

Internet Week ショーケース in 札幌

CDNの仕組みとトラフィック分析
～CDN事業者が語る～

2023/07/20
JOCDN株式会社
青野 慧志郎



アジェンダ

- 自己紹介
- CDNの仕組み
- CDN事業者視点のトラフィック分析

アジェンダ

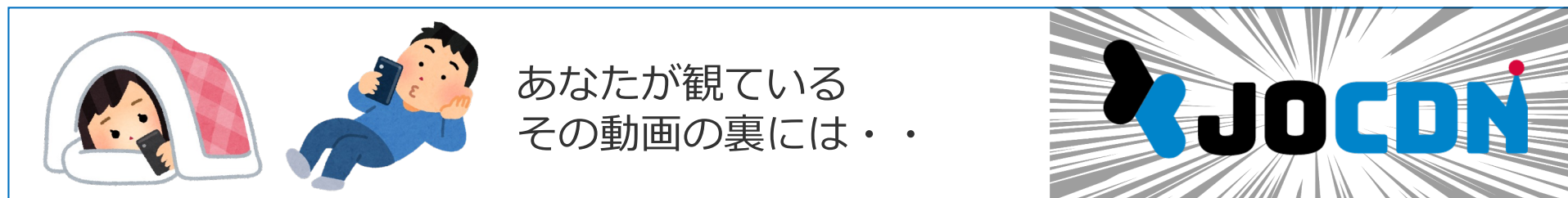
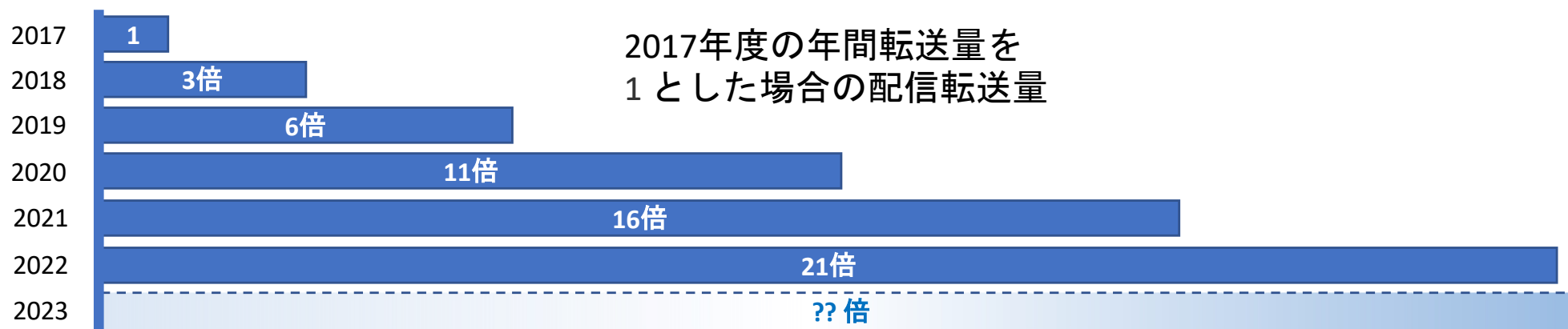
- 自己紹介
- CDNの仕組み
- CDN事業者視点のトラフィック分析

自己紹介

- **名前**
 - 青野 慧志郎 (あおの けいじろう)
- **所属**
 - JOCDN株式会社 技術部
 - インターネットイニシアティブから兼務出向 (= IJ にも在籍)
 - どちらも配信系のお仕事
- **仕事**
 - 自社CDNサービスの運用全般
 - 運用チームのマネジメント

JOCDNとは

- 2016/12 日本テレビとIIJの合併で設立
 - 現在は在東名阪民放局、NHK、WOWOWも出資
- (主に) 動画配信事業者様向けのCDNサービスを提供
 - 放送局系の動画配信サービスなどにご利用
 - 最近ではWebやゲーム配信にも実績あり

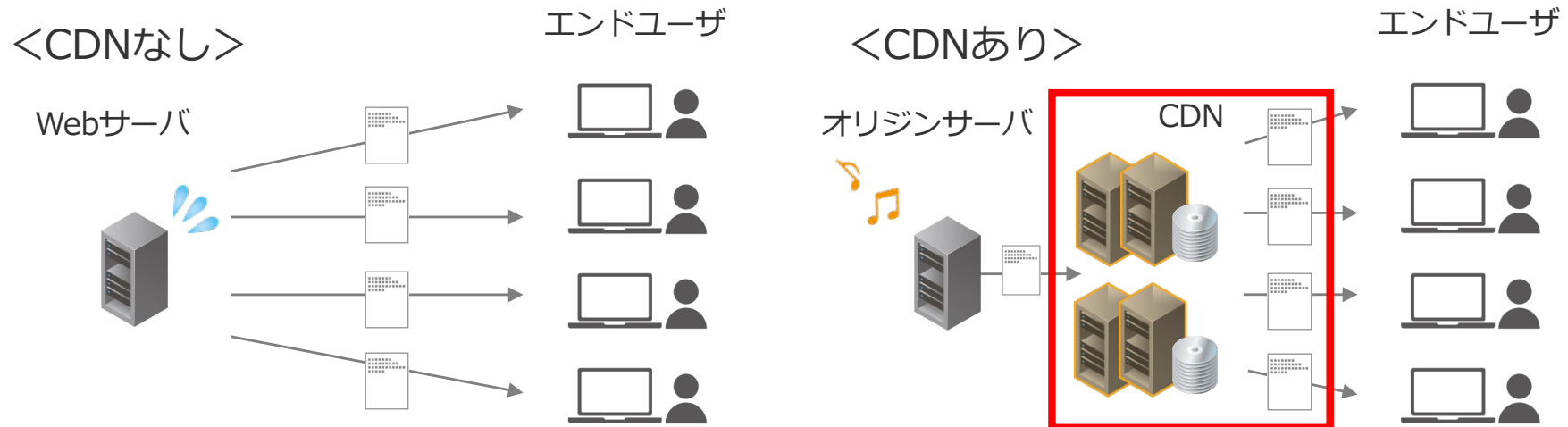


アジェンダ

- 自己紹介
- **CDNの仕組み**
- CDN事業者視点のトラフィック分析

CDN (Content Delivery Network) とは

- 効率的かつ高いパフォーマンスで、安定したコンテンツ配信を実行するための仕組み



- CDNがコンテンツをキャッシュ・配信することでオリジンの**負荷軽減**と**高速配信**を実現
- 一般的に**従量課金**であるため、費用を抑えつつ**突発的なアクセス**にも耐える構成が可能

例えば・・・

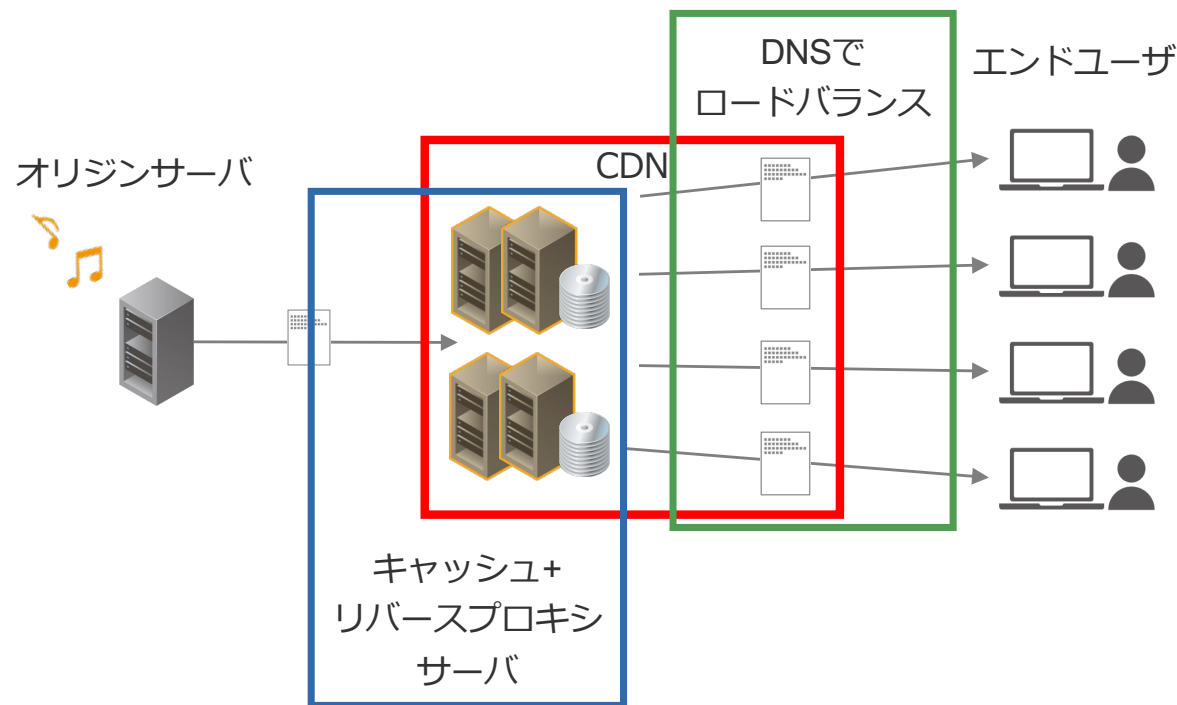
- 短時間で高負荷になるサイト（災害情報、天気予報、交通情報、ニュースなど）
- アクセスの多いWebサイト
- 容量が大きい動画ストリーミング配信
- 大規模なライブ配信

などで利用されています

CDNの仕組み

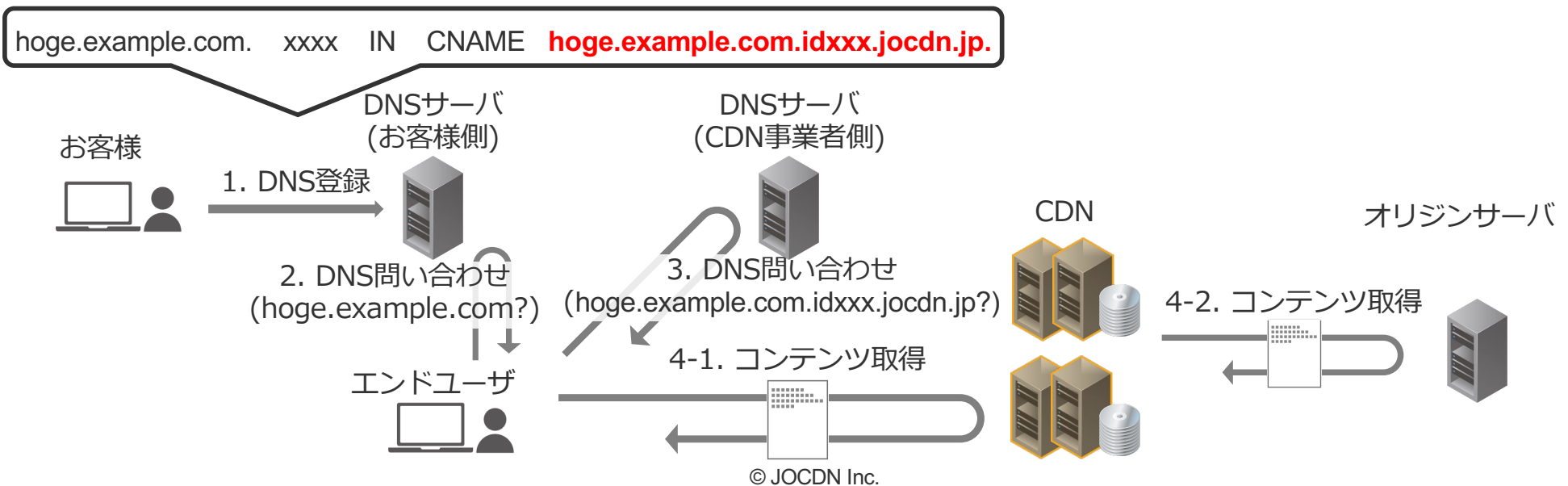
要は CDN とは・・・

- キャッシュ + リバースプロキシサーバ
- それを大量に横に並べて DNS でロードバランスした設備



CDNを利用した配信の流れ

1. CDN事業者が払い出したFQDNをCNAMEとしてDNS登録
2. エンドユーザがホスト名を名前解決するとCNAMEが返る
3. CNAMEを名前解決するとCDNのサーバのA/AAAAが返る
4. CDNのサーバがエンドユーザにコンテンツをレスポンス
 1. キャッシュあり: キャッシュを返す
 2. キャッシュなし or 期限切れ: オリジンから取得しそれを返す



CDNの設定 (事業者により異なる)

- **基本設定**

- オリジン
- 配信方法
 - HTTP and/or HTTPS
 - 証明書は持ち込み or Let's Encrypt
 - TLS versions (1.0~1.3)
 - HTTP versions (1.1, 2)
 - IP versions (v4, v6)

- **より効率的なキャッシュ管理**

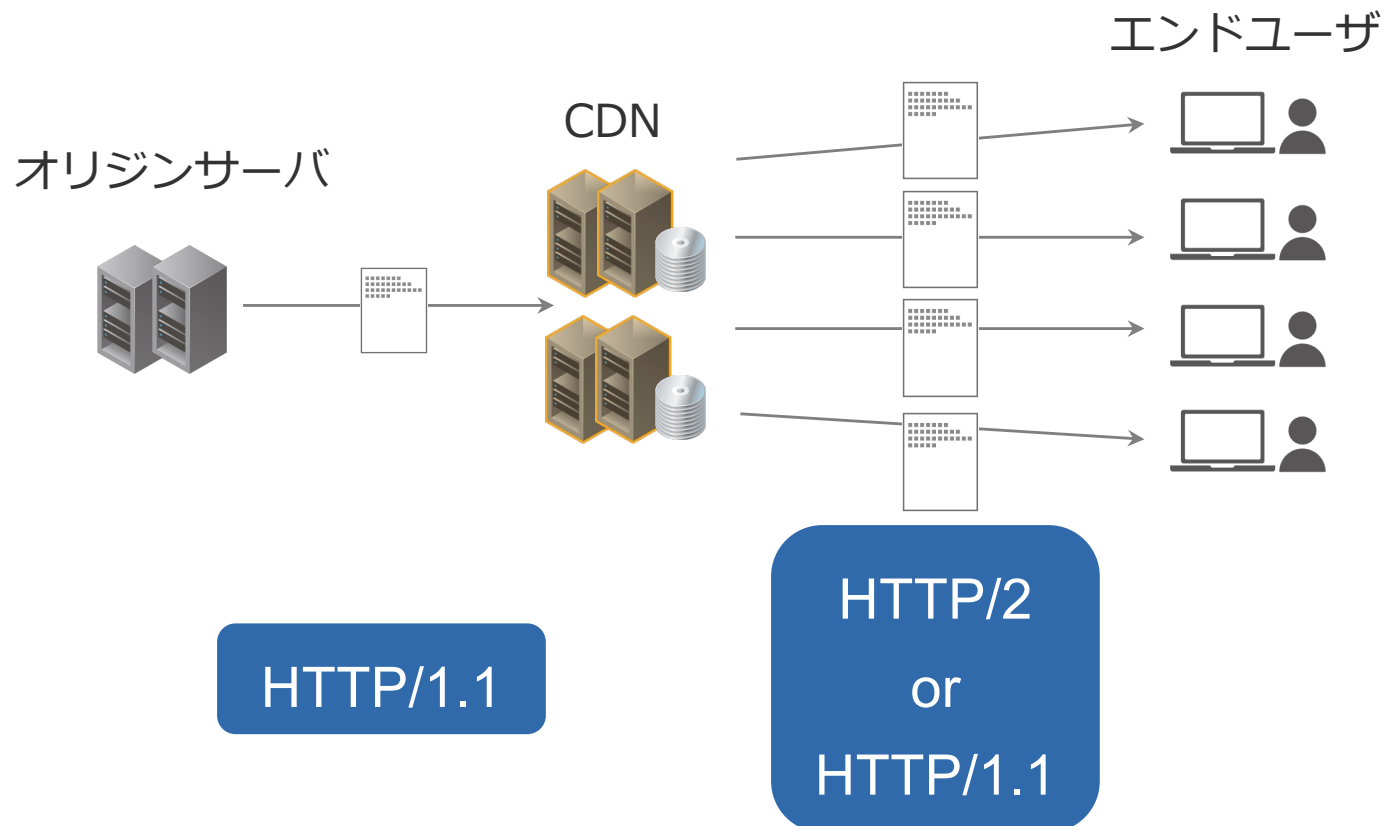
- TTL
 - Cache-Control header
- キャッシュキー
 - クエリ
 - HTTP header

- **その他**

- header 制御、アクセス制御、ログ、レポート・・・

CDNを利用する機能的なメリット

- **オリジンが対応していない方式で配信できる**
 - HTTP/2 や IPv6 で配信したい！ とか
 - エンドユーザ - CDN 間と CDN - オリジン間の通信は別物
 - 逆に HTTP/1.1 や IPv4 だけにすることも可
 - IPv4 だけにすると名前解決時にAAAAを返さなくなる



CDN 内での障害発生時の挙動

- **サーバ**
 - 名前解決時に当該サーバのA/AAAAを返さなくなる
 - 常時死活監視している
- **DC拠点**
 - DC拠点に加えて、CDN事業者ドメインの権威DNSサーバも分散・冗長化
- **CDNまるごと**
 - CDNという仕組み自体が冗長性を含んでいるがそれでも障害は起きうる
 - マルチCDN、バックアップCDN
 - CDN共通の機能しか使えない、挙動が異なる等に注意
- **オリジン**
 - 単一障害点になりがち
 - オリジンフェールオーバー
- **CDNを利用することがサービス冗長化、DR対策にもなる**

最近のCDN動向

- 標準化の動きが目立つ


- Cache-Status ([RFC 9211](#))

- CDN が付与している（と思われる）header の例

- J-Stream

Via:	JSTCDN
X-Cache:	HIT/S
X-Cache-Age:	28859/43200
X-Origin-Date:	Tue, 18 Jul 2023 10:23:21 GMT

- CloudFront

Via:	1.1 0ae700b4d5aa148e2d31504815385bb0.cloudfront.net (CloudFront)
X-Amz-Cf-Id:	tEdbfC31lJOWaNoXleEwIj9Wyk9qn6FvHgReBqfu1HvuvNtFvG1HA==
X-Amz-Cf-Pop:	NRT57-C4
X-Cache:	Miss from cloudfront
X-Nginx-Cache:	EXPIRED 

- fastly

X-Cache:	HIT, HIT
X-Content-Type-Options:	nosniff
X-Imgix-Id:	6ebf3dc656b3fd10f6cbd09bed409389e1fc0182
X-Imgix-Render-Farm:	01.148560
X-Served-By:	cache-sjc1000120-SJC, cache-itm18823-ITM

- キャッシュHIT/MISSを表すX-Cacheの仕様を標準化
- HIT/MISS、TTL、転送結果の他、多段アクセス（ブラウザ、CDN、リバースプロキシ、オリジン）の場合もリストで表現
- Cache-Status: OriginCache; hit; ttl=1100,
"CDN Company Here"; hit; ttl=545

最近のCDNネタ

- 標準化の動きが目立つ

- Targeted HTTP Cache Control ([RFC 9213](#))

- Cache-Control

- ブラウザにもCDNにも有効。CDNによっても挙動が異なる

- ブラウザとCDNとオリジン手前のリバースプロキシで設定を細かく分けたい、とかが不便

- CDN-Cache-Control

- CDN ピンポイントで効く Cache-Control

- 他にも・・・

- An HTTP Cache Invalidation API (draft)

- Invalidation (Purge) 用 API の標準化

- HTTP Cache Groups (draft)

- Cache-Groups response header

- Cache-Group-Invalidation response header

- キャッシュに対する tag 付けのようなもの

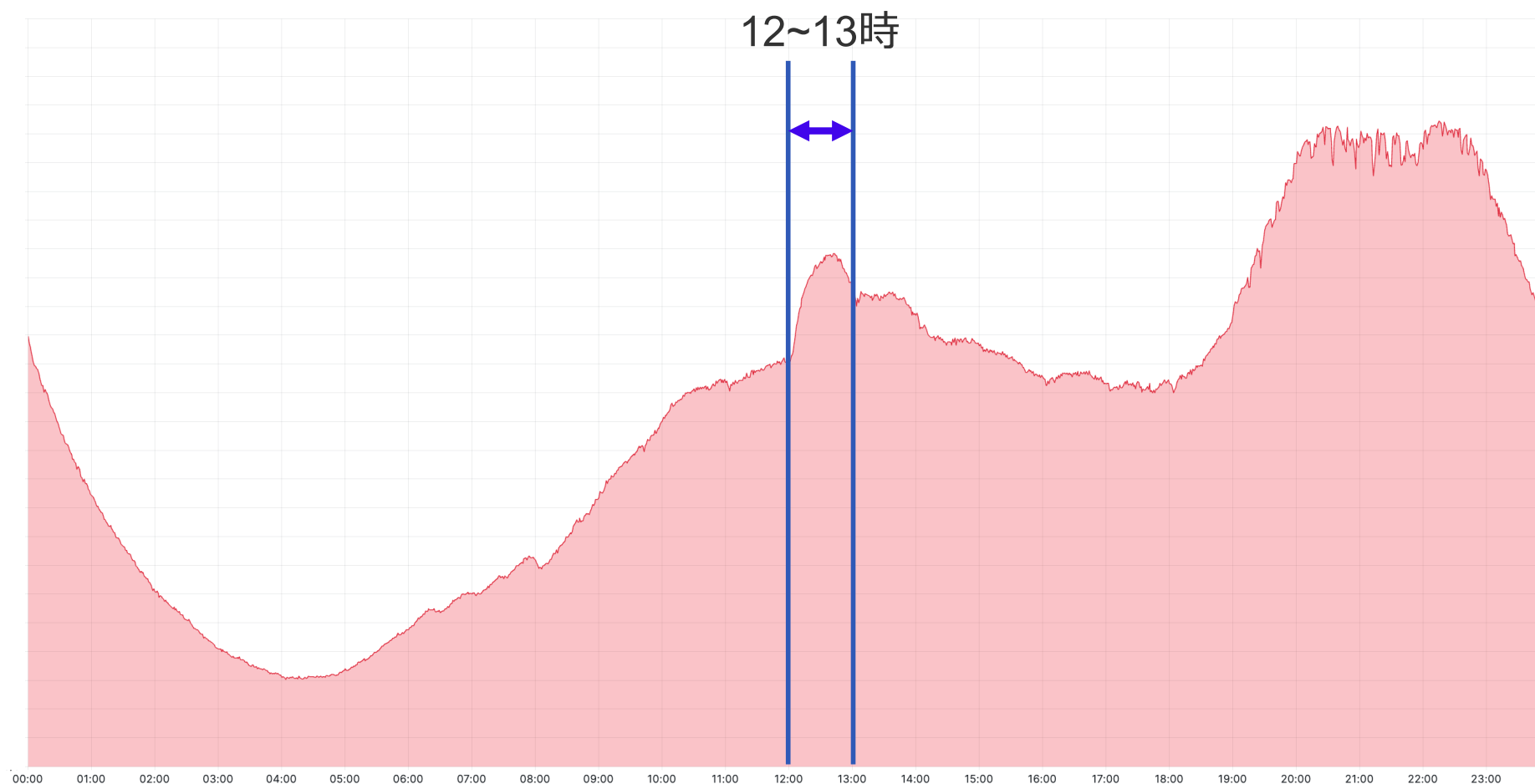
アジェンダ

- 自己紹介
- CDNの仕組み
- **CDN事業者視点のトラフィック分析**

トラフィック総量 (全体, 1d)

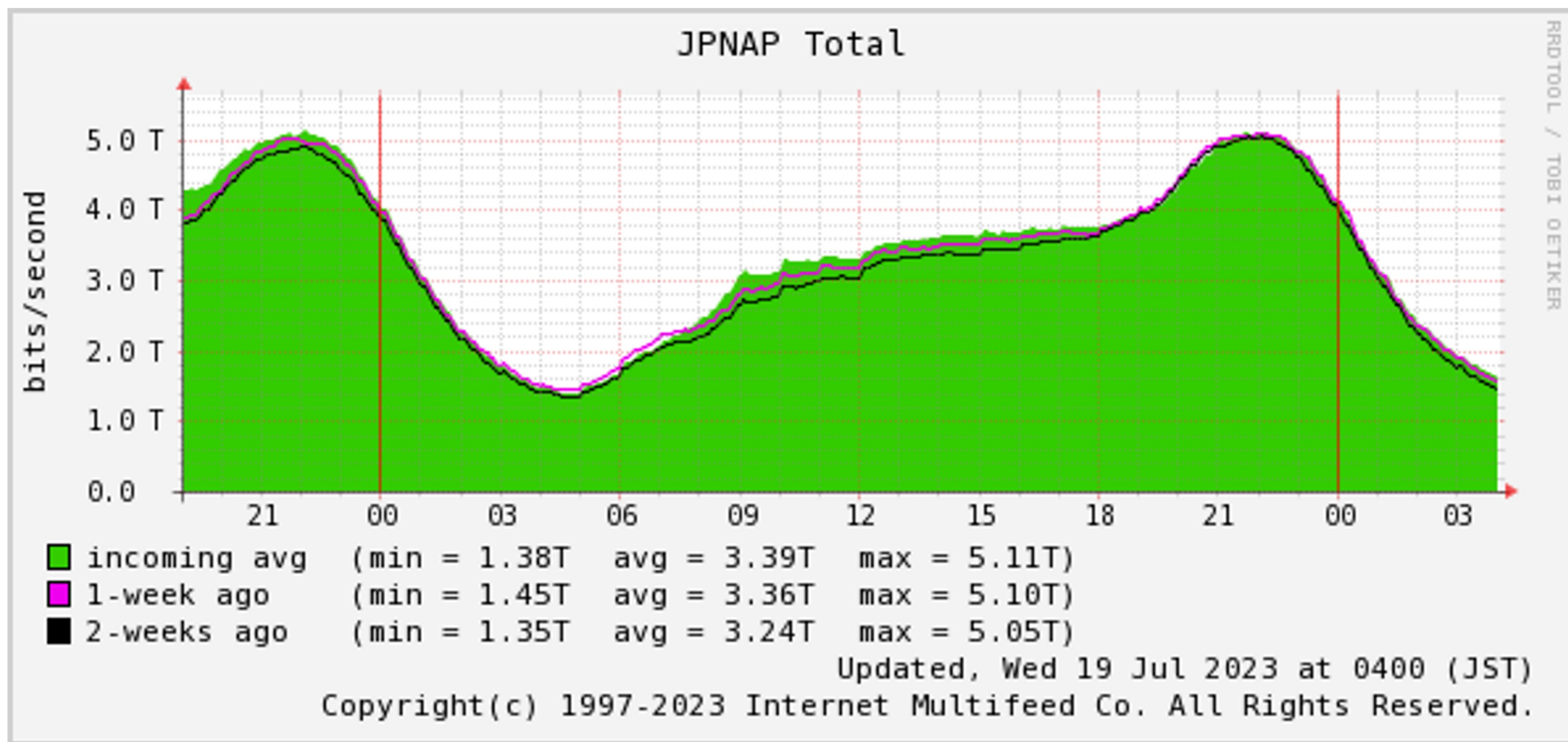
- 最近の平日
- 明け方に減、夜間にピーク (インターネット全体と同傾向)
- 昼休みのピークが特徴的 (休日にはなくなる)
 - 昼休みにサクッとスマホで動画視聴？

Front Traffic



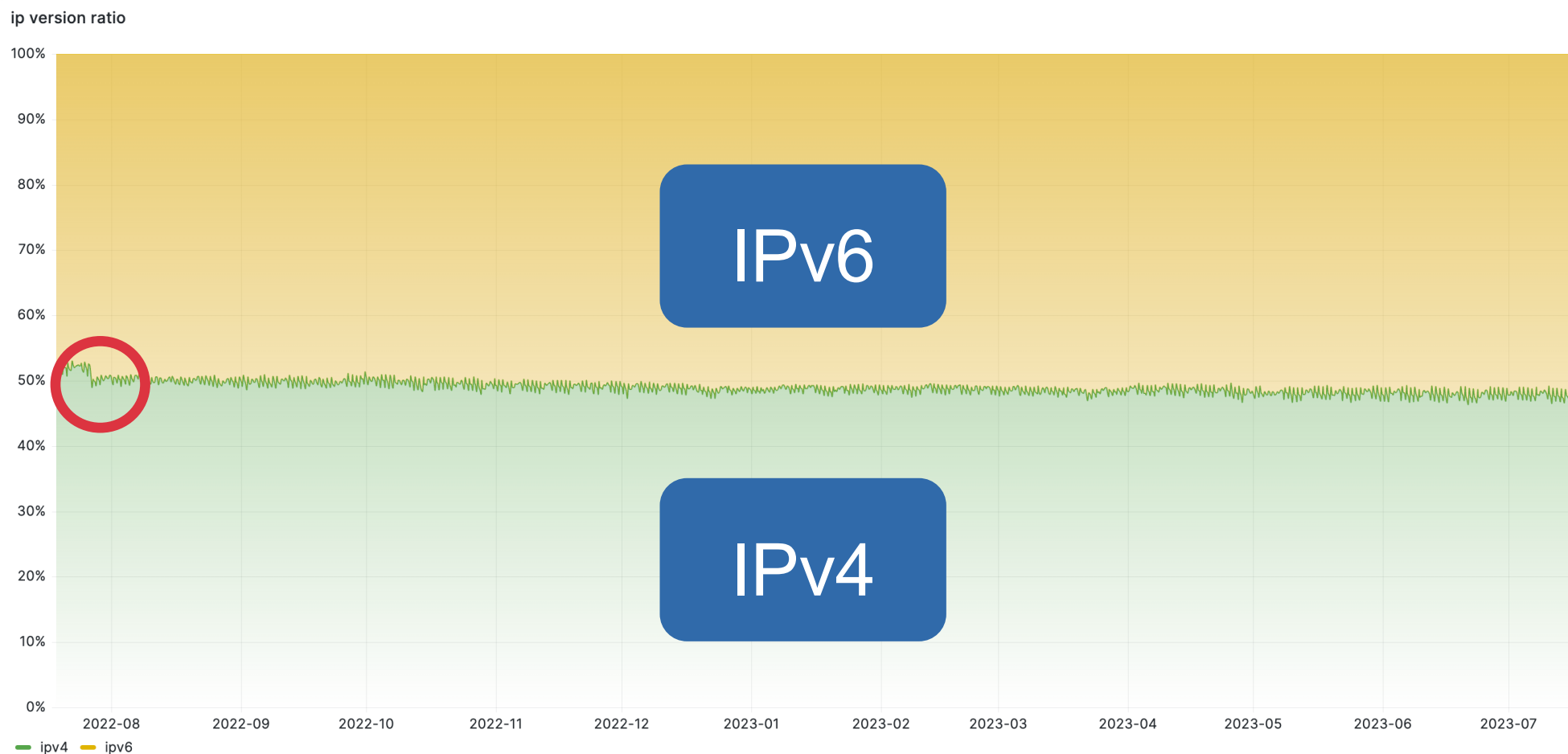
(参考) JPNAP全体のトラフィック

- <https://www.jpnap.net/ix/traffic.html>



IPv4/v6割合 (全体, 1y)

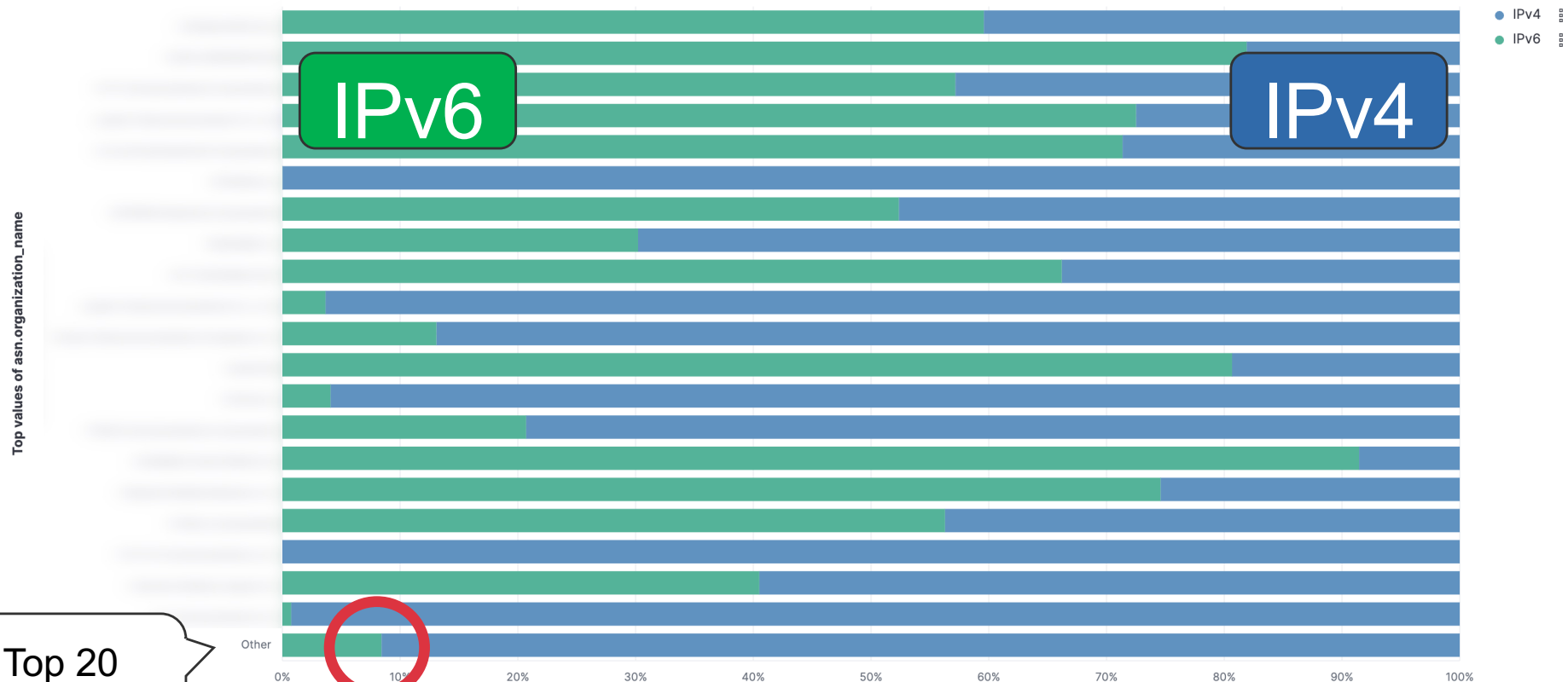
- ほぼ半々。IPv6 がじわじわ増加し、最近は50%超え
- 2022/07下旬にやや大きな変動
 - 某社での大規模リニューアルの影響？



IPv4/v6割合 (ISP別, 1d)

- ISP間で差があるが、アクセス数上位ほどIPv6が多い傾向
- Top 20 以下(地方ケーブルTV等) ではIPv4が9割以上
 - IPv6設備への投資状況による？

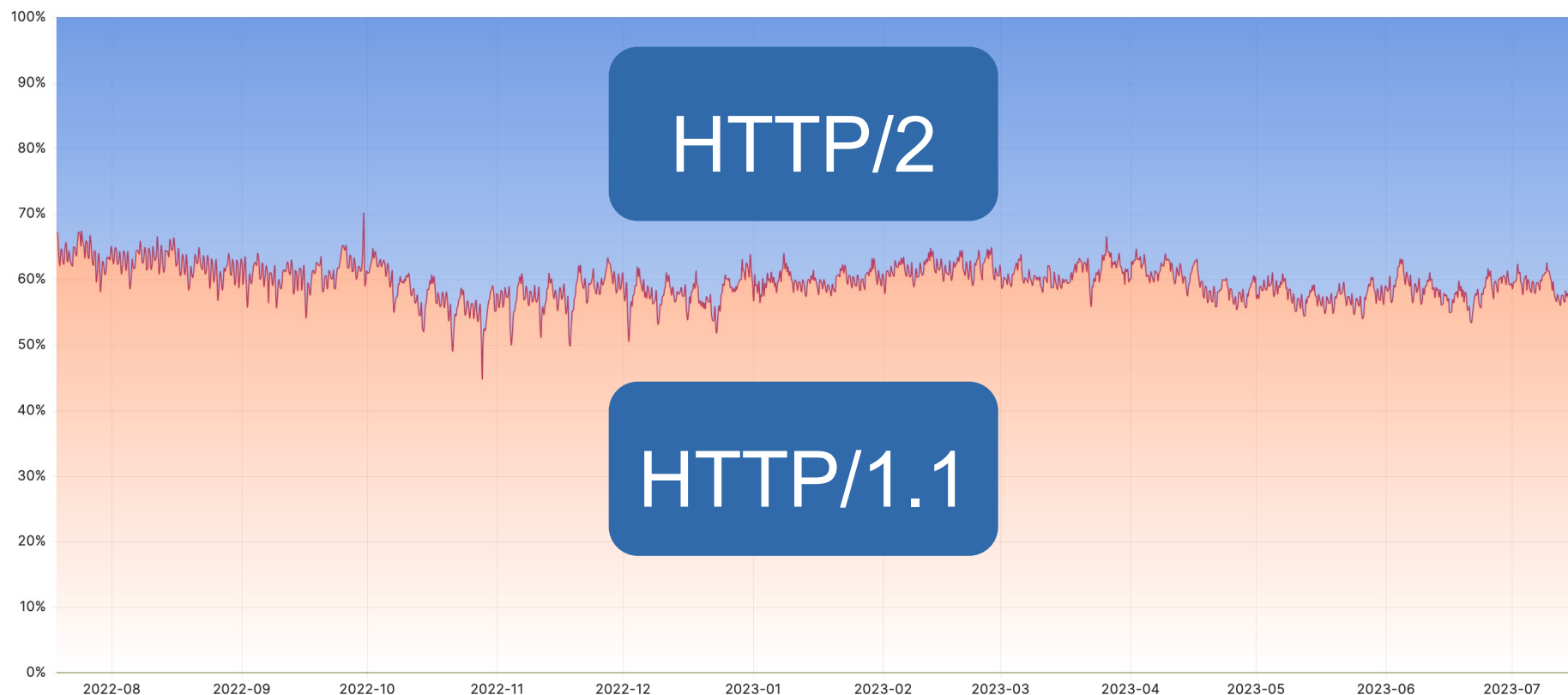
IPv4/v6 ratio per AS



Top 20
以下の合計

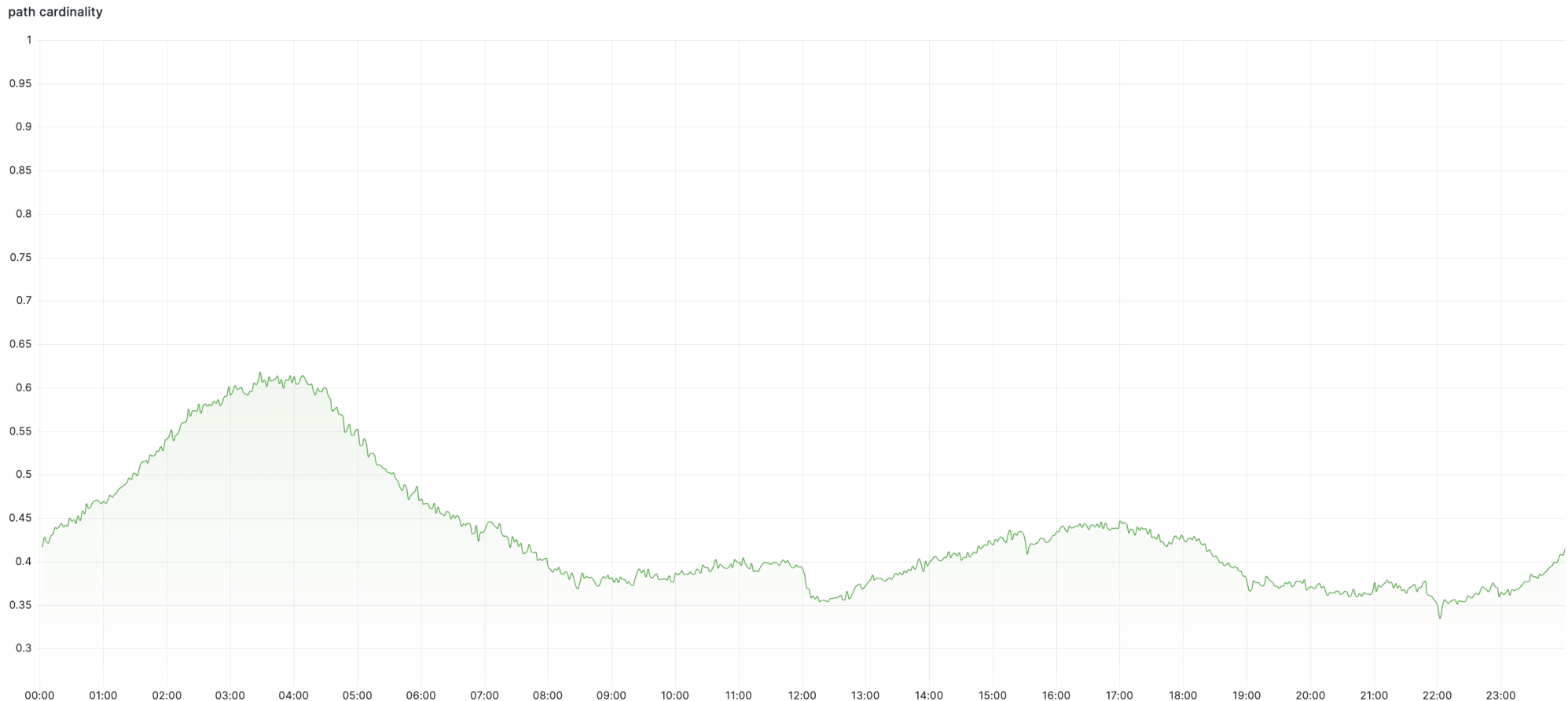
HTTP/1.1, 2割合 (全体, 1y)

- HTTP/1.1 の方がやや多め
 - HTTP/2 が 2/3 というデータもある
 - <https://httparchive.org/reports/state-of-the-web#h2>
- **CDN で HTTP/2 の on/off が可能 (default off)**
 - 動画配信（ファイルサイズ大）では恩恵が少ない？
 - 特殊なデバイスの対応が困難？



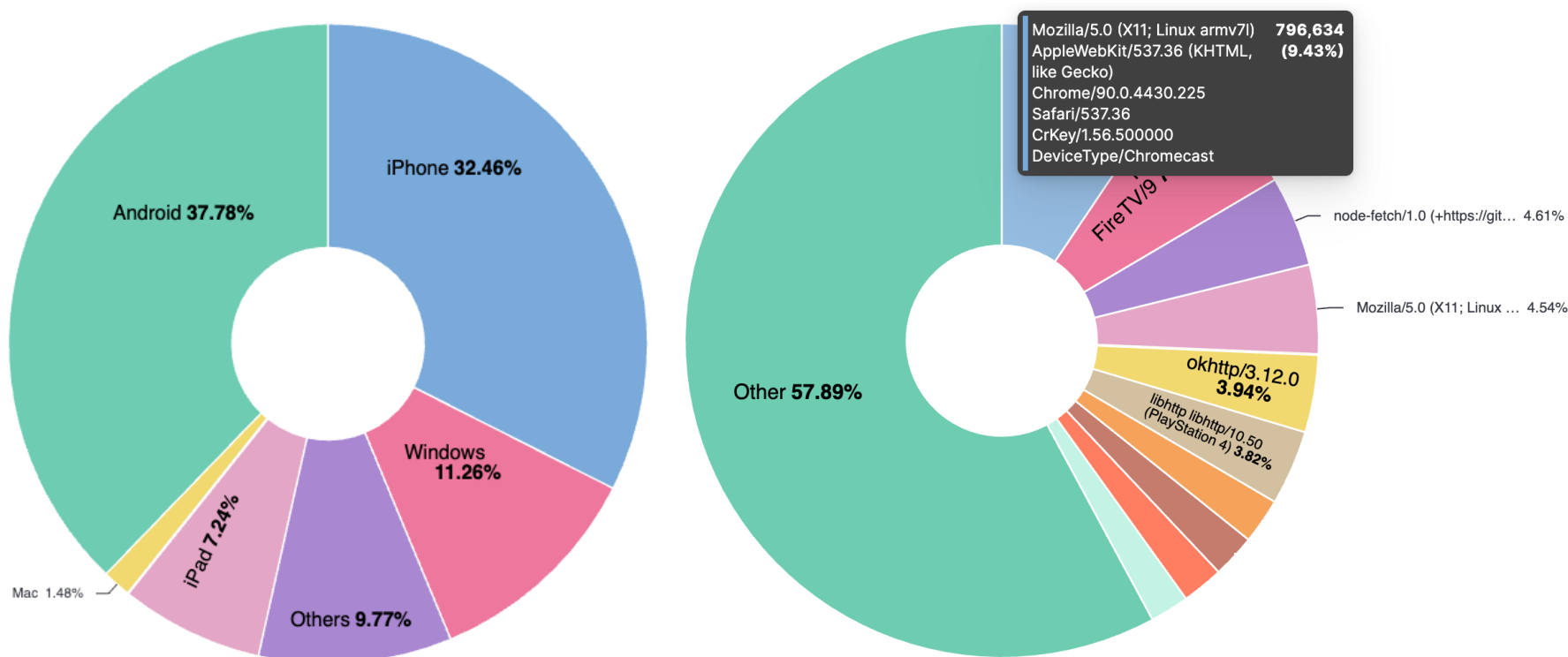
path cardinality (全体, 1d)

- 最近の平日
- path cardinality = GET する path のばらつき具合
 - 小さいほどばらつき小 = 特定のコンテンツに集中
 - ちょうどトラフィック量そのものと真逆の動き



User-Agent (全体、1d)

- (無数にあり分類しきれないので目安程度に)
- スマホ + タブレット + PC で約9割
 - iPhone, iPad, Android, Mac, Windows
 - ただし Android には一部の FireTV やテレビも含む
- 他は スティック型デバイス (Chromecast, FireTV)、ゲーム機 (PlayStation), テレビ、セットトップボックスなど

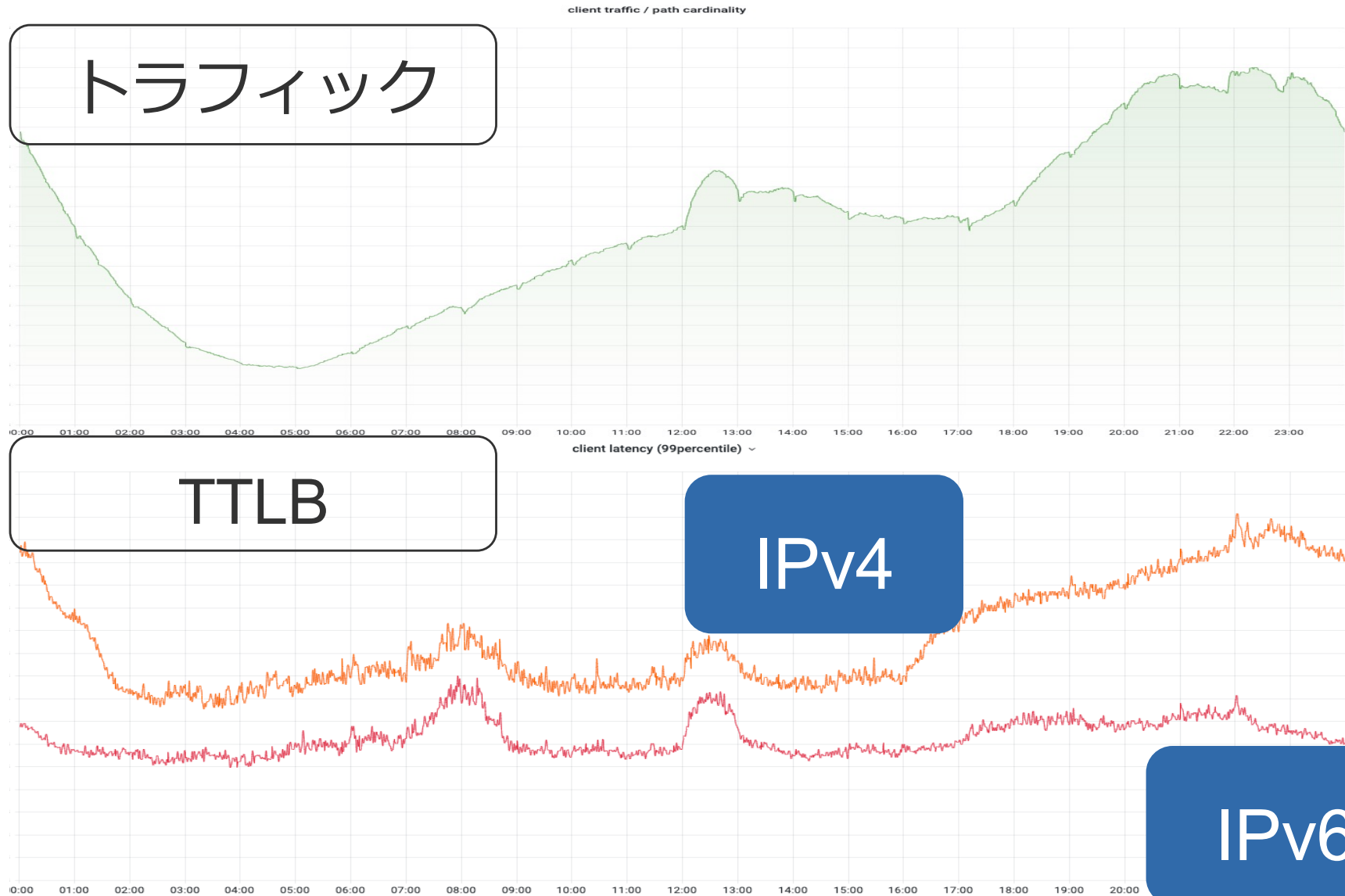


CDN事業者から見た品質指標

- **TTLB (Time To Last Byte)**
 - ファイルをすべて送信しきるまでの時間
 - 動画再生に必要なタイミングまでにレスポンスを返せなければ、動画は一時停止 = rebuffering
 - 速く・安定してレスポンスを返すことが大事
- **TTFB (Time To First Byte)**
 - ファイルを送信し始めるまでの時間
 - キャッシュヒット or NOT?
 - メモリから返した or NOT?
 - ディスク読込して返した場合はディスクIO性能で変動
 - NVMe < SSD < HDD

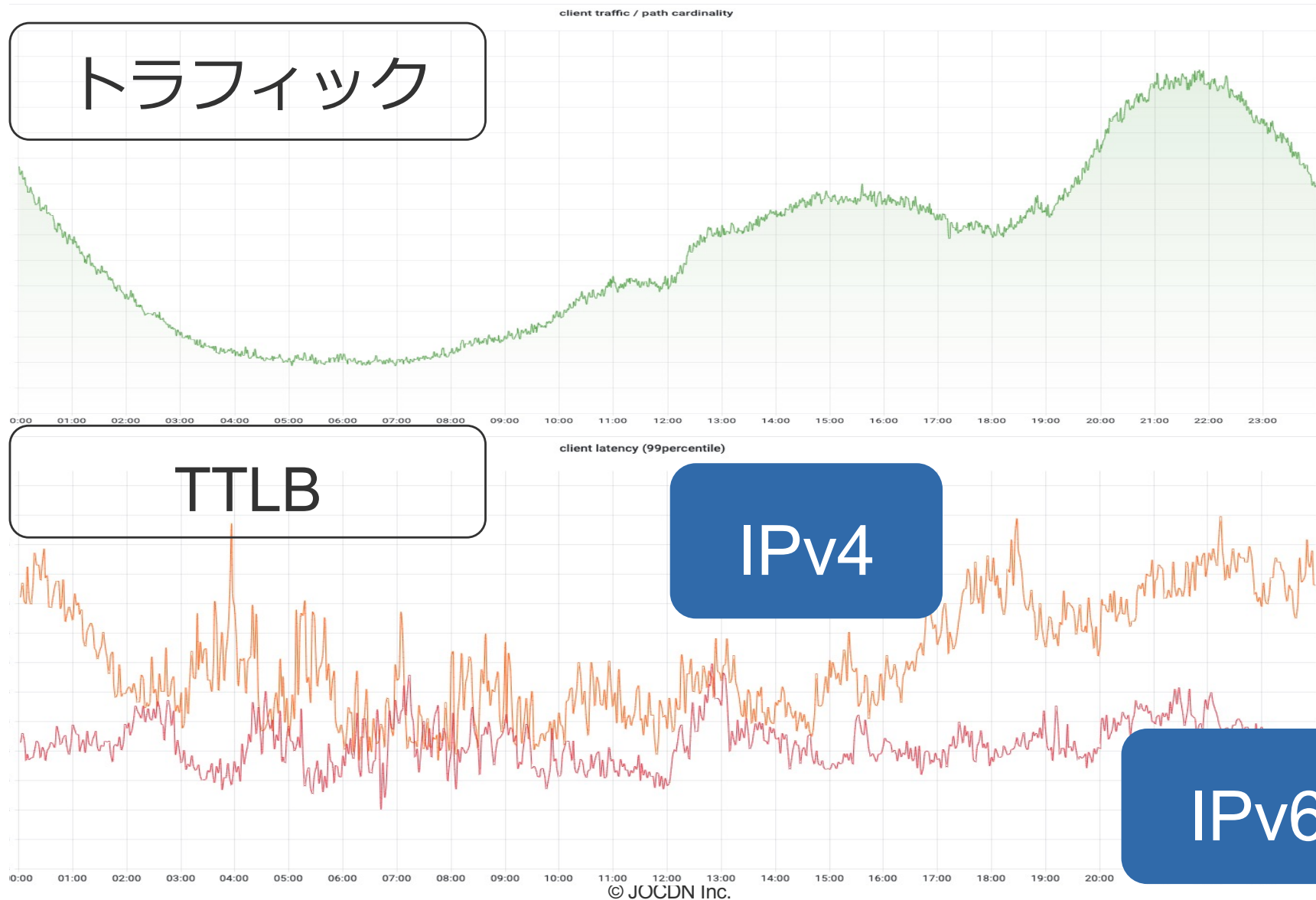
CDN全体のトラフィックとTTLB (1d)

- なんとなくIPv6の方が速そう (雑)
 - 「プロトコルの性質としてIPv6の方が速い」ではない



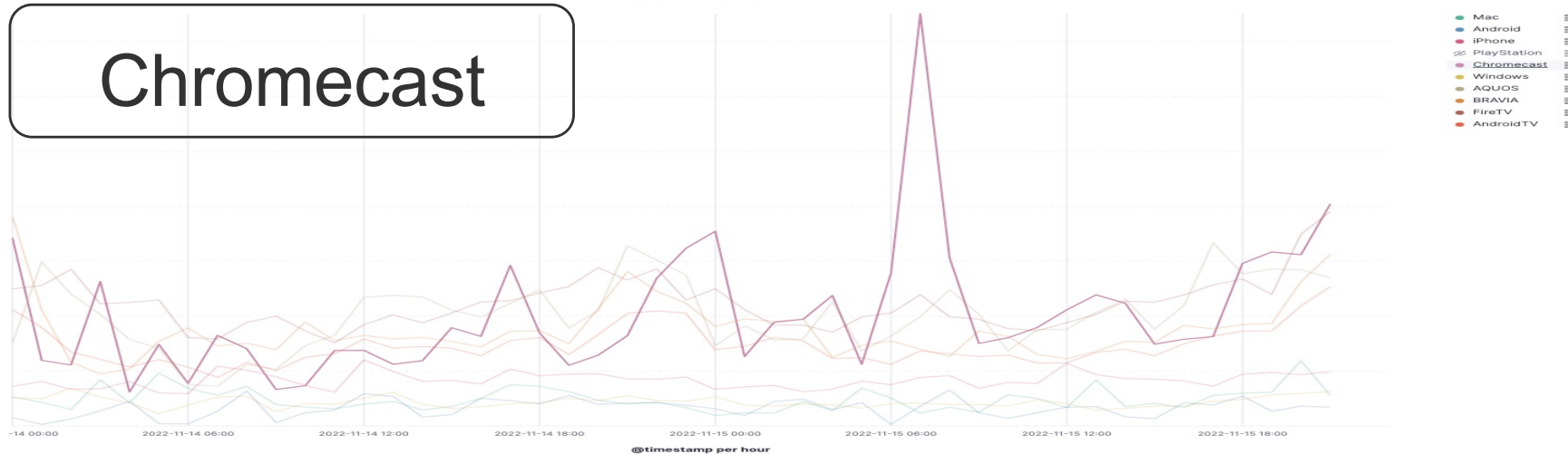
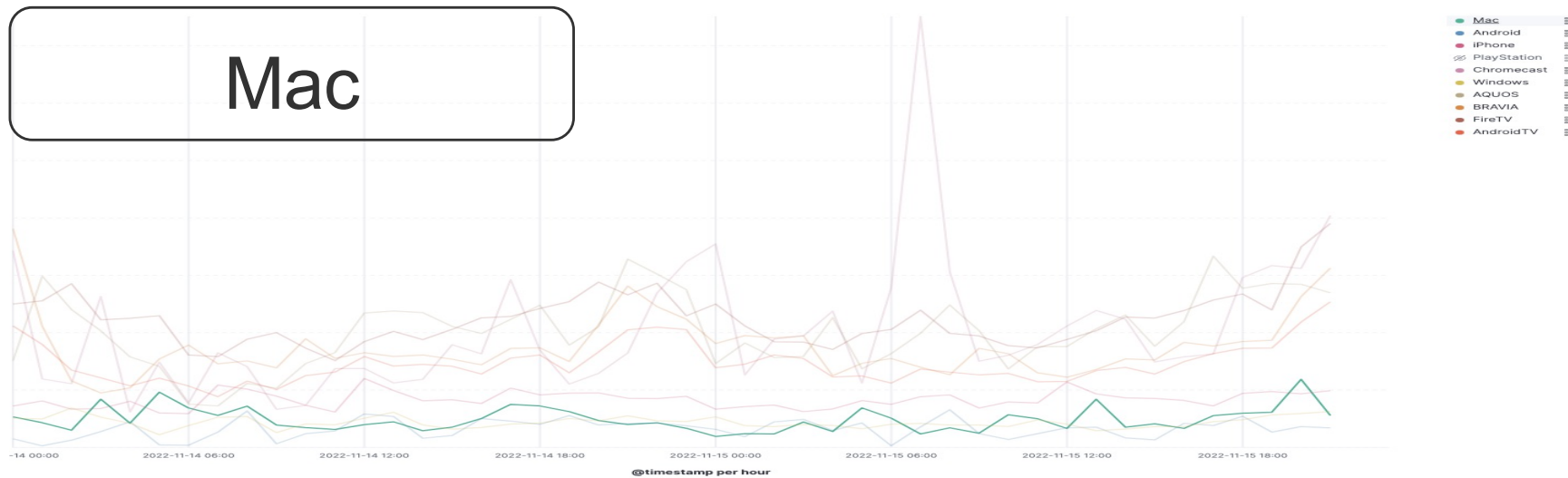
あるFQDNのトラフィックとTTLB (1d)

- 空いている時間帯（明け方等）はIPv4/v6の差は少ない
- 混雑している時間帯（夜間等）は顕著な差が出る



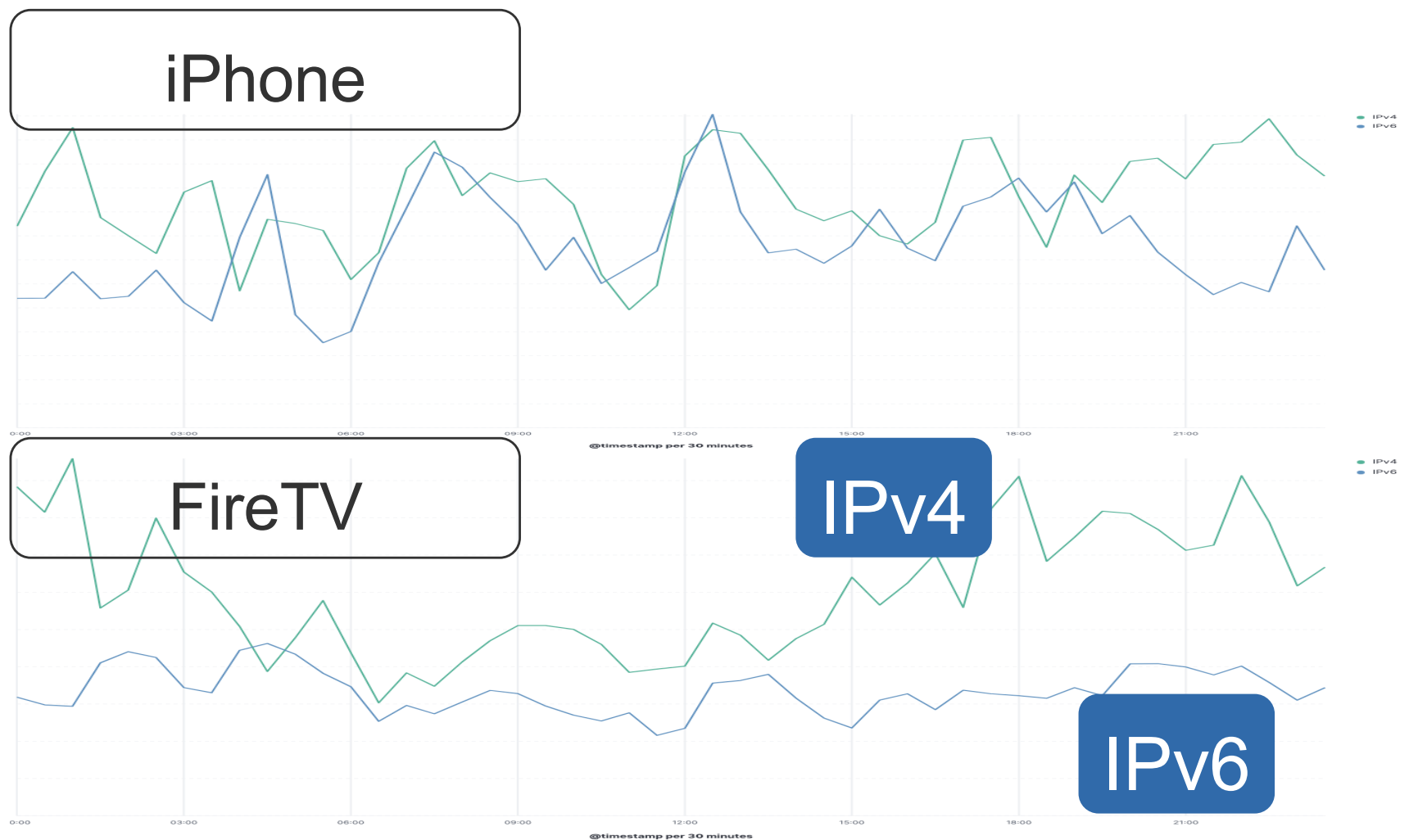
あるFQDNのデバイス別TTLB (1d)

- スマホやPCは速い・安定
- Chromecast, FireTV やTV系デバイスは遅い・不安定
 - HW的に貧弱なのでTLS復号処理が重い？



あるFQDNの使用回線別TTLB (1d)

- 主にモバイル回線を使用するiPhoneのIPv4/v6差は少なめ
- 宅内（固定回線）で使用するFireTVは顕著な差あり
 - PPPoE の網終端装置での混雑問題？



あるFQDNのISP別TTLB (1d)

- 某ISPで毎日同じ時間からTTLBが悪化
 - 帯域制御をかけている？
 - IPv6だと影響なし



IPv4

IPv6

当日限り

まとめ

- CDNを使用すると安定した大規模配信が可能となるだけでなく、様々な機能が利用でき、サービス冗長化にも寄与
- CDNの標準化が進んでいる

- CDN事業者視点でトラフィックを眺めると、世の中の動画視聴やISP設備の状況が垣間見えて興味深い
- IPv6の方が速そう（雑）
 - デバイス・ISPなど多要素が関連するので一様ではない