

インターネットにおけるパケット転送のCAモデルとウィンドウ制御

龍谷大学理工学部 廣清八千代, 西成活裕

1. はじめに

インターネット通信でのパケット転送は現在 TCP/IP プロトコルに従って行われている。このプロトコルにはパケットの渋滞、つまり輻輳を避けるために様々なアイデアが盛り込まれている。しかし、ネットワークの混雑は日常しばしば体験することであり、これからインターネットの利用のさらなる増加が予想される中、輻輳の根本的解決は重要な課題になっている。そのために本研究では、複雑な TCP/IP プロトコルを直接考えるのではなく、まずセルオートマトン(CA)により本質を抽出したモデル化を行い、その単純化したモデルを利用して輻輳のメカニズムや、その解決のためのパケット制御について考えてゆく。

2. 基本となるCAモデル

CA モデルを構築する前に、まずパケット転送についていくつかの単純化を行う。パケットの通るルートは途中のルーター内の IP プロトコルによって動的に決められるが、簡単のために、ひとつの通信が終わるまでそのルートは固定されていると仮定する。そして輻輳が発生してもまずはウィンドウ制御は行わず、ホストのウィンドウサイズ W は常に一定とする。また途中のルーターには、他のローカルネットワークからのパケットは来ないとする。各ルーターでは、そのバッファ内に存在するパケットはできる限り次のルーターに送ろうとし、もしも送り先のルーターのバッファがいっぱいになっていれば、そのパケットは次のルーター内で棄却するとする。以上より CA モデルは以下の通りになる。

1. ルーターからの単位時間あたりの最大パケット可能送信数を S_n とし、ルーター内のバッファの最大メモリ数を L_n とする。時刻 t に n 番目のルーターのバッファに存在するパケット数 B_n^t ($n = 1, \dots, K-2$) の変化を表す式は

$$B_n^{t+1} = B_n^t + \min(S_{n-1}, B_{n-1}^t, L_n - B_n^t) - \min(S_n, B_n^t) \quad (1)$$

で表される。ただし、 $n=0$ と $n=K-1$ はそれぞれ送信ホスト A、受信ホスト B とし、途中ルーターは $K-2$ 個あるとする。

2. あるルーターにおいてパケットが棄却されたとすると、そのパケットはすぐホスト A に戻されると考える。時刻 t において途中のルーターで棄却された総パケット数 r^t は

$$r^t = \sum_{n=1}^{K-2} \max(\min(S_{n-1}, B_{n-1}^t) - (L_n - B_n^t), 0) \quad (2)$$

で表され、この分だけホスト A でパケットが増加する。

3. ホスト A は、初めにウィンドウサイズ W の分のパケットをホスト B に向けて送信する。

そして、ホスト B から確認応答が到着しだい、再び W のパケットを送信するとする。ネットワークが輻輳になると、途中でパケット棄却量が多くなり、確認応答がしばらく届かなくなる。この場合、タイムアウト時刻を設定し、この時間までに確認応答が来なければ、W だけのパケットを送るものとする。このタイムアウト時刻は、ルーター数に比例して設定する。

4. ホスト B の時間発展は以下の式で与えられる。

$$B_{K-1}^{t+1} = B_{K-1}^t + \min(S_{K-2}, B_{K-2}^t) \quad (4)$$

つまり、ホスト B からの応答確認パケット数は毎時 $\min(S_{K-2}, B_{K-2}^t)$ となる。

以上が本研究で用いるインターネットの基本的 CA モデルである。ただし、このままでは輻輳は発生しない。実は各ルーターはそれぞれ他のサブネットと結合しており、そこから来るパケットも処理しなければならない。そしてこれが輻輳をもたらすのである。各ルーターに到着する他からのパケットを J_n^t 表すことにすると、ルーターのモデルは (1) の代わりに、

$$B_n^{t+1} = B_n^t + \min(S_{n-1}, B_{n-1}^t, L_n - B_n^t - J_n^t) - \min(S_n, B_n^t) \quad (5)$$

と表すことができる。ここで、 J_n^t に関してはポアソン分布でパケットが到着すると考え、

$$J_n^t = \min(\text{poisson}(\lambda), L_n - B_n^t) \quad (6)$$

とする。ただし、 $\text{poisson}(\lambda)$ は平均値 λ のポアソン分布乱数とする (図 1)。

ここで、次の事実を指摘しておこう。(1) より最大バッファ数を $L_n = 1$ とおくと、このモデルはルール 48 のセルオートマトンになっていることがわかる。車のモデルはルール 184 と関係があるが、パケット転送はルール 48 であることが興味深い。

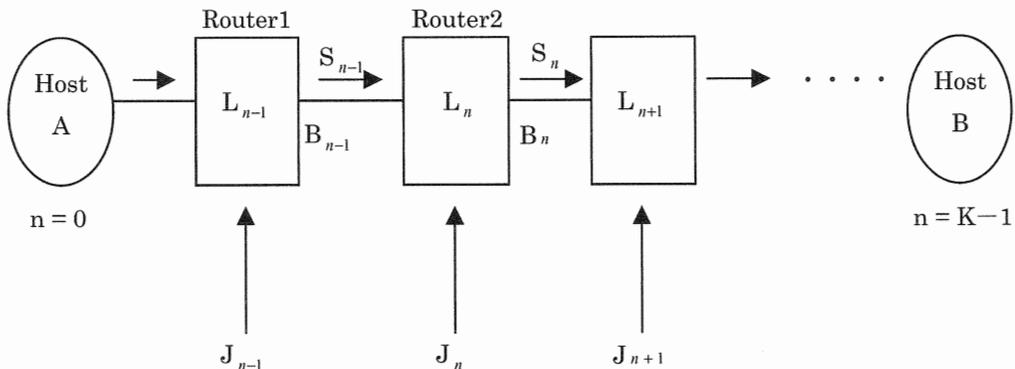


図 1 : インターネットの拡張 CA モデル

3. シミュレーション結果

図2はパケットの転送時間を計測した結果である。ホストAの初期パケット数はすべて7とし、ルーター数は30とする。そして、ホストAのウィンドウサイズWを2から7まで変え、ポアソン分布の平均値を0から0.25刻みで4.75まで増やしていき、それぞれの場合について総転送時間を調べた。データは10回の計算の平均値を記してある。網がけの部分是最も転送時間が短かった場合で、ポアソン分布の平均値の増加、すなわち混雑の度合いが増すにしたがって、ウィンドウ数が小さい方が転送時間が短くなる傾向があるのがわかる。

	W=7	W=6	W=5	W=4	W=3	W=2
0.0001	29	58	58	58	61	87
0.2501	29	58	58	58	61	87
0.5001	31.9	58	58	58	61	87
0.7501	40.6	58	58	58	61	87
1.0001	34.8	58	58	58	61	87
1.2501	46.4	58	58	58	61	87
1.5001	60.9	58	58	58	61	87
1.7501	58	58	58	58	61	87
2.0001	58	58	60.9	58	61	87
2.2501	60.9	60.9	58	58	66.2	87
2.5001	64.1	63.8	58	66.9	63.6	87
2.7501	75.4	69.6	70.1	58	69.1	87
3.0001	75.9	78.5	78.3	67	71.4	90.1
3.2501	78.6	75.4	84.9	85.9	74	87.9
3.5001	98.9	85	98.3	95.3	71.7	90.5
3.7501	96.5	115.4	95	120.6	96.7	116.4
4.0001	119.4	146.5	118.4	135.7	105	127.2
4.2501	143.5	164.6	147	129.1	135.7	134.4
4.5001	184.4	170.1	166.6	176.3	156	162
4.7501	200.6	205.4	198.4	191	207.8	230.3

図2： パケット転送にかかる総時間とウィンドウサイズ、サブネットの影響

この結果により、ウィンドウ数が小さいほど、混んでいるときはサブネットからの邪魔なパケットの影響を受けずに、つまり棄却量も少なく送れる事が分かる。また、ネットワークがすいているときは棄却が無いので、初めから大きなウィンドウサイズで送れば早く送れることも分かる。混雑時には、棄却再送のせいで大きなウィンドウサイズのままでは不利になることが明らかになった。

4. ウィンドウ制御

最後に、ウィンドウを一定にせず、輻輳に応じて変化させることを考えよう。こうすることで、輻輳になってもウィンドウサイズを小さくせずに、常に大きなウィンドウサイズで送れるようにすることはできるだろうか。その結果が図3である。ここで、ウィンドウサイズは棄却量(2)に応じて変化させるものとし、この(2)の値がある臨界値 r_0 をこえたら、輻輳対策としてウィンドウサイズをその次の時刻だけ動的に W_d だけ下げる、というものである。

	W=7	W=6	W=5	W=4
0.0001	29	58	58	58
0.2501	29	58	58	58
0.5001	29	58	58	58
0.7501	29	58	58	58
1.0001	34.8	58	58	58
1.2501	46.4	58	58	58
1.5001	55.1	58	58	58
1.7501	55.1	58	58	58
2.0001	58	58	58	58.3
2.2501	60.9	63.8	58	58.6
2.5001	60.9	73.3	69.6	61.5
2.7501	67.2	63.8	69.6	63.8
3.0001	73.4	72.5	78.6	67.6
3.2501	76.8	79.4	79.1	76
3.5001	83.8	88.5	92.1	90.7
3.7501	109.6	88.9	104.1	94.3
4.0001	119.5	109.8	115.1	124
4.2501	159.6	136.4	148.7	143.2
4.5001	188.8	160.2	186.3	176.9
4.7501	185.4	205.6	231.8	228.1

図3： パケット転送にかかる総時間とウィンドウサイズ、サブネットの影響 ($r_0=1$ 、 $W_d=3$)

これにより、強い輻輳がおこっていた $\lambda = 3$ 以上で、再び W が大きいほうが総転送時間が短い場合が出現しているのが分かる。つまり、ウィンドウ制御がうまく働き、輻輳が確認された時のみウィンドウサイズを絞ることによって、初めからウィンドウサイズを絞らなくても早く送れることが分かった。

参考文献：Katsuhiko Nishinari, Trans. Mat. Res. Soc. Jan., Vol. 26(1) (2001) pp.453-456.