

制御センタを持つネットワークのプロトコル

三木彬生, 萩野隆彦, 近谷英昭
(国鉄・鉄道技術研究所)

1. まえがき

国鉄では現在100台以上のコンピュータが、旅客、貨物、運転、経営、研究、教育の各分野で使用されている。その約1/4を占める大型コンピュータは、ほとんど旅客、貨物の予約、新幹線の運転管理等の大規模なオンラインシステムとして、それぞれ専用のネットワークを持って稼働している。

これらのネットワークを共用化し、さらにはシステム間の有機的な結合による総合的な情報システム形成の機運が高まり、部内外で検討が進められている。ここでは、トラヒックの高いビジネス型のオンラインシステムが主流を占めるネットワークのひとつの形態として提案された、集中制御センタを持った分散形構成について、その制御方式とプロトコルについて述べる。

2. ネットワークの構成

2. 1. データの特性

国鉄のオンラインシステムは、MARS（みどりの窓口）に代表されるように、中央に大きな処理装置を持ち、全国的に分散した端末が中央と交信する形が多い。中央システムは現在、ほとんどが東京、国立周辺に集中しているため、ほとんどのデータは同じルートを通っている。またデータ長は平均約170字で90%が500字以下のメッセージであるが稀に数10k~数100k字のデータも存在する。

2. 2. サブネット

ネットワーク構成の際、最も問題となるのは、サブネットの形態である。トラフィック容量、信頼性、拡張性の点から分散形のネットワークが適しているが、上のようなデータの中央集中傾向、販売システムにおける対顧客の“公平性”等を考慮すると、単純な分散型ではなく、階層を持った分散形が最適と思われる。すなわち、サブネットを構成する交換ノード（IMP）は、中央システムをホストとする中央IMPと、地方に置かれるコンピュータをホストとする地方IMPの2種類が存在する。これらのIMPには原則的な差異はないが扱うトラフィック量、ルーティングの方法などに違いができる可能性がある。

一方サブネットを効率的に働かせるためにはルーティング、フロー制御等を効果的に行う必要がある。分散形構成の場合はこれらの機能が各IMPに分散しているため、網の信頼性が向上している反面、これらのアルゴリズムの複雑さがマイナスの効果をもたらしている。国鉄ではトラヒックのパターンが時刻によって変化する上、変動の中が大きいので、分散した、自律的な制御機構だけでは充分に対処できないと見られ、サブネット内にサブネット制御センタ（SCC: Subnet Control Center）を設けることとした。その上、SCCは（原則として）全てのIMPと、IMP間の情報パスとは物理的に独立な、制御用パスを介して接続される。この制御用パスは情報パス（48 kb/s）よりも低速の回線（例えば2.4 kb/s）で充分である。制御用パスが情報パスと論理的のみならず物理的にも独立しているため、最も制御の必要なサブネットの混雑時や回線のダウン時に制御が不能になるという事態

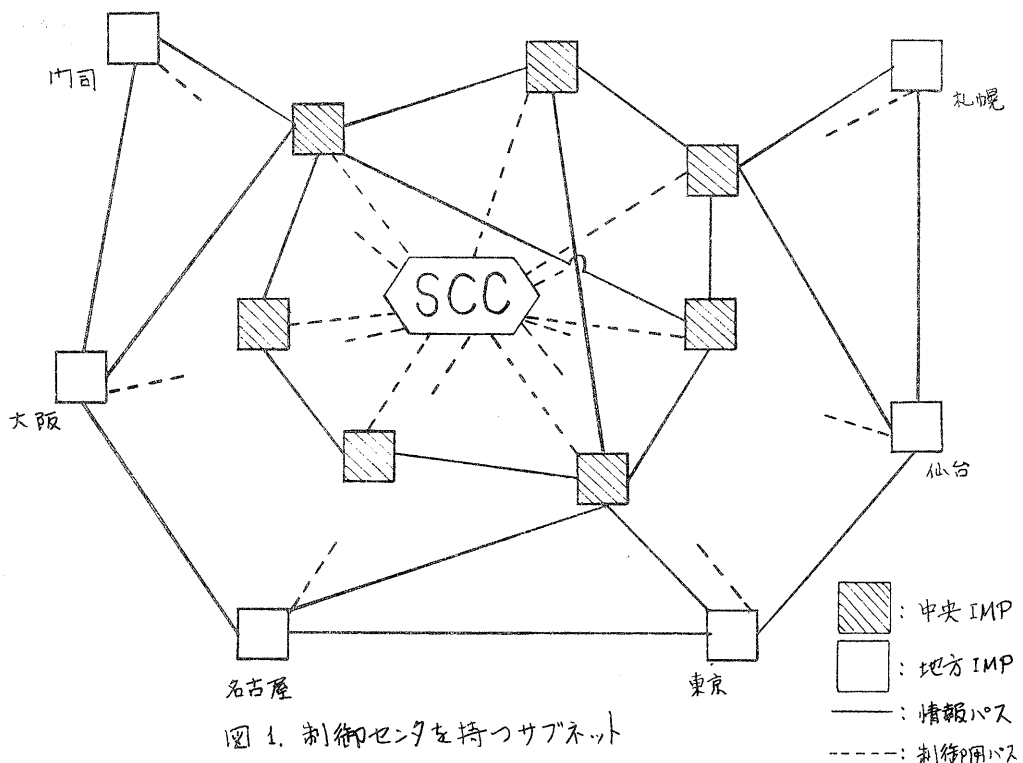


図 1. 制御センタを持つサブネット

が回避される。

なお、SCC の存在により、網全体としての信頼度が低下することのないよう、SCC 又は制御用パスのダウン時もサブネットは自律的に動作するよう考慮されている。以上、サブネットのトポロジの要約をすると以下ようになる。

- (1) IMP は中央 IMP, 地方 IMP の 2 種類存在する。
- (2) 地方 IMP は中央 IMP のいずれかに直接の情報パスを持つ。
- (3) 中央 IMP は分散形のネットワークを構成している。
- (4) 各 IMP はサブネット制御センタに対し情報パスと独立の制御用パスを持つことを原則とする。

2. 3 TCP

TSS 指向のシステムでは、端末を 1 個のプロセスとみなして、それぞれがホストコンピュータとリンクを構成して通信をすることが多いが、多数の端末が、ホストの稼働中、常時接続されているオンライン・ビジネスシステムにおいては、これらの端末をひとまとめにして制御するためのプロセスを置くことが有利である。このプロセスを TCP (Terminal Control Processor) と呼び、ネットワークに加入するすべての独立した端末はこの TCP を介することとする。したがって ARPANET における TIP に相当する IMP はサブネットには存在しない。TCP はサブネットから見るとホストの一種であり、後述するように、ホスト-ホストプロトコルによるフロー制御の重要な機能を分担する。

3. 機能の分担

ネットワークで行うべき機能とその分担を図2に示す。ここではそのうち主要な3項目について、設計思想を述べる。特にサブネットに関する部分はSCCとIMPの機能分担が問題となる。

3. 1. ルーティング

SCCの存在下では網全体の情報が比較的容易に入手できる可能性があるため、ルーティングの決定に自由度が大きくなる。かといってIMPが各パケットの到着毎にSCCにルート選択を依頼するという方法は応答性、信頼性等から不適当であり、IMPだけでかなりの程度自律的にルート選択を行う必要がある。

ルーティングの問題点は

(1) 個々の到着パケットに対して次の出力パスを割当てること。

(2) 上の決定を下すための基準をよえること

に命じられる。(1)はいわばフォアグラウンドの作業であり、ある時点である同一の性格を持ったパケットに対してルートの選択肢が一気に定まっているか、複数個あるか、さらに後者の場合最終的にどのような方針でルートを決定するかが問題となる。(2)はバックグラウンド的な、ルーティングの方法論であって、時間変化

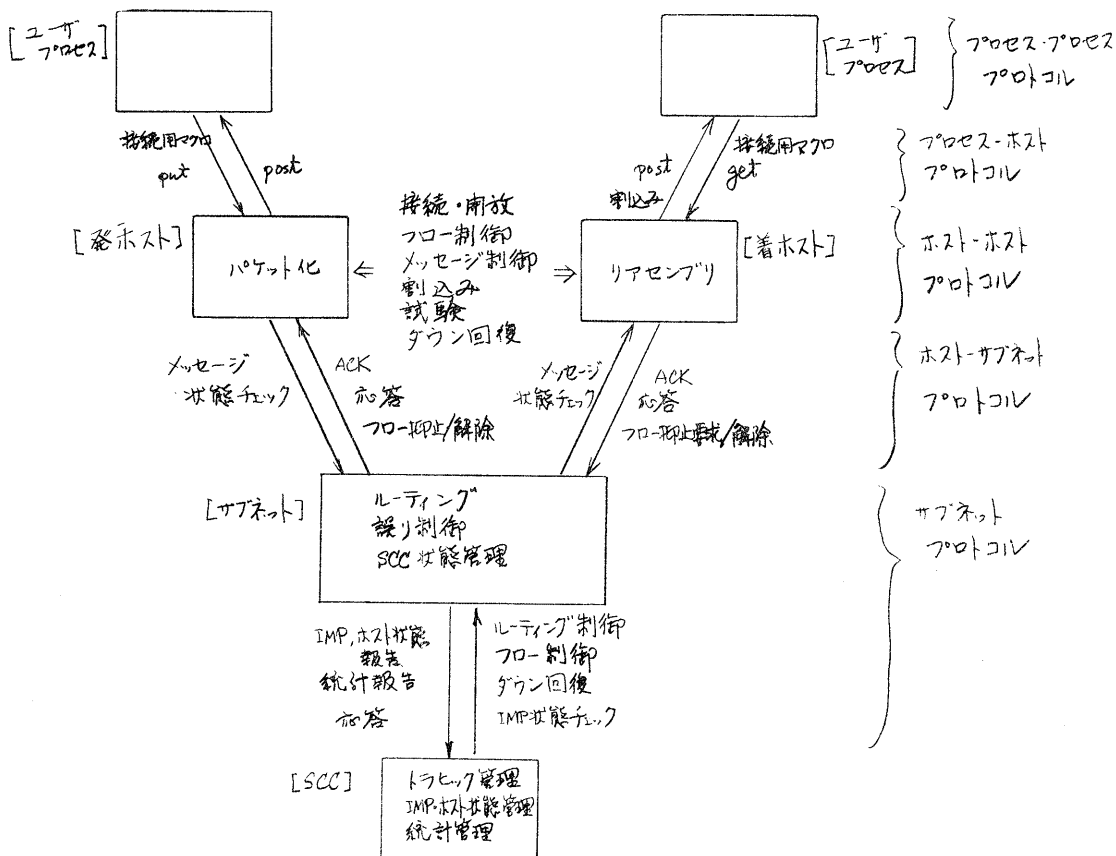


図2. ネットワークにおけるプロトコルの機能分担

あるいはトラヒックに対する適応性の有無が問題となる。SCCの存在するこのネットワークでは(1)をIMPプログラム、(2)をSCCプログラムに機能分担させることが適切と思われる。

IMPプログラム ルート決定時に選取肢が複数存在することは、計算オーバーヘッドがある反面、ネットワークのダウンの一因であるパケットのバーストがあるIMPに集中する確率を下げることに、情報パスや隣接IMPのダウンに対する対処がそれほどクリティカルな問題にならない等の利点がある。このサブネットのようにSCCにルーティング方針の決定権を委譲した場合、そのダウン時を考慮して複数選取肢を持つことが必要であろう。この際、ルート決定の方法は、ランダムやラウンドロビンの決定方法が考えられている。

SCCプログラム ARPA等で採用されている、ディレイベクトルの交換によるを採用することも可能である。ただしこの際、サブネット全体に関する情報が一度に把握できるため、ローカルな逐次近似的な繰返しより効率よい制御が可能である。その他、国鉄のようなビジネス指向のネットワークでは、トラヒックのパターンとその変動があらかじめ予想できるため、何種類かのルーティングのパターンを用意しておき、時間帯毎に切換えたり、トラヒックの観測により最もマッチしたパターンを選取する[†]等の方法が考えられる。

3.2. フロー制御

前述したように、ネットワークのトラヒックの大部分が大きなオンラインシステムで決定されるため、フロー制御はホスト-ホストプロトコルで行うこととした。

このフロー制御を行う、てもサブネットの能力以上に流入するパケットを抑止するための制御が必要である。この部分はサブネット内のプロトコルと、それに対応するホスト-サブネットプロトコルで分担させる。

ホスト-ホストプロトコル、サブネットプロトコルによるフロー制御は次節で述べる。

3.3 パケット化およびリアセンブリ

リアセンブリがネットワークのかなり大きな問題であることは広く知られており、このネットワークも例外でない。国鉄の場合、かなり高いトラヒックを確実に処理する必要があること、ホストに大型のシステムがあり、処理、バッファ等はホストに任せようであること、端末は全てTCPを経由する等の理由から、リアセンブリはサブネットの外に出し、サブネットは単純なデータグラムのサービスのみの行うこととした。ただし既存のシステムで新たにNCPを組み込むことが困難な場合、ホストに処理能力や容量上の余裕がない場合等は、リアセンブリホストあるいはNCPホストをFEP的に仲介させる必要がある。

4. プロトコル

4.1. プロトコルの階層

図2はプロトコルの階層とその機能を示した図であるが、同一レベル内のプロトコルは更に組合化した論理的な階層構造を持つ。例えばホスト-ホストプロトコルでは、i) 接続、ii) メッセージ取扱い、iii) パケット化とリアセンブリ[†]このパターンマッチング法の適用は、東大・猪瀬教授の示唆による。

の3レベルが考えられる。ここでは、ホスト-ホストおよびサブネットプロトコルについて、その機能的な面(コマンド類)について概述する。

4.2. ホスト-ホストプロトコル

4.2.1. 機能

このネットワークにおける機能は次のようなものである。

- a) ユーザプロセス間のメッセージの伝達
- b) メッセージのパケット化とリアセンブリ
- c) メッセージの順序制御(指定による)
- d) プロセス間のフロー制御
- e) 割込み制御
- f) テスト

4.2.2. プロセス間の論理的結合(パイプ)

ネットワークを介して2つのプロセスが通信を行うには、あらかじめ両プロセスの間に、パイプと呼ばれる機構を用いて論理的に接続しておく必要がある。パイプは方向性と容量を持っており、逆方向にメッセージやコマンドが伝達されることはない。したがってプロセス間には必ず1対のパイプが設定される。

パイプの容量はその中を同時に通り得るメッセージ数、換言すれば送信側が未だ受取通知を受けていないにもかかわらず送出できるメッセージ数である。一般のメッセージはこの

パイプの容量によるフロー制御を受けるが、他のコマンド類はフロー制御を受けない。

NCPは自ホストが持つ受信パイプを管理するテーブルを持つ。さらに容量を以上、または

は各パケットメッセージを取扱うパイプについては専用のバッファ領域を確保し、リアセンブルはここで行う。送信および容量1の1パケットメッセージの受信には、ホスト全体のバッファを共用する。

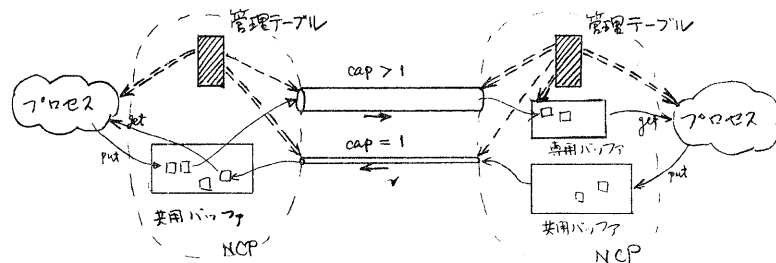


図3. プロセス間の結合(パイプ)

4.2.3. パイプ制御

ホスト-ホスト間コマンドを表1にあづる。これらのコマンドはパイプ制御、メッセージ制御、その他(割込み等)に合けられる。

オ1のグループはパイプの制御に関するもので、パイプの設定、切断を行う他、トラフィックの変動に対処するため、パイプの容量の変更を可能にするものである。しかしこの変更はトラフィックの変動に応じて送信側が要求するのではなく、プロセス、あるいはNCPのバッファの余裕により受信側が変更するものである。典型的な通信手順を図4に示す。

4. 2. 4. メッセージ制御

これに属するコマンドはメッセージ、およびこれに対する応答、督促である。

メッセージ(パケット)には、パイプ番号の他、メッセージの保障と順番管理のための通番、リアセンブリのためのパケット情報が入っている。最後のパケット情報はさらに一段下のレベルのプロトコルとして独立に扱うのが論理的であるが、NCPの処理の都合上同レベルで扱っている。

メッセージのフローは、HDLCのダブルナンバリングに似た形でメッセージに付けられた通番(RN, SN)と送受側で管理している番号(SH, SL; RH, RL)によってウィンドウ方式で制御される。

システムの中には、メッセージの順番が問題とにならないものもあるため、順序管理はオプションとして指定される。

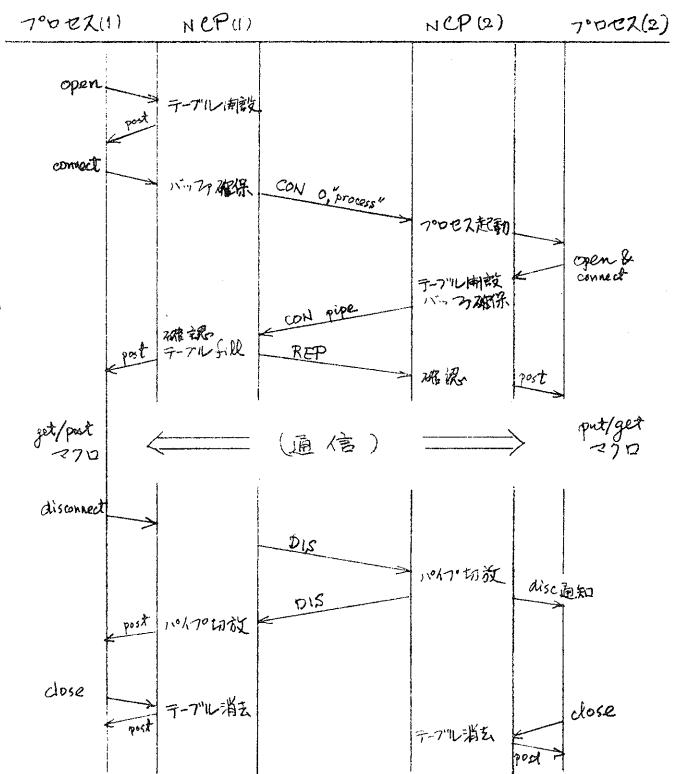


図4 典型的な通信手順

表1 Host-Host プロトコル

command	abbr.	parameters
[pipe control]		
connect	CON	[0, process], host*, pipe*, process*, rec. pipe cap., send message size, admission
reply	REP	pipe, host*, pipe*
change pipe	CHP	pipe, host*, pipe*, rec. pipe cap.
reply CHP	REP	pipe
disconnect	DISC	pipe, host*, pipe*
[message and message control]		
message		pipe, rec. thru no, all, send thru no, packet no, total packet no, message length, text
acknowledge	ACK	pipe, rec. thru no, all
request	REQ	pipe, rec. thru no, requesting thru no.
[interruption, etc.]		
break in	BRK	pipe, text(break command)
free message	BRK	0, process, text
test	TES	0, host*, pipe*, text
echo	ECH	pipe, text

(* のついたパラメータは自分の post/pipe を示す)

なお、パラメータの all 項は、単独および一括ACKを兼用できるようにしたもので、図5、6の例では*で示されている。

4.2.5. 割込み; その他

通常のメッセージの他に割込み、試験のためのコマンドが用意されている。この他、“フリーメッセージ”として、データグラムの1パケットメッセージが存在する。これは放送的な用途に使用される他、NCPが網を管理するホストに臨時に問合せをするときにも使用できる。

図7にエコー試験の手順を示す。

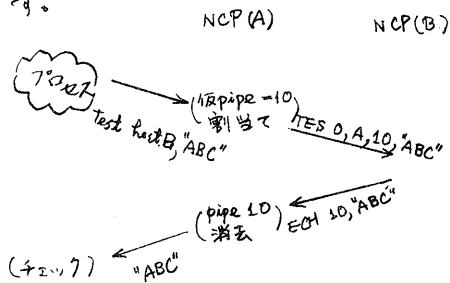


図7. エコー試験

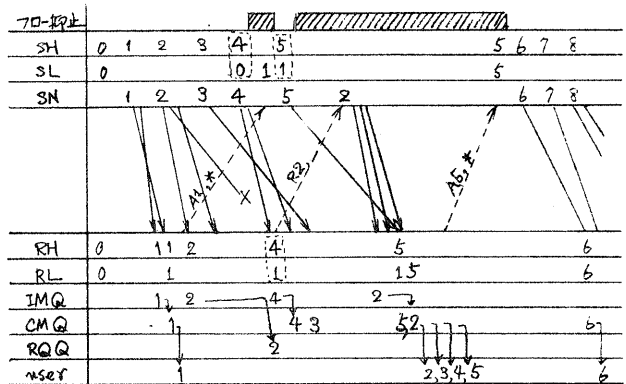


図5. 順序管理のあるフロー制御 (バッファ容量=4)

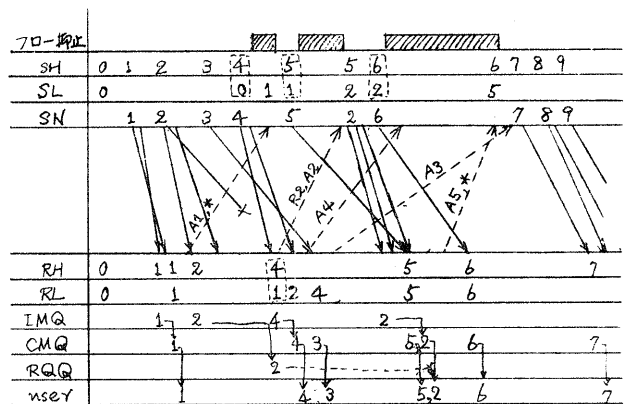


図6. 順序管理のないフロー制御 (バッファ容量=4)

4.3. サブネットプロトコル

4.3.1 機能

ここで述べるプロトコルはサブネットの中でも比較的高位な、SCC, IMP同士のコマンドであって、次の機能を持つ。

- ルーティング
 - フロー制御
 - サブネットの状態監視
 - 統計収集
 - その他サブネット, IMPの初期化, ダウン回復等
- サブネット制御コマンド表に示す。

4.3.2 ルーティング

IMPはSCCからのRUTコマンドにより与えられたルーティングテーブルに従ってルート選択を行う。このテーブルを決定するための情報は主としてIMPが定期的に報告するDLYコマンドによって与えられるが、SCCはその他のサブネットコマンドや外部要因を考慮することができる。

表2 サブネット制御コマンド

command	abbr.	direction	parameters
[flow control]			
stop flow	STP	↓	time, {imp, host imp
send	SND		
help	HLP	↑	time, {imp, host imp
relaxed	RLX		
discard	DCD	↓	time, {imp, host, survivability imp, —, survivability —, survivability
[state check]			
how are you	HAY	↑ ↔	time
	YAH	↓ ↔	time
down	DWN	↑	time, imp, {host line (neighboring IMP)
up	UPP		
SCC down	SCD	↔	time
SCC up	SCU	↔	time
[routing]			
route table	RUT	↓	time, routing table
delay	DLY	↑	time, imp, line, delay time
[statistics]			
	STA	↑	time, imp, statistical information

↓ : from SCC to IMP
↑ : from IMP to SCC
↔ : between IMPs
(commands concerning IPL
&c. are omitted)

4.3.3 フロー制御

フロー制御の大部分は前項で述べたホスト-ホストプロトコルで行うが、これで制御しきれない場合については、パケットをサブネットに流入させないための手段が必要である。SCCの中央集権的な制御はこの点で特に有効に作用すると思われる。

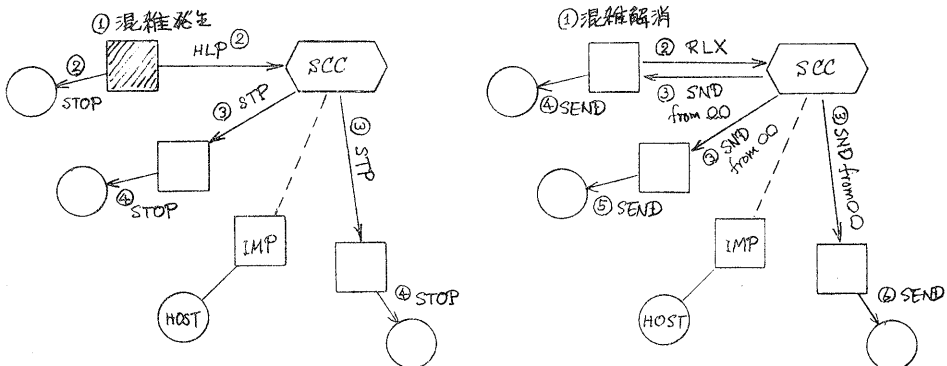


図 8. 典型的なフロー抑止(左)と再開(右)

IMP から混雑の報告を受けた SCC は適当な IMP (全 IMP でもよい) に対し、フロー抑止コマンドを送出する。これを受取った IMP はパラメータにより、特定の IMP あるいはホストに向かうパケットのサブネットへの流入を防ぎ (着 IMP 規制)、またはホストからの全パケットの受付を停止する (一斉規制)。これとは別に混雑した IMP は独自にホストからのパケットの受付を停止する (発 IMP 規制)。

フローの再開も同様な手順で行うことができるが、全 IMP が一斉にパケットを受付ると、バーストによりサブネットが再びロックアップを起す危険がシミュレーションにより指摘されている。このため SCC からの SND コマンドは再開時刻を指定してゆるやかな立ち上がりさせることができる (図 8)。

フロー制御を行っても、サブネットがロックアップ状態になることがある。この場合は IMP 内にたまったパケットをディスクカードコマンドで、生存優先度 ("survivability" と呼ぶ) の低いものから棄て去る。ディスクカードコマンドは、あるホスト又は IMP がダウンした場合にも、それを目的とするパケットを棄て去る場合にも使用される。

4.3.4 状態監視とダウン対策

SCC, IMP は互に (SCC は定期的に、IMP 同士は両者間にとり合いつけない時) HAY (How are you) コマンドと、それに対する YAH コマンドを交換して、隣接 IMP 又は回線の障害を検出し、その状態を SCC に集約する。SCC はこれらを統合して、回線、IMP のダウン、あるいはサブネットのロックアップを判断し、対応する処置 (ルーティングの変更、IMP のプログラムロード、パケットのディスクカード等) を行う。

SCC, IMP 間の制御用パスの障害時には、対応する IMP は、隣接 IMP の情報パスを経由して SCC と通信する。このような IMP を ICI (indirectly controlled IMP) と呼び、通常の IMP (DCI: directly controlled IMP) と区別する。

SCC のダウンは、IMP が SCC の HAY をタイムアウトにすること等で検出する。この場合、網は SCC ダウンモードの動作に移る。特にフロー制御が発 IMP 規制のみになるため、サブネット全体のスルーポットを犠牲にしてロックアップの危険を下げる方法がとられる。すなわち、未だ解除されていない SCC からの規制コマンドは解除した上、

a) 発 IMP 規制を通常時より低い混雑度で開始する。

b) さらに IMP が混雑した場合、生存優先度の低いパケットを棄て去る。

c) ルーティングテーブルはダウン前のものをそのまま継続する。

等の方法がとられる。

4.3.5 ホストとしての SCC

SCC は本来サブネットを効率よく動作させるためのものであり、ホストから見れば存在を意識する必要のないものであり、さらには不可視的存在でなくてはならない。しかし SCC はサブネットの制御のため各種の統計を取っており、ARPA の NMC 的働きもしている。これらの統計は他のホストが必要となるとき、サブネット外に取り出せる必要がある。この目的のために新たにホスト-サブネット (SCC) プロトコルを作ることは好ましくない。したがって、SCC のうち一般のホストと通信する部分を論理的にサブネット制御部分と分離し、ホストとして IMP と接続する。前者 (ホスト SCC) と後者 (サブネット SCC) との間はチャネル結

合、あるいはファイルシェアの形で統計データが受け渡され、必要ならホスト SCC で加工されて、他ホストに渡される。インプリメントの際はホスト SCC を独立にする、サブネット SCC と同一マシンとする、他の適当なホストと同一にする等の方法が考えられる。

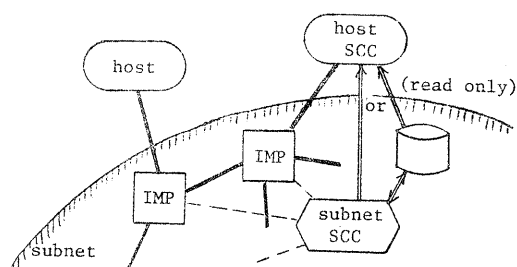


図 9. Host SCC

4.3.6 その他のサブネットコマンド

サブネットコマンドとしてはこの他に IMP のテーブル類の初期化、IMP ダウン回復のためのプログラムロード指示等のオフライン的なものがあるが、ここでは省略した。

5. あとがき

分散形のネットワークに集中制御を行うという一見矛盾した発想は、SCC を仮定しないネットワークについて行ったシミュレーションの検討段階でできたものであり、その妥当性が完全に裏付けされたものではない。SCC を含めた形での検証を進めたい。基本的にはネットワークは SCC のない形（前節でいえば SCC のダウンモード）で完全に動作すべきものであり、SCC の存在によってその動作がより高効率、安全な方向に行くという構成でなければならないであろう。国鉄においては回線が自営である点も、SCC の設置に有利に働いている。

おわりにこれらのプロトコル案について御検討いただき、貴重な御意見を戴いた、鉄道通信協会コンピュータ結合方式研究委員会の猪瀬委員長を始めとする委員諸氏、御討論戴いた幹事諸氏に謝意を表する次第である。

参考文献

- 1] 鉄道通信協会：“コンピュータ結合方式研究委員会報告書” 1976.3
- 2] 萩野他：“国鉄ネットワークにおけるフロー制御について” 当研究会資料 2
- 3] 三木：“共用網上の専用コンピュータ・ネットワーク” 情報学会 C.N 講習会(85.2)
- 4] 近谷：“国鉄におけるコンピュータ・ネットワーク形成の考え方”
電気学会 C.N. シンポジウム予稿(78.11)
- 5] 萩野他：“コンピュータ・ネットワークのシミュレーション” 鉄研連報 No.76-47