ChromeのTCPクエリ問題

山口崇徳@IIJ
2023/06/23 DNS Summer Day 2023

何があった?

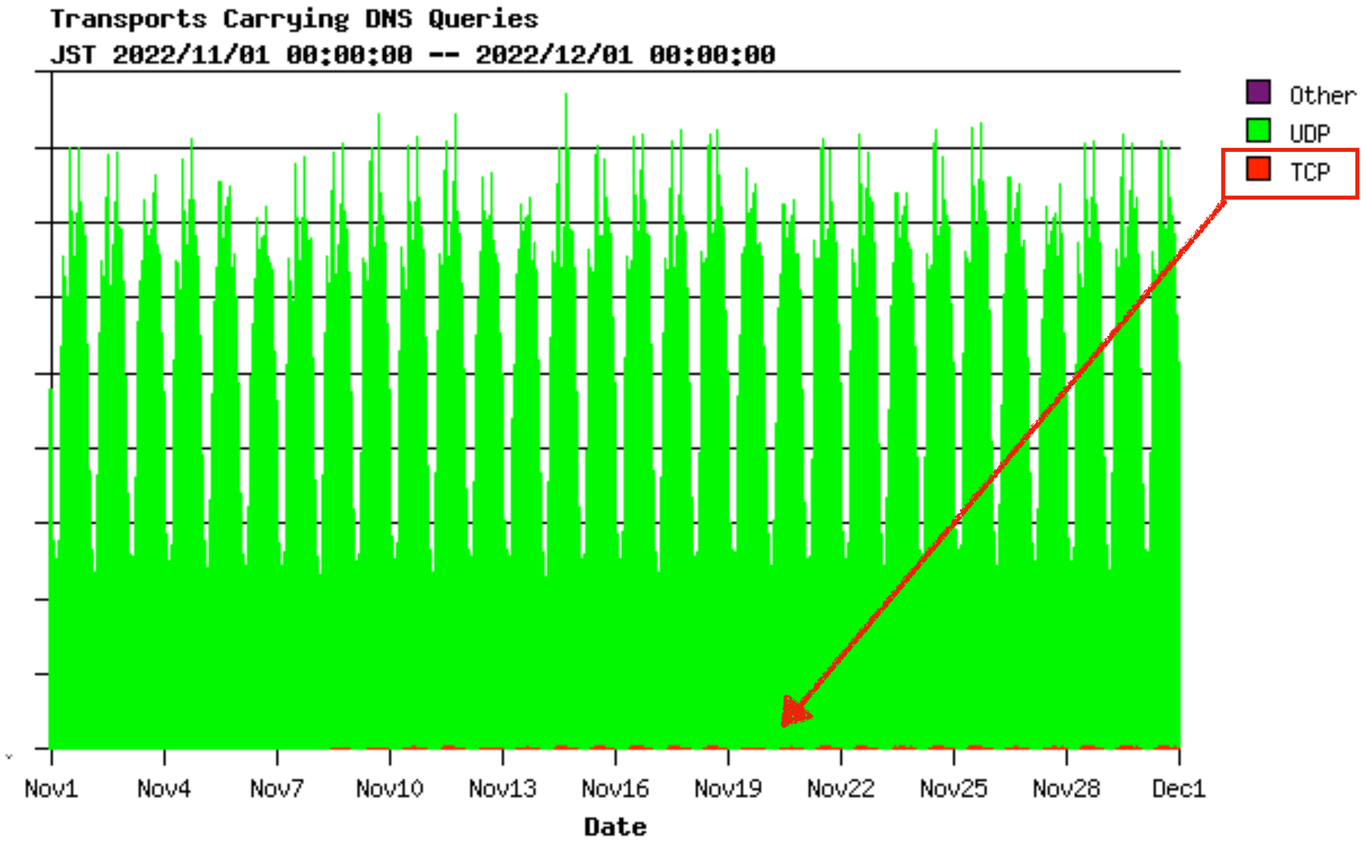
- 2022年11月ごろから、Chromeでネットにつながりにくくなる現象が発生
  - Edgeなど、Chromeベースの他ブラウザでも
- 2023年2月ごろから報告が多くなる
  - https://support.google.com/chrome/thread/188653227?hl=ja
  - 特定ISPのユーザで多く発生
  - それ以外のISPでも多くはないが報告がある

起きていたこと

- Chromeの出すDNSクエリがなぜか突然UDPからTCPに切りかわる
  - 一度TCPになると再起動するまでUDPに戻ることはない
- 家庭用ルータのDNS機能がTCPクエリをうまく扱えないとネット接続が不安定に
  - そういう機種をユーザに配ってるISPでは多数のユーザがひっかかる
    - https://www.commufa.jp/information/2023/20230209210O
    - https://faq.commufa.jp/faq/show/3536?site_domain=default
  - 好きなルータを買って使ってね、というISPでは特定機種を使ってるユーザのみ
  - https://bugs.chromium.org/p/chromium/issues/detail?id=1413620

IIJキャッシュサーバでの観測

- 2022年11月以降、それまでほとんどなかったTCPクエリが増える
- TCPでクエリを受けてもUDPでフォワードするルータもあると思われるので、実際にChromeが投げていたTCPクエリの量はこのグラフからはわからない



ルータベンダーからの案内

- いくつかのベンダーがそれらしき情報を出している
  - (時期と対処内容から推測してこの件だろうと判断したけど違ってたらごめんなさい)

回避方法

- TCPクエリの扱いに問題のあるDNSサーバを使わない
  - 家庭用ルータのDNS機能を使わず、ISPのキャッシュDNSサーバやpublic DNSを使う
  - 家庭用ルータをTCPクエリを問題なく扱える機種に交換する、ファームウェア更新する
- キャッシュDNSサーバ運用者向け
  - 多数のTCPクエリをちゃんと受けられるかどうか確認
  - とくに、unboundはincoming-num-tcpのデフォルト値が非常に小さいので注意

11月に何があったのか

- 11/2に“Enable AsyncDNS on Windows by default”なるコミット
  - https://github.com/chromium/chromium/commit/629950a2fe003db7d96667c2cb8e902faea63217

3 chrome/common/chrome_features.cc

```
@@ -99,7 +99,8 @@ BASE_FEATURE(kAppShimNewCloseBehavior,
 99  99  // Enables the built-in DNS resolver.
100 100  BASE_FEATURE(kAsyncDns,
101 101                "AsyncDns",
102     - #if BUILDFLAG(IS_CHROMEOS) || BUILDFLAG(IS_MAC) || BUILDFLAG(IS_ANDROID)
    102 + #if BUILDFLAG(IS_CHROMEOS) || BUILDFLAG(IS_MAC) || BUILDFLAG(IS_ANDROID) || \
    103 +     BUILDFLAG(IS_WIN)
103 104                base::FEATURE_ENABLED_BY_DEFAULT
104 105  #else
105 106                base::FEATURE_DISABLED_BY_DEFAULT
```

- バージョンでいうと109以降 (2023/01/10 リリース)
  - 11月から観測されはじめて2月から報告が増えるという実測とも一致

AsyncDNSとは

- Chromeの新しいbuilt-in DNSリゾルバ
  - HTTPSレコードに対応 ← HTTP/3に必要
- 11月のコミットはAsyncDNSをデフォルト有効に変更しただけで、AsyncDNS自体のコード修正ではない
  - 先行して有効になっていたMacやAndroidではこのような問題は起きていない
  - つまり、Windows版固有の問題

回避方法その2

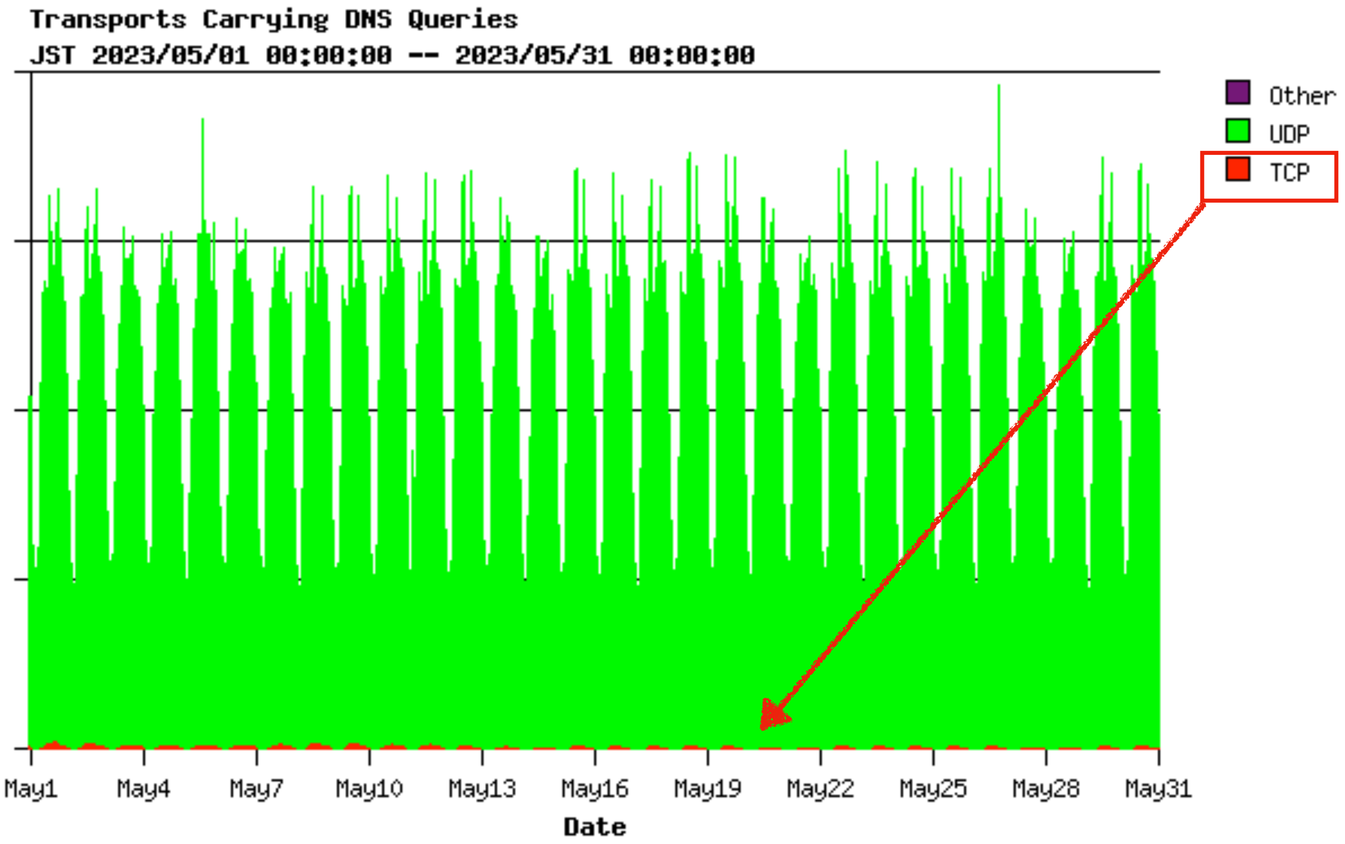
- AsyncDNSが有効にされたことが原因なら、無効にすればよい
  - chrome://flags#enable-async-dns をfalseにする
    - このフラグはおそらく将来的には廃止されて強制的に有効になってしまうと思われるが、それまでにchromeが修正されるならば問題ない
    - 先行してAsyncDNSが有効になったMac/Androidはすでにこのフラグは存在しない
  - AsyncDNSで実現された機能が使えなくなる
    - HTTPSレコードのクエリが出なくなる
  - Edgeにはこのフラグはないっぽい…

- AsyncDNSが何をやってるのか
- なぜWindows版だけ問題が起きるのか

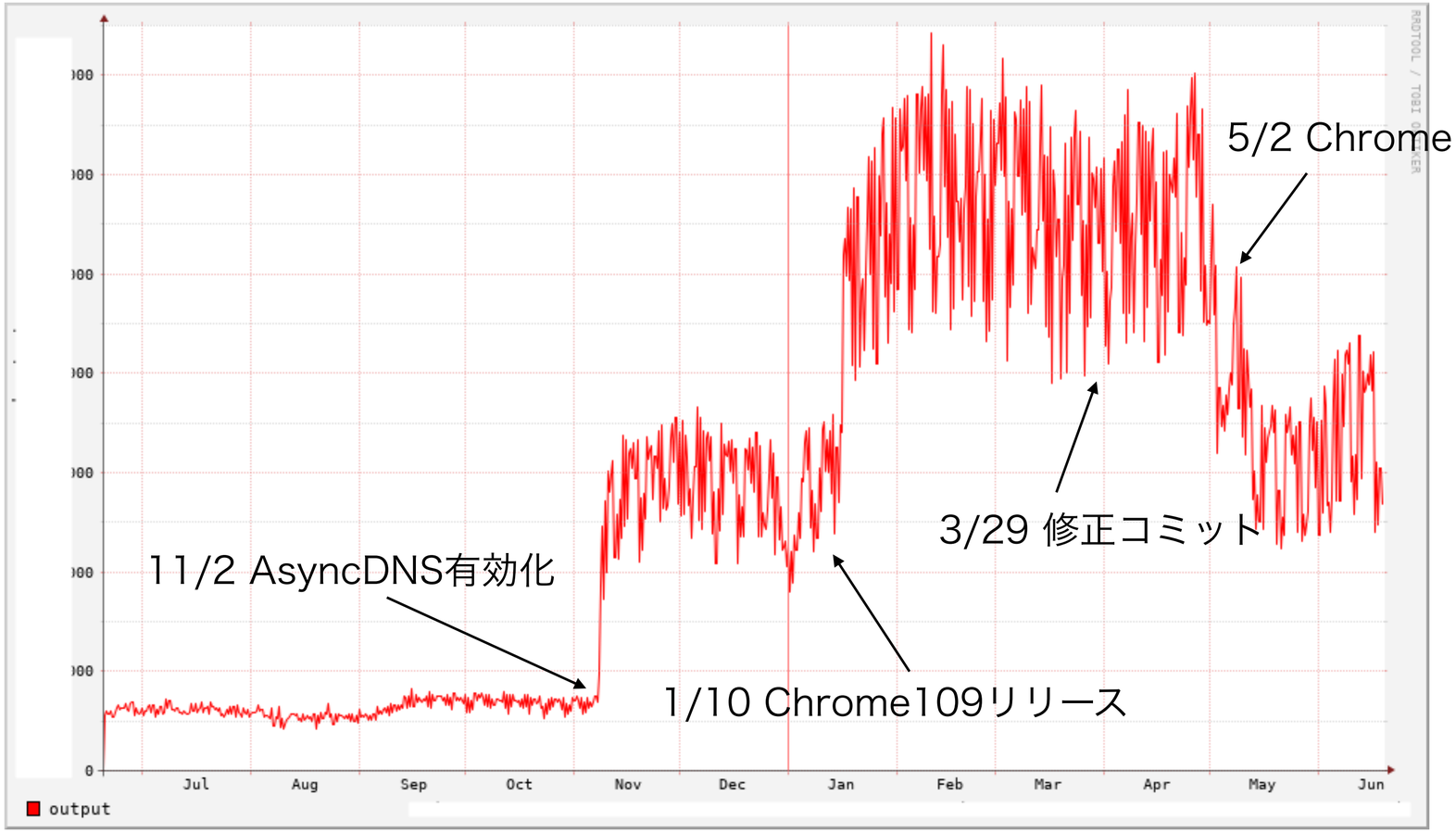
- 草場さんよろしくー

その後

- 3/29 修正コミット
- 5/2 Chrome113としてリリース
- キャッシュサーバでの観測では、TCPクエリは減っている
- が、なくなってはいない
- 問題は解決していないと思われる



TCPクエリの推移


5/2 Chrome113リリース
3/29 修正コミット
11/2 AsyncDNS有効化
1/10 Chrome109リリース
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
output

修正版の問題点

- ソースポートの重複は直近256クエリ中2回まで、越えたらエントロピーが低いと判断して以後TCPでクエリ → 3回までに緩和
- コミットログには「Windowsはポート番号の範囲が狭いからひっかかりやすい」とあるが、これは事実誤認
  - https://github.com/chromium/chromium/commit/59d686c1417b5aea7b1d94a28bac45d8d8f26fe0
  - エフェメラルポートのレンジはWindows、Macともに49152-65535 (16384個)で、Windowsだけとくに狭いわけではない
    - RFC6056における”traditional”なレンジ(2.1節)
    - ちなみにRFC6056における推奨レンジは1024-65535 (64512個)

ソケットキャッシュ(1)

- UDPポートの消費を抑えるために、同一のサーバへの通信であれば、ソースポートをキャッシュして同じポートを使いまわしするWindowsの機能、らしい
  - スタブリゾルバが送るクエリのソースポートが固定されてしまう
  - cacheじゃなくてreuseという方が適切な気がする
  - https://lists.dns-oarc.net/pipermail/dns-operations/2023-March/021979.html
- 「直近m回中n回重複したら」というアルゴリズムが有効なのはポート割り当てがランダムな場合
  - 積極的に再利用される場合にnを増やしても解決にならない

ソケットキャッシュ(2)

- Microsoftの公式な情報が見つからない…
- っていうか、MS以外の情報もdns-operationsに流れたメールぐらい
  - このメールもけっこう間違いが多い
    - Chrome105 (2022/08) から、とあるけど実際は109から
    - Windows11から、とあるけどWin10でも観測されている
  - しかし、ポート番号再利用の仕組みが存在することは間違いなさそう

なぜソケットキャッシュ? (1)

- UDPはNATと相性が悪い
  - UDPにはTCPのFINのような明確な終了パケットがない
  - NAT箱は終わった通信を変換テーブルから消すタイミングがわからない
    - TCP FINのような終了パケットを検知してテーブルから消すのではなく、タイムアウトで消す
- QUICもNATと相性が悪い
  - UDPベースのプロトコルなので
  - QUICには接続断を知らせる手続きがあるが、暗号化されていてNAT箱は検知できない
  - JANOG48: QUICとNATと https://www.janog.gr.jp/meeting/janog48/lt4/

なぜソケットキャッシュ? (2)

- 接続を送受信IPアドレスやポート番号などで管理していた従来のTCP、UDPと異なり、QUICはコネクションIDで管理
  - ポート番号は見ていないので、UDPソケットを再利用しても問題が起きない
  - 再利用すればNATテーブルのエントリは増えない
- HTTP/3を筆頭に、今後QUICの利用が急増する
  - NAT機器の負担を下げるためにUDPソケットキャッシュを実装したのでは、と想像
  - ほんとにそういう理由なのかはMicrosoftに聞かないとわからん
  - とはいえUDPはQUIC以外でも使うということをMicrosoftが知らんはずないし…

Chrome以外では?

- ソケットキャッシュはWindowsの「OSとしての機能」っぽい
- ならば、Chrome以外でもソケットキャッシュされるのでは?
- キャッシュDNSサーバでソケットキャッシュされるとマズい
  - ソースポートが固定される → キャッシュポイズニングの脆弱性

キャッシュDNSサーバ

- 挙動は確認できてません…
  - Microsoft DNS Serverは手元に実験に使えるライセンスがなかった…
  - BINDは手元の環境ではインストーラがなぜか異常終了して動かせなかった…
  - だれか確認しておしえてください…
- もしもWindows用のキャッシュDNSの出すクエリがソケットキャッシュされるようならば、誰かが見つけてすでにCVEが出てるんじゃないかなぁ
  - CVEが出てないってことはだいじょうぶなはず…(願望)
- Windows版BINDは9.16が最後(2024/Q1 EoL)

その他DNSクエリを出すアプリ

- 手元の環境(Win11 pro 22H2 22621.1848)での確認結果
  - 同一リビジョンのWindowsですら動作が異なることがあるようなので参考程度に
  - とくに、**この環境ではChromeでソケットキャッシュの動作を確認できていません**
- ソケットキャッシュ確認
  - Firefox114、Resolve-DnsName(powershell)
- ソケットキャッシュ確認できず
  - Chrome110～114、Edge114、nslookup
- 異なるアプリ間でソケットキャッシュされるかは不明(あってもおかしくない)

ソケットキャッシュじゃないけど

- 手元の環境(Win11 pro 22H2)では、ポート番号の再利用ではなく、前回使われたポート番号に+1しただけのポートが使われることがある
  - digを3回実行 →
  - nslookupも同様

```
10:20:03.971505 IP 192.168.11.21.62338 > 192.168.11.123.53: 19896+ [1au] A? www.iij.ad.jp. (54)
10:20:03.971785 IP 192.168.11.123.53 > 192.168.11.21.62338: 19896$ 1/0/1 A 202.232.2.180 (58)
10:20:08.443174 IP 192.168.11.21.62339 > 192.168.11.123.53: 63000+ [1au] AAAA? www.iij.ad.jp. (54)
10:20:08.443460 IP 192.168.11.123.53 > 192.168.11.21.62339: 63000$ 1/0/1 AAAA 2001:240:bb81::10:180 (70)
10:20:18.719108 IP 192.168.11.21.62340 > 192.168.11.123.53: 52565+ [1au] MX? iij.ad.jp. (50)
10:20:18.719373 IP 192.168.11.123.53 > 192.168.11.21.62340: 52565$ 1/0/1 MX omgi.iij.ad.jp. 10 (59)
```

- ポート番号をひとつ知ることができれば、次に使われるポート番号を外部から予測可能になるので、ポイズニングに脆弱

まとめ

- ChromeのTCPクエリ問題を調べてみました
- ポイズニング防止のためにポート番号をたくさん使いたいChromeと、消費を抑えたいWindowsの衝突
- チケットはクローズされちゃったし、最近はあまり報告も聞かないけど、問題はまだ解決していないと思われる
- Chrome以外にも、UDPなアプリを動かす場合は気をつけたほうがいいかも
- Microsoftさん情報だして…!