網輻湊時のフロー制御の一評価

川村　克彦[i]、Douglas Browning[ii]、五反田　隆広[i]、佐久間　幹郎[i]
（[i]沖電気工業㈱　[ii]Princeton大学）

1.はじめに

網を利用することを考えた場合、次のことに注意する必要がある。

i)　データ発生源は、不特定多数
ii) 使用されるリンク経路は、未定
iii)各リンク内でのデータ発生量は、一定水準以下であるが、Node 間でのデータ発生量は、未定

このことから、網利用者側や網提供者側から、網利用形態や状態について把握出来ない場合が多い。

そのため、例とえば、あるNode の一つのPort に対して、データの流れが集中するという輻湊状態が起こる可能性が高くなる。その後、そのPort 内の処理がデータの流れに追い付かずに、データの紛失を引き起こす。またデータ発生源では、データの紛失に対して再送等の制御を行うので、輻湊状態にあるPort の輻湊が解除されないという、通信の飽和状態に陥ってしまう。そして、最悪の場合、その輻湊を起こしたPort を中心として、網全体に輻湊が広がってしまう。

通常、利用者側で網内の輻湊の対策を行うことは不可能な場合が多い。そのため、その対策は、網の提供者に任せられることになる。そこで、網内にフロー制御を導入し、その効果的な利用方法について、一例を挙げて評価してみた。

2.フロー制御と通信効率

2.1.スループットと遅延

輻湊が引き起こす最大の欠点は、通信遅延が大きくなることである。逆に、輻湊が発生していないことを判断するために、通信遅延の大きさが一つの基準となる。

通常スループットを向上させると遅延が大きくなる。また、スループットの上限は、100 %であり、その時の遅延は、限り無く大きくなる。そこで、スループットと遅延の関係をもう少し詳しく見ると、低負荷においてはほぼ比例関係にあると見なされるが、高負荷においてはスループットの向上の割合より遅延の伸びの割合の方が、飛躍的に大きくなっている。

そこで、スループットと遅延が、ほぼ比例関係を保っているところでの最大のスループット値を、最適スループットとして、高負荷のもとでの安定性の基準とする。この値は、Packet 網については、70%程度である。（グラフ0　参照）

2.2.フロー制御とその問題点

輻湊状態を解除するための基本的な考え方は、輻湊を起こしたPort に、輻湊が解除される程度にまで、データの流れを制限することである。具体的な手段としては、データ発生源に対し、制御情報を流すことである。

つまり、データの流れを制御するために、データを流すという方法を採るのである。この二つの矛盾したデータの流れ

が同一経路上に混在するため、次のような問題が起こる。

i) 制御情報の流れは、データの流れの妨げになる。つまり必要以上の制御情報が網内に氾濫した場合、本来のデータの流れがその分妨げられる。

ii) i)の逆で、制御情報の量が少ない場合、網内のデータの流れは刻一刻と変化するためその変化に対応出来ず、網内のデータの流れを制御出来なくなる場合がある。

iii)通信の飽和状態に陥った後では、制御情報を流すために使用するBufferすら確保出来ない状態にあるため、輻湊解除は実現しにくい。

iv) 制御情報を流すための遅延が必ず存在するため、制御効果の現われるまでには、ある程度の時間が必要となる。そのため、極く短時間発生する輻湊については、制御することが、不可能となる。さらに、その輻湊を制御しようとしたことで、輻湊の直接の原因となったデータの流れに対してでなく、他のデータの流れを妨げることになる。

以上の様な問題の他にも次の様な問題も解決する必要がある。

v) フロー制御機能の複雑さと効果
フロー制御を実現する方法は、出来る限り簡素化されていること。

vi) 制御の完全性
なるべく多くの場合の輻湊を解除する。

2.3.網内の望ましいデータの流れ

フロー制御を網に導入することによって、輻湊状態の解除を主として、網全体は、次の機能を持つことが望ましい。

i) データの流れが多い場合、最適スループットを維持する。
輻湊解除のため、制御情報が適度に流れている状態

ii) データの流れが少い場合、最大スループットを提供する。
輻湊の解除よりデータの流れを重視し、あまり制御情報を流さない状態

2.4.フロー制御例

ここで、フロー制御方式の例とその完全性について、簡単にまとめてみた。[1]

。Input Buffer Limit
入力Node側で、入力データ発生量を監視、制限する。

。End−to−End
入力Nodeと出力Node間でRR(Receive Ready)の送受信を行う。RRの送受信に一定以上の時間が必要な場合、入力Nodeは、データの流れを制限する。

。Congested Node Control
入出力、中継の区別を問わず、輻湊を起こしたNodeから入力Nodeに対し制御情報を流す。それをうけて入力Nodeは、入力データの流れを制限する。

2.5 制御の完全性

次の様なデータの流れを考えてみる。

i) 複数Node から単一Node の同一Port にデータが流れる場合
ii) 大量のデータの流れを中継している場合

。Input Buffer Limit
i)、ii) 共に制御不可能である。

。End－to－End
i)については、制御可能である。
しかし、、ii) では、そのNode で提供する本来の経路が混雑しているため制御効果が薄れる。それ故、最終的には輻湊状態に陥る。

。Congested Node Control
i)、ii) 共に制御可能である。

3. シミュレーション モデル

3.1 網のモデル

網は、図１の様にモデル化した。各々のモデル要素は、次の役割を持つ。

・ノード: 全てのノードは、発信［S］中継［R］、さらに受信［D］ノードとしての機能を持つ
・網利用者: 入力待ち行列、データの発生源であり、またデータ入力制限を受けるポートでもある。
・入力ポート: 入力待ち行列、ノード間転送のためのポート
・出力ポート: 出力待ち行列、ノード間と網利用者の両用のポート
・処理部: 入力データの経路を決定し、ポートからポートへデータを移したり、出力ポートの待ち行列の長さを監視して制御情報を発生したりするプロトコル部である。

図１にデータの流れの例を実線矢印で示してある。

3.2 フロー制御のモデル

網内で輻湊が起きた場合、それを解消しようとして網は次の様に動作する。

図１のデータ通信において［R］ノードの（O）で輻湊が発生する。（P）がそれを検出すると、その（P）は、［S］ノードに対して制御情報を送信する。［S］ノードの（P）が、それを受信すると［R］に対するデータ送信を一時中止させる。

制御情報の送信の様子は、破線矢印で示してある。

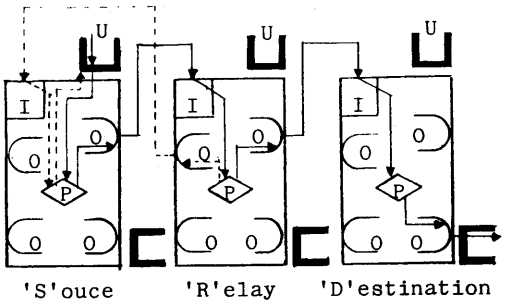


図１ モデル

記号概説

（U）： 網利用者
（I）： 入力ポート
（P）： 処理部
（O）： 出力ポート

3.3.評価するフローモデル

。輻湊の測定

一定周期で輻湊の判断を行う。

。輻湊の判断

制御を簡素化するために、Threshold (T)による制御方式をとる。つまり、出力待ち行列の長さが、Tを越える場合輻湊中であると判断する。

。Buffer 上限

使用Buffer は、有限Buffer を考えBuffer Size (B)を越えるデータについては、廃棄する。

。遅延

最適スループットに対する通信遅延(I)は、3.5(sec)を評価の目安にした。

。制御情報

輻湊の原因となった全てのデータ発生Node に対してのみ制御情報を流す。

。データ発生

Poisson分布に従う。また、各々のリンク内での最大のデータ発生量は、スループット値で70%である。

。最小間隔

データの流れが、Node 間で要する時間を単位として、制御情報の発生から次の制御情報の発生まで待たされる時間である(S)。その間で、輻湊の判断が行われていても無視される。

。制御水準

データ発生Node では、制御情報を受け取るとデータの流れを一時停止する。その後、除々に以前のデータの流れに戻す。そして、どの位でもとに戻すのかを輻湊の測定の周期を単位として、回数で示す。

N回で以前のデータの流れに戻る場合、N水準の制御を行ったとする。その中のそれぞれの値は、水準0から水準N－1で呼び、水準0は、制御を受ける以前のデータの流れを表わし、また水準N－1は、0を表わす。

つまり、制御情報を受け取ると、水準N－1の状態になり、それから、水準N－2と続き、最後には、水準0に戻る。

3.4.シミュレーション方法

Node 内のPort をBuffer 上限を持つMarkovian待ち行列として表現し、その状態遷移を調べることによって行う。具体的には、次の様な状況を設定する。

i) 各Node がある時間から、あるPort に対して一斉にデータを流し始める。そして、そのPort に輻湊状態を発生させる。

ii) その後、そのの状態が、安定な状態になってゆく様子を観察する。

3.5.評価し得る項目

フロー制御を評価する場合、制御の動作からその効果が現われるまでの遅延が、重要な要因を含んでいる。その上で、次に示す要因が、網全体のデータの流れの変化に、どの様に影響を及ぼすかを調べる必要がある。

i) 輻湊Node 側の動作

輻湊を検出したNode が、データ発

生源Node に対して、どれだけの個数の制御情報を単位時間に流すか、つまり最小間隔をいくらにするのが最も効果的であるのか

ii) データ発生源Node の動作
制御情報を受け取ったデータ発生源Node では、どれだけの期間、どの位、データ発生を押さえる（減らす）か、つまり、制御水準をどの位にするか

iii)制御情報の伝達速度
制御情報に優先権を与えた場合、制御情報の伝達速度が大きくなると考えられる。その時、網に与える影響はどの位になるのか、つまり、優先権の導入は、どの位効果をもたらすか

iv) 制御水準の遷移
N水準の制御を行った場合、各水準 i が、どの様に遷移するのが効果的か、つまり、水準 i の値を定める関数は、どの様なものが良いのか

4.結果と解析

。最小間隔　（グラフ１，２ 参照）

輻湊の判断が行われると同時に制御情報を流した場合、データの流れが安定するまでの間に、かなり大きな変化を繰り返している。これでは、制御情報による十分なデータの流れの制御が出来ているとは言えない。これは、必要以上に制御情報が作られているためである。

このことは、網の動きを詳しく調べることによって容易に理解出来る。

i) 初めの輻湊が測定されると、その Node は、データ発生源に対して、制御情報を流す。

ii) 制御情報が、データ発生源に到着するまでの間、データの流れは減少しないため、輻湊は更に進行する。

iii)データ発生源がデータの流れを減少させるが、輻湊がある程度進行しているため、輻湊状態が解除されるまでには、至っていない。そこで、さらに制御情報を流すことになる。

iv) iii)は、輻湊解除の直前まで行われる。そのため輻湊が解除された後、データ発生源ではデータをほとんど流さない状態が続くことになる。

v) データの流れが急速に少なくなったため、輻湊を起こしたPort 内の待ち行列の長さが更に短くなっている。そして、待ち行列の長さが再び長くなり輻湊の判断が行われる頃には、データ発生源では初めの輻湊が測定される直前のデータの流れに近付く。そのため、再び i) に近い状態に戻ることになる。

この動作は、輻湊が解除される直前まで制御情報が流され続けることに原因がある。そこで、制御情報を流す量を適度に減らすことで、データの流れの変化量を小さくすることが出来る、と考えられる。

グラフ１によると、最小間隔を次第に大きくすることで変化量が小さくなってゆくことが、分かる。

しかしながら、最小間隔が４を越える

あたりから再びデータの流れの変化量が大きくなり始める。これは、制御情報の流れる量が少くなすぎるため、輻湊を解除出来なくなったためである。

制御情報の発生は、輻湊の判断がされている場合以外は起きない。そして制御情報を流すだけで網全体のスループット制御を試みる場合、その発生量に頼らざるを得ない。しかしながら、以上の考察で判る様に制御情報の発生量は、網の動作の安定性を定める要因となっていることが判る。そのため、網全体のスループットの制限については、他の要因に頼る必要がある。

。入量制限　（グラフ３，４　参照）

制御水準段階が大きくなるほど、一回の制御によってデータの流れの平均が小さくなる。つまり、制御水準段階を増すと、スループットが低下する。

制御情報の量では、スループットを制限出来ない。そこで、網のスループットを制限するためには、制御水準段階を変化させる必要がある。あるＰort に対するデータ発生源の数が多くなるほど、制御水準段階を殖やさなければ、最適スループットが得られない。そして、その値は、データ発生源個数の約３倍程度である。

例外として、データ発生源の数が１の場合は、３水準の制御を行うと、最適スループットより低く制御されてしまう。この場合は、データの流れの総量が、最適スループットを越えていない。つまり、この状態では、何も制御しないのが一番良い結果を得られる。

また、データ発生源数が等しい場合、平均距離が長居ほど、制御の効果が緩やかになるため、スループット遅延ともに大きくなっている。

。低負荷　（グラフ５　参照）

低負荷時には、要求したスループット通りのスループットが得られている。これは、単純に、各Ｎode で輻湊が起こりにくいという理由で説明される。つまり、フロー制御機能自体が働かず、データの流れが、網によって押さえられる割合が小さくなったからである。よって、低負荷時の動作は、最小間隔、制御水準にほとんど関係しない。

。優先権　（テーブル１　参照）

優先権を導入した場合、優先権により制御情報の伝達速度が高まるため、遅延が小さくなる。そのため、網内のＰort が安定になるまでの時間が短い。しかしながら、スループット値は、あまり向上していない。

。制御水準　（テーブル２　参照）

Ｎ水準の中の水準ｉの値を色々と変化させて測定した結果がグラフ　　である。各水準ｉの値が水準０に片寄った場合、スループットと遅延は幾分良くなる程度である。しかし、各水準ｉの値が水準Ｎ－１に片寄った場合、スループットと遅延は、劣化する。これは、データの流れを低く押さえ、そして、一定時間後、制御される以前のデータの流れに戻すからである。このためデータの流れに大きな変化が出て来る。この変化は、グラフ２において、最小間隔を４より小さくした時と類似した動作を示すためあまり良い制御方法とは言えない。（最小間隔を参照）

5. おわりに

この評価を通して次の様なことが判かった。

i) 制御情報の最低の発生間隔は、Node 間通信遅延の4倍程度が良い。

ii) 制御情報を受けた入力Node は、競合するリンク数と通信遅延の3倍を掛けた時間でもとのデータ発生量に戻る場合が、スループットと遅延の関係が良い。

i)については、殆どの場合、一様な値を固定的に与えることで、実現されるであろう。しかし、ii) については、その時折の状態で決まるため、制御情報の内に競合するリンク数を記入しておく必要がある。

*参考文献

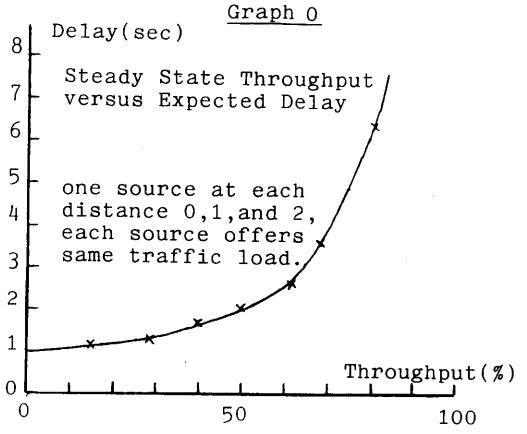
(1)

L. Pouzin

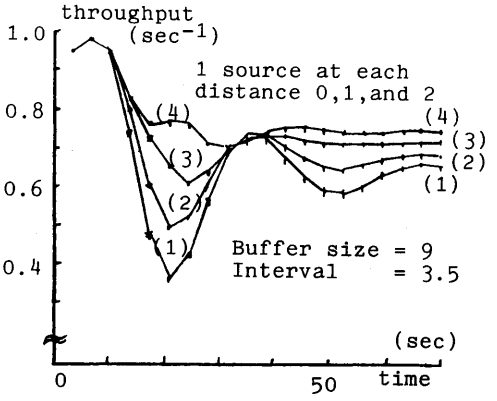
" Methods, Tools, and Observation Flow Control in Packet-Switched Data Network"

IEEE Transactions on Communication Vol. Com-29, 4 (April, 1981)

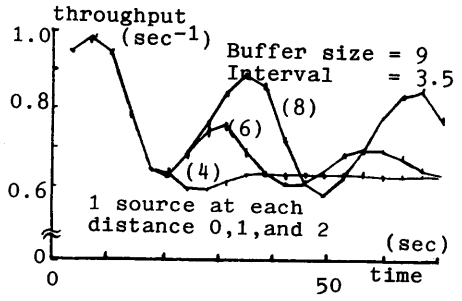
Graph 0



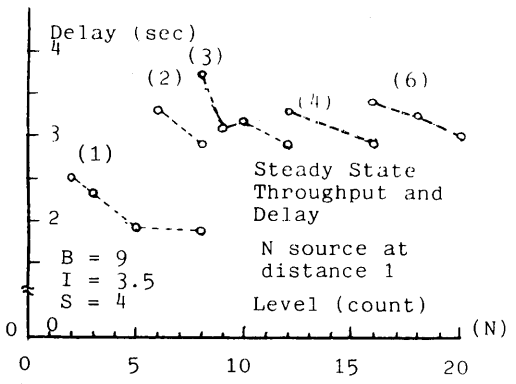
GRAPH 1



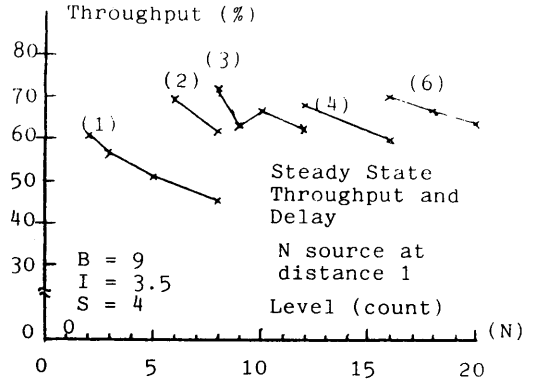
GRAPH 2



Graph 3



Graph 4



Graph 5

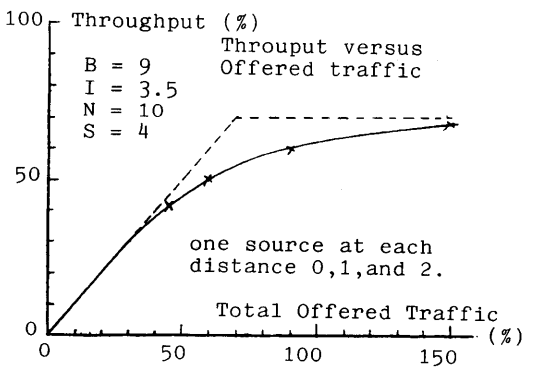


TABLE 1
PRIORITY VERSUS NON-PRIORITY
CONTROL MESSAGE STEADY-STATE
THROUGHPUT AND DELAY

B=9 I=3.5

(1) Number of active sources at distance
(2) Non-priority Throughput
(3) Non-priority Delay
(4) Priority Throughput
(5) Priority Delay

N	S	(1) 0 1 2	(2)	(3)	(4)	(5)
8	2	1 1 1	.6682	3.261	.6626	3.197
		0 3 0	.6487	3.611	.6535	3.318
8	4	1 1 1	.7486	3.738	.7362	3.606
		0 3 0	.7546	4.174	.7257	3.706
10	4	1 1 1	.6949	3.335	.6929	3.282
		0 3 0	.6985	3.713	.6669	3.261
12	4	1 1 1	.6537	3.052	.6300	2.863
		0 3 0	.6555	3.374	.6236	2.961

Graph 6

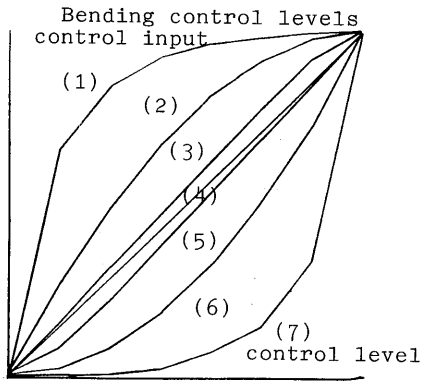


TABLE 2
ALTERNATIVE VALUES OF CONTROL LEVELS

N	S	Bending	Throughput	Delay
8	4	(1)	.5388	3.170
		(2)	.6247	3.180
		(3)	.7033	3.432
		(4)	.7362	3.606
		(5)	.7587	3.785
10	4	(6)	.7112	3.430
		(7)	.8676	4.778

B=9 I=3.5

One source at each of the distances 0, 1, and 2.