

フロー制御の性能解析における OR 理論を用いたアプローチ

小沢 利久

駒澤大学経営学部経営学科

〒 154-8525 世田谷区駒沢 1-23-1

E-mail: toshi@komazawa-u.ac.jp

概要 インターネットは既に最も身近なコミュニケーション手段となっているが、その中でパケットの流量を制御しているのが TCP (Transmission Control Protocol) のフロー制御である。この機能は、ネットワークの混雑を回避・緩和するとともに、利用者に対する公平なネットワーク資源の配分を目指している。TCP フロー制御を評価・分析するための方法は、解析的手法によるもの、コンピュータシミュレーションによるもの、実データによるものに大別される。本論文では、この中の解析的手法に関する幾つかのアプローチについて解説していく。

キーワード: コンピュータ通信ネットワーク, インターネット, TCP, 輻輳制御, フロー制御, 性能評価, 確率過程, 待ち行列モデル, 最適化問題

1. はじめに

インターネットに代表されるコンピュータ通信ネットワークでは、データをパケットと呼ばれる単位に分割して転送している [9]。パケットをアドレスに従って適切な方向に送りだしているのがルータであり、そのルータをリンク（通信回線）で結んでネットワークが構成されている。ルータやリンクは不特定多数のユーザから送信されたパケットによって利用される可能性がある。その為、あるルータやリンクに利用が集中し、転送すべきパケット数とそのルータやリンクの転送能力を超えると、ルータでのパケット転送待ち時間が増加したり、バッファが溢れてパケット廃棄が起きたりする。これが輻輳と呼ばれている現象である。

輻輳状態では、転送遅延の増加やパケット廃棄の発生によって通信サービスの質が低下し、さらには通信ネットワークの転送効率（単位時間当たり有効に転送されたデータの量）も低下する可能性がある。輻輳制御とは、このような輻輳の回避、緩和を狙った通信制御である。コンピュータ通信で用いられている輻輳制御の方法は、明示的または非明示的に得られた通信網の輻輳情報をもとに、送信パケット数を送り側の端末（コンピュータ）で調節するものである。このように、輻輳制御とはパケットの流れを制御するものであり、フロー制御とも呼ばれている。以下、それに合わせて輻輳制御とフロー制御を同じ意味で用いる。インターネットを初めとして、現在のコンピュータ通信の多くが TCP (Transmission Control Protocol) のフロー制御を輻輳制御のために用いている。

この解説文は、フロー制御の性能を理論的に解析する方法を、著者の知識の範囲でまとめたものである。フロー制御としては TCP フロー制御を中心に考えているが、一部、それには限定されない一般的な設定の下での解析方法も示す。以下、2 節では TCP フロー制御について簡単にまとめる。3 節以降では、3 つの節に分けて各種解析方法を説明する。この分類はモデルの中におけるネットワークの表現方法に従っている。3 節は、ネットワークをパケットの損失過程として表現した解析方法である。ユーザについては特定の 1 ユーザに注目し、そのユーザの挙動を TCP フロー制御のプロトコルに従ってできる限り忠実に表現している。パケットの発生過程は、一定間隔とランダムな場合を考える。4 節は、ネットワーク内のボトルネックリンクのみに注目し、それを待ち行列モデルとして表現する解析方法である。ユーザについてはそのボトルネックリンクを利用する複数のユーザを考え、ユーザ間での競合を考慮する。ただし、個々のユーザの挙動はかなり簡略化した表現となっている。ベースとなる手法に応じて、マルコフ連鎖によるモデル化、プロセッサシェアリングモデルによるモデル化、最適化問題としてのモデル化を考える。5 節では、ネットワークを容量を持ったリンクから構成されたグラフとして表現する解析方法である。パケットはデータの流れ（流体）として表現され、各リンクの容量を各ユーザに配分する最適資源配分問題が定義される。フロー制御は、その最適資源配分問題の最適解を求めるアルゴリズムとして与えられる。

2. TCP フロー制御

2.1. 基本動作

TCP フロー制御 [2, 5, 16] は、オクテット単位でつけたデータのシーケンス番号を基にスライディングウィンドウでデータ送信量を制御している。データ送信は MSS (maximum segment size) 単位で行なわれ、これがパケットに対応する。データの受信側はパケット受信毎に ACK (acknowledgement) を送信側に返す。ACK には次に受信を期待するデータのシーケンス番号が入っている。スライディングウィンドウ (以下、単にウィンドウ) の大きさ W を決めるパラメータには輻輳ウィンドウ (congestion window; $cwnd$) と受信側ウィンドウ (receiver; $rwnd$) があり、

$$W = \min\{cwnd, rwnd\}$$

により W の値が決定される。TCP のフロー制御は、輻輳ウィンドウ $cwnd$ を大きくしていき、パケット損失が検出されたら輻輳と判断して $cwnd$ を小さくする制御法である。

2.2. スロースタートフェイズと輻輳回避フェイズ

輻輳ウィンドウ $cwnd$ を大きくしていく仕方は 2 つのフェーズに分れる [2, 5]。ひとつは送信開始時など、 $cwnd$ の値が指数関数的に増加していくフェーズであり、スロースタート (slow start) と呼ばれている。もうひとつのフェーズは輻輳回避 (congestion avoidance) と呼ばれているもので、 $cwnd$ の値は線形的に増加する。両フェーズのどちらを取るかを決めているのがスロースタート閾値 (slow start threshold; $ssthresh$) である。具体的には次のようになっている。

$$\begin{aligned} cwnd \leq ssthresh \text{ (スロースタート)} &\Rightarrow cwnd \leftarrow cwnd + MSS \text{ (ACK 受信毎に)} \\ cwnd \geq ssthresh \text{ (輻輳回避)} &\Rightarrow cwnd \leftarrow cwnd + MSS \text{ (RTT 経過毎に)} \end{aligned}$$

$cwnd$ の初期値は IW (initial window) であり、 $IW \leq 2 \times MSS$ に設定される。 $ssthresh$ の初期値については特に指定はない。RTT (round trip time) とは、パケットを送信してその応答が返ってくるまでの時間である。スライディングウィンドウの原理からわかるように、 W が変化しなければ RTT の間に $\lfloor W/MSS \rfloor$ 個のパケットを送信できる。このように、RTT は TCP フロー制御を理解する上で非常に重要な役割を持つ。スロースタートフェイズでは ACK 受信毎に $cwnd$ が 1 増えるため、結果的に RTT 毎に $cwnd$ が 2 倍になる。輻輳回避フェイズにおける $cwnd$ の更新には

$$cwnd \leftarrow cwnd + \frac{MSS^2}{cwnd} \text{ (ACK 受信毎に)}$$

の式が用いられる場合もある。

2.3. タイムアウト再送

パケット損失を検出する方法には幾つかある [2, 5]。最も基本的な方法は、タイムアウト再送 (timeout retransmit) と呼ばれるものである。これは、パケット送信毎にタイマーをセットし、そのタイマーが時間切れになるまでに受信確認ができなかったらそのパケットは損失したとみなす。タイマーの時間は RTT の測定値をもとにして設定されている。タイムアウトになると、

$$\begin{aligned} ssthresh &\leftarrow \max \left\{ \frac{FrighSize}{2}, 2 \times MSS \right\} \\ cwnd &\leftarrow LW \end{aligned}$$

と設定され、スロースタートフェイズで送信が再開される。 LW (loss window) は通常 MSS に設定される。 $FrighSize$ とは、送信はしたが受信確認はされていないパケットに対応するデータサイズである。送信の再開ではまず初めに損失パケットが再送される。もし、その再送パケットがタイムアウトになると、タイマーの時間を r 倍 (通常は $r = 2$) にし、以下、再送が成功するまでこれを繰り返す Karn のアルゴリズムが使われる。

2.4. 早期再送と早期回復

もうひとつのパケット損失検出方法は早期再送 (fast retransmit) と呼ばれている方法である [2, 5]。ACK には次に受信を期待するオクテットのシーケンス番号が入っているが、送信途中にあるパケットが損失すると、それ以後に受信されたどのパケットに対する ACK にも損失したパケットを表すシーケンス番号が入ることになる。このような ACK を重複 ACK (duplicated ACK) という。重複 ACK はパケットの受信順序が入れ替わった時にも起きる可能性がある。そこで、順序入れ替えとパケット損失を区別するため、通常、3 つの重複

ACK を受信したらパケット損失と判断する方法をとっている。これが早期再送によるパケット損失の検出方法であり、パケット損失検出と同時に

$$ssthresh \leftarrow \max \left\{ \frac{FlightSize}{2}, 2 \times MSS \right\}$$

に設定する。この後の送信再開の方法には、 $cwnd \leftarrow LW$ としてスロースタートから始める方法 (TCP Tahoe が採用) と次の早期回復 (fast recovery, TCP Reno が採用) がある [2, 5].

重複 ACK はパケット損失後でもパケット送信が成功したことを示しており、輻輳の程度はそれほどひどくないと考えられる。そこで、損失パケットを再送し、その再送が成功したら (正確には、新たなシーケンス番号の入った ACK を受信したら),

$$cwnd \leftarrow ssthresh$$

として輻輳回避フェイズから送信を再開するのが早期回復である。(実際の制御はもう少し複雑であり、未確認パケット数から重複 ACK 数を引いた値を網内に滞留しているパケット数と考え、それが $ssthresh$ 以下であれば新たなパケットを送信するといった処理を行っている。)

2.5. AQM と ECN

上記で示したことは、ネットワークを構成するルータはフロー制御に対して何ら積極的なかわりを持たないということを前提としている。単にバッファから溢れたパケットを廃棄するだけという前提である。これに対し、TCP フロー制御の効率化のためにルータが積極的にかかわるための仕組みが AQM (active queue management) と ECN (explicit congestion notification) である [4].

ECN とは、パケットヘッダにある CE ビット (congestion experimented bit) を用いて、ルータがパケット送信端末に輻輳情報を送るための方法を規定したものである。輻輳情報はそのルータを通過したパケットに書き込まれ、そのパケットを受信した端末から ACK に乗せられて送信側端末に送られる。輻輳情報を受け取った端末は、現段階では、パケット損失検出時と同じ動作をすることとしている。

AQM とは、バッファでのキュー長をもとに、受信端末に向けた輻輳情報を生成する仕組みである。例えば、閾値を設け、キュー長がそれを超えたら輻輳情報を生成するといった方法があげられる。輻輳情報の伝達は ECN による場合もあるし、従来通りパケットを廃棄して間接的に伝えることもある。代表的な例は RED (random early detection) である [4, 11]. RED では、平均キュー長の推定値に依存した確率に従ってランダムにパケットを選び、そのパケットに輻輳情報を付与する (そのパケットを廃棄する) ものである。

2.6. その他

以上に示した他にも様々な方法が標準化あるいは提案されている。以下にその一部を示す。

Delayed ACK これは ACK によるトラヒックを減らすなどの目的で、複数の受信パケットに対して一つの ACK を返すものである [1]. 通常は 2 つのパケットに対してひとつの ACK を返す。重複 ACK に対しては適用されない。

SACK option 再送を効率的に行えるよう、受信側で既に受信済みのデータブロックに関する情報を ACK に乗せて送信側に送るための方法を規定したもの [3].

NewReno 早期再送・早期回復では、RTT の間に複数のパケットが損失になると早期再送を複数回繰り返すことで $ssthresh$ の大きさが必要以上に小さくなる可能性がある。それを改善するために提案されたのが NewReno という方法であり、不必要に $ssthresh$ を小さくしないための仕組みが取り入れられている。

注意 TCP フロー制御において、ウィンドウサイズの単位は 1 オクテットである。しかし、以下の説明では理解を容易にするために 1 パケットを単位として考える。 W/MSS を W で置き換えたと考えればよい。

3. 特定ユーザに注目した解析方法

3.1. 基本的考え方

TCP フロー制御におけるユーザとネットワークの相互依存関係は、ユーザからネットワークへのパケット受け渡しとネットワークからユーザへのパケット損失情報の受け渡しとして捉える事ができる。パケット損失情報は重複 ACK やタイムアウトによりもたらされる情報である。この章で考えるモデルは、パケットとパケット損失情報の受け渡し点で特定ユーザとネットワークを切り離し、そのユーザの挙動のみを極力忠実にモデル化しようとする。ネットワークはそのユーザとは独立に存在し、ユーザに対してパケットが損失したか否かのみを情報として与えるものとしてモデル化される。

性能評価指標はスループットを考える。ここでスループットとは、単位時間当りに送信できる最大パケット数の長時間平均である。すなわち、 $M(t)$ を時間区間 $[0, t]$ での送信パケット数（それが受信側に届いたかは問わない）として、スループット \bar{X} は

$$\bar{X} = \lim_{t \rightarrow \infty} \frac{M(t)}{t} \quad (1)$$

で与えられる。スループットを考えている事から、データの発生に対しては次の仮定をおく。

(仮定 1-1) 常に送信すべきデータがあり、パケットとして送信可能であれば即座に送信される。

TCP フロー制御については次の仮定をおく。

(仮定 1-2) スロースタートフェイズは無視する。すなわち、タイムアウトが起きても、パケット送信の再開は常に $cwnd = ssthresh$ の輻輳回避フェイズからであるとする。

この仮定はモデル化を容易にするためであるが、スロースタートフェイズではウィンドウサイズが指数関数的に増加することから比較的短時間で輻輳回避フェイズに移行するという理由による。輻輳回避フェイズでのパケット送信再開からパケット損失検出後の次の輻輳回避フェイズ再開までの間を PWC (packet window cycle) と呼ぶ事にする。一つの PWC はちょうど一つの輻輳回避フェイズを含み、複数の早期再送や早期回復、複数のタイムアウト待ちを含む可能性を持つ。 n 番目の PWC ($n \geq 1$) に対して次の記号を定義する。

- T_n : PWC の開始時点 ($T_1 = 0$)
- U_n : PWC の期間長
- Y_n : PWC 中に送信したパケット数
- \hat{W}_n : PWC で初めて損失となったパケットを送信した時のウィンドウサイズ (単位はパケット数)

次に、時間軸を RTT (round trip time) 毎に区切り、各区間を RTTP (round trip time priod) と呼ぶ事にする。1 番目の RTTP 開始時点を時点 0、 n 番目の RTTP 開始時点 ($n \geq 1$) におけるウィンドウサイズ (単位はパケット数) を W_n とする。仮定 1-1, 1-2 を考慮し、次の仮定をおく。

(仮定 1-3) n 番目の RTTP ではちょうど $\lfloor W_n \rfloor$ 個のパケットが送信される。

この仮定は、ウィンドウサイズ分のパケット送信時間と比較して RTT が十分長い場合を想定している。仮定 1-3 より、時点 0 から n 番目の RTTP 終了時点までに送信されたパケット数を M_n とすると、

$$M_n = \sum_{k=1}^n W_k, \quad n \geq 1, \quad (2)$$

となる。また、RTT 長が一定値 rtt であるとし、パケットを流体と考え、時点 t でのウィンドウサイズ $W(t)$ を各 RTTP でのウィンドウサイズ W_n を結んで得られる曲線で表現すれば、

$$M(t) \approx \int_{s=0}^t \frac{W(s)}{rtt} ds, \quad (3)$$

という近似式を得る。よって、どちらの表現を用いるにせよ、ウィンドウサイズの解析によってスループットが得られる。以下では、この点を基に、パケット損失の発生が周期的な場合とランダムな場合についてスループットの導出法を述べていく。

3.2. 周期的にパケット損失が起きるモデル

仮定 1-1, 1-2, 1-3 を前提とする。ただし、パケットは流体と考え、表現は式 (3) を用いる。さらに次の仮定をおく。

(仮定 2-1) パケット損失率を p として、パケット損失は $1/p$ パケット送信した毎に起きる。

(仮定 2-2) パケット損失は常に早期再送によって検出される。パケット損失検出後は早期回復が成功し、ただちに輻輳回避フェイズに入る。ただし、早期再送、早期回復の期間は解析では無視する。

(仮定 2-3) Delayed ACK を考慮し、 b パケット毎に一つの ACK が返信されるとする。パケットを流体と考え、仮定 1-2, 2-2 より、 $W(t)$ は常に傾き $1/(brtt)$ で増加するものとする。ただし、 rtt は RTT の平均である。

(仮定 2-4) 受信側ウィンドウサイズは無視する。

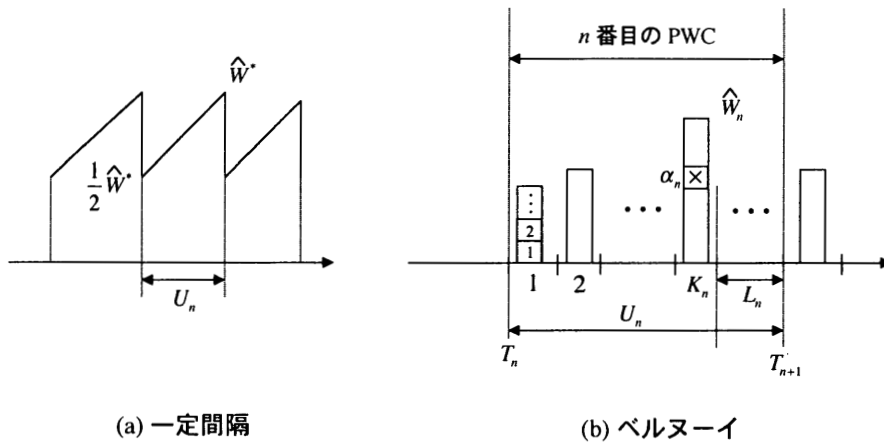


図 1: ウィンドウサイズの変化

ここで $1/p$ はパケット損失がパラメータ p のベルヌーイ過程に従って起きるとした場合の平均パケット損失間隔 (パケット数) である. 以上の仮定より, 十分時間が経過した後は, \hat{W}_n がある一定値に近付くと考えられる (図 1(a) 参照). その値を \hat{W}^* とすると, PWC はウィンドウサイズ $\hat{W}^*/2$ で始まって \hat{W}^* で終わる事から次の関係式が得られる.

$$U_n = \frac{\hat{W}^*}{2} brtt$$

$$Y_n = \frac{1}{2} \left(\frac{\hat{W}^*}{2} + \hat{W}^* \right) U_n$$

仮定 2-1 より, $Y_n = 1/p$ であるから

$$\bar{X} = \frac{Y_n}{U_n} = \frac{1}{rtt} \sqrt{\frac{3}{2bp}} \quad (4)$$

が得られる [26]. この式は, 非常に単純化されたモデルから導出されたものではあるが, 次節で見ると, タイムアウトが起きないような状況ではより一般的な条件から導出された式とほぼ同じ値を与える.

この式 (4) から次の事が導かれる.

- スループットは平均 RTT に反比例する.
- スループットはパケット損失率の平方根に反比例する.

また, 式 (4) はパケット損失率とスループットの関係を与えている事から次のような利用法が考えられる.

- ユーザの立場から: 何らかの方法で rtt と p の推定値が得られれば, あるサイズのデータを送信するのにかかる時間の期待値を見積もる事ができる.
- ネットワーク設計の立場から: rtt として何らかの代表値を与える事で, スループットの設計目標値をパケット損失率の設計目標値に変える事ができる.

3.3. パケット損失がランダムに起きるモデル

3.3.1. モデル化と解析方法

仮定 1-1, 1-2, 1-3 を前提とし, 次の仮定の下, より精密なモデル化と解析を試みる (基本となる考え方は文献 [26] による).

(仮定 3-1) パケット損失率を p とし, パケット損失率はパラメータ p のベルヌーイ過程に従って起こる.

パケットを流体として近似する事は行わず, PWC の開始時点はある RTTP の開始時点と一致するものとする. n 番目の PWC ($n \geq 1$) に対して次の記号を定義する (図 1(b) 参照).

- $\hat{W}_n^{(0)}$: PWC 開始時のウィンドウサイズ
- K_n : PWC において初めてパケット損失がおきた RTTP の番号 (PWC の最初の RTTP を番号 1 とする.)
- α_n : PWC において初めて損失となったパケットの番号 (PWC の最初のパケットを番号 1 とする.)

- r_{nj} : PWC における j 番目の RTTP の期間長
 - γ_n : PWC において、早期再送・早期回復が成功すれば値 0、タイムアウトになれば値 1 をとる確率変数
 - L_n : PWC において、 K_n 番目の RTTP 終了から次の PWC 開始までの期間長
 - β_n : PWC において、 K_n 番目の RTTP 終了から次の PWC 開始までに送信したパケット数
- さらに次の仮定をおく。

(仮定 3-2) RTTP の期間長は互いに独立で同一の分布に従い、他の事象とも独立である。 $r_{tt} = E(r_{nj})$ とする。

(仮定 3-3) γ_n は \hat{W}_n には依存してもよいが、他の事象とは独立で斉時的である。

(仮定 3-4) L_n, β_n は $\hat{W}_{n-1}, \hat{W}_n, K_n, \alpha_n, \gamma_n$ には依存してもよいが、他の事象とは独立で斉時的である。

(仮定 3-5) $\hat{W}_{n+1}^{(0)} = \max \left\{ 2, \lfloor \frac{\hat{W}_n}{2} \rfloor \right\}$

(仮定 3-6) 受信側で受信された全てのパケットに対して ACK が返信される。

仮定 3-5 は、 W_n が常に整数値を取るようにし、仮定 1-1 と $ssthresh$ の更新アルゴリズムを反映させたものである。仮定 3-6 は説明を簡単にするための仮定であり、本質的ではない。 $n \geq 1$ に対して次の関係式が成り立つ。

$$\hat{W}_n = \min\{rwnd, \hat{W}_n^{(0)} + K_i - 1\} \quad (5)$$

$$U_n = \sum_{j=1}^{K_i} r_{nj} + L_n \quad (6)$$

$$Y_n = \sum_{j=1}^{K_i} \min\{rwnd, \hat{W}_n^{(0)} + j - 1\} \quad (7)$$

ただし、 $rwnd$ は受信側ウィンドウサイズ、 \hat{W}_0 は $\{\hat{W}_n\}$ の初期値とする。

仮定 1-2, 3-6 より、 K_n 番目の RTTP までは RTTP 毎にウィンドウサイズが 1 ずつ増加する。この事と仮定 3-1, 3-5 より K_n の分布は \hat{W}_{n-1} のみに依存する形で与える事ができる。よって、式 (5) より $\{\hat{W}_n\}$ はマルコフ連鎖となる。このマルコフ連鎖は既約で正再帰的であるとし、状態空間を S 、推移確率行列を $\hat{P} = (\hat{p}_{kl})$ 、定常分布を $\hat{\pi} = (\hat{\pi}_k)$ とする。同様な考察により、 U_n と Y_n の分布は \hat{W}_{n-1} と \hat{W}_n にのみ依存する形で与える事ができる。そこで、連続時間確率過程 $\{Z(t)\}$ と $\{R(t)\}$ を

$$N(t) = \sup\{n - 1 | T_n \leq t, n \geq 1\}$$

$$Z(t) = \hat{W}_{N(t)}$$

$$R(t) = \sum_{n=1}^{N(t)} Y_n$$

と定義すると、 $\{Z(t)\}$ はセミマルコフ過程、 $\{R(t)\}$ はその上で定義された報酬過程となる。 $\tau_k = E(U_n | \hat{W}_{n-1} = k)$ 、 $r_k = E(Y_n | \hat{W}_n = k)$ とし、 $Y_n \geq 0$ を考慮する事で次の定理が得られる。

Theorem 1 $\sum_{k \in S} \hat{\pi}_k \tau_k < \infty$, $\sum_{k \in S} \hat{\pi}_k r_k < \infty$ と仮定すると

$$\lim_{t \rightarrow \infty} \frac{R(t)}{t} = \frac{\sum_{k \in S} \hat{\pi}_k r_k}{\sum_{k \in S} \hat{\pi}_k \tau_k} \quad (8)$$

が成り立つ。 □

この定理より、 $R(t) \leq M(t) < R(t) + Y_{N(t)+1}$ であることから、

$$\bar{X} = \lim_{t \rightarrow \infty} \frac{M(t)}{t} = \frac{\sum_{k \in S} \hat{\pi}_k r_k}{\sum_{k \in S} \hat{\pi}_k \tau_k} \quad (9)$$

となる。よって、仮定を満たす範囲で TCP フロー制御のアルゴリズムをモデル化すれば式 (9) よりスループットが得られる。次に簡単な例を示す。

3.3.2. 解析例

タイムアウトを考慮した簡単なモデルを考える。そのため、以下の記号と仮定を導入する。

- δ_n : n 番目の PWC の K_n 番目の RTTP において損失したパケット数 ($1 \leq \delta_n \leq \hat{W}_n$)
- C_n : n 番目の PWC における再送タイマーの設定時間

(仮定 3-7) $\{C_n\}$ は独立で同一の分布に従い、他の事象とも独立とする。

(仮定 3-8) (a) $\hat{W}_n \leq 3 \Rightarrow \gamma_n = 1, L_n = C_n, Y_n = \alpha_n + \hat{W}_n - 1$

(b) $\hat{W}_n \leq 4, \delta_n = 1 \Rightarrow \gamma_n = 0, L_n = r_{n, K_n+1}, Y_n = \alpha_n + \hat{W}_n$

(c) $\hat{W}_n \leq 4, \delta_n \geq 2 \Rightarrow \gamma_n = 1, L_n = r_{n, K_n+1} + C_n, Y_n = \alpha_n + \hat{W}_n$

仮定 3-8 (a) は、 $\hat{W}_n \leq 3$ なので重複 ACK が 2 個以下となり、早期再送にはならない場合である。1 回のタイムアウト後に必ず次の PWC に入るとした。パケットは、最初に損失となったパケット以降、 $\hat{W}_n - 1$ だけ送信されるとした。(b) は RTTP 内での損失パケット数が 1 の場合で、この場合は必ず早期再送・早期回復が成功するとした。早期再送のため 1 RTTP だけ間をあけて次の PWC に入る。早期再送を行なうので最初に損失となったパケット以降送信したパケット数を \hat{W}_n とした。(c) は複数のパケットが同じ RTTP 内で損失したため、早期再送は行なったが、早期回復が失敗し、タイムアウトになった場合に対応する。

解析を見やすくするため、マルコフ連鎖 $\{\hat{W}_n\}$ の初期分布を定常分布 $\hat{\pi}$ にとった定常過程を考える。この時、 $E(U_1) = \sum_{k \in S} \hat{\pi}_k r_k$, $E(Y_1) = \sum_{k \in S} \hat{\pi}_k r_k$ より、

$$\bar{X} = \frac{E(Y_1)}{E(U_1)}$$

となる。また、式 (6) と仮定 3-8 より、

$$Y_n = \alpha + \hat{W}_n - 1(\hat{W}_n \leq 3),$$

$$U_n = \sum_{j=1}^{K_n} r_{nj} + \gamma_n C_n + 1(\hat{W}_n \geq 4)r_{n, K_n+1}.$$

ここで、 $1(A)$ は A が真の時に値 1、偽の時に 0 をとる関数 (指示関数; index function) である。これより次が得られる。

$$E(Y_1) = \frac{1}{p} + E(\hat{W}_0) - \Pr(\hat{W}_n \leq 3)$$

$$E(U_1) = (E(K_1) + \Pr(\hat{W}_n \geq 4))rtt + E(\gamma_1)E(C_1)$$

ただし、 $E(\gamma_1) = \sum_{k \in S} \hat{\pi}_k \frac{kp(1-p)^{k-1}}{1-(1-p)^k}$ である。説明は省略するが、このモデルでは直接 $\hat{\pi}$ を計算するよりも、RTTP 毎のウィンドウサイズ過程 $\{W_n\}$ の定常分布 π を求め、その結果を用いて $\hat{\pi}$, $E(\hat{W}_0)$, $E(K_1)$ を求める方が容易である。図 2 は、常に $\gamma_n = 0$ とした場合 (常に早期再送と早期回復が成功するとした場合) について、 p の値を振らせた時のスループットである。比較の為に、式 (4) で求めた値も示す。この図より、式 (4) で得られた結果は p が十分小さければ非常に精度のよいものである事がわかる。

ここでは仮定 3-1 を前提に、パケット損失がベルヌーイ過程に従って起きるとした。しかし、セミマルコフ過程としての定式化からもわかるように、パケット損失がもっと一般の過程に従うとした定式化も可能である (例えば、文献 [7] などを参照)。

4. ボトルネックリンクを待ち行列として表したモデル

4.1. 基本的考え方

TCP フロー制御はパケット損失を契機としてウィンドウサイズを減少させる方法であった。パケット損失の原因は、伝送誤りやルータバッファでのパケット廃棄など様々なものが想定される。3 節ではパケット損失の要因を特に限定しなかったが、ここではルータバッファでのパケット破棄のみに注目する。これにより、ネットワークは、各リンクをサーバとする待ち行列ネットワークとして捉える事ができる。しかし、複数の待ち行列の挙動を考慮した解析は困難である事から次の仮定を入れる。

(仮定 1-1) パケットを送受信しているユーザ間にはひとつのボトルネックリンクが存在し、パケット損失はそのリンクのバッファでのみ起こる。

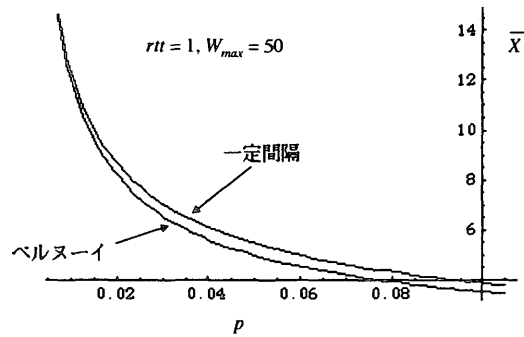


図 2: 常に早期再送と早期回復が成功するとした場合スループット ($rtt = 1$)

この仮定の下、以下ではマルコフ連鎖によるモデル化、プロセッサシェアリングモデルによるモデル化、最適化問題としてのモデル化を示す。ボトルネックリンクのみではなく、ネットワーク全体に注目した解析は 5 節で示す。

4.2. マルコフ連鎖としてのモデル化

4.2.1. モデル化

3 節で述べたように、スループットは RTTP 毎のウィンドウサイズの挙動を解析する事で得られる。そこで、ここではウィンドウサイズの挙動をマルコフ連鎖で表し、それを解析する事でユーザ間の競合の様子や、有限の時間スケールで見た場合のスループットの変動について見ていく [28]。仮定 1-1 の他に、まずは次の仮定をおく。

(仮定 2-1) ボトルネックリンクを通過する TCP コネクションの数を固定値 m とする。

(仮定 2-2) 全ての TCP コネクションは同じ RTT を持つものとし、その値を固定値 rtt とする。

TCP コネクション数が変動する場合については 4.3 で扱う。仮定 2-2 はモデル化を容易にする上で本質的な仮定である。3 節での解析結果からわかるように、この仮定はスループットが RTT に反比例する点を無視している。ただし、RTT を固定値にすることはやはり 3 節の結果からわかるように、スループットの期待値を考える上では本質的ではない。時間軸を RTT 毎に区切り、1 番目の RTTP 開始時点を時点 0、 n 番目の RTTP 開始時点 ($n \geq 1$) におけるコネクション i ($i = 1, 2, \dots, m$) のウィンドウサイズ (単位はパケット数) を $W_i(n)$ とする。以下では、TCP コネクション i とユーザ i を同一視する。3 節と同様に次の仮定をおく。

(仮定 2-3) 各ユーザは常に送信すべきデータがあり、パケットとして送信可能であれば即座に送信される。

(仮定 2-4) パケット損失は常に早期再送によって検出される。パケット損失検出後は早期回復が成功し、ただちに $cwnd = ssthresh$ の輻輳回避フェイズに入る。ただし、早期再送、早期回復の期間は解析では無視する。

(仮定 2-5) 各コネクションは共通の受信側ウィンドウサイズ $rwnd$ を持つ。

(仮定 2-6) 受信側で受信された全てのパケットに対して ACK が返信される。

(仮定 2-7) コネクション i では、 n 番目の RTTP でちょうど $\lfloor W_i(n) \rfloor$ 個のパケットが送信される。

$S = \{2, 3, \dots, rwnd\}$ を $\{W_i(n)\}$ の状態空間とする。仮定により、 $W_i(n)$, $n \geq 0$, $i = 1, 2, \dots, m$ の変化は次で与えられる。

$$W_i(n+1) = \begin{cases} \min\{W_i(n) + 1, rwnd\} & n \text{ 番目の RTTP でコネクション } i \text{ のパケット損失なし} \\ \max\{2, \lfloor \frac{W_i(n)}{2} \rfloor\} & n \text{ 番目の RTTP でコネクション } i \text{ のパケットが損失} \end{cases}$$

ボトルネックリンクは次のようにモデル化する。

(仮定 2-8) ボトルネックリンクにおいて、RTTP 当りに転送可能なパケット数を c とし、 n 番目の RTTP で送信されたパケットの内、最大 c 個が転送され、それ以外は廃棄される。廃棄されるパケットはルータへの到着順に関係がなくランダムに選ばれる。

仮定 2-8 は、バッファでの送信待ちパケット数を状態として持たなくて済むようにすることでモデルを簡単にするための仮定である。以上の仮定により、 $\mathbf{W}(n) = (W_1(n), \dots, W_m(n))$ はマルコフ連鎖となる。さらに、各 i に対して状態の集合を次のように定義する。

$$\bar{k}(i) \equiv \{(k, j_{i+1}, \dots, j_m) \mid j_{i+\ell} \in S, \ell = 1, 2, \dots, m-i\}, k \in S$$

推移確率行列 \mathbf{P} を $W_1(n)$ に注目してブロック化すると $\mathbf{P} = (\mathbf{P}_{\bar{k}(1), \bar{\ell}(1)})$ は以下のように表される。

$$\mathbf{P}_{\bar{k}(1), \bar{\ell}(1)} = \begin{cases} \mathbf{B}_k(1), & \ell = \max\{2, \lfloor \frac{k}{2} \rfloor\}, k \in S \\ \mathbf{A}_k(1), & \ell = \min\{rwnd, k+1\}, k \in S \\ \mathbf{O}, & \text{otherwise} \end{cases} \quad (10)$$

ここで、各ブロックの次数は $rwnd^{m-1}$ であり、 \mathbf{O} は要素が全てゼロの行列を表す。 $\mathbf{B}_k(1)$ はパケット損失によりウィンドウサイズが $\max\{2, \lfloor \frac{k}{2} \rfloor\}$ に減少する場合を、 $\mathbf{A}_k(1)$ は輻輳回避フェイズによりウィンドウサイズがひとつ増加する場合を表している。 \mathbf{P} は block-Hessenberg 型であり、かつ、各行に高々 2 つの非ゼロブロックしか現われない。 \mathbf{P} の構成から類推できるよに、 $\mathbf{B}_k(1), \mathbf{A}_k(1)$ も \mathbf{P} と同様のブロック構造を持ち、それらブロックもまた同様なブロック構造を持つ。よって、 $\{W_1(n), \dots, W_{i-1}(n)\}$ が与えられた時の、 $W_i(n)$ に注目した推移確率行列をレベル i の推移確率行列と呼ぶことにすると、その表現は次で与えられる。

$$\begin{aligned} \mathbf{A}_j(i; \alpha_1, \dots, \alpha_{i-1}) &= \left(\mathbf{A}_{\bar{k}(i), \bar{\ell}(i)}^{(j)}(i; \alpha_1, \dots, \alpha_{i-1}) \right) \\ \mathbf{B}_j(i; \alpha_1, \dots, \alpha_{i-1}) &= \left(\mathbf{B}_{\bar{k}(i), \bar{\ell}(i)}^{(j)}(i; \alpha_1, \dots, \alpha_{i-1}) \right) \\ \mathbf{A}_{\bar{k}(i), \bar{\ell}(i)}^{(j)}(i; \alpha_1, \dots, \alpha_{i-1}) &= \begin{cases} \mathbf{B}_k(i+1; \alpha_1, \dots, \alpha_{i-1}, A_j), & \ell = \max\{2, \lfloor \frac{k}{2} \rfloor\}, k \in S \\ \mathbf{A}_k(i+1; \alpha_1, \dots, \alpha_{i-1}, A_j), & \ell = \min\{rwnd, k+1\}, k \in S \\ \mathbf{O}, & \text{otherwise} \end{cases} \quad (11) \end{aligned}$$

$$\mathbf{B}_{\bar{k}(i), \bar{\ell}(i)}^{(j)}(i; \alpha_1, \dots, \alpha_{i-1}) = \begin{cases} \mathbf{B}_k(i+1; \alpha_1, \dots, \alpha_{i-1}, B_j), & \ell = \max\{2, \lfloor \frac{k}{2} \rfloor\}, k \in S \\ \mathbf{A}_k(i+1; \alpha_1, \dots, \alpha_{i-1}, B_j), & \ell = \min\{rwnd, k+1\}, k \in S \\ \mathbf{O}, & \text{otherwise} \end{cases} \quad (12)$$

ここで、 α_i はレベル i で A タイプ（注目するコネクションのウィンドウサイズが輻輳回避フェイズにより増加するタイプ）と B タイプ（注目するコネクションのウィンドウサイズがパケット損失により減少するタイプ）のどちらのブロックに属し、かつ、そのブロックがどの行にあったのか（ j 行であれば注目するコネクションが j 個のパケットを送信）を表す。例えば、レベル i で j 行目の A タイプのブロックに属していた場合、 $\alpha_i = A_j$ となる。ところで、レベル m での各ブロックはスカラーとなるので、

$$\begin{aligned} \gamma(\alpha_1, \dots, \alpha_{m-1}, A_j) &\equiv A_j(m; \alpha_1, \dots, \alpha_{m-1}), j \in S, \\ \gamma(\alpha_1, \dots, \alpha_{m-1}, B_j) &\equiv B_j(m; \alpha_1, \dots, \alpha_{m-1}), j \in S, \end{aligned}$$

とおく。この $\gamma(\alpha_1, \alpha_2, \dots, \alpha_m)$ は、各コネクションが送信したパケット数とそのパケットが損失したかどうかを指定した時に、その事象が起きる確率を表す。

以下では、マルコフ連鎖 $\{\mathbf{W}(n)\}$ は既約で非周期的であるとし、 $\mathbf{W}(n)$ の分布を $\mathbf{p}(n)$ 、定常分布を $\boldsymbol{\pi} = (\pi_{i_1, \dots, i_m})$ とする。 $\mathbf{p}(1) = \boldsymbol{\pi}$ にとつた定常過程を $\{\tilde{\mathbf{W}}(n)\} = \{(\tilde{W}_1(n), \dots, \tilde{W}_m(n))\}$ とする。

4.2.2. スループット

ここでは全て定常過程で考える。仮定 2-7 より、時点 0 から n 番目の RTTP 終了時点までにコネクション i が送信したパケット数を $\tilde{M}_i(n)$ 、その間におけるスループットを $\tilde{X}_i(n)$ とすると、

$$\tilde{M}_i(n) = \sum_{k=1}^n \tilde{W}_i(k), \quad n \geq 1, \quad (13)$$

$$\tilde{X}_i(n) = \frac{1}{rtt} \frac{\tilde{M}_i(n)}{n} \quad (14)$$

となる（各コネクションの特性が同じである事からこの値はコネクションに依存しない）。また、長時間平均としてのスループット \bar{X} と長時間平均としてのリンク使用率 $\bar{\rho}$ は次で与えられる。

$$\bar{X} = \frac{E(\tilde{W}_1(0))}{rtt} \quad (15)$$

$$\bar{\rho} = \sum_{i_1 \in S} \dots \sum_{i_m \in S} \min\{c, \sum_{j=1}^m i_j\} \pi_{i_1, \dots, i_m} \quad (16)$$

ところで、コネクション 1 に注目すると、 $\tilde{M}_1(n)$ は $\tilde{W}_1(j)$ を時点 j での到着客数とみなした場合の区間 $[1, n]$ での総到着客数であることから、離散時間 BMAP の結果 [27] を応用してその分布を計算することができる。まず、 $n \geq 0, k \geq 0$ に対して以下の記号を定義する。ただし、 $\tilde{M}_1(0), \tilde{W}(0)$ は計算のために設定したものであり、 $\tilde{W}(0)$ の分布は定常分布 π であるとする。

$$\begin{aligned} V(n, k) &= (V_{\bar{i}(1), \bar{j}(1)}(n, k)) \\ V_{\bar{i}(1), \bar{j}(1)}(n, k) &= \Pr(\tilde{M}_1(n) = k, \tilde{W}(n) = \bar{j}(1) \mid \tilde{M}_1(0) = 0, \tilde{W}(0) = \bar{i}(1)) \\ C_i &= (C_{\bar{j}(1), \bar{k}(1)}^{(i)}), \quad i \in S \\ C_{\bar{j}(1), \bar{k}(1)}^{(i)} &= \begin{cases} B_j(1), & j = i, k = \max\{2, \lfloor \frac{i}{2} \rfloor\} \\ A_j(1), & j = i, k = \min\{rwnd, i + 1\} \\ O, & \text{otherwise} \end{cases} \end{aligned}$$

$V(k, n)$ は次の手順で計算される。

Step 1: $V(0, 0) = I$

Step 2: For $n \geq 1$ and $n \leq i \leq nrwnd$, $V(n, i) = \sum_{j=\max\{1, i-(n-1)rwnd\}}^{\min\{rwnd, i-(n-1)\}} V(n-1, i-j) C_j$

この $V(k, n)$ を用いて $\tilde{M}_1(n)$ の分布は次で与えられる。

$$\Pr(\tilde{M}_1(n) = i) = \pi^T V(n, i) \mathbf{1}, \quad n \leq i \leq nrwnd \quad (17)$$

この分布を用いて、 $\tilde{X}_1(n)$ の分散は次で与えられる。 $(\tilde{X}_1(n))$ の平均は \bar{X} となる。

$$\text{Var}(\tilde{X}_1(n)) = \frac{1}{(nrwnd)^2} \text{Var}(\tilde{M}_1(n)) \quad (18)$$

4.2.3. 数値例

ここでは最も基本的なユーザ数が 2 の場合 ($m = 2$) について、 \bar{X} と $\tilde{X}_1(n)$ の標準偏差 $\text{SD}(\tilde{X}_1(n))$ を示す。 $rwnd = 1, c = 6, nrwnd = 6$ とする。計算より、 $\bar{X} = 3.1$ となり、これは $c/m = 3$ よりも若干大きくなっている。図 3 には $\text{SD}(\tilde{X}_1(n))$ の推移を示す。

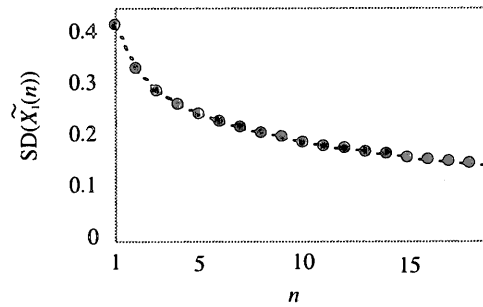


図 3: $\text{SD}(\tilde{X}_1(n))$

4.3. プロセッサシェアリングモデルによるモデル化

4.3.1. 基本モデル

TCP フロー制御は、パケット損失の検出によってネットワークの混雑状況を把握し、その混雑の程度に合わせてデータの送出量を調整するための仕組みであった。これは、ネットワークの立場からは、ネットワーク資源をユーザに割り当てるための制御とみることもできる。ここで以下のような理想化された状況（仮定）を考える。

(仮定 3-1) ネットワーク資源はそれを利用するユーザに対して常に公平に割り当てられる。

(仮定 3-2) あるユーザがデータの送信を開始または終了したら即座に割り当ての修正が行われ、新たな公平な割り当てが実現される。

(仮定 3-3) 全てのネットワーク資源が常に有効に利用される。

このような仮定の下, 仮定 1-1 に従って, ネットワーク内のひとつのボトルネックリンクに注目し, そのリンクをサーバ, そのリンクを通して転送されるデータを客としたプロセッサシェアリングモデル (processor sharing model; PS モデル) をフロー制御の評価モデルとして考える. PS モデルとは, 系内に n 人の客がいる場合には各客を $1/n$ のサーバ能力で同時に処理する待ち行列モデルである. PS モデルにおける系内客数 L はデータ転送中のユーザ数に対応し, 客の応答時間 D はデータの転送時間に対応する. (よって, このモデル化ではパケットという概念は陽には表れない.) PS モデルでは, 客 (データ転送要求) が到着率 λ のポアソン過程に従って到着し, サービス時間 (データ長 / ボトルネックリンク速度) が平均 h の一般分布に従えば, 定常状態において

$$\Pr(L = k) = (1 - \rho)\rho^k \quad (19)$$

$$E(D(x)) = \frac{x}{1 - \rho} \quad (20)$$

が成り立つ. ただし, $D(x)$ はサービス時間が x である客の応答時間 (データ転送時間に対応) であり, ρ はトラフィック密度 $\rho = \lambda h$ である. この結果からスループット X は

$$X = \frac{x}{E(D(x))} = 1 - \rho \quad (21)$$

で与えられる. ここで, X はボトルネックリンクの転送能力を 1 として計ったスループットである. 例えば, ボトルネックリンクの速度を C byte/sec とすれば, $X \cdot C / MSS$ がパケットスループットに対応する.

TCP フロー制御の挙動を考えた場合, 仮定 3-1 ~ 仮定 3-3 には様々な近似的要素が入っている. 以下, 簡単にまとめる. PS モデルの妥当性については文献 [12] がシミュレーションによる考察を行っている.

- 仮定 3-1 について: ウィンドウサイズが時間とともに変化していることから, ネットワーク資源の割り当ても時間とともに変化する. よって, この仮定はそのような短い時間スケールでの変化を平滑化した長い時間スケールでのモデル化であり, 比較的大きなデータを送信する場合には妥当な仮定であると考えられる. 仮定 3-1 はさらに, RTT の違いによるスループットの違いを生むような非公平な資源割り当てを考慮していない. これについては, サーバ能力の割り当て比率が客のクラス毎に異なるモデル [35] を用いた解析などがなされている. しかし, そのようなモデルは後に説明する対称性を持たないため, 解析は容易ではない.
- 仮定 3-2 について: スロースタートフェイズやユーザの増減後に新たな平衡状態に達するまでの過渡的期間を考慮していない. これもそのような過渡的な期間での影響が相対的に小さくなるような場合 (例えば, 大きなデータを送信する場合) に妥当な仮定と考えられる.
- 仮定 3-3 について: ウィンドウサイズの変動によって生じる資源の空き (効率的に利用されない部分) を無視している. これについては, 後の 4.3.3 でもう一度考える.

ところで, PS モデルにおいて上記のような単純な結果が得られるのは, PS モデルが対称性を満たしているからである. このことから, PS モデルを用いた解析では, 対称性という性質が保存される範囲でモデルをどこまで一般化できるかがポイントとなる.

4.3.2. 対称性と待ち行列ネットワーク

ここでは対称性を持つノードからなる待ち行列ネットワークの性質を文献 [10, 17] に従ってまとめる.

N 個のノード, I 種類の客クラスからなる待ち行列ネットワークを考える. また, ノード j において系内客数が n_j であった場合, 各客はポジション ℓ , $1 \leq \ell \leq n_j$ にいるものとする. $C_j(\ell)$ をノード j のポジション ℓ にいる客のクラス, $C_j = (C_j(1), \dots, C_j(n_j))$ とし, ネットワークの状態を $C = (C_1, \dots, C_N)$ で表すことにする. 対称性を持つ待ち行列 (対称待ち行列; symmetric queue) は以下で定義される.

Definition 1 次の 3 つの条件を満たす場合, そのノード (ノード j とする) は対称待ち行列であるという.

- (A1) 客のサービス時間はクラスに依存した一般分布に従う. S_{ju} をクラス u の客のサービス時間とし, 期待値を $E(S_{ju}) < \infty$ とする.
- (A2) サービス能力 (単位時間当りに処理できるサービス時間) は系内客数 n_j に依存してよい. $\phi(n_j)$ を系内客数が n_j の時のサービス能力とする. ただし, $\phi(n_j) > 0$, $n_j \geq 1$.
- (A3) 系内客数が n_j の時に新たに到着した客は確率 $\delta_j(\ell, n_j+1)$ でポジション ℓ に入る. ただし, $\sum_{\ell=1}^{n_j+1} \delta_j(\ell, n_j+1) = 1$. 新たに到着した客がポジション ℓ に入った場合, ポジション ℓ, \dots, n_j の客はそれぞれポジション $\ell+1, \dots, n_j+1$ に移る.

(A4) 系内客数が n_j の時にポジション ℓ にいる客が受けるサービス能力の割合は $\delta_j(\ell, n_j)$ である。ポジション ℓ の客が退去した場合、 $\ell + 1, \dots, n_j$ の客はそれぞれポジション $\ell, \dots, n_j - 1$ に移る。 □

対称待ち行列には、PS モデル、無限サーバモデル、損失モデル（待ち合い室のないモデル）などが含まれる。

ノード間での客の移動はマルコフ型ルーチングを考える。 $r_{ju, kv}$ をノード j にいるクラス u の客がノード k へクラス v の客となって移動する確率とする。 $r_{ju, kv}$ は以下を満たすものとする。

$$\sum_{k=1}^N \sum_{v=1}^I r_{ju, kv} = 1, j = 1, \dots, N, u = 1, \dots, I \quad (22)$$

ノード 0 はネットワークの外部を表す。外部からの到着は各ノードの各クラス毎に、互いに独立なポアソン過程に従うとし、 λ_{ju} をノード j へのクラス u の客の到着率とする。他のノードから移動してきた客も含め、ノード j へのクラス u の客の平均到着率を α_{ju} とする。 $\{\alpha_{ju}\}$ は次のトラフィック方程式の解として与えられる。

$$\alpha_{ju} = \lambda_{ju} + \sum_{k=1}^N \sum_{v=1}^I \alpha_{kv} r_{kv, ju}, j = 1, \dots, N, u = 1, \dots, I \quad (23)$$

以上の設定の下、次の定理が得られる。

Theorem 2 サービスポジションを状態として持った待ち行列ネットワークを考える。各ノードは次のどちらかであるとする。

- 対称待ち行列。
- 対称待ち行列ではないが、客のサービス時間がクラスに依存しない指数分布に従う。

この時、ネットワークの定常分布は

$$\pi(C) = \prod_{j=1}^N \pi_j(C_j) \quad (24)$$

で与えられる。ただし、 $\rho_{ju} = \alpha_{ju} E(S_{ju})$, $\rho_j = \sum_{u=1}^I \rho_{ju}$,

$$\pi_j(C_j) = b_j^{-1} \prod_{\ell=1}^{n_j} \frac{\rho_j^{C_j(\ell)}}{\phi_j(\ell)}, \quad b_j = \sum_{n=0}^{\infty} \frac{\rho_j^n}{\prod_{\ell=1}^n \phi_j(\ell)}.$$

ただし、 b_j の右辺は収束するものとする。 □

L_{ju} を定常状態においてノード j にいるクラス u の客数、 L_j をノード j の系内客数、 D_{ju} をクラス u の客のノード j での滞在時間とすると、この定理とリトルの公式より次が得られる。

$$E(L_j) = b_j^{-1} \sum_{n=0}^{\infty} \frac{k \rho_j^n}{\prod_{\ell=1}^n \phi_j(\ell)} \quad (25)$$

$$E(D_{ju}) = \frac{E(L_j)}{\rho_j} E(S_{ju}) \quad (26)$$

$D_{ju}(x)$ をサービス時間が x であるクラス u の客のノード j での滞在時間とすると、クラス u をサービス時間毎に細分化したクラスを考えることで次を得る。

$$E(D_{ju}(x)) = \frac{E(L_j)}{\rho_j} x \quad (27)$$

4.3.3. モデルのバリエーション

式 (25) で与えられる $E(L_j)$ はデータを送出している平均ユーザ数に対応する。また、クラス u の客がノード j で単位時間当りに受けたサービス量をスループット X_{ju} とすると、式 (27) より、 X_{ju} は次で与えられる。

$$X_{ju} = \frac{x}{E(D_{ju}(x))} = \frac{\rho_j}{E(L_j)} \quad (28)$$

この式から、例えば、ほとんどのユーザが TCP を利用しており、データを転送していない時はコネクションを設定していないのであれば、リンクの使用率 (ρ_j) と TCP コネクションの平均同時接続数 ($E(L_j)$) からスループットを推定できることが分かる。

式 (25) と式 (28) より、定理 2 の条件を満たす範囲内であれば、データを送出している平均ユーザ数やスループットが容易に求められることになる。以下、このような観点からモデルのバリエーションを示す。

- (1) 受信側ウィンドウサイズ ($rwnd$) を考慮したモデル：受信側ウィンドウサイズによって ACK なしで送出できるパケット数が制限される場合には、ボトルネックリンクの容量を使いきれない可能性がある。これを PS モデルに置き換えると、ひとりの客が受けられる最大のサーバ能力に制限があるモデルとなる。全体のサーバ能力を 1 とし、ひとりの客が受けられる最大のサーバ能力が客に関係なく $r \leq 1$ であったとする（これは $rwnd$ の値が各ユーザで同じ場合に相当する）。この時、 $n^* = \max\{n | nr \leq 1\}$ として、

$$\phi(n) = \begin{cases} nr, & n \leq n^* \\ 1, & \text{otherwise} \end{cases}$$

とおくと、この PS モデルは対称待ち行列となる。

- (2) 速度の遅いアクセスラインを考慮したモデル：モデムを利用した場合など、ボトルネックリンク以外に通信速度を制限する要因がある場合にも、(1) と同じ現象の生じる可能性がある。この場合も、制限された通信速度の値が各ユーザで同一であれば (1) と同様に対称待ち行列となる。
- (3) 仮定 3-3 を考慮したモデル：ウィンドウサイズの変動によって生じる帯域の無駄についても、無駄になる帯域の割合を同時にデータを送信しているユーザ数 n の関数 $w(n)$ として与えることができれば、 $\phi(n) = 1 - w(n)$ と置くことで対称待ち行列になる。
- (4) データ発生に構造を持たせたモデル：例えば、ユーザがホームページを見る場合において発生するデータ転送要求は、そのような行為を行っている期間（これをセッションと呼ぶ）があり、その期間内で幾つかのデータ転送要求が発生するという構造を持っている [15]。セッションの発生はポアソン過程に従うとし、セッション内では一つのデータ転送が終了したらある期間（OFF 期間）経過後、次のデータ転送を行うものとする。送信するデータ量と OFF 期間は互いに独立で任意の分布に従うものとする。ユーザのクラスを設けてデータ量や OFF 期間の分布がそれに依存するとしてもよい。また、セッション内の何番目のデータ送信か、何番目の OFF 期間かによって分布を変えてもよい。このような設定の下、ボトルネックリンクを PS モデルで表したノードと、OFF 期間を無限サーバモデルで表したノードを持つ待ち行列ネットワークを考える。ノード間のルーチングはそれがマルコフ型ルーチンになるよう、適当に与える。こうしてできた待ち行列ネットワークは定理 2 の条件を満たす [12]。よって、ボトルネックリンクでのスループットは式 (28) で与えられる。

4.4. 最適化問題としてのモデル化

この部分と次の 5 節では、TCP フロー制御を離れ、AQR と ECN をもっと一般化した枠組みの中でフロー制御について考えていく。特にこの部分では、仮定 1-1 を前提に、通信による効用と輻輳によるコストを導入してフロー制御を最適化問題として定式化し、分析する [29, 30]。最適化問題としての定式化に関する基本的考え方は次の 5 節で説明する。ここで取り上げる手法と次節で取り上げる手法との違いは、前者がバッファにおける送信待ちパケット数の挙動を確率モデルとして表しているのに対し、後者ではパケットを流体として捉え、その流れを確定的モデルとして表している点にある。この表現の違いは、前者ではボトルネックリンクのみが考慮されているのに対し、後者ではネットワーク全体を考慮できるという違いを生んでいる。

4.4.1. モデルの構成

ここでは、次の枠組みでモデル化を行う。

(仮定 4-1) ネットワークから輻輳情報がユーザに送られ、ユーザはそれをもとにパケットの送信レートを調節する。

(仮定 4-2) ネットワークからユーザに送られる輻輳情報に遅れはない。

仮定 4-1 は TCP フロー制御における CI ビットの利用などを想定しているが、ここでは 2 値情報だけではなく、任意の情報が送れるものとする。ここでの解析の目標は、ネットワークにおける信号の生成方法と、その信号を基にしたユーザのパケット送信レートの調節方法を定めることにある。さらに、各ユーザのスループットに差をつける為の仕組みを導入することにある。（これは、各ユーザが異なる効用関数を持つことで実現される。）ボトルネックリンクと効用、コストは次のように設定する。

(1) ボトルネックリンク

(仮定 4-3) ボトルネックリンクを通して通信を行っているユーザ数を固定値 m とする。

(仮定 4-4) 各ユーザは互いに独立な非斉時ポアソン過程に従ってパケットを送信する。 $\lambda_i(t)$ を時点 t におけるユーザ i のポアソン強度（パケットの送信レートに対応）とし、 $\lambda(t) \equiv (\lambda_i(t))$ とする。

(仮定 4-5) ボトルネックリンクにおけるパケットの送信時間は互いに独立で平均 $1/\mu$ の指数分布に従う。パケットの送信時間はパケットの発生過程とも独立である。

これらの仮定の下、ボトルネックリンクはバッファサイズ $K < \infty$ の単一サーバ待ち行列としてモデル化される (図4)．バッファにあるパケットには送信中パケットも含む．ユーザを含め、このモデル全体をシステムと呼ぶことにする．以下では、時点 t においてバッファにあるパケット数を $L(t)$ とする．

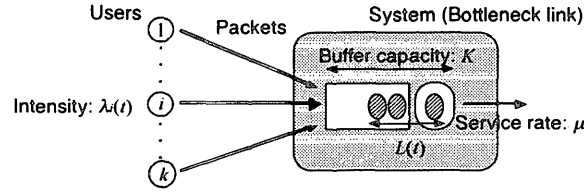


図4: 解析モデル

(2) ユーザ効用

ユーザ i が通信によって受け取る効用をスループットの関数として表す．まず、 $A_i(t)$ を区間 $[0, t]$ においてユーザ i が送出したパケット数とする．この時、パケット数を基準に考えたこの区間におけるスループットの平均は以下で表される [8]．

$$E \left[\frac{1}{t} \int_0^t 1(L(t-) < K) dA_i(t) \right] = E \left[\frac{1}{t} \int_0^t 1(L(t) < K) \lambda_i(t) dt \right], \quad (29)$$

ここで、 $1(\cdot)$ は指示関数 (indicator function) であり、平均は $\{A_i(t)\}$ が強度 $\lambda_i(t)$ のポアソン過程となるような確率測度による平均とする．式 (29) より、 $1(L(t) < K) \lambda_i(t)$ を時刻 t における瞬間的なスループットと考え、次の仮定をおく．

(仮定 4-6) ユーザ i の効用は関数 $U_i(\cdot)$ を用いて $U_i(1(L(t) < K) \lambda_i(t))$ で与えられる．

(仮定 4-7) $U_i(\cdot)$ は単調増加で微分可能、厳密に上に凸 (strictly concave) であるとする．

仮定 4-7 のような $U_i(\cdot)$ の設定はコンピュータ通信における順応性のあるトラフィック (elastic traffic) [34] を表す効用として用いられているものである．

(3) 輻輳コスト

バッファでのパケット送出が先着順 (FIFO) であるとし、バッファ内パケット数 $L(t)$ を、時点 t に送出されたパケットの遅延を間接的に表す量と考える．送信されたパケットにはパケット毎に $R(L(t))$ の輻輳コストが生じるものとし、式 (29) と同様の考え方に従い、次を仮定する．

(仮定 4-8) 時点 t において、ユーザ i は輻輳コスト率 (単位時間当りの輻輳コスト) $R(L(t)) \lambda_i(t)$ の輻輳コストを被る．

(仮定 4-9) $R(\cdot)$ は上に有界な非負単調増加関数であり、ユーザには依存しない．

ここで、 $R(K)$ は廃棄されたパケットにかかるコストに対応している．仮定 4-9 で $R(\cdot)$ をユーザには依存しない関数として与えたことは、後で示すように、信号 (輻輳情報) をユーザに依存しない形で生成するために本質的な仮定である．

4.4.2. 最適化問題

ユーザ i に対する時点 t での制御パラメータを $u_i(t)$ とし、 $\mathbf{u}(t) \equiv (u_i(t))$ とする．パケット送信レートの制御は $\lambda_i(t) = u_i(t)$, $t \geq 0$ の形で行われるものとする． $u_i(t)$ の制御範囲を $B_i \equiv [b_{i,0}, b_{i,1}]$, $0 \leq b_{i,0} < b_{i,1} < \infty$ とし、 $B \equiv \prod_{i=1}^k B_i$ とする．コスト最小化問題として定式化するために、 $U_{i,max} \equiv U_i(b_{i,1})$ と置き、ユーザ i のコスト率 (単位時間当りに被るコスト) とシステム全体でのコスト率を次で定義する．

$$C_i(L(t), \lambda_i(t)) \equiv \left\{ U_{i,max} - U_i(1(L(t) < K) \lambda_i(t)) \right\} + R(L(t)) \lambda_i(t), \quad (30)$$

$$C(L(t), \lambda(t)) \equiv \sum_{i=1}^k C_i(L(t), \lambda_i(t)). \quad (31)$$

定義より、コスト率は常に非負の値を取る．平均コストの期待値を次で定義する．

$$J(\mathbf{u}, \ell_0) \equiv \lim_{T \rightarrow \infty} E_{\mathbf{u}} \left[\frac{1}{T} \int_0^T C(L(t), \mathbf{u}(t)) dt \mid L(0) = \ell_0 \right]. \quad (32)$$

ここで、 $\mathbf{u} \equiv \{u(t)\}$ であり、 E_u は制御パラメータに依存した期待値であることを示す。次の最適化問題を考える。

$$[\text{OP1}] \quad J^*(l_0) \equiv \inf_{\mathbf{u}} J(\mathbf{u}, l_0) \quad (33)$$

後で示すように、 $J^*(l_0)$ は l_0 に依存しない形で与えられる。以下では、 $J^*(l_0)$ を最適化問題の最適値、その最適値を与える制御 \mathbf{u}^* を最適制御と呼ぶ。

4.4.3. 最適制御

平均期待コストを最小化するマルコフ決定過程の最適制御（最適政策）を導出するのと同様な方法 [33] により、最適化問題 [OP1] が最適制御を持つための十分条件が次で与えられる [30]。

Theorem 3 最適化問題 [OP1] について、次を満たす有界な関数 $G(\ell)$, $0 \leq \ell \leq K-1$ と定数 g が存在するとする。

$$g = \inf_{\mathbf{v} \in \mathbf{B}} \left\{ \sum_{i=1}^k v_i 1(\ell < K) G(\ell) - \mu 1(\ell > 0) G(\ell-1) + C(\ell, \mathbf{v}) \right\}. \quad (34)$$

ここで、 $\mathbf{v} \equiv (v_i)$ である。この時、次を満たす \mathcal{R}^k 値関数 $\mathbf{v}^*(\ell) \equiv (v_i^*(\ell))$, $0 \leq \ell \leq K$ が存在する。

$$\mathbf{v}^*(\ell) = \arg \min_{\mathbf{v} \in \mathbf{B}} \left\{ \sum_{i=1}^k v_i 1(\ell < K) G(\ell) + C(\ell, \mathbf{v}) \right\}. \quad (35)$$

これを用いて、最適制御 $\mathbf{u}^*(t)$ の少なくともひとつは $\mathbf{u}^*(t) = \mathbf{v}^*(L(t))$ で与えられ、最適値 J^* は $J^* = J^*(l_0) = g$ であたえられる。□

もし、定理 3 の条件が成立すれば、バッファ内パケット数過程 $\{L(t)\}$ は出生死滅過程となる。 $L(t) = \ell$ の時の増加率は $\sum_{i=1}^k v_i^*(\ell)$ で与えられ、減少率は μ で与えられる。

4.4.4. フロー制御の構成

定理 3 より、最適化問題 [OP1] の下では、各ユーザは式 (35) に従ってパケットの発生率を調節すればよいことがわかる。もし、式 (35) で表される最適解 $\mathbf{v}^*(\ell)$ の値が常に \mathbf{B} 内部にあれば、最適解 $\mathbf{v}^*(\ell)$ は ℓ , $0 \leq \ell \leq K-1$ 、に対して次を満たす。

$$U'_i(v_i^*(\ell)) = G(\ell) + R(\ell), \quad 1 \leq i \leq k. \quad (36)$$

この式の右辺はユーザに依存しないことから、それをフロー制御のための信号 $p(\ell) \equiv G(\ell) + R(\ell)$ と見なす。この信号を用いて、ユーザ i は、 $L(t) < K$ の時、 $u_i^*(t) = \min\{b_{i,1}, \max\{b_{i,0}, U_i'^{-1}(p(L(t)))\}\}$ に従ってパケットの発生率を調節する。ユーザ i にとって、この制御方法は次の最適化問題と同値である。

$$[\text{UP}] \quad \inf_{v_i \in B_i} \left\{ v_i p(L(t)) - U_i(v_i) \right\}. \quad (37)$$

この最適化問題 [UP] は、文献 [18, 19] におけるユーザ効用最大化問題と同様なものであり、そこでは信号 $p(\ell)$ が影の価格 (shadow price) に対応している。説明は省略するが、 $p(\ell)$ が $L(t) = \ell$ の時にパケットをひとつ送信したことによってシステムが被るコストの増加分であることを理論的に示すこともできる。式 (35) より、 $L(t) = K$ の時は $u_i^*(t) = b_{i,0}$ となるので、 $p(K)$ には十分大きな値を設定するものとする。以上の結果より、理想化されたフロー制御が次のように得られる。

- ネットワークは、 $L(t)$ の値が変化するごとに信号 $p(L(t))$ を全ユーザに送る。
- 信号 $p(L(t))$ を受け取ったユーザは最適化問題 [UP] を解いてパケットの発生率を決める。

このフロー制御では非常に多くの信号が生成される可能性があり、現実的ではない。ここでの議論はまだ理論的モデルの中だけのものであり、実現可能性を考慮した検討が必要である。

4.4.5. 数値例

ユーザ i の効用関数を $U_i(x) = a_i x^{0.5}$, $a_i > 0$ とし、輻輳コストを表す関数は常にゼロ ($R(\ell) = 0$) であるとす。 (よって、式 (30) より、効用の最大化、言い換えれば、効用関数で変換されたスループットの最大化のみを考えた問題となる。) 全てのユーザに対して $b_{i,0} = 0.1$, $b_{i,1} = 10$ とする。図 5 は、各ユーザが異なる効用関数を持つ場合について $v_i(\ell)$ と $p(\ell)$ を示したものである。先の議論より、 $v_i(\ell)$ は $L(t) = \ell < K$ の時の、ユーザ i の瞬間的なスループットに対応する。ユーザ数 k は 10 とし、バッファサイズは $K = 20$ 、平均パケット送信時間は $\mu^{-1} = 0.1$ とする。ユーザ 1 からユーザ 5 に対しては $a_i = 10$ とし、その他のユーザに対しては $a_i = 15$ とする。この図より、効用関数を適当に設定することでユーザ間にスループットの差を設けることができるのがわかる。

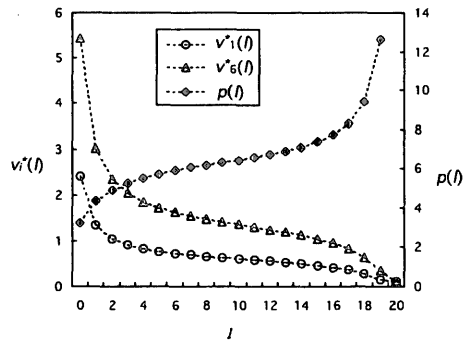


図 5: 各ユーザが異なる効用関数を持つ場合の結果

5. 最適資源配分問題としての解析方法

5.1. 基本的考え方

フロー制御を含めた通信制御の検討に経済学の考え方やモデルを適用する試みが多くなされている。Shenker[34]は通信によって得られるユーザの効用 (utility) をもとに将来的なネットワークのデザインについて論じている。MacKie-Mason[25]らは、各ユーザの効用を、そのユーザに割り当てられた財の量 (パケットの送信速度に相当) と利用する通信設備の使用率 (輻輳の状況を間接的に表す量) の関数として与え、通信ネットワーク全体での効用を全ユーザの効用の和として定義している。この定式化に従えば、各ユーザに割り当てられる財の量は通信ネットワーク全体での利益を最大化する量として与えられ、通信の価格 (price) は輻輳の限界費用 (marginal congestion cost) として与えられる。この定式化は、輻輳によるパケットの遅延や廃棄を経済学における外部性 (externality) と捉え、それを輻輳の費用 (congestion cost) とし、通信網全体での効用を、ユーザに割り当てられた財のみに依存するユーザ効用の合計から輻輳の費用を引いたものとして捉え直すこともできる。MacKie-Mason らの検討はフロー制御の方法に主眼を置いたものではなく、商用インターネットの出現により生じた課金 (pricing) の課題 [24, 25] について論じようとしたものである。しかし、通信ネットワークのフロー制御において、この価格は通信ネットワークの輻輳情報と見なすことができ、価格によって需要を変化させる仕組みはフロー制御に対応すると考えることができる。このような考え方を明示的に取り入れたのが Kelly らによる検討 [13, 18, 19, 20], Key らによる検討 [21], Golestani らによる検討 [14] であり、関連して Low らの検討 [22, 23] がある。

これらの検討は基本的に以下の仮定に基づいている。(これらの文献では必ずしもこの枠組みに当てはまらない内容の検討も行っているが説明は省略する。)

- (仮定 1-1) パケットを流体と考える。(よって、以下ではパケットという表現をやめ、データと呼ぶことにする。)
- (仮定 1-2) ネットワークはデータを送信する複数のユーザとそのデータを転送する複数のリンクからなる。リンクにはバッファはなく、転送容量を超えたデータは単純に廃棄される。データの伝送遅延は考慮しない。
- (仮定 1-3) ネットワークから輻輳情報がユーザに送られ、ユーザはそれをもとにデータの送信レート (単位時間当りに送信するデータ量) を調節する。
- (仮定 1-4) 各ユーザには常に送信すべきデータがあり、常に調節された送信レートいっぱいデータを送信する。
- (仮定 1-5) 輻輳情報はリンク毎に、そのリンクに対する各ユーザからのデータ転送要求量の総和に基づいて生成される。

以上の仮定からわかるように、このモデルは各ユーザへの確定的なリンク容量配分問題となる。そこで、各ユーザのデータ転送レートに関する効用と、各リンクに対するデータ転送要求量の総和がそのリンクの容量を超えたことにより生じるコスト (ペナルティ) を考える。リンク容量配分問題は、コストの総和から効用の総和を引いたものを最小化する最適化問題として定式化される。以下で示すように、フロー制御の方法は、この最適化問題の解を繰り返し計算により求める手順として表される [19, 20, 21]。この数学モデルは通信網全体における輻輳制御の最適性を表すモデル (社会的最適性を表すモデル) と、その最適性を実現するために通信網が輻輳情報を生成するモデル及びその情報に応じたユーザ行動を表すモデルで構成される。(実際の輻輳制御では、これらの他に、網内で生成された輻輳情報をユーザに通知するための方法が重要となるがここでは考えない。)

5.2. 社会的最適性を表すモデル

通信ネットワークを構成するリンクの集合を J とする。輻輳費用について次の仮定をおく。

(仮定 2-1) リンク $j \in J$ に対するデータ転送要求量の総和が y の時にかかる輻輳費用は $C_j(y)$ である。輻輳費用に対しては加法性が成り立つ。

(仮定 2-2) $C_j(y)$ は微分可能であり、その微分 (限界費用) を $p_j(y) = \frac{d}{dy} C_j(y)$ とする。 $p_j(y)$ は正の値を取り、 y に関して連続かつ強い意味での単調増加 (strictly increasing) である。輻輳の限界費用に対しては加法性が成り立つ。

ユーザの集合を R とする。ここでユーザとは通信網内のある経路に沿ってデータを送信しているコンピュータとする。もし、同じコンピュータから複数の送信先にデータを送信していたならばそれらは異なるユーザとして扱う。ユーザ $r \in R$ が使用するリンクの集合を $\{j | j \in r\}$ 、データ送信レートを x_r とする。 A_{jr} を $j \in r$ ならば 1、そうでなければ 0 の値を取る記号とし、行列 $A = (A_{jr})$ 、列ベクトル $x = (x_r)$ を定義する。効用について次の仮定をおく。

(仮定 2-3) ユーザ r は需要 x_r に対して効用 $U_r(x_r)$ を持つ。効用に関しては加法性が成り立つ。

(仮定 2-4) $U_r(x_r)$ は $x_r > 0$ において単調増加で強い意味で上に凸な関数 (strictly concave function) である。また、 $U_r(x_r)$ は連続微分可能性であり、境界条件 $\lim_{x_r \rightarrow +0} U'_r(x_r) = \infty$ と $\lim_{x_r \rightarrow \infty} U'_r(x_r) = 0$ を満たす [20]。

この仮定に従う効用関数は、インターネットにおいて順応性のあるトラフィック (elastic traffic) [34] と呼ばれているサービスに対応する。

通信ネットワークの社会的最適性 (social optimum) とは、効用の総和から輻輳費用の総和を引いた値を最大化するようにリソースの配分を行うことであると、次の最適化問題を設定する。

$$\begin{aligned} & \text{maximize} \quad U(x) = \sum_{r \in R} U_r(x_r) - \sum_{j \in J} C_j \left(\sum_{r \in R} A_{jr} x_r \right) \\ & \text{over} \quad x \geq 0 \end{aligned} \quad (38)$$

5.3. 輻輳情報を生成するモデル

ユーザ r が時刻 t で受け取る輻輳に関する情報を $f_r(t)$ とする。輻輳情報 $f_r(t)$ には 5.2 で定義した輻輳の限界費用を用いる。まず、リンク j の時刻 t における輻輳の限界費用 $\mu_j(t)$ を次式で与える。

$$\mu_j(t) = p_j \left(\sum_{s \in R} A_{js} x_s(t) \right) \quad (39)$$

ユーザ r に対する輻輳情報はそのユーザが使用しているリンクから送られるとする。限界費用に対する加法性を仮定していることから、 $f_r(t)$ を次で与える。

$$f_r(t) = \sum_{j \in r} \mu_j(t) = \sum_{j \in r} p_j \left(\sum_{s \in R} A_{js} x_s(t) \right) \quad (40)$$

輻輳の限界費用を価格 (price) と考えることで、この輻輳情報 $f_r(t)$ はユーザ r に課せられた通信の価格と見ることもできる。価格は通信を利用する人に対してインセンティブを与えるものであり、輻輳制御に関する重要な要素になり得ることも考えられる。しかし、ここではそれが形式化されユーザ行動モデルの中に組み込まれているとし、通信に係わる経済的行動は考えないものとする。

5.4. ユーザ行動を表すモデル

現在のインターネットでは、TCP と呼ばれるプロトコル (通信規約) に従い、各コンピュータが送信するパケットの量を輻輳状況に合わせて自動的に調整する仕組みとなっている。従って通信の開始から終了までのユーザ行動を表すモデルはこのプロトコルによって完全に記述される。サービス利用者としての人の介入は通信の開始と場合によってはその強制的な終了の指示のみである。このようにコンピュータ通信網の輻輳制御とは、ある理想化された消費者のみが存在する需要と供給の調整モデルと見なすことができる。そこではプロトコルによる規定が理想化された消費者象に対応する。

今、ユーザ r が時刻 t での輻輳に関する情報 $f_r(t)$ を利用してその需要 $x_r(t)$ を変化させるとし、その変化を表す関数を $F_r(x_r(t), f_r(t))$ と置く。時間を連続に取る場合は需要の変化が十分滑らかであるとしてユーザ行動モデルを次のように表す。

$$\frac{d}{dt} x_r(t) = F_r(x_r(t), f_r(t)) \quad (41)$$

また、時間を離散に取る場合は以下で表す。

$$x_r(t+1) - x_r(t) = F_r(x_r(t), f_r(t)) \quad (42)$$

5.5. モデルの例

5.5.1. TCP フロー制御

ここではTCPフロー制御について、関数 $F_r(x_r(t), f_r(t))$ の具体的な形を見ていく。非常に多くの仮定が設定されるため、説明が煩雑になるのを避け、ここでは何が仮定かを分けて記述することはしない。

ユーザ r に注目し、時点 t におけるウィンドウサイズを $W_r(t)$ (r.v.) とする。 rtt をパケットの往復時間 (round trip time, 簡単のため固定値と仮定) とし、パケット数を基準として量った時点 t における単位時間当りの送信データ量 $X_r(t)$ (r.v.) を次で与える。

$$X_r(t) = W_r(t)/rtt \quad (43)$$

$W_r(t)$ は、送信したパケットが通信網内で廃棄されたという輻輳情報を受け取ると半分 ($W_r(t)/2$) に減らされ、そのパケットが受信されたという輻輳情報を受け取ると $1/W_r(t)$ だけ増やされるとする。そこで n 番目の情報を受け取った時刻を t_n とし、その時の輻輳情報を $I_r(t_n)$ とする。この $I_r(t_n)$ は、パケットが廃棄された場合は 1 の値を取り、その他の場合は 0 の値を取る確率変数とする。この時、 $W_r(t)$ の変化は次式で表される。

$$W_r(t_{n+1}) - W_r(t_n) = \frac{1}{W_r(t_n)}(1 - I_r(t_n)) - \frac{W_r(t_n)}{2}I_r(t_n) \quad (44)$$

さらに、コンピュータ通信網の状態は各ユーザの輻輳ウィンドウサイズの組 $W(t) = (W_r(t), r \in R)$ で与えられると仮定する。式 (44) はその状態の変化を規定するものである。そこで、 $W(t)$ の平均的な挙動を見るために、現在の状態が与えられたという条件の下で式 (44) の条件付き平均を求める (次式)。ただし、 $w = (w_r, r \in R)$ とする。

$$E(W_r(t_{n+1}) - W_r(t_n) | W(t_n) = w) = \frac{1}{w_r} - \left(\frac{1}{w_r} + \frac{w_r}{2}\right)E(I_r(t_n) | W(t_n) = w) \quad (45)$$

式 (45) において、 $E(I_r(t_n) | W(t_n) = w)$ は、ユーザ r のパケットが廃棄される条件付確率を表している。この条件付確率が状態 w にのみに依存して決まるものとし、 $\tilde{f}_r(w) = E(I_r(t_n) | W(t_n) = w)$ と置く。さらに、次式に従って変化する (非確率的な) 過程を $w(t) = (w_r(t), r \in R)$ と置く。

$$w_r(t_{n+1}) - w_r(t_n) = \frac{1}{w_r} - \left(\frac{1}{w_r} + \frac{w_r}{2}\right)\tilde{f}_r(w) \quad (46)$$

これを輻輳ウィンドウサイズの平均的な挙動を表すモデルと考え、さらに式 (43) と同様に時刻 t における平均的な送信データ量 $x_r(t)$ を $x_r(t) = w_r(t)/rtt$ で与える。

次に、文献 [20] に従って式 (46) を連続時間モデルで近似し、 $x_r(t)$ の変化率を与える。まず、時間 rtt の間に $w_r(t)$ 個のパケットが送信されることから、式 (46) の変化が起きる間隔は平均的に $rtt/w_r(t) = 1/x_r(t)$ となる。よって、 $x_r(t)$ ($t = t_n$) の変化率は $f_r(t) = \tilde{f}_r((x_r(t)rtt, r \in R))$ として次式で近似される。

$$\frac{d}{dt}x_r(t) \approx \frac{x_r(t_{n+1}) - x_r(t_n)}{1/x_r(t)} = \frac{1}{rtt^2} - \left(\frac{1}{rtt^2} + \frac{x_r^2(t)}{2}\right)f_r(t) \quad (47)$$

この式がTCPフロー制御のユーザ行動を表すモデルとなり、 $f_r(t)$ が輻輳情報となる。輻輳情報についてはさらに次のように考える。リンク j でのパケット廃棄率はそのリンクを通過する送信データ量の総和 y のみによって決まるとし、それを $p_j(y)$ と置く。さらに、パケット廃棄率に対して加法性が成り立つと仮定すると、 $f_r(t)$ は式 (40) で与えられる。文献 [20] によれば、式 (47) でユーザ行動モデルが与えられた場合、 $x(t) = (x_r(t), r \in R)$ はある状態 $x^* = (x_r^*(t), r \in R)$ に収束し、その状態 x^* は次の最適化問題の解として与えられる。

$$\begin{aligned} & \text{maximize } U(x) = \sum_{r \in R} \frac{\sqrt{2}}{rtt} \arctan\left(\frac{x_r rtt}{\sqrt{2}}\right) - \sum_{j \in J} \int_0^{\sum_{r \in R} A_{j,r} x_r} p_j(z) dz \\ & \text{over } x \geq 0 \end{aligned} \quad (48)$$

これをTCPフロー制御の社会的最適性を表すモデルと考えれば、ユーザの利得及びリンクの輻輳費用は次で与えられる。

$$U_r(x_r) = \frac{\sqrt{2}}{rtt} \arctan\left(\frac{x_r rtt}{\sqrt{2}}\right), \quad C_j(y) = \int_0^y p_j(z) dz \quad (49)$$

5.5.2. 支払意思額による方法

ここでは文献 [19, 20] で提案されている方法を示す。式 (39) で定義された $\mu_j(t)$ はリンク j における輻輳の限界費用であり、これを影の価格 (shadow price) と考える。この時、ユーザ r に課せられる単位時間当りの料金は $x_r(t) \sum_{j \in r} \mu_j(t) = x_r(t) f_r(t)$ となる。次に、ユーザ r の単位時間当りの支払意思額 (willingness to pay) を $w_r(t)$ とし、これをユーザ r の限界効用 $U'_r(x_r)$ を用いて次で与える。

$$w_r(t) = x_r(t) U'_r(x_r(t)) \quad (50)$$

均衡状態では料金と支払意思額が一致すると考えられることからユーザ行動を表すモデルとして次を与える。

$$\frac{d}{dt} x_r(t) = \kappa_r (w_r(t) - x_r(t) f_r(t)) \quad (51)$$

ここで κ_r は $x_r(t)$ の変化の速さを調整するパラメータである。このユーザ行動モデルによって $x(t)$ が収束する状態 x^* は式 (38) の最適化問題の解として与えられる。

以上の例からも分かるように、コンピュータ通信の輻輳制御法を考えるということは各ユーザの行動モデルを考えることであり、ユーザの戦略を考えることでもある。このようなゲーム論的な視点からのシミュレーションや実験も幾つか試みられている [13, 21]。

6. おわりに

パケットフロー制御の理論的解析法について幾つかの手法を取りあげ解説してきた。解説では、何を前提としているのかを極力明確にするように努めた。仮定が何であるかは、結果の正しい利用には必須な情報である。また、次の研究課題を暗示するものでもある。理論的解析では仮定が多くなる傾向にあるが、逆に何が仮定なのかをかなり明確に示すことができるとも考えられる。最後に、ここで取りあげた方法は提案されている方法のほんの一部であり、他にも様々な視点からの検討がなされている。

参考文献

- [1] RFC1122: Requirements for Internet Hosts – Communication Layers (1989).
- [2] RFC2001: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms (1997).
- [3] RFC2018: TCP Selective Acknowledgment Options (1996).
- [4] RFC2481: A Proposal to Add Explicit Congestion Notification (ECN) to IP (1999).
- [5] RFC2581: TCP Congestion Control (1999).
- [6] RFC2582: The NewReno Modification to TCP's Fast Recovery Algorithm (1999).
- [7] E. Altman, K. Avrachenkov, and C. Barakat, A stochastic model of TCP/IP with stationary random losses, SIGCOMM 2000 (2000).
- [8] P. Brémaud, *Point Processes and Queues*, Springer-Verlag (1981).
- [9] D. Comer (村井 楠本訳), TCP/IPによるネットワーク構築 Vol. I, 共立出版, 1993.
- [10] X. Chao, M. Miyazawa, and M. Pinedo, *Queueing Networks; Customers, Signals and Product Form Solutions*, John Wiley & Sons, New York (1999).
- [11] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking, 1, 4 (1993), 397–413.
- [12] S. B. Fredj, T. Bonals, A. Proutiere, G. Régnié, and J. W. Roberts, Statistical bandwidth sharing: a study of congestion at flow level, SIGCOMM 2001 (2001).
- [13] R. J. Gibbens and F. P. Kelly, Resource pricing and the evolution of congestion control, Automatica, 35 (1999).
- [14] S. J. Golestani and S. Bhattacharyya, A class of end-to-end congestion control algorithms, in Proc. Sixth International Conference on Network Protocols (1998).
- [15] K. Ishibashi, T. Kimura, and T. Ozawa, A measurement-based performance evaluation method for IP networks and its implementation, Telecommunications System, 15 (2000), 203–215.

- [16] Van Jacobson, Congestion avoidance and control, SIGCOMM '88 (1988), 314–329.
- [17] F. P. Kelly, *Reversibility and Stochastic Networks*, John Wiley & Sons, New York (1979).
- [18] F. P. Kelly, Charging and rate control for elastic traffic, *European Transactions on Telecommunications*, 8 (1997), 33–37.
- [19] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, Rate control for communication networks: shadow price, proportional fairness and stability, *Journal of the Operational Research Society*, 49 (1998), 237–252.
- [20] F. P. Kelly, *Mathematical modelling of the Internet*, Forth International Congress on Industrial and Applied Mathematics, Edinburgh (1999).
- [21] P. B. Key and D. R. McAuley, Differential QoS and pricing in networks: where flow-control meets game theory, *IEE Proceedings Software* (1999).
- [22] S. H. Low and D. E. Lapsley, Optimization flow control, I: basic algorithm and convergence, *IEEE/ACM Transactions on Networking* (1999).
- [23] S. H. Low, Optimization based congestion control for the Internet, *IEEE Network* (1999).
- [24] J. K. MacKie-Mason and H. R. Varian, Pricing the Internet, in *Public Access to the Internet*, edited by B. Kahin and J. Keller, MIT Press, Cambridge (1995), 269–314.
- [25] J. K. MacKie-Mason and H. R. Varian, Pricing congestible network resources, *IEEE J. of Selected Areas in Communications*, 13, 7 (1995), 1141–1149.
- [26] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, The macroscopic behavior of the TCP congestion avoidance algorithm, *Computer Communication Review*, 27(3) (1997).
- [27] M. Neuts and J.-M. Li, An algorithm for the $P(n, t)$ matrices of a continuous BMAP, *Matrix-analytic methods in stochastic models* edited by S. R. Chakravathy and A. S. Alfa, Marcel Dekker, New York (1996), 7–19.
- [28] 小沢, 朝香, TCP フロー制御の過渡的特性の評価, 情報通信ネットワークの新しい性能評価法に関する総合的研究, 京都 (1998), 175–182.
- [29] 小沢, 輻輳の限界費用と輻輳情報 (コンピュータ通信網の数学モデルとその応用), 情報通信ネットワークの新しい性能評価法に関する総合的研究, 仙台 (2000), 150–159.
- [30] T. Ozawa, Dynamic programming type solution for a simple packet flow control model, *Symposium on Performance Models for Information Communication Networks*, Kyoto (2001), 75–84.
- [31] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, *SIGCOMM '98* (1998).
- [32] J. Padhye and S. Floyd, On inferring TCP behavior, *SIGCOMM 2001* (2001).
- [33] S. M. Ross, *Applied Probability Models with Optimization Applications*, Holden-Day (1970).
- [34] S. Shenker, Fundamental design issues for the future Internet, *IEEE J. on Selected Areas in Communications*, 13, 7 (1995), 1176–1188.
- [35] Yashkov, S. F., Processor-sharing queues: some progress in analysis, *Queueing Systems*, 2 (1987), 1–17.