

迂回設計と輻輳回避

Matsuzaki 'maz' Yoshinobu

<maz@ij.ad.jp>

輻輳

- 流しそうめん
 - すかさか
- わんこそば
 - 帯域いっぱい
- 大皿料理
 - 輻輳



輻輳の困りどころ

- パケットロス
 - 食べ残し
- 遅延
 - 温かい料理が冷めちゃう

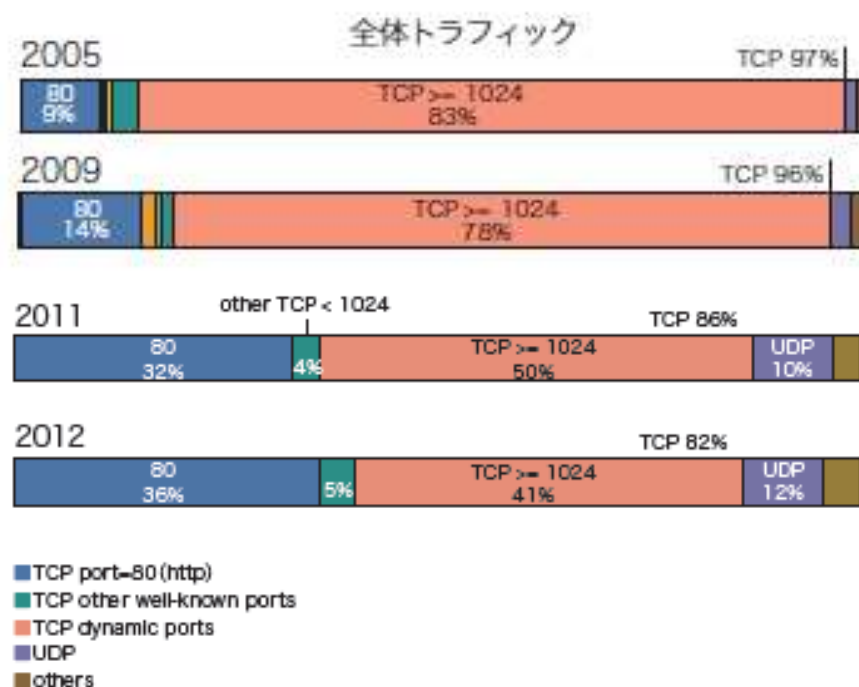
→早飯、完食が求められる

対策

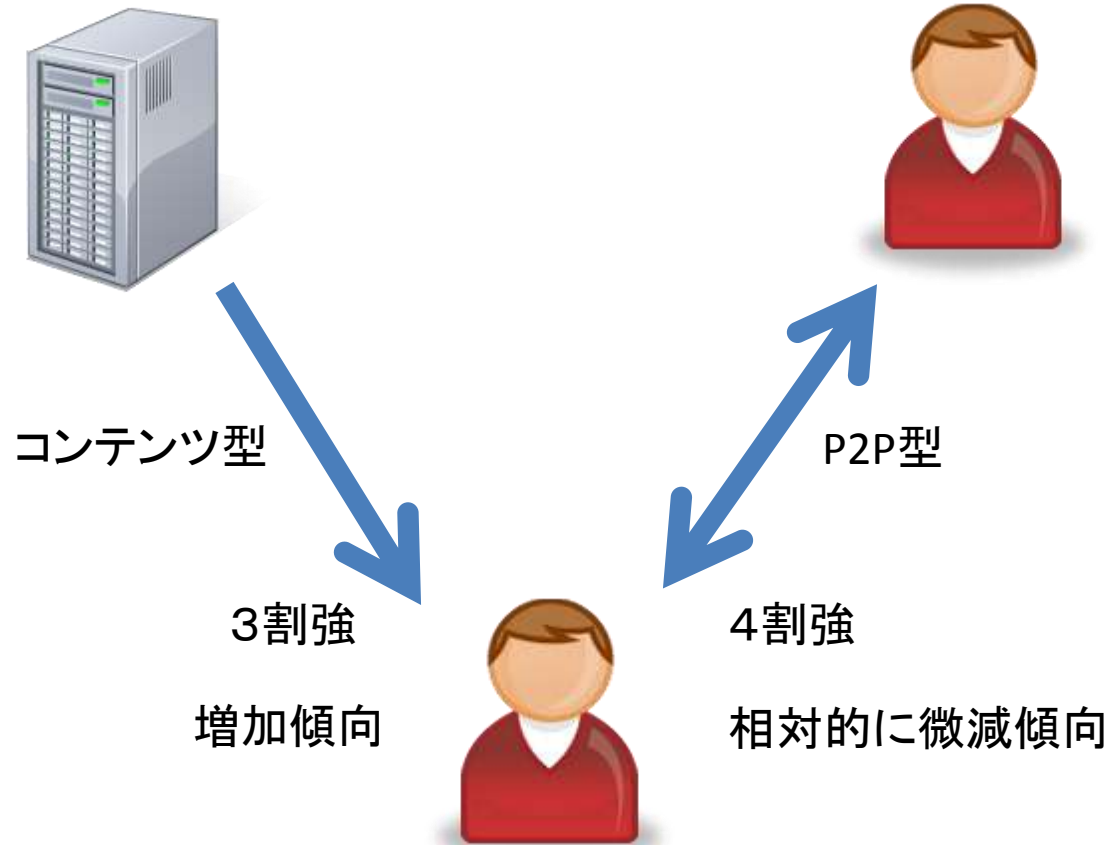
- 早く食べられるようにする
 - 回線を多く
 - 帯域を太く
- 万が一、誰かが倒れても大丈夫なように
 - 冗長設計
 - 迂回設計

ブロードバンドトラフィックの変遷

- ダウンロードが多い
- P2Pと思しき割合の減少
- tcp/80の増加
 - アップロードの少ないユーザで顕著



ブロードバンドトラヒック

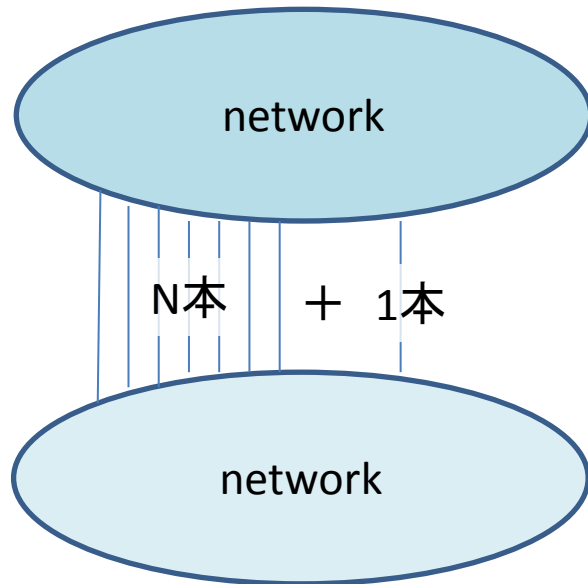


P2P型とコンテンツ型

- P2P型トラヒック
 - 同じアプリケーションのユーザ同志で通信
 - ユーザ数に比例するので読みやすい
 - AS毎に多重効果で傾向が出てくる
- コンテンツ型トラヒック
 - 配信元の制御で送信経路、送信元が変わる
 - 配信元がASをまたいで移動することもある
 - 受信先はだいたいコンシューマ

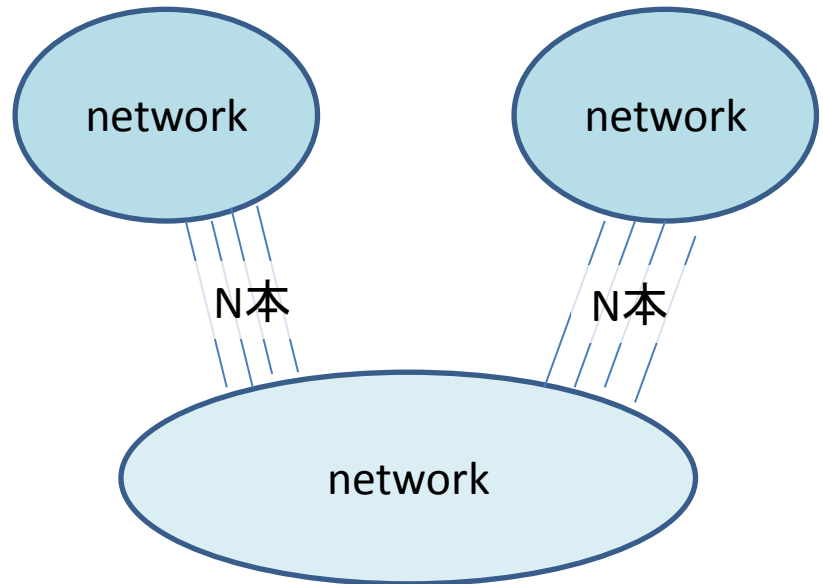
迂回設計

$N+1$



隣接ネットワークと冗長構成

$N*m$



異なるネットワークをまたいで
冗長構成

N+1 迂回設計が有効な範囲

- 迂回後も同じ様に流れようとするトラヒック
 - 送信元、受信先、総量が同じ
 - 途中経路に大きな変更が無い → 直接接続とか
- 細かな制御ができるトラヒック
 - 送出トラヒック

→ P2P型トラヒック、コンテンツ送信では
N+1設計が良く動く

コンテンツ受信向けの冗長構成

- コンテンツ配信元と頑張って帯域増強
 - 他に回らないように
 - 帯域増強しすぎるとトラフィックが回ってくることも
- コンテンツ流入経路を絞りこむ
 - 必ず上流ISPから流入するようにする
 - コンテンツが流入しそうなASとはピアしない

迂回設計は経路の絞り込み

- 絞れば絞るほど、効率的な冗長帯域
 - $N+1$
- 絞れなければ、流れそうな所に冗長帯域
 - $N*m$

迂回設計の考え方

- 効率重視
 - 積極的な迂回設計
 - トラヒックの流れを絞り込み、冗長帯域を最少化
- 影響重視
 - 消極的な迂回設計
 - 流れちゃ困るところを避ける
 - 迂回の影響範囲を限定的にする

冗長帯域どうしてますか？

- 専用帯域を確保
 - N+1
- 共用帯域を確保
 - IX経由でバックアップ
- 他のネットワークに迂回
 - 上流ISPやピア先
- 縮退運用



CDNキャッシュのバックアップは？

- CDNキャッシュなんて導入しない
 - 相互接続で頑張る
- CDNキャッシュ自体を頑強に
 - サーバの二重化
- CDNの配信元とも接続しておく
 - う回路の確保
- どこかにバックアップ
 - CDN屋さん任せ



見えないトラフィック対応は？

- トランジットを提供しているISPの悩み
 - 見えないけど、流れてくるかもしれないトラフィック
 - 送信元、受信先がネットワーク内に無い
- 他のネットワーク経由で交換されてる
- CDNキャッシュ
- 何か対応してますか？



輻輳対応は？

- それでも起きちゃうことはある
 - 予想外のトラヒック
 - 早すぎるトラヒック増加
- どんな対応してますか？
 - 短期的
 - 長期的



東京壊滅したらどうする？

- 東京での相互接続全滅
 - 通信できる？
 - 輻輳する？
 - トラヒックがなくなる？



考えどころ

- 最低限、到達性は確保したい
 - 相互接続ポイントの分散
 - 広報経路の生成
- いざって時に使える手段
 - IXの余剰帯域
 - サービス設備の分散

まとめ

- 障害等の影響をユーザに波及させないために迂回設計はとっても大事
- 輻輳を長期的に放置すると、良いことないよ
- 主要なトラフィック要因をきちんと観測しながら対応していこう！

