode<=>Tor=>Webサイト のデータセットでは、すべてのクライアント・サーバペアについて我々は、そのクライアントサーバ間の地理的な最短経路にもっとも近い5%のリレーノードを保有している仮想の敵をエミュレートした。そして、我々は敵がその5%のノードを25倍ずつ複製し、より支配を拡大するという仮想のモデルを構築した。我々はこの仮想モデルのTorネットワークにおいて、集合化した場合としなかった場合においてWSPアルゴリズムを動かした。どちらの場合も、最低でもひとつの妥協された横断経路を、WSPによって選択される可能性を、我々は算出した。この値は、(敵が5%すべてのノードを保有していた場合)敵の保有ノードを、WSPが選択経路の一部に混入させてしまう確率の上限をあらわす。

図6の(a)ではクライアントサーバ間で異なる格子サイズでリレーを集合として分割した場合のその上限を比較している。我々はすべての格子の辺の長さを0.25～4まで(経度と緯度の違い観点から計測した)変化させ、どちらの場合でも、敵の支配下のリレーを通る経路の割合を算出した。その結果、辺の長さを2とした場合の格子では、それより短い長さの時と比べて、特別大きく敵の影響を抑えることができた。また、それ以上の格子幅の延長はあまり意味がないということも分かった。
次に、我々は集合化した後のWSPの、敵の攻撃に対する耐性について評価する。図6の(b)はクライアントサーバ間の経路を、次の3つの種類の妥協経路を含むケースに分けて分布させたものである。

1) WSPを、PlanetLabNode<=>Tor=>Webサイト のデータセットについて、集合化も25倍成長モデルも適用させずに普通に走らせた場合
2) WSPを、同データセットについて、2x2の格子で集合化させて25倍成長モデルで走らせた場合
3) WSPを、同データセットについて、集合化なしで25倍成長モデルで走らせた場合

図における1)と3)を比べると、集合化なしの場合は、敵は支配リレーを単純に25倍の速度で複製し、35%～65％の割合でTorのリレー経路に自分の支配リレーノードを混入させることができている。
それとは対照的に、リレーが2x2の格子幅で集合化されていた場合は、敵はリレーを複製させたところで何の利得も得られていないことが判明した。それでも、このようなリレーの集合化はWSPの経路選択アルゴリズムにとっては、遅延という部分において悪い影響を与える。なぜなら、クライアントと宛先サーバ間の間の地理的に直線的な位置に近い場所にいくつかのリレーノードが存在する場合、通常のWSPではそのいくつかのノードをピンポイントに好んで選択することができるからである。一方で、集合化させたリレーにおいては、WSPはその集合の中からノードを選択せねばならない。ゆえに、図6の(c)で示されているとおり、通常のWSPに比べて、集合化させたWSPでは選択された経路の地理的な距離がおおよそ15％、中央値平均で増加してしまっている。このようなリレーの集合化による経路長の膨張は、敵が高確率で支配下のリレーのトラフィックを傍受するためには複数の地点でリレーノードを構築しなければならなくなるという負担を大きくさせるための、我々が我慢しなければならない妥協である。最後に、我々は今回のWSPで得られた遅延削減効果について評価する。我々はTorクライアントソフトをWSP経路選択アルゴリズムで実装し、それを使用してTorネットワークを介してTOP200のウェブサイトに対して50のPlanetLabノードからアクセスした。それぞれのクライアントサーバの組み合わせに対して我々はWSPを5回実行し、5個のHTTP HEADリクエストにおける遅延の中央値平均を算出した。我々はそれと同じことを通常のTorクライアントソフトで再度行い、それにより選択された5つの経路のそれぞれについて、5個のHTTP HEADリクエストにおける中央値平均を算出した図6の(d)は通常のTorクライアントソフトを使った場合とWSPを使った場合について、クライアントサーバ間で係争した遅延の分布を表している。この図では、WSPは通常の時と比べて25%分、中央値平均で遅延を削減できていることが示されている。

C. 複数存在しうる宛先サーバに関しての説明

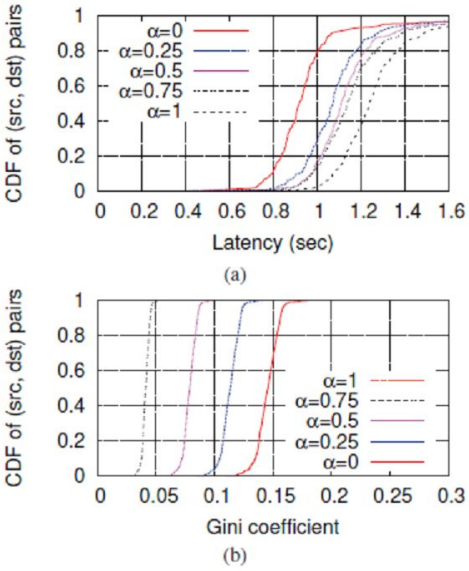

Fig. 8. Increasing the value of α when using WSP results in (a) higher latencies and (b) greater entropy of path selection.

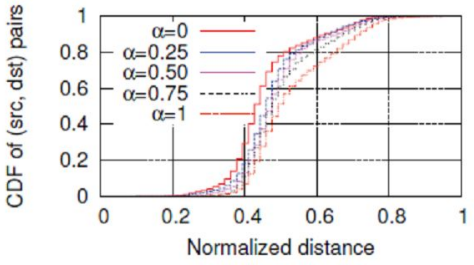

Fig. 9. End-to-end distances on paths chosen with WSP when using α to tailor the set of relays from which we select entry guards.

これまで、我々はWSPを宛先サーバはたった一つの地点に存在していると仮定して解説した。しかし実際には、(Webサーバのような)対話的な通信を行う宛先サーバは大抵地理的に複数の地点に分散して多重化されている。そのような場合はユーザはホストネーム(DNSで名前解決された宛先サーバ)を指定する。Webサービスプロバイダーはそのエンドホストに近い場所にあるサーバのIPアドレスをそのDNSルックアップで返す。これはユーザが宛先サーバと通信するためにTorを使用した場合は、Torリレーにおける「exitノードの時点での」DNS解決後の宛先サーバと通信することになることを意味する。したがって、WSPがある候補経路のエンドツーエンドの距離推定した場合、exitノードでリダイレクトされる特定のIPアドレスの地点について考慮しなければならない。しかしながら、経路選択をする時点において、すべてのexitノードの候補について宛先サーバのDNSルックアップ情報を取得することは非現実的である。それをするにはクライアントソフトは全てのexitノードについてTorリレーを構築せねばならずない(Torリレーを介さずに普通にDNSルックアップをすることはユーザの匿名性を意味のないものにさせる)。毎回、選択され得るすべてのexitノードについて、一個ずつTorリレーを構築することは、Torネットワークそのものに大きな負荷をかけてしまう上に、数十秒の時間を食うことになるので、これは遅延が少ない経路を選択する利益を無駄にしてしまうことになる。

その代わりに我々は、互いに15個の地理的に分散されたPlanetLabノード達にDNSルックアップサービスを構築した。クライアントがWSPを実行する必要がある場合、WSPは宛先サーバのホストネームからDNSルックアップサービスを提供する各Planetノードに対してDNSを検索するリクエストを送信する。クライアントソフトはこれらのリクエストを前もって構築しておいたTorリレー(たとえば、一般のTorのように起動時に構築されるTorリレー)を使って送信する。クライアントソフトはPlanetノードに対するこのDNS名前解決リクエストを送信するのに、HTTPSを使う。これはなぜなら、Planetノードと通信するTorリレーのexitノードで盗聴されることよって、ユーザがどの宛先サーバと通信しているのかをDNSリクエストを見られることによって推定されることを防ぐためである。クライアントが宛先サーバにおける(複数の)IPアドレスのセットを取得したら、我々はそのIPアドレスのセットの中から最も地r的に宛先サーバと近いexitノードの候補を仮定する。こうして、我々が宛先サーバへの経路を抽出するために続けてWSPを実行するときは、我々が信じる、exitノードがリダイレクトされる宛先サーバのIPアドレスをもとに、exitセグメント(exitノードと宛先サーバまでの距離)を使用し、エンドツーエンドの距離を算出することができる。このWSPアルゴリズムの改造の効果を評価するために、我々はQuantcsastからトップ100のWebサイトを考え、それらのWebサイトはPlanetLabノードからDNS名前解決を行った場合複数の地点に配置されたIPアドレスを返すことに焦点を当てた。そして我々はTorネットワークにおける、これらのWebサイトについて50個のPlanetLabノードをクライアントとして遅延を測定した。我々は2つのケースについて遅延を測定した。一つ目のケースは、上記で示したような15の地理的に分散したPlanetLabノードを、宛先サーバのIPアドレスを解決するために使用したWSPを実行した場合である。二つ目のケースは、宛先サーバはたった一つのIPアドレス、をランダムに選ばれたexitノードにおけるDNS名前解決によって返すものと仮定しWSPを実行した場合である。図7はこれらの2つのケースについて、測定した遅延の具合を比較している。これによると、分散された宛先サーバによって、ポテンシャル分布は中央値で15%の遅延削減経路となる可能性がある事実が考えられる。

D. 遅延と匿名性のトレードオフ

リレーを集合化させることは、多くの経路選択において、妥協する可能性を減らすのにも関わらず、WSPのもつより短い経路への趣向性は経路選択のエントロピーを自然に下げてしまう。どのユーザも遅延を少なくするためにエントロピー（匿名性）を犠牲にしたくはないだろう。したがって、我々はWSPにおける経路選択を一つの α というパラメータによってチューニングできるようにした。ユーザは α を0から1の間に設定することができる。0という値はもっとも遅延が少ないという意味であり、1という値はもっとも匿名性が高いという意味である。我々はこのパラメタ「 α 」をWSPに以下のように組み込んだ。前述したとおり、全ての経路候補についてエンドツーエンドの距離を算出した後、WSPは全ての、経路のエンドツーエンドの最大長の距離を各経路同士の距離から算出し重みとして計算する。WSPがある経路を選択する確率はその経路の重みに比例することになる。我々はこの経路の重み w を代わりに $w^{(1-\alpha)}$ として計算する。つまり、$\alpha = 0$ のときは、WSPは前述の通り、もっとも最短の経路を好んで選択するような、通常の動作をするようになる。これは言い換えれば、$\alpha = 1$ のときは $w^{(1-\alpha)}$ は1となるので、すべての経路はたった1の重みをもち、あらゆる経路がランダムに選択されるようになることを意味する。0から1の間の値を α にとった場合は、適当な低遅延か高匿名のどちらかに偏りをして経路選択をするようになる、図8は α を変化させたときの遅延と匿名性への効果を示している。

(a)では $\alpha = 0, 0.25, 0.5, 0.75, 1.0$ の各場合において、4-Bで示したような手順（5つのHTTP HEADリクエストをトップ200のWebサイトに対して50のPlanetLabノードをクライアントとして送信したときの遅延の中央値）でしたときのグラフである。匿名性のに対応する分散を補足するために、我々は"*Gini coefficient metric*"[31](ジニ係数。以前にもTorの経路選択の匿名性を計測するために使用されている。例:[5])を使用する。ジニ係数とは値同士の傾斜（格差）の指標である。ジニ係数において0という値は「完全な平等（すべての値は等しい）」を示し、一方、1という値は「完全な不平等」を示す。我々はこのジニ係数という計量を、PlanetLabNode<=>Tor=>Webサイト のデータセットにおける各（クライアント・宛先サーバ）ペアについて、WSPによってその間にある各経路の候補が選択されうる確率の格差を使用して算出する。図8の(b)は、α を高くした場合ジニ係数は対応して低くなり、これらの経路選択における確率の格差が少なくなったことにちゃんと対応していることが見ることができる。最後に我々はこの値 α をEntryノードを選択するためにも使う。統計的な特定攻撃を避けるために、通常のTorクライアントソフトは、クライアントソフトがスタートアップ[16]するときにランダムに選択された3つのノードのリストを継続的に使用するようにEntryノードの選択を制限している。そして、クライアントがセットアップしたすべてのTor経路はこの3つの"*Entry Guards*"の中の人をTorリレーのEntryノードとして以後選択するようになっている。当然予想されるように、このEntryノードに関する制約は匿名性の向上につながるが、WSPの低遅延な経路選択を妨げる結果となる。たとえばもし、その3つのEntryGuardsが、クライアントからも宛先サーバからも地理的に離れた場所に選ばれた場合はそれらの間の経路はやむを得ず回りくどい経路となってしまう。したがって、我々が求める、遅延をとるか匿名性をとるかという選択の可用性を経路選択に与えるという目標を達成するために、我々はこれらのEntryGuardsの選択を以下のように変更する。まず、上の方で示した、Torリレーの集合化を行った後、それらの内Entryノードの候補を持つすべての集合について、クライアントとの地理的距離についてソートする。そして、ソート後の配列のうち、最も近い上位 $(g + \alpha * (100 - g))\%$ の集合から3つの集合をランダムに選択する。その後、その3つの集合から一つずつノードをランダムに選択し、それらを最初の3つのEntryGuardsとして抜き出す。この時、g は変動可能な値であり、今回の実装では $g = 20$ という値をとった（上位20%）。したがって、$\alpha = 0$ のとき(最も遅延を少なくする設定にした場合)は完全にランダムに、そのソート後上位20%からEntryGuardsをクライアントからリレーへのノードとして選択することになる。これは、$\alpha = 0$ の時、すべてのEntryGuardsの候補の20%という極めて大きな部分集合から選択されたEntryGuardsにより、高い匿名性を得られつつ、回りくどいルーティングの可能性を最小にすることを意味する。言い換えれば、$\alpha = 1$ を、ユーザが、最大レベルの匿名性を得るために、選択した場合、EntryGuardsの選択は、(匿名性において)現時点における最善の実践である「すべてのEntryリレーノードの候補からランダムに抽出する」という手法に回帰する。図9はこの集合化したEntryノード選択アルゴリズムの、経路のエンドツーエンドの距離を算出済みの、PlanetLabNode<=>Tor=>Webサイト のデータセットについて行ったときの、効果を示している。α を増大させるにつれて、EntryGuardsのランダム性が比例して増大し、結果として選択する経路の地理的距離も長くなっているのが見て取れる。

---

[31] Gini, Corrado. "Measurement of inequality of incomes." *The Economic Journal* 31.121 (1921): 124-126.

5. 自律システム検知

次に、2つ目の、通常のTorクライアントソフトの欠点を対処する(Torリレーにおいて、自律システム(AS)がクライアントとEntryノードの間、Exitノードと宛先サーバの間の両方の間でトラフィックの相関を求めることができるような経路を意図的に避ける)。本論文ではTorネットワークに対して何の働きかけ(改造)も必要としないことを前提にしているため、全てのTorリレーノードとクライアント、全てのTorリレーノードと宛先サーバの間のルーティングを知ることによりそのような経路を避けることはできない。したがって、我々は以下に、クライアントはどのようにして、先ほど述べたような検閲自律システムが存在する可能性がある経路を判定し避けるために、ローカルでインターネットのルーティングを推定するのかというのを議論する。

Algorithm 1 Pseudocode of AS set estimation algorithm.

```
Inputs: AS graph G, AS three-tuples set T, source S, destination
  D, AS path length L
Shortest_Path(G, T, D)
Queue Q
List Node PossibleSet
List Node AS_set
S.hops = 0
Add S to Q
while Q is not empty do
   cur ← Q.pop
   cur.added ← 0
   Add cur to PossibleSet if cur ∉ PossibleSet
   for n ∈ cur.neighbors do
      Skip n if (cur.parent, cur, n) ∉ T
      Skip n if ∄ m ∈ n.neighbors such that m.pathLength +
      cur.hops + 2 = L
      if n has ancestor p with p.pathLength <
      p.parent.pathLength then
         Skip n if n.pathLength > cur.pathLength
      end if
      n.hops = cur.hops + 1
      Add n to Q
      cur.added += 1
   end for
   if cur.added = 0 then
      Decrement n.added for every ancestor n of cur
   end if
end while
for n ∈ PossibleSet do
   Add n to AS_set if n.added > 0
end for
return AS_set
```

A. 自律システム集合の推定

任意のIPアドレス間の自律システムレベルの正確なルーティング推論は、既存の技術[32],[11],[33],[12],[13]、難しい。したがって、あるEntryノードとExitノードの組のリレーが検閲自律システムの可能性を示すかどうか評価するとき、「Tor経路におけるEntryノードとExitノードの両側における自律システムレベルの経路推測」という手法は排除する。代わりに、我々は「インターネットを通る時の自律システム達の集合の候補はそのセグメントのインターネットのルーティングそのものである」という推測手法をとる。これにより、クライアントとEntryノードの間・Exitノードと宛先サーバの間に中継するルータか何かが無いかをチェックすることで、その経路の検閲自律システムの潜在性を判断することができる。そのような自律システム集合の推論をTorクライアントソフトで可能にするために、クライアントソフトに3つの入力をさせなくてはならない。一つ目に、「自律システムの内部的相互リンク」のようなインターネットの自律システムレベルの接続形態の情報を使う。二つ目に、インターネット上のすべてのTorリレーとすべてのエンドホスト間の自律システム経路長の推論を要する。この入力情報は、BGPによって選択された自律システムベースの経路は概ね自律システムの接続形態における最短の経路長よりも長い[13]ので必要となる。後に示すが、自律システムの経路長は自律システムの経路そのものと比べて、より少ない容量で、より強固に安定して保持することができる。三つ目に、以下に示すような3つの自律システムの組を、自律システムによって採択されたルーティングポリシーを表すために保持させる。このような自律システムレベルの接続形態 と 送信元 S と宛先 D の間の自律システム経路長 L を与えられたとき、その接続形態の中で S と D の間の自律システムが中継する L というポリシー準拠なルーティングにのっとった、ある自律システムからなる S から D へのルーティングトラフィックによって、自律システム達をまとめる。ここで、すべての自律システムレベルの接続形態における経路は各自律システムのルーティングポリシーに則っていないという理由から、ポリシー準拠について強調する。それゆえ、ポリシー準拠な経路上の自律システムだけを想定していることを保証するために、iPlane Nano[13]から3つの自律システムの組を使ったテクニックを拝借する。自律システム経路の測定した値のコレクション(BGPフィード[27],[28]から得て、自律システム経路へのtracerouteの結果[34], [12])から、あらゆる自律システム経路にみられる3つの連続した自律システムの順番を

[32] Lee, DK et al. "Scalable and systematic Internet-wide path and delay estimation from existing measurements." *Computer Networks* 55.3 (2011): 838-855.
[33] Qiu, Jian, and Lixin Gao. "Cam04-4: As path inference by exploiting known as paths." *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE* 27 Nov. 2006: 1-5.
[34] "Archipelago Measurement Infrastructure - Caida." 2006. 5 Jun. 2013 <http://www.caida.org/projects/ark/>

判別し、それらを3つの組の集合として追加する。たとえば、AS1-> AS2 -> AS3 ->AS4 -> AS5という自律システム経路が与えられた場合、

(AS1, AS2, AS3), (AS2, AS3, AS4), (AS3, AS4, AS5)という3つの組の3つのタプルの集合にまとめることができる。このようなある3つの組(A,B,C)の自律システムのタプルは、B というノードを通過するAから来たトラフィックはCへとルーティングしようとする、というルーティングポリシーを表している（言い換えると、BはCから来たルーティング情報をAへと送信する）。我々はそのような3つの組の自律システムの集合をBGPフィードを集計することで生成し、このデータを約1MBで賄えることができた。ここで、インターネットのルーティングが非対称的になることが少ないが、(要するに、送信元Sから宛先サーバDへのルーティングはDからSへのルーティングとは異なること)、我々はここではルーティングを非対称であるとみなし、自律システムベースの経路の中で見つかったすべての3組の(A,B,C)のタプルの集合の中に(C,B,A)というようなタプルを加えていることに注意してほしい。送信元Sと宛先サーバDの持つ2つのIPアドレスの間の自律システム経路長Lの推測のあと、我々は以下の2つの手順のアルゴリズムによってSとDの間のルーティングの間に発生しうる自律システムの集合を推測する。まず一つ目の手順として、我々は宛先サーバDを含む自律システムと、そのほかの全ての自律システムとの間の最短経路長を、ダイクストラの最短経路アルゴリズムによって算出する。我々はここで、算出される最短経路長が含むあらゆる連続した自律システムのタプルが、我々が導いた自律システムの3つのタプルの集合の中に完全に内包されていることを保証するために、標準のダイクストラのアルゴリズムに変更を加える。次に、そのネットワークトポロジの全ての自律システムを判断し、あらゆる一つ隣の自律システムから宛先サーバDへの経路長の集合を算出することを可能にする。二つ目の手順として、我々は送信元Sから改良済み幅優先探索アルゴリズム(BFS)を用いて自律システム集合の出力を判断する。幅優先探索アルゴリズムを実行するにあたって、ある自律システムAの隣に、送信元Sからk回のホップ数にあたる場所に位置するとき、宛先サーバDまでの経路長が
(L − k − 1)となるような隣の自律システムを持つ、ある自律システムBだけを通過させる。加えて、インターネットのvalley-free[35]性を、一度でも、ノードAからAよりも宛先サーバDへの短い最短経路をもつ隣のノードBへとルーティングした以降は、いかなるノードにおいても宛先サーバDへの最短経路が長いほうのいかなる隣のノードを通過しないようにアルゴリズムを修正することによって幅優先探索アルゴリズムに強制させる。それに加えて、もう一度、入力である3組の自律システム集合を、(C,B,A)という自律システム集合がその入力に含まれる場合にのみ、Aの隣のノードであり、幅優先探索アルゴリズムにおける親ノードがCである、Bというノードを通過するように強制させる。図. アルゴリズムⅠ（自律システムグラフG, 3組の自律システムタプルの集合T, 送信元S, 宛先D, そしてそれらの間に推定された自律システム経路長, を入力としている）はこのアルゴリズムの疑似言語を要約している。

[35] Gao, Lixin. "On inferring autonomous system relationships in the Internet." *Global Telecommunications Conference, 2000. GLOBECOM'00. IEEE* 2000: 387-396.

B. 検閲自律システムの避

Torのクライアントが宛先サーバへの経路を選択するとき、上記に述べた手法を用いて、クライアントと3つのEntryGuardsの間のすべての経路とExitノードと宛先サーバの間のすべての経路について存在する自律システム集合を決定しなければならない。後者の経路の集合については自律システムの集合を独立して計算することはない。その代わり、最初の手順で一度でもAS集合推定アルゴリズムを走らせたなら、以降は2番目の手順である幅優先探索アルゴリズムを各Exitノードについて独立して行う。これにより、(クライアント,Entryノード),(Exitノード,宛先サーバ)の経路が空でないASを縦断するようなEntryノードとExitノードの組み合わせを避けることにより、検閲自律システムを潜在的に含む可能性のあるすべての経路について考えることから離れることができる。このアルゴリズムは検閲ASを含む複数の経路を(1000のExitノードを選択した場合であっても)約3秒以内にふるい落とすのとができる。

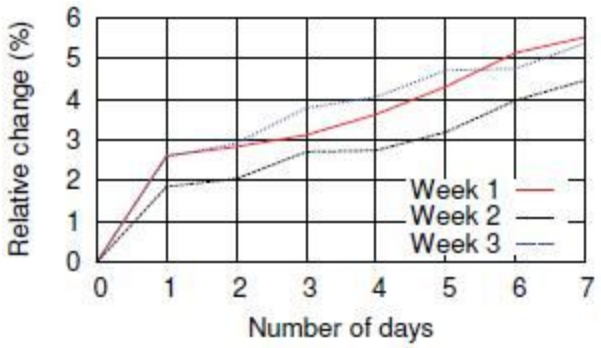

Fig. 10. Relative changes in AS path length data across days.

効率的な計算の観点のほかにも、この手法はクライアントがローカル環境で自律システム集合を推論する際にダウンロードするデータを最小化する。初めに自律システム間接続経路の集合と3つの自律システムの組み合わせの集合のデータはそれぞれざっと1MB程度のサイズでありこれらのデータセットが変動することは稀である。次にインターネット上に存在するすべてのTorリレーノードとすべてのエンドホストはせいぜい600のグループのBGP集合にグループ化することができ、グループ各々も50KB程度である[25][12]。したがって、LASTorはユーザに対してすべてのTorノード-エンドホストペア間の自律システム経路情報を合計30MBダウンロードさせる。

我々は自律システム経路長とそのデータの安定性を取得しこれらの推測したデータのサイズを、毎日iPlane[36]がPlanetLabノードからインターネットの末端を示すすべてのプレフィックスのIPアドレスについて行っているtracerouteのデータを使用して、評価する。我々はこのデータを2011年の7月から3週間に渡って解析した。各日、我々はすべてのtracerouteの結果をそれらの対応する自律システムレベルの経路に変換し、それらについて、自律システム経路長すなわちその経路の中に見られた自律システムの数を算出した。一つ目に、我々はそのtracerouteで中継した各経路のうち、8つ以上の自律システムを中継する経路は全体の0.05%以下であることを見つけた。つまり、すべての自律システム経路長は3ビットにおさまるということであり、これにより、ユーザに対して約11MBの自律システム経路長データを最初にダウンロードさせればよいということになる。考察している毎週の区切りに、その週の最初の曜日で観測された自律システム経路長とその週のほかの曜日に観測された自律システム経路長を比較した。我々はその経路のうちday0(=週の初めの日)とday$i$(週のi番目の日)について、異なる自律システム経路長を持つものの割合を計算することで比較を行った。図.10で示されているとおり、自律システム経路長は1週間経過してもたった5%しか変化しないことがわかる。これより、要するに、我々のモデルでは、自律システム間の経路情報、3つの自律システムの組集合、そして自律システム経路長にわたるデータをクライアントに初めに13MBダウンロードさせることになる。これは事前算出された「すべてのTorノードとクライアントおよび宛先サーバ間に存在する自律システム経路たち」のデータをおよそ1/40にまで減らしたことになる。そしてこれはクライアントに、1週間に一度最大でも1.5MBの差分データをダウンロードさせるだけで、データを最新に保たせることができることを意味する。

[36] "iPlane: Datasets." 2010. 29 Jun. 2013 <http://iplane.cs.washington.edu/data/data.html>

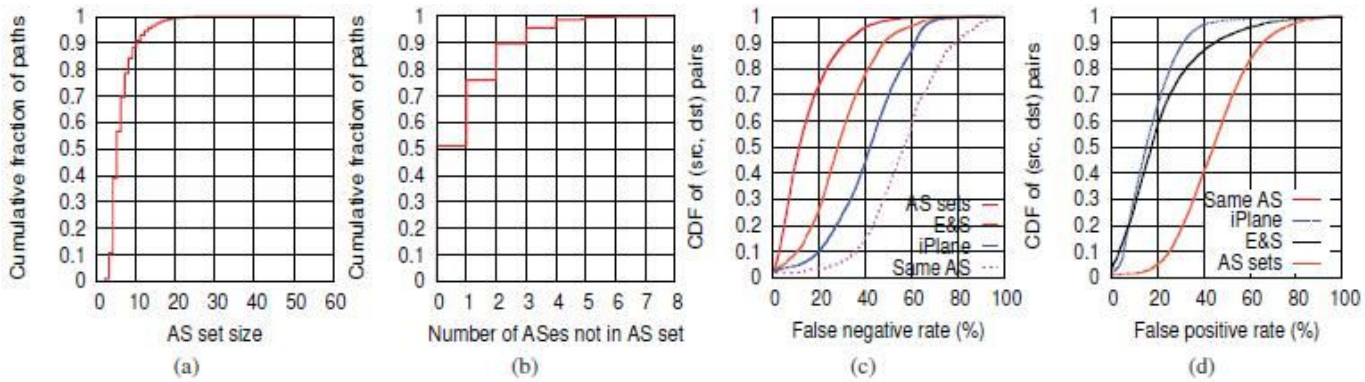

Fig. 11. (a) Distribution of predicted AS set sizes, (b) accuracy of predicted AS sets encompassing actual AS paths, and distribution of (c) false negative and (d) false positive rates in predicting the existence of snooping ASes.

C. 検閲自律システム検知機能の評価

次に、我々は2つの観点からLASTorの自律システム集合の推測機能を評価する。はじめに、我々は推測された自律システム集合が現実に存在する自律システム経路を正確に網羅できているかを試験した。そのために、PlanetLabNode<=>Tor<=>Webサイト のデータセットについて、PlanetLabノードからTorノードまでの経路の自律システム集合を推測した。図.11(a)と図.11(b)は推測された自律システム集合は概して軽量であることを示している。90パーセンタイルは10個以下の自律システムであり、たった一個の実在する自律システムが含まれていない(推測失敗)、推測された自律システム集合はその経路のうち75%程度であった。次に、PlanetLabNode<=>BGP<=>ランダムWebサイト のデータセットを、自律システム集合が潜在する検閲自律システムの、予測を可能にする自律システム集合の妥当性について検証するために使った
(我々はPlanetLabNode<=>Tor<=>Webサイトのデータセットにおいてexitノードと宛先サーバの間の自律システム経路をもっていない。また、PlanetLabNode<=>PlanetLabNode<=>Webサイトのデータセットは今回の解析にとってはいくらかの偏りがある)。我々は、PlanetLabNode<=>BGP<=>ランダムWebサイト のデータセットにおいて、すべてのクライアントおよび宛先サーバのペアについて、すべてのEntryノードとExitノードのリレーの組合わせを、共通する自律システムをEntryセグメントとExitセグメントに持っているものと、持っていないものに分けた。その後、我々の手法で推定された自立システム集合の経路に乗っかっていない自律システムの数で検閲自律システムの推定値を割り、偽陰性率を算出した。図.11(c)は偽陰性率が中央値で11%であることを示している。これは「iPlaneの推定した自律システム経路を使う方法」・「注釈[10]で示されている手法("E&S"軸)」またはエンドホストの自律システムとTorリレーの自律システムにのみ注目した場合("the same AS"軸)の中央値偽陰性率である28～57%という数字と比較すべき数値である。その反面、図.11(d)では、自律システム集合がほかの手法に比べてそれよりも大きい偽陽性率(検閲自律システムを持たない経路を、検閲自律システムを一つ以上持っていると我々のアプローチで検出した割合)を示している。しかしながら、図.12でみられるように、潜在的な検閲自律システムをもつ経路の割合はほとんどの送信元<=>宛先サーバ間で低い。つまり、中央平均で45%もの候補経路を削減したとして、それでもかなりの量の経路の選択岐をWSPアルゴリズムに与えることができている。

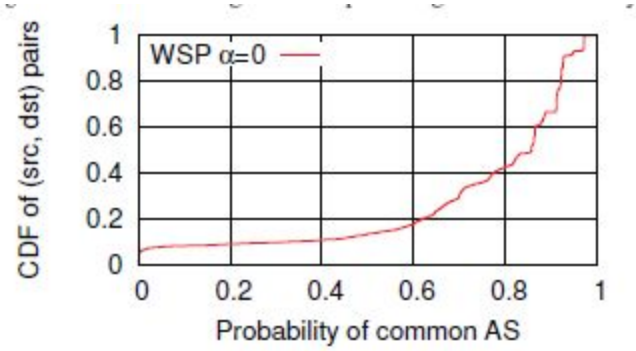

Fig. 12. The probability of existence of snooping ASes across (src, dst) pairs in the *PL-BGP-Rand* dataset.

D. 自律システム検知機能が与える経路の遅延時間の影響

最後に我々は自律システム検知の取り組みによるWSPによって得られた経路の遅延の影響について評価する。WSPは今のところ、我々のトラフィック相関攻撃によってクライアント<=>サーバのペアが推測できるような自律システムの暗黙の横断を避けるという自律システム集合推測アルゴリズムによって検出されたものを無視させられるという理由から、すべての考えうる候補経路の一部分から経路を探索しなければならない。候補経路のうち検閲自律システムを含む部分は経験的に概して小さいものであるにもかかわらず、我々の検出アルゴリズムの高い偽陰性率はその部分であるとみなされる経路を著しく減らす。したがって、我々は再度WSP(α＝0にして状態で)を、50のPlanetLabノードからTop200のWebサイトに対してTorネットワークを使用して遅延の計測を行った。

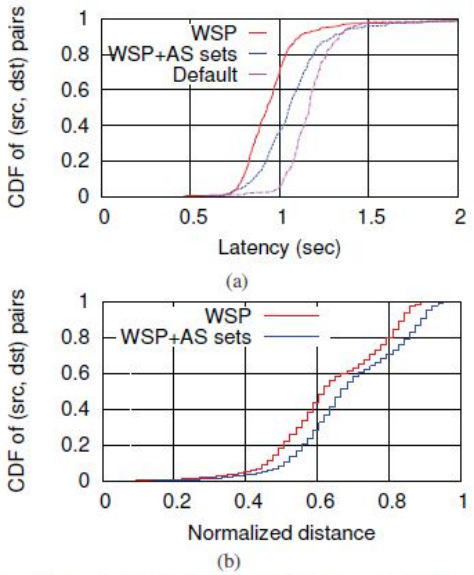

Fig. 13. Comparison of (a) latencies and (b) normalized geographical distance along paths chosen with WSP ($\alpha = 0$) with and without AS-awareness.

図.13(a)はWSPを、自律システムを考慮しない設定にして実行した場合と、WSPを使用せずに普通のTorクライアントで実行した場合に起こる、遅延の度合いを比較している。この図から、検閲自律システムを避けるために経路を間引くと、遅延がわずかに増加することが見て取れる。図.13(b)は、この遅延の増加はWSPが選択する経路の長さが自律システム集合で計算した時に増加したからであることを示している。将来的には、この自律システムを考慮したWSPを使用するときの遅延をさらに減らし偽陰性率を下げることを追及しようと考えている。

6. 実装

我々は「経路上の遅延を削減する」「経路選択を調整可能にする」「経路選択に自律システムを考慮した仕組みを組み込む」というこれまでに出たすべてのアルゴリズムをLASTor(Torクライアントソフトウェア)に実装する。この章ではLASTorの経路選択アルゴリズムを要約し、我々の実装の全体像を提供する。

A. クライアントの動作

普通のTorクライアントソフトウェアは起動の初めと、その後ユーザがある特定の宛先サーバにTorを介して接続するときに数個のTor回路をセットアップし、Torクライアントソフトウェアはユーザのトラフィックを、セットアップした経路の内一つを使用してルーティング[24]する。LASTorはこの仕組みに関しては普通のTorクライアントのものを真似る。加えて、一度LASTorがユーザが通信を行いたい宛先サーバを記憶すると、LASTorはすぐさまその経路を自律システムを考慮したWSPアルゴリズムによって選択し、新たにその経路についてTor回路をセットアップし、宛先サーバに対するユーザのトラフィックをこの新しい回路に変移させる。こうして、LASTorは普通のTorクライアントソフトウェアが、ユーザが宛先サーバの地理的に近くで通信した時に匹敵するくらいの、遅延で済むようになる。もしユーザが宛先サーバとの通信を長引かせた場合（例えば、ユーザが一つのWebサイトの複数のWEbページを閲覧している場合）、ユーザの通信のほとんどでLASTorは著しく遅延を減少させることができる（すなわち、一度でもLASTorがユーザのトラフィックをWSPによって選択した回路にスイッチングした時点で）。ある特定の経路を選択するために、LASTorは自律システムを考慮した（調整可能な）WSPアルゴリズムを次に示すステップを順番にして実行する。

・初期化するとき、LASTorクライアントソフトウェアはすべての利用可能なTorリレーを収集し、入力された設定の α の値を用いて、LASTorは3つのEntryGuardsをクライアントにもっとも近い（$20 + \alpha * 80$）％のリレーの集まりからランダムに選び出す。

・宛先サーバへの経路を選択しなければならなくなったとき、LASTorは宛先サーバのホスト名を、分散されたDNS作引きを提供するサーバノードの集合を使って、解決する。これらのDNS解決リクエストは初期化のときにクライアントと接続を確立したTor回路の内の一つを介して送信される。

・LASTorはすべてのクライアント＜＝＞EntryGuard および すべてのExitリレーノード＜＝＞宛先サーバの間の経路に関して自律システム集合を推測し、宛先サーバにマッチしたIPアドレスにもっとも近いExitリレーノードの候補をすべて算出しマッピングする。

・その後LASTorは互いに素なEntryセグメントとExitセグメントを持つという条件を満たす自律システム集合について、3つのクラスタを介してすべての経路についてエンドツーエンドの距離を算出する。そして、そのエンドツーエンドの距離と入力値 α の値によってそれらのクラスタレベルの経路が実際に選択されるかどうかの確率を算出、その確率にしたがって一つのクラスタレベルの経路が選ばれる。

・その選ばれたクラスタレベルの経路の中から、一つのノードレベルの経路をランダムに選択し、それを宛先サーバへのTor回路として確立する。

B. 普通のTorクライアントの改造

我々はLASTorを普通のTorクライアント上で実装する。普通のTorクライアントのコントロールポートにアクセスするために、Javaアプリケーションを実装した。この「コントロールポート」とはTorクライアントをその標準プロトコル[37]によってモニターもしくは操作するために使うことができるポートである。コマンドをコントロールポートに伝えることで、Javaアプリケーションはすべての利用可能なリレーの「説明」などの情報を取得することや、Torクライアントに Tor回路を確立させたり切断させたりするように操作することや、回路にストリームを付加することや、TorのDNSキャッシュを削除することができる。回路をセットアップするために、我々のプログラムは初めにTorのコントロールポートを使用して関連する情報を読み込み、それを調整可能な経路選択アルゴリズムに入力として提供する。その後もう一度コントロールポートを使用して、望みの回路を構築するためにコマンドを入力する。我々はLASTorに

1. 経路選択に使う α の値
2. DNS解決サービスを提供するノードのリストが書かれたファイル

の2つの入力するべき設定を実装する。

C. 入力データセット

自律システムを考慮した調整可能なWSP経路選択アルゴリズムを実行するために、Javaプログラムはいくつかのデータセットを必要とする。一つ目は、MaxMind[19]サービスから入手するIPアドレスの地理的な情報を示した地図、IPgeolocationデータベースである。二つ目は、一番初めにLASTorが実行されたときにダウンロードする

1. インターネットトポロジの自律システムレベルの表
2. ポリシー準拠の経路かどうか確認するために使う3つの自律システムを一緒にしたタプルのセット
3. Torリレーとエンドホストの間にあるすべての自律システム(BGPアトムの精度にまでグループ化したもの)経路長のスナップショット

この内初めの2つのデータセットはさまざまな情報源[27][28][12][34]から自律システム経路を集計することで一つにまとめる。自律システム経路長を推測するために、我々はクエリをiPlane[38]に発行する。我々はiPlaneがおおむね1000個のクエリを毎秒ごとに処理できることを見つけたので、自律システム経路長の算出に必要な情報である6000万個のIPアドレスのペア(600個のTorリレー付きBGPノードと5万個のすべてのエンドホストからなるBGPノード、双方向)を、毎日、iPlaneからダウンロードすることができる。先述の通り、3つのデータセットの全ては13MBytes以内のサイズに収まる。これらのデータセットはすべてのクライアント間で同じものであるため、クライアントがダウンロードしたこのデータセットそのものもクライアントの匿名性を阻害するものではなく、クライアントはBitTorrentのようなP2Pのファイル共有システムによってもダウンロードし、互いに情報交換することもできる。つまり、このデータセットをダウンロードするときにiPlaneのようなデータセットを持つ中央サーバの帯域を圧迫することを避けることも可能である。通信帯域幅が制限されたクライアントであっても、中央サーバからこのデータセットの該当する一部分のデータだけをダウンロードすることが可能である(たとえば、人気のWebサイトとの通信に必要な自律システム経路の情報だけをダウンロードする、など)。そして、毎週クライアントは約1.5MBytesの自律システム経路の更新差分をダウンロードして、まれに自律システム間の接続状況と自律システムのう3つの組のタプル集合をアップデートするだけでよい。これらのアップデートはクライアントの所有するデータのバージョンに依存するため、中央サーバからダウンロードする。LASTorが必要なすべてのデータセットにおいて、我々はクライアントに、普通のTorクライアントソフトウェアが使用している完全性保証に似た手法を利用することでダウンロードしたデータの完全性を検証できるようにすることができる。 — データセットの暗号ハッシュをTorのWebサイトにPOSTするだけ。

---

[37] "dir-spec.txt - Tor Project." 2011. 27 Jul. 2013 <https://gitweb.torproject.org/torspec.git/blob/HEAD:/dir-spec.txt>

[38] "iPlane: Measurements and Query Interface." 2006. 27 Jul. 2013 <http://iplane.cs.washington.edu/iplane_interface.pdf>

7. 考察

この章では、もしLASTorがTorに広く導入されたときにTorの拡張機能が必要とするさらなる負荷分散効果と遅延減少効果について考察する。

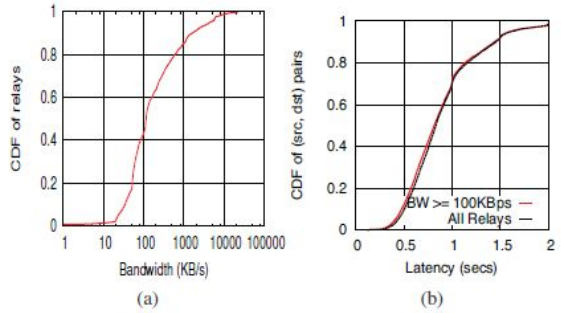

Fig. 14. (a) Distribution of bandwidth across Tor relays, and (b) comparison of end-to-end latencies with and without taking relay bandwidth into account; median latency across 5 paths are shown.

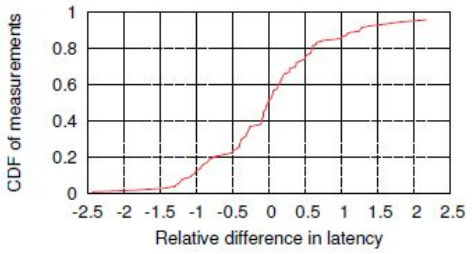

Fig. 15. Variation across time of latencies on paths spanning similar end-to-end geographical distances.

A. 動的ロードについての説明

WSPは著しくTorにおける通信の遅延を減少させるということを示したが、それでも通常の公衆インターネット経路を使用した通信と比較して、やはり著しいオーバヘッドが残っている。したがって、さらに遅延を減らすために、WSPアルゴリズムによる経路選択アルゴリズムを使用した伝搬遅延を減らす方法以外に、各リレーの経路選択に必要となるロードを考慮し、待ちパケット行列遅延を最小化することが必要である。ここに、それをするためにした我々の努力の副産物的な結果を記す。まず、図.14(a)に示したように、広い範囲に散らばったTorリレーの通信帯域幅を監視した。したがって、我々はこのうち高帯域幅のリンクをリレーの間に持つというリレーに経路選択を制限することで待ちパケット行列遅延を削減できる可能性があるかどうかを調査した。これらを実験するために、我々は50個のPlanetLabノードからトップ200のWebサイトにTorネットワークからアクセスしたときにおける、ある2つのケースについて経路遅延を計測した。我々は初めにリレーを100KBytes/s以上の帯域幅を持つものからランダムに選びだし、その後その選択を繰り返した。待ちパケット行列遅延を双方の設定で同じに保つため、100KBytes/s以上の帯域幅で全てのリレーに対してリレーを抽出し、我々は対応するEntry,Middle,Exitノードを同じ地理的な地点から、リレーの帯域幅の制限なしで選びだした。双方の経路選択手法のために、我々は5つの異なる経路のすべてのクライアント＜＝＞宛先サーバペアについての遅延を計測した。図.14(b) の ”All Relays”の曲線と ”BW ≧ 100 KBps”の曲線 は5つの選択された経路における遅延の中央値平均の分布は通信帯域幅を考慮した場合と考慮しなかった場合で同じであったことを示している。今回のケースでは、我々はそれぞれのリレーの決定に”推測された”通信帯域幅を使用した— 普通のTorクライアントソフトウェアが経路選択を実行するために使用する値 —が、Torディレクトリサーバが供給する各リレーの、別に推定された、通信帯域幅を使ったときの結果と似た結果になることを見つけ出した。次に、我々は与えられた経路における遅延のバリエーションについて検証した。我々は20のクライアント＜＝＞宛先サーバのペアをランダムに選択し、選択したそれぞれに対して、同じエンド・ツー・エンドの地理的な距離を持つ互いに素な2つの経路、3つのTorリレーを横断する経路について考えた。それぞれのクライアント＜＝＞宛先サーバペアについて、30分に一度それらのペアのために選択された経路で測定した遅延と、最初と2番目に選択された経路で測定した遅延との相対的な差異を計測（我々はランダムにすべてのクライアント＜＝＞宛先サーバペアのために選択された2つの経路を、すべての計測値を回るように軌道修正）した。図.15 はその日に計測された遅延毎の違いにバリエーションがあることを示している。すべてのクライアント＜＝＞宛先サーバペアのために選択された経路のペアが全く同じ地理的距離をもっているにも関わらず、より低遅延な通信を提供する経路は時間とともに著しく変化することが見て取れる。したがって、これらの結果はリレーを変更するだけ—リレーに、負荷状態を細かい精度の時間ごとに記録させ報告させるか、リレーに新しいパケットキューアルゴリズムを導入するか—で待ちパケット行列遅延が削減できる—これは我々のゴールであるTorクライアントソフトの向上により即座に遅延を減少させるという目的の範疇ではない—ことを示しているだろう。今日のTorリレーの実相に関していうならば、リレーの選択にそれらの帯域幅を考慮しバイアスをかけることはTorリレーのスループットを向上させるかもしれないが、しかしそれはインタラクティブな通信の遅延を減少させることはないだろう。

B. 負荷分散(ロードバランシング)

現在の一般的なTorクライアントは、経路のリンク帯域幅を確率的な重みにした方法で、ランダムに経路を選択する。その結果として、あるリレーを横断するすべてのTorのトラフィックは概してそのリレーアクセスのリンク帯域幅に相関する。こうしてリレー同士で負荷分散が行われている。

これと裏腹に、もしLASTorが広く使用されるようになれば、これらのTorリレーにおける負荷分散のシステムは著しく歪曲してしまうかもしれない。もしユーザがLASTorを、αを0に近い値に設定して使用した場合、LASTorはクライアントが通信しようとする宛先サーバとクライアントとのエンドツーエンドの距離が少ないリレーを横断するようにバイアスをかける。言い換えれば、もしすべてのユーザがLASTorをαを1に設定した状態で使用した場合、すべてのリレーに対して均等に負荷がかかるようにランダムに経路が選択される結果となる。これは(先に示した図.14(a)にあるように)リレー同士のリンク帯域幅を著しく歪曲してしまい、望まれないことである。これらの問題に対処するには今回の研究の範囲外であるさらなる調査を必要とするが、我々はTorリレーの負荷分散のバランスを脅かさずにLASTorの使用を普及させられるための、2つの提案を示す。

一つ目、我々はTorをBitTorrentのようなバルク転送に使用しているユーザに対しては、今現在使用している普通のTorクライアントソフトウェアを使用することを推奨する。バルク転送はTor[1]におけるトラフィックの用途の大多数を占めており、普通のTorクライアントソフトウェアは、負荷分散のために、Torリレーのリンク帯域幅に従ってTorリレーをうまい具合に分布させ、負荷分散することが見込まれる。ある特定のプロトコルに則った経路選択による、匿名性の欠落問題にはさらなる調査が必要である。

二つ目、LASTorの経路選択アルゴリズム自体、Torリレーのアクセスリンク帯域幅を考慮するように改良しなければならない。しかしながら、それを行うには、LASTorクライアントを使っているTorユーザの使用しているαの値の分布を調べ、その平均を発見しなければならない。この分布を発見することは各ユーザの匿名性を保護したうえでの調査によって可能となるかもしれない。LASTorの経路選択アルゴリズムはそうすることで、―シンプルにエンドツーエンドの地理的な距離を考慮した設定するだけでなく―Torリレーのアクセスリンク帯域幅や各ユーザのαの値の分布も考慮して微調整できるようになるかもしれない。

8. 先行研究

我々の研究は次の3つの軸によって進められている。

1. Torのパフォーマンスを向上させる
2. Torの匿名性を向上させる
3. 自律システム経路の推論

我々はこれらの3種の軸にそって、関連した研究について述べる。

Torのパフォーマンスの向上

Torのパフォーマンスを向上させるために、*Sherr* [6] [39] らは、リレー間のリンクを考慮したリレー選択のコンセプトをベースとした経路選択アルゴリズムを提案した。この手法ではクライアントは、経路のセグメントに渡って算出された経路のメトリック(例：遅延・通信帯域幅)に基づいて、各経路の「コスト」を集計する。その後、このコストを経路選択アルゴリズムの「重み」として使用し、その重みに基づいた確率的な決定により経路を選択する。シミュレーションのおかげで、彼らは彼らの手法がこれまでに延べられた「Torパフォーマンスの向上」という方針において、より良いパフォーマンスをたたき出したことを示した。これらのパフォーマンスの向上効果を得るためには、自身の使用する情報を拡散させるTorリレー—たとえば、ネットワーク共同協力システム—について述べた。しかしながら、パフォーマンスの向上のために分散システムであるTorリレーを改造し、そのような情報を拡散させることは、ささいなリスクであるとは言えない。そういうことで、我々はTorリレーの改造なしに遅延を減少させられることに焦点をあてた。加えて、*Sherr*らは彼らの手法のによる匿名性を評価するために、経路上に存在する横断した自律システムの数を計算し、「横断した自律システムの数が少ないほどより良い匿名性を得られる」と解釈した。しかしながら、我々は明示的にこれを、「そのリレーのEntryセグメントとExitセグメントに共通した自律システムが現れるような経路を避けることでより良い匿名性を得られる」という風に解釈した。*Panchenko*[7]らは、Torのパフォーマンスを向上させるために、2つのアルゴリズムを提案した。一つ目に、遅延を減らすために、彼らは全てのTorリレーペアの遅延を計測し、その後、その経路のエンドツーエンドの遅延に沿って確率的に経路を選択する。二つ目に、スループットを重視したアプリケーションを考慮するために、彼らは各リレーの利用可能な帯域幅を推論するために受動的な計測を行い、その後、予想されたエンドツーエンドのスループットに従って経路を抽出する。これもしかしながら、全てのTorリレーの改造がこれらの手法の実装には不可欠なものである。Tor上の接続のほとんどが対話的なトラフィックによるもの[1]であるので、我々は遅延を減少させることと、それをいかにしてクライアントサイドのソフトウェアの改造だけで実装してみせるかに焦点をあてた。参考文献[40]著者は、地理距離の多様性がToのパフォーマンスに与える影響について研究し、向上されたパフォーマンスと匿名性のトレードオフの関係を発見した。彼らは地理的により狭い多様性をもったリレーノード同士はリレー回路をセットアップする際に匿名性を低くしてしまう可能性があるということ、地理的により広い多様性をもったリレーノード同士は強い匿名性を確保するのに重要な要因であることを発見した。我々は、似たように、低遅延の経路を好んで選択したときに発生する匿名性の欠落を示したが、よい匿名性が遅延の減少の引換えとして失われないように、好みによってこの経路選択をチューニング可能にした。SnaderとBorisov[5]はTorのクライアントがいかにして、経路選択時のパフォーマンスと匿名性のトレードオフを行えるのかを示した。しかしながらSnaderとBorisovはTorにおけるスループットの向上—彼らの評価基準は1MBのサイズのファイルのダウンロードによるものである—に焦点をあてているのに対して、我々は遅延に焦点をあてている。我々は低遅延の経路を選択することで、いくつかのスループットを最適化する際には必要のないいくつかのテクニック — たとえば、EntryGuardsの慎重な選択や、地理的に分散された宛先サーバ — の必要性を保証する。DefenstaTor[8]はTorリレーのトラフィック制御を、トラフィックの混雑状況に関連した待ちパケット行列遅延を減らすように、改造することで、Torの遅延を減少した。我々はTorリレーの改造を必要とせずに伝搬遅延を減らす、補助的な手法を追求した。

---

[39] "iPlane: Measurements and Query Interface." 2006. 7 Aug. 2013 <http://iplane.cs.washington.edu/iplane_interface.pdf>

[40] Panchenko, Andriy, Lexi Pimenidis, and Johannes Renner. "Performance analysis of anonymous communication channels provided by Tor." *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on* 4 Mar. 2008: 221-228.

自律システムを考慮した経路選択

2004年、FeamsterとDingledine[9]はTor回路の両方のエンドサイドで盗聴する自律システムの問題について調査するためにTorネットワークについて研究した。一つ目に、彼らは、同じ自律システムの中に複数の異なるIPアドレスが付与されていることと、Torクライアントソフトウェアは同一の自律システムから2つのリレーを選択することを避けるべきであるということを示した。二つ目に、彼らは、自律システムがTor回路の両端のペア(クライアント・宛先サーバ)をその支配下に置き、検閲できる確率は10%～30%とさまざまであるということを発見した。この確率を減らすために、彼らは自律システム経路の決定をするためにBGPフィードを受動的にモニタリングする手法を提案した。しかしながら、彼らは、クライアントがどのようにしてBGPフィードからBGPのルーティングテーブルに関する最新の情報を取得するべきなのか、肝心な部分について明確にしなかった。しかし、彼らの研究結果を利用し、我々は計算時間と保存領域の複雑さにとらわれずに自律システムを考慮した経路選択の研究をすることができた。その後、2009年にEdmanとSyverson[10]は、FreamsterとDingledineの調査で数えられてから、Torリレーの数が非常に多くなってきたにも関わらず、トラフィックの両サイドで接続状況を監視できる自律システムに出くわす確率はそこまで減っていないことを示した。このような検閲自律システムの発生を防ぐために、彼らはすべてのTorリレーサーバ管理者に対してRIB(Routing Information Bases)に基づいた自律システムのスナップショットを取得するように提案した。そうするとクライアントは自律システムトポロジのスナップショットを、クライアントからEntryノード・Exitノードから宛先サーバまでの間の自律システム経路の内、互いに素な自律システム同士を持つ、経路を選択するのに使える。我々の評価によって示した通り、我々の用いた手法では、EdmanとSyversonの提案した手法より、検閲自律システムを見逃すという確率を格段に減らしている。

自律システム経路の推論

いくつかのシステムやアルゴリズムが、インターネットにおいて、任意のIPアドレスの間に存在する自律システム経路の推論をしてきた。これらの手法は大まかに2つの種類に分けることができる。一つは、入力として膨大な量のデータを使用するが自律システム経路をCPU資源的に効率よく推測することができる、[12],[32],[33]の手法の組み合わせである。このような手法は自律システム経路の推測値をクエリすることができるというホスティングサービスとして運用するのが理想であるが、その経路情報のクエリを送信するということそのものがクライアントの匿名性を脅かすものとして、Torにおいては選択すべきものではない。もう一つは、[11],[13]のような、入力としてのデータは少量—PoPレベルのインターネットの・自律システムレベルの トポロジ程度—であるが法外なCPU計算処理が必要なものである。検閲自律システムを避けるためにこのような手法を経路選択に使用することは遅延の少ない経路を選択するということに関して議論を引き起こす。これらの先行研究における自律システム経路の推論の欠点を踏まえて、我々は必要なランタイムと必要な入力データの”両方”を抑えた新たな手法を開発した。

その他の関連研究

Torネットワークに関するいくつかの研究[41][42] [1] は、Torユーザの主な(地理的な)発信源と、Torで利用されるトラフィックの主な種類(HTTP,BitTorrent,E-Mailなど)を位置づけた。これらの調査研究は、HTTP通信がTorの総トラフィックのごく一部であるのにもかかわらず、HTTP通信がTorのコネクションにおける大多数を占めていることを明らかにした。つつまり、ほとんどのTorユーザにとって、「遅延」はスループットよりも重要であるということである。我々の知る限り、我々は今日のTorにおいて、クライアントサイドのTorソフトウェアの改造だけで遅延の向上手法を示した、最初の研究である。Hotter[21]らは、クライアントによって使用されたTor回路における遅延を知ることで、クライアントの匿名性が失われるという調査をした。我々の研究の補足として、この研究は我々の”調整可能な自律システムを考慮したWSP経路選択アルゴリズム”の観点から再度検証する必要がある。我々は「クライアントがWSPを用いて経路を選択している」という情報は、経路の遅延状況が明らかになった場合、クライアントの情報をさらに漏洩させてしまうことになると推測する。

---

[41] Blond, SL. "One Bad Apple Spoils the Bunch: Exploiting P2P Applications to ..." 2011. <http://arxiv.org/abs/1103.1518>

[42] Loesing, Karsten, Steven J Murdoch, and Roger Dingledine. "A case study on measuring statistical data in the tor anonymity network." *Financial Cryptography and Data Security* (2010): 203-215.

9. 結論と今後の課題

Torは低遅延の匿名通信のための匿名ネットワークとして、今日、最も広く使われているが、遅延と潜在する自律システムによるトラフィック相関攻撃の脅威は、今日のTorの可用性にとって非常に問題である。これまでのTorのスループットを向上させる試みは、対話的通信をサポートしていない。また、Torリレーノードに改造を加えるという試みは、Torの開発者たちに負担を強いることになり、いまだに実装までには至っていない。本論文では、大幅な遅延の削減と今日のTorネットワークにおいてTorリレーに一切改造を要さずに検閲自律システムを防御できることを証明するために、新たなTorクライアント、"LASTor"を開発した。1万ものクライアント・宛先サーバペアにおける計測に基づき、LASTorは中央値平均で25%の遅延の削減が可能であることを示した。これらの遅延の削減効果を提供するために、我々はEntryGuardを注意深く選択するとともに、冗長化された宛先サーバについても考慮している。我々はさらに、ある経路における潜在的な検閲自律システムを検出する、記憶領域的にも計算時間的にも効率的な手法を開発した。その上我々はユーザに対して遅延と匿名性のトレードオフを簡単に選択できるように、LASTorの経路選択をチューニング可能なものにした。我々は、LASTorを自由に使用できるようにしようと考えている。また、[12][13]のような、リレーから得られる測定データを必要としない―つまり、Torリレーの改造をせずに遅延を削減できる―遅延推測手法の応用についても調査を続けていくつもりである。