

Self-similar トラフィックの重なりによる ボトルネックパフォーマンスの評価

小野明花梨、中嶋卓雄
九州東海大学

{70mie101@stmail, taku@ktmail}.ktokai-u.ac.jp

Web サーバのファイルサイズ分布などに Self-similar な特性が見つかっており、ネットワークトラフィックモデルが Poisson モデルから self-similar なモデルへ変化している。本研究では、一定数のクライアントが関節を作りながら線形的に伸びていくトポロジーについて考え、その伸びに伴うトラフィックの重なりによるボトルネックパフォーマンスについてシミュレーションにより評価する。結果から、第一に、 α が 1 に近い self-similar なトラフィックのほうがロス率が低い。第二に、ボトルネックでのロス率は、クライアント数とは比例せず、送信元のロスの発生に依存しながら、ある程度の大きなクライアント数によって出現する。第三に、数少ない非常に大きなファイルサイズの転送にスループットは影響を受ける。などの現象を発見できた。

Evaluation of Bottleneck Performance under Multiplexing Self-similar Traffics

Akari Ono, Takuo Nakashima
Kyushu Tokai University

{70mie101@stmail, taku@ktmail}.ktokai-u.ac.jp

The observation of the file size distribution of Web server has the self-similar property, changing the network traffic model. In this research, we construct the network topology which the 32 clients are organized in a caterpillar graph with 4 articulation points, each containing 8 clients, and discuss the bottleneck performance multiplexing network traffics using the simulator. As the results, we observed the following properties. Firstly, the traffics are close to 1, meaning the self-similar traffics, have low loss rates. Secondly, the loss rate on the bottleneck link is not proportional to the number of clients, and appears at the case of larger the number of clients depending on the losses on the source on the traffic flow. Finally, a few numbers of large files affects the throughput on the bottleneck link.

1. はじめに

Leland ら [1] の革新的な研究により、時間軸のスケールに対して、異なるスケールにおいても不変なバースト性を持つ self-similar なトラフィックが実際のネットワークで発見されている。現在、ネットワークトラフィックの self-similar な特性はネットワークのモデル化やパフォーマンスの評価において幅広く採用されている。しかし、self-similarity とネットワークパ

ラメータの関係はおもにアプリケーション層に近い文脈で議論されてきた。

一方、Park ら [2] はシミュレーションによって、ネットワーク資源、例えばボトルネック帯域やバッファ容量を変化させながら、self-similar なネットワークのパフォーマンスを評価してきた。十分に容量が大きなバッファサイズを持っている環境においては、 α が 1 に近づくことにより、パケットロス率が少しずつ増加することを観測している。このことは、self-similarity が増加

することによってパフォーマンスが低下することを示している。しかし、これらの実験はボトルネックの帯域などの制約と関係しているにも関わらず、その関連性については議論されていない。我々はボトルネックの制約が非常に大きなパケットロスを導く直前において、 α が1に近づくことによりパケットロス率が増加することを検証している。さらに我々のシミュレーションによる研究[6]では、ボトルネックの制約が解消されるにつれて、 α が1に近づけばスループットが上昇することが観測されている。このように、self-similar なトラフィックについては、パフォーマンスゲインとなる可能性もある。

本研究では、ボトルネック資源の変化について、トポロジーを変化させることにより、重複するトラフィックを生成させ、個々の self-similar な特性とネットワークパフォーマンスとの関連性について議論する。

以下、本稿は次のように構成する。まず、第2章では、今回採用したネットワークトポロジーについて説明し、シミュレーションにおいて仮定したトラフィック特性について説明する。シミュレーションにより特性を抽出し、その結果を第3章で議論する。第4章ではまとめと今後の検討項目についてまとめる。

2. ネットワークモデルとシミュレーション

2.1. ネットワークモデル

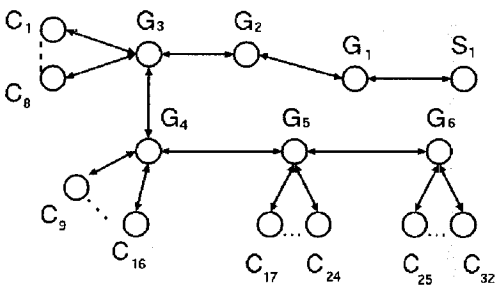


図1. ネットワークトポロジー

図1に示すように、ネットワークのノードは、 S で記述している1台のサーバ、 G による6台のゲートウェイおよび C で表した32台のクライアントの要素からなる。リンクは図1のように接続され、各ゲートウェイが節状にクライアントをバックボーンに接続しているトポロジーを考える。ボトルネックリンクは、 G_1 と G_2 のリンクであり、トラフィックとしてはダウンストリー

ムトラフィック、すなわち G_1 から G_2 へのトラフィックを計測の対象とした。

Crovella ら[3]が指摘しているように self-similar トラフィックはおもに Web サーバのファイルサイズやユーザセッションの間隔がその要因となっている。我々も Web サーバのファイルサイズの分布が、Heavy-tailed な特性を持っていることを検証してきた[5]。インターネットで転送される平均ファイルサイズは10Kbyte前後が多く、このことは平均的なファイルは10個のTCPパケットに満たないパケットで構成されていることを意味している。TCPの輻輳制御で考えると、大部分のファイル転送がスロースタートフェーズで終了することを意味している。一方で、インターネットでは非常に大きなファイルが転送されている。本研究では、この self-similar な特性を通常利用されている Pareto 分布を使って表わし、シェープパラメータ： α の値によって、long-range dependence なのか、short-range dependence なのかを表すことにする。

ファイルサイズ分布の典型的な分布は、 α の値が1と2の間にあり、その平均が10Kbyte、ミディウム値は約2.5Kbyteである。シミュレーションにおいては、このような Pareto 分布によりファイルサイズ分布が構成されていると仮定する。また、今回は新しいコネクションの確立間隔は指数関数的であると仮定する。

2.2 シミュレーションセットアップ

本研究では、ネットワーク性能評価でも利用されているシミュレータである、LBNLのネットワークシミュレータ (ns-2)[4]を利用する。

各クライアントとゲートウェイ間およびサーバとゲートウェイ間は、帯域が10Mbyte、遅延が15msecとし、ゲートウェイ間の帯域および遅延については、可変とする。また、リンクのキューサイズは十分な大きさと考えられる3000パケットとする。クライアントとサーバ間ではTCPによるコネクションが張られ、その上位層でFTPによりファイルサイズが Pareto 分布のファイルを送送する。分布については α の値を、1.05, 1.35, 1.65 および1.95と4つの値に関してシミュレーションすることにより、long-range dependence なトラフィックと short-range dependence なトラフィックとの相違を検証する。また、そのコネクションに対応したフローは各クライアントに対してそれぞれ50個のフローを

発生させる。シミュレーション時間は 20 秒としているが、それは前述したフローがほぼ終了する時間を設定している。また、それらの TCP コネクション到着は Poisson プロセスに従っているものとする。また、TCP の輻輳制御のエージェントは NewReno を利用する。

3. シミュレーション結果

シミュレーションにおいては、トラフィックの重なりを考察するため、まず 8 クライアントのみによるデータ転送を考え、それを 16 クライアント、24 クライアントおよび 32 クライアントへと増加させていった。

3.1 ロス率

パケットロス率はサーバ (S) およびゲートウェイ (G1) において発生したので、それぞれについて考察する。まずゲートウェイ間の帯域を 5Mb と小さくし制約を受けている状態について考察する。

図 2 に帯域が 5Mb、 $\alpha=1.05$ の場合のサーバ上でのロス率の変化を示す。図から明らかかなように、クライアント数が 32 の場合のみ、立ち上がり時に若干のロスが発生しているが、他の少ないクライアント数の場合ではまったくロスは発生していない。このように self-similar なトラフィックの場合には、個数としてはファイルサイズが小さなファイルが多く存在するため、ロス率の発生も抑えられていると思われる。

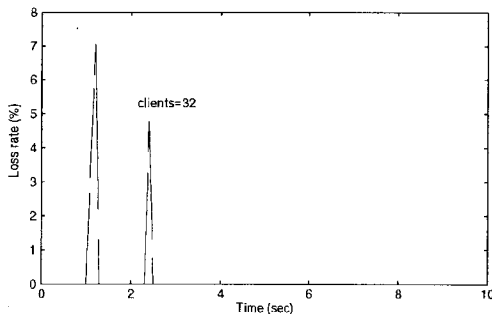


図 2. $\alpha=1.05$, 帯域が 5Mb の場合のサーバでのロス率の変化

図 2 と同様の条件で、 $\alpha=1.65$ とした場合のロス率の変化を図 3 に示す。このように、 α が 2 に近づく、すなわち short-range dependence な特徴を持つに伴って、ロス率もかなり増加することが分かる。この原因であるが、異なる α に対して、平均が同じとした場合には、 α が 1 に近い場合には、

比較的ファイルサイズが小さなファイルが数多くでき、 α が大きくなるにしたがって、そのファイルサイズも大きくなることが判明している。ただ、 α が 1 に近い場合には、非常に大きなファイルも存在するため、それがパフォーマンス全体に影響する場合もある。

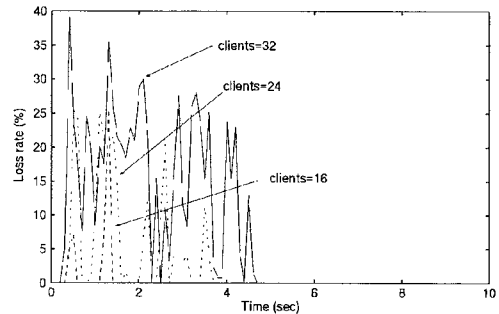


図 3. $\alpha=1.65$, 帯域が 5Mb の場合のサーバでのロス率の変化

図 3 から、帯域が制限されており α が 2 に近い場合には、クライアント数が 16 を超える段階からサーバ側でロスが頻繁に発生していることが確認できる。特に、クライアント数が 32 となれば、シミュレーションを開始してから 5 秒程度までは、10% を超えるロスが確認できる。この $\alpha=1.65$ 、帯域が 5Mb を基準にして、条件を変えながらロス率の変化を以下で検討する。

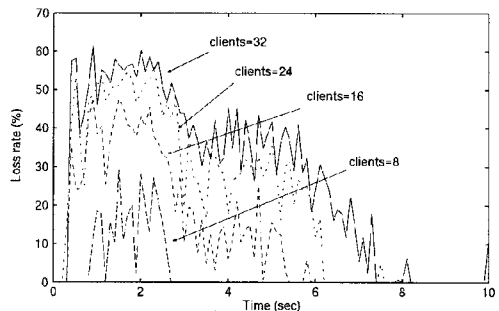


図 4. $\alpha=1.65$, 帯域が 10Mb の場合のサーバでのロス率の変化

図 3 と比べて帯域を倍、すなわち 10Mb に増やした場合のロス率の変化を図 4 に示す。単純には帯域が増えればロス率は減少すると思われるが、TCP のセルフクロッキングの機能により、より転送データを増加させる方向に制御が動き、逆に転送元のサーバ上ではロス率が 5Mb の場合と比べて大きくなっていることが確認できる。転

送量を増加させる制御はこのようなロスの発生によりしだいに抑制され、それにともないロス率も減少している。このような現象はクライアント数が 8 という非常に少ない数であっても発生している。

また、クライアント数が 8 ごと線形に増加しているが、ロス率は線形に増加するのではなく、クライアントの増加率と比べてその増加は抑えられている。

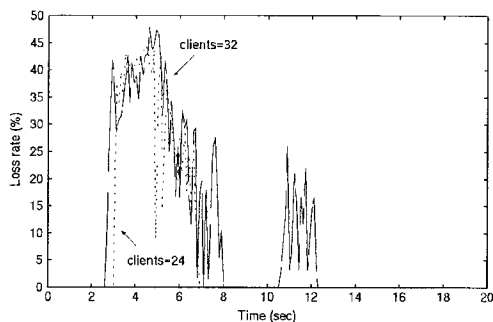


図 5. $\alpha=1.65$, 帯域が 5Mb の場合のゲートウェイ(G1)でのロス率の変化

今まで TCP のデータ送信元でのパケットロス率について検討してきたが、次に、ダウストリームトラフィックのボトルネックの送信元となるゲートウェイ 1 (G1)でのパケットロス率について議論する。

クライアント数が少なく、例えば 16 個であれば、サーバではパケットロスが発生しているにもかかわらず、G1 においてロスは発生しておらず、フローの送信元に近いホストの方がロス率は高いことが確認できる。しかし、この現象はシミュレーション開始に近い段階、すなわちフローの始まりに近い段階に限定され、サーバ上では 6 秒以降ではロスが発生していないにもかかわらず、G1 では、クライアント数が 32 の場合には 10 秒以降においてもロスが発生している。また、2 秒から 6 秒の間においては、クライアント数が 24 の場合と 32 の場合において、同程度のロス率を確認できた。このことは、ボトルネックのゲートウェイにおいてトラフィックフローの増加に比例してロス率の増加が決まるのではなく、ゲートウェイにたどりつく前の TCP のフローの制御の状態に依存していることがわかる。フローが開始した初期の時間帯においては、TCP が送信元においてロスを発生させながら、フローの量を制御し、2 秒ほど経過して、ゲートウェイにおいてロスの発生が開始する。送信元では、ロスが継続して発生しながらも、フローが制御される方向に向

かって、ゲートウェイでは少しずつロス率が減少していく。これらの現象は、ある程度大きなクライアントに対応したフローには同様に働き、ボトルネックとなるゲートウェイでのロス率にそれほどの差異は発生させていない。いったん、TCP の送信元でフローが落ち着くと、送信元でのロスの発生はなくなり、ゲートウェイでロスが発生する。この場合は、クライアント数が大きな場合にのみ発生することが確認できた。 $\alpha=1.65$ の場合しか図示していないが、送信元の場合と同様に、 α が 2 に近づけばロス率も増加する傾向にある。ただし、線形に増加するのではなく、例えば、 $\alpha=1.35$ と 1.65 の場合のロス率がそれほど異なる場合も存在する。

3.2 スループット

2 番目の指標としてスループットについて検討する。今回はボトルネックリンクでのトラフィックの重なり注目しているので、ダウンロード元の G1 におけるスループットを計測している。これは G2 が受信したパケットによってスループットを計算しており、前述したロスしたパケットはカウントしていない。また、その時間帯に何個のセッションが張られている途中なのかについて、データを収集したので、それを図示する。ただし、このセッション数は終了したセッション数となっており、途中のセッション数については含まれていない。

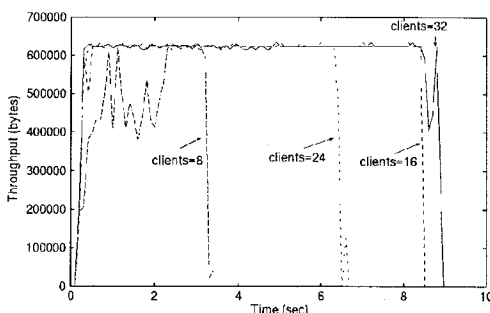


図 6. $\alpha=1.05$, 帯域が 5Mb の場合の G1 でのスループットの変化

まず、図 6 に $\alpha=1.05$ 、帯域が 5Mb の場合のスループットの変化を示す。また、その時のセッション数を図 7 に示す。図 7 から、クライアント数が増加するにしたがって、セッション数も増加することが分かる。図 6 からは、10 秒以内にすべてのフローが終了しスループットが 0 になっているため、図 7 に示したセッション数はすべてのフロ

一を示していることになる。

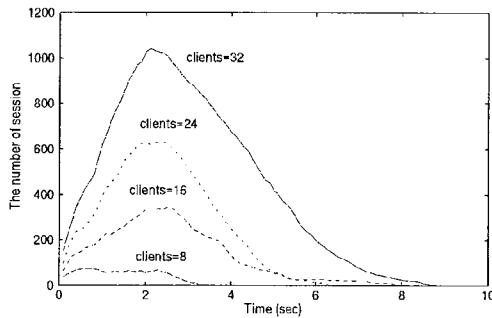


図7. $\alpha=1.05$,帯域が5Mbの場合のG1でのセッション数

ここで、図6を見ると、クライアント数が24のフローが、16個のフローより早く終了していることが分かる。一般的には16の方がフローの数が少ないため、早く終わると予測できるが、この場合は、 $\alpha=1.05$ の場合であり、非常に大きなファイルが存在する可能性がある。具体的にデータをチェックしたところ、クライアント数が16の場合、非常に大きなファイルの存在により、ほんの1,2セッションが継続して流れており、そのフォローが全体のスループットを構成し、結果として32クライアントの場合よりも長時間フォローが存在していた。この結果から、 α が1に近いself-similarなトラフィックでは、わずかのフローにより全体のフローの継続時間が延びる可能性があり、時間としてみるとクライアント数が多い場合よりも、長く継続する可能性があることが発見できた。

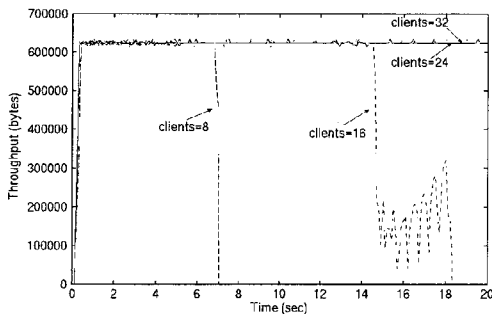


図8. $\alpha=1.65$,帯域が5Mbの場合のG1でのスループットの変化

図8は、 α を2に近い方に近づけ $\alpha=1.65$ とした時のスループットを示し、その時のセッション数を図9に示している。図8から、クライアント数が24と32の場合のみ

だ帯域全体が利用されており、フローが終了していないことが分かる。そのことを前提として考えると、図9ではクライアント数が24のセッション数が、32の場合を超えている理由としては、まだ32のセッションは途中のものが大半で、単にシミュレーション時間内に終わったものが少ないため、総数では24より時間的に変化するセッション数が少なくなっている、ことにより説明できる。

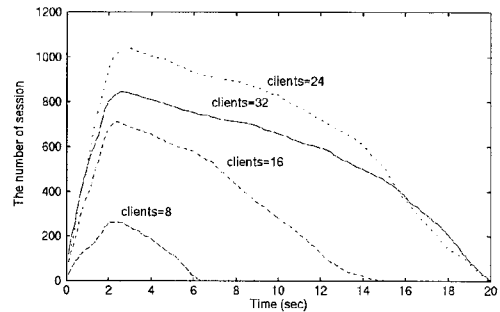


図9. $\alpha=1.65$,帯域が5Mbの場合のG1でのセッション数

セッション数の時間的変化を図7の $\alpha=1.05$ の場合と比較すると、同じ条件であっても、 α が2に近くなれば、セッションも長く継続し、終了時間も遅くなる。これは α が2に近い場合のファイル分布が、平均値が同じであれば、比較的大きなサイズのファイルが多くなることが原因となっている。図8に示されているクライアント数が16で、シミュレーション時間が14秒から18秒までの間の振動は、数少ない非常に大きなファイルが存在し、それらへの転送のみでボトルネックリンクが占有された場合であり、スループットは振動する。

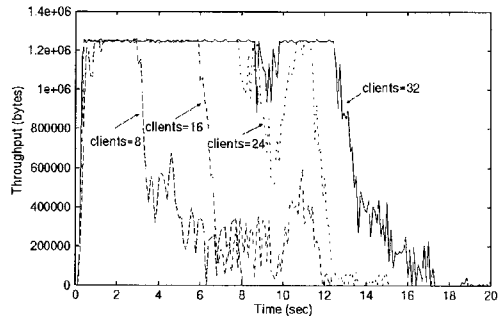


図10. $\alpha=1.65$,帯域が10Mbの場合のG1でのスループットの変化

図10に、図8の場合の2倍の帯域をボ

トルネックリンクに割り当てた場合のスループットの変化を示す。またその時のセッション数の変化を図11に示す。スループットの値がシミュレーション時間内において0に収束しており、このことからすべてのフローを構成するセッションが終了していることがわかる。それを前提にして、図11を見れば、図9ではクライアント数が24の場合が32の場合のセッション数を上回っていたが、すべてのセッションが終了した図11から、32のほうがセッション数が多くなり、14秒程度でほとんどのセッションが終了していることが読み取れる。

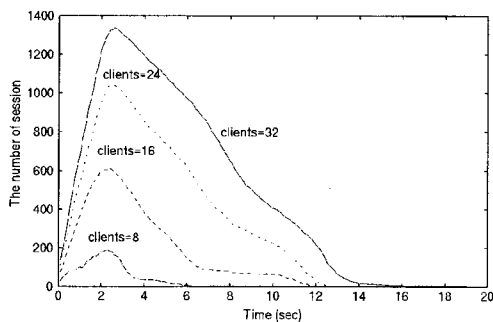


図11. $\alpha=1.65$,帯域が10Mbの場合のG1でのセッション数

図10に示すスループットの変化において、いったん帯域がすべて利用された後、どのクライアント数の場合においても振動が観測できる。これは、前述したように数少ない非常に大きなファイルの存在により帯域をすべて利用せず、転送量が頻繁に変化することによって発生している。

4. おわりに

本研究では、データ分布が Self-similar な特性を持ち、一定数のクライアントが閾値を作りながら線形的に伸びていくトポロジーについて、そのボトルネックにおけるネットワークパフォーマンスについてシミュレーションにより考察した。

実験結果から、第一に、 α が1に近い self-similar なトラフィックのほうがロス率が低い。第二に、ボトルネックでのロス率は、クライアント数とは比例せず、送信元のロスの発生に依存しながら、ある程度の大きなクライアント数によって出現する。第三に、数少ない非常に大きなファイルサイズの転送にスループットは影響を受けるなどの現象を発見できた。

今後は、TCP 輻輳ウィンドウの観測を通して self-similar なトラフィックに対して

パフォーマンスを向上させるパラメータを見つけていきたい。また各ゲートウェイにおけるキュー長を観測することにより、パフォーマンスを向上させるキュー制御方式について検討していきたい。

参考文献

[1] Leland, W., Taqqu, M., Willinger, W. and Wilson, D.: On the Self-Similar Nature of Ethernet Traffic (Extended Version), IEEE/ACM Trans. on Networking, Vol.2, No.1, pp.1-15, February (1994).

[2] Park, K., Kim, G. and Crovella, M. E.: The Protocol Stack and Its Modulating Effect on Self-similar Traffic, Self-Similar Network Traffic and Performance Evaluation, K.Park and W.Willinger, Eds., Wiley-Interscience, New York, pp.349-366, (2000).

[3] Crovella, M. and Bestavros, A.: Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes, IEEE/ACM Trans. on Networking, Vol.5, No.6, pp.835-845 (1997).

[4] UCB/LBNL/VINT groups, UCB/LBNL/VINT Network Simulator, <http://www.isi.edu/nsnam/ns/>, May (2001).

[5] Nakashima, T.,Tsuichihara, M.: Stability in Web Server Performance with Heavy-Tailed Distribution, Proc. of 9th International Conference of Knowledge-Based Intelligent Information and Engineering Systems(KES 2005), Vol. 1, pp.575-581, (2005).

[6] Nakashima, T.: Self-similar Property for TCP Traffic under the Bottleneck Bandwidth Restrainment, Proc. of Second International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IHMSIP 2006), pp.193-196 (2006).