解　説

コンピュータ・ネットワークにおける高信頼化技術†

鍛治　勝三††

1. はじめに

コンピュータ・ネットワーク[1]の規模が拡大し，適用の場が広まり，多様なあるいは不特定多数の利用者がネットワークに参加する段階に至ると，そのネットワークの故障は利用者に多大な損害や著しい影響を与え，場合によっては人命に係わる重大事故を招きかねない．これらの故障を事前に阻止し，安全を維持する対策が近年とくに重要視されるようになってきた．

多数の異なった要素から構成されるコンピュータ・ネットワークは，複雑で大規模なシステムであり，その故障や障害の種類も多種・多様である．その故障原因の究明にも複雑かつ高度な判断機能を必要とするのみならず，これらの故障・保守のためにサービスが中断することは許されない．コンピュータ・ネットワークにおいては，発生しうる各種の故障を未然に予防する手段を講ずるとともに，故障が発生した場合その影響を及ぼす範囲を極力少なくし，速やかに回復させる必要がある．

また，コンピュータ・ネットワークの保守を的確に，容易に，そして迅速に行うために，ネットワークの稼動時に発生した故障状況をソフトウェアで克明に記録しておき，構成要素ごとに故障発生分布や故障原因を周期的に解析し，間欠障害が多発している機器や回線を重点的に行う予防保守，故障発生時の自動切分け，障害のために切り離された装置の修復後のネットワークへの自動復帰機能などの対策が不可欠である．

以上のような対策が施され，故障によって性能は低下するがネットワーク全体がダウンすることのないコンピュータ・ネットワークは，特に“フォールト・トレラント・コンピュータ・ネットワーク”(Fault-Tolerant Computer Networks) と呼ばれる．

本稿では，コンピュータ・ネットワークで採用されている高信頼化（フォールト・トレラント）技術の内，状態監視機能（第2章）と保守機能（第3章）について解説する．

2. 状態監視機能

2.1 状態監視機能の目的

コンピュータ・ネットワークは，図-1 に示すようにホスト・コンピュータ，スイッチング・ノード，通信回線（リンク），そして端末など多種・多様なたくさんの構成要素から成り，かつ広域に分散しているためさまざまな利用者や環境の影響を受ける．このため，単独のコンピュータ・システムに比し，誤り，故障，デッドロック，あるいはロックアップなどの起る確率も高く，かつその種類も多様であり，より高度な状態監視および診断機能が必要である．

以下では，コンピュータ・ネットワーク全体の状態監視の中枢である NCC，パケット（メッセージ）の送達を司る通信サブネット，そしてリソースの供給源であるホスト・コンピュータに分けて，各々の状態監視機能について述べる．

2.2 NCC (Network Control Center)

コンピュータ・ネットワーク中のトラフィックの量および分布は，利用者の利用状況により急速に変化す

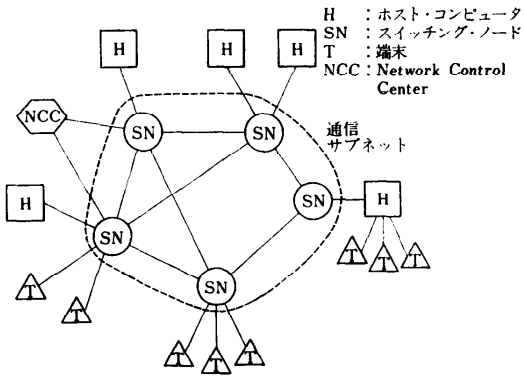

図-1　コンピュータ・ネットワークの形態

† A Survey of Methods of Achieving Fault-Tolerant Computer Networks by Katsuzo KAJI (Japan Information Processing Development Center).

†† (財)日本情報処理開発協会

る可能性がある．さらにネットワーク内には多数の構成要素が存在し，それらが動的にアップあるいはダウンし，また利用者によるコネクションのオン・オフもランダムに発生する．したがってネットワークの構成やトポロジが動的に変化するとともに，その稼動状況も刻々変化する．既存のコンピュータ・ネットワークにおいては，通信サブネットのスイッチング・ノードが隣接ノードおよび自回線の状態，ネットワーク・トポロジやトラフィックの変化，ノードの温度など自然環境変化を常時監視し，NCC (Network Control Center)[2] あるいはスーパバイザ・ステーション (Supervisor Station) と呼ばれる統轄局に一定時間ごと（周期的）に，あるいは特定のイベントが発生するたびにこれらの変化を報告する．報告を受けた NCC は，その状態をログ・ファイルに記録し，スイッチング・ノードや回線の故障などのような重大なイベントに対しては，NCC にあるコンソールにその旨を表示したり，赤ランプを点滅したり，あるいはアラームを鳴らしたりして，操作員に警告する，という方式をとるものがある．

NCC はネットワーク運用管理の中枢であり，NCC のダウンが即ネットワーク全体のダウンとならぬ対策が必要である．ネットワーク中にただ一つの NCC しか設けない場合には，NCC 故障の発生によるネットワーク全体への影響を考えると，NCC 全体の二重化あるいはクラスタ化，NCC へ行く回線の二重化あるいは2つ以上のスイッチング・ノードに接続し，できるかぎり冗長性を持たせる必要がある．

また集中型の NCC でも TYMNET[3] のように NCC を複数箇所（4カ所）に設置し，常時稼動させておき，一時期にはただ一つの NCC のみがネットワーク全体を監視する方式も有効である．したがって，NCC に要求される信頼度をよく踏まえた上で，コストとの兼合いを十分に検討して対策を立てる必要がある．

2.3 通信サブネットの状態監視機能

（1） スイッチング・ノードにおける状態監視機能

ネットワーク中のトラフィック量が極端に増加したり，あるいは一つのスイッチング・ノードに大量のトラフィックが集中したりすると，ネットワークが輻輳状態や過負荷状態に陥りやすくなり，スイッチング・ノードの交換用のバッファが不足し，リアセンブリ・ロックアップや蓄積交換ロックアップと呼ばれるデッドロック現象が必然的に起ってくる．これらの状態が発生すると，ネットワークのパフォーマンスが極度に低下するので，それらに対する対策（フロー制御と呼ばれる）が必要である．

したがって，ネットワークの稼動状態を常時監視し，パフォーマンスに悪影響を与える因子を制御する機構を設けることにより，ネットワークの信頼性，可用性，そしてパフォーマンスを向上させることが可能となる．

しかしながら，状態監視機能は通常スイッチング・プログラムと同じノード上に存在し，そのノードの CPU，メモリ，入出力装置などのリソースを借用しなければならず，状態監視のためのオーバヘッドが正規のスイッチング・プログラムの動作速度に影響を及ぼさないように配慮する必要がある．

（2） 自ノードの状態監視と完全性

各スイッチング・ノードは，プログラムやデータなどが格納されている自己のメモリ内容の完全性を保障するために周期的(数分ごと)に，あるいはクリティカル・プロセスの実行直前にメモリ・ブロックのチェック・サム検査を実施し，誤りを検出した場合には，全メモリ内容を NCC に転送し，検診を行ってもらう．これは，ダウン・ライン・ダンピング (Down-line Dumping) と呼ばれる．

また自ソフトウェア・システムの完全性を強化するため，テスト時に検出されなかったソフトウェア誤り (residual software error) や，ハードウェアの検出されなかった間欠故障によってデータの一部が壊され不完全になる (data mutilation) ことによってソフトウェア・システムが破壊されないように，データの値や構造に冗長性* を持たせるだけでなく，命令にも積極的に冗長性** (instruction redundancy と呼ばれる) を持たせた自己検査機能ソフトウェア (Self-Checking Software) を採用したシステムもある[4]．

（3） 隣接ノードの状態監視

隣接する二つのスイッチング・ノード間では，定期的にあるいは一定期間往来するトラフィックがない場合には，お互いの通信機能の完全性を相互監視・検証するため，必要十分なメッセージを授受し合う．

たとえば，ARPANET では，往来するパケット・フローがない場合には，スイッチング・ノードである IMP (Interface Message Processor) は隣接ノード間相互に “Hello” パケットと “I-hear-you” パケット

* たとえば，定マーク符号 (*nCm* コード)，double-linked list.
** たとえば，relay-runner, positive check.

を交換する．これらのパケットには，自 IMP の内部データ構造の完全性，自 IMP に結合しているホスト・コンピュータの完全性，そして自回線系の完全性の周期的検査結果を含ませることが可能である．DEC 社の DNA (Digital Network Architecture)[5] では，一定期間往来するトラフィックがない場合には，NOP (No Operation) メッセージと呼ばれる人工的なトラフィックを発生させ，そのメッセージを受信したノードは，それを棄却するプロトコルを制定している．

（4） 伝送誤りの監視

スイッチング・ノードは隣接ノードにパケット送出後，一定期間確認応答を待つ．一定期間内に相手ノードからの確認応答が到着しない場合には，そのパケットを再送する．規定回数回だけ再送を試みても相手ノードから確認応答が返ってこない場合には，回線系かまたは相手ノードの故障とみなし，その旨を NCC に報告する．

またスイッチング・ノードは隣接ノードより誤りのあるパケットを受信する場合がある．パケットの伝送誤りを検出するために，一般に CRC (Cyclic Redundancy Check) をパケット中に含ませている．CRC は通常，ノードの CPU とは独立の通信回線のインタフェース・ハードウェアで発生させて，パケットに埋め込まれる．ただし回線インタフェースが簡易な場合には，CRC はノード内のソフトウェアで発生させられる．この場合，原理的には，パケット誤りの原因が伝送誤りなのか，発信ノード故障であるかの切分けが不可能となる（詳細は **3.1**（2）で述べる）．

伝送誤りの場合，受信側ノードは発信側ノードへ NAK (Negative Acknowledgment), REJ (Reject), SREJ (Selective Reject) 信号などを応答として返し，再送を要求する．あるいは，発信側の応答時間監視により，発信側自身で再送を開始する．再送して正しくパケットが受信された場合は間欠故障，そして数回（3回程度）再送を試みてもまだ伝送誤りが検出された場合は固定故障として NCC に報告する．

（5） デッドロックの防止

大規模ネットワーク環境下での蓄積交換ロックアップなどのデッドロックを完全に防止する経済的で，実用的な通信プロトコルはまだ知られていない．デッドロックの防止策としては，次のようなフロー制御方式を採っている．

ネットワーク中のホスト・コンピュータやスイッチング・ノードのバッファは有限個しかないため，ネットワークに入来するトラフィックすべてを無条件になんら規制せずに受け入れることはできない．各スイッチング・ノードは，自ノード中の蓄積交換用のバッファの使用状況を監視し，残りのバッファの個数が一定量以下に達した時には，到着したパケットを棄却し，接続されている各回線の伝送容量に応じて最低限度個の入力用バッファを確保し，蓄積交換ロックアップなどの発生を防止する．そして，パケットを棄却した時には，その旨を NCC に報告する．

（6） 分散制御の問題点

フロー制御，輻輳制御，そして経路制御のようなネットワーク制御アルゴリズムの運用に関してネットワーク全体に同等に権限が委譲されている場合，ある一つのスイッチング・ノードの故障による誤動作がネットワーク全体に波及し，他のノードに悪影響を与える場合がある．ここでは適応経路制御について考えてみる．

適応経路制御は ARPANET などいくつかの代表的ネットワークで採用されており，自ノードの回線系故障が回復した時，過負荷状態に陥った時，あるいは反対に過少負荷状態に陥った時など，各ノードは相互に経路情報としてこの事態を通知し，それを受け取ったノードは自ノードの経路表を調整する方式である．経路情報は隣接ノード間で交換され，その情報により経路表が更新される．経路に影響を及ぼす経路情報は，ノードからノードへ伝播され，ネットワーク中の全ノードの経路表を更新させる．この方式の問題点は，あるノードで誤りが発生した場合，誤った経路情報が他のノードへ伝播・波及してしまうことにある．

ARPANET では，この種の誤りを検出するため，ソフトウェアによって重要なパケットおよびプログラムに対してチェックサムを組み込むように変更された．ソフトウェア・チェックサム機構により受信したパケットに誤りを検出した IMP ソフトウェアは，相手 IMP のソフトウェアか，あるいはハードウェアに故障があると推定し，誤りを起した情報のコピーを NCC に転送する．さらに，自 IMP のプログラムにチェックサム誤りを検出した IMP は，主メモリの誤りの部分のコピーを NCC に転送し，NCC に検診を委託し，直ちに隣接 IMP からのプログラム・リロードを要求するように修正された．これにより，ある1つの IMP の故障による誤動作が，誤った経路情報を伝播させ，ネットワーク中の正常な他 IMP へ悪影響を及ぼすことを防止することが可能となった．

2.4 NCC のホスト状態監視機能

NCC はスイッチング・ノードだけでなく，重要なホスト・コンピュータの稼動状態も監視する．ホスト・コンピュータ側のローカル・サービスが正常に運用されている場合，ネットワークとの結合が正常でなくても，各ホスト・サイトの操作員がそれに気付かない場合がある．NCC は常時ホスト・コンピュータの状態を監視し，ダウンした場合に警報を鳴らす．NCC の操作員は，そのダウンが予定されているものかどうかをスケジュール表で確認する．もし予定されたダウン（たとえば定期保守）以外の場合は，NCC の操作員は，そのサイトの操作員に電話し，ホスト・コンピュータがネットワークと結合してないことを通知し，至急結合させ，もしホスト・コンピュータ故障の場合には何時頃復旧するかを問合せる．ホスト・コンピュータ側のネットワーク・インタフェース・ハードウェアが故障のため，ホスト・コンピュータとの交信が不可能の場合，NCC の操作員が介入することにより，この種の故障の検出が早まる．

3. 保守機能

コンピュータ・ネットワークにおける通信サブネットの保守機能は，診断機能と修復・回復機能の二つに大別される．以下にこれらの機能をいくつかの例を含めて紹介する．

3.1 診断機能

（1） 自ノードの診断

スイッチング・ノードは自己のソフトウェアに誤りを検出した場合，まず NCC や他のノードの力を借りず自己のみで診断を行う．またハードウェア故障の場合は，独立の診断用ハードウェア機構が付加されている特殊な場合を除き，ネットワーク中の他の構成要素の力を借りて診断を行う必要がある．たとえば，ARPANET では，IMP が主記憶の故障を検出した時は，主記憶の故障部分のコピーを NCC に送り，NCC に故障診断を委託する．その IMP は隣接 IMP よりプログラムをリロードし，運用を続行する．その間，NCC はそのコピーとオリジナル・プログラムとを照合し，故障箇所を究明する．

（2） 隣接ノードの診断

あるスイッチング・ノードが隣接ノードとの通信で故障を検出した場合について考えてみる．

まず行わなければならないのは，通信故障の原因が隣接ノード自体の故障か，あるいはモデムなどを含めた回線系の故障に起因するかの切分け作業である．隣接ノードへ行く迂回路がある場合には，その迂回路を使用して隣接ノードと通信テストを行う．

故障原因が回線系の場合には，モデムにループバック診断機構が内蔵されておれば，故障原因が回線自身なのか，それとも回線インタフェースなのかの決定に非常に役立つ．この場合，さらにループバック診断機構がリモート側から制御可能であることが望ましい．たとえば，DEC 社の DNA では，モデムをループバック・モードにし，NODE INIT (Node Initialization) メッセージを送り，ループバック（折返し）して返ってきたメッセージを受信し，送信したメッセージと一致するか否かを検証している．

次に隣接ノード自体の故障と判断された場合の方式としては，たとえば，テスト・メッセージを送り，想定される故障箇所に関するデータをレスポンスに含ませて返送させ，その内容を検証する．あるいは特別なメッセージを送り，受信したノードはそれをトリガにして自分自身の診断手順を起動するなどの手法が考えられる．

（3） 診断データベース

診断作業を迅速に行うため，メンテナンス・ディレクトリと呼ばれるデータベースを NCC に持たせるシステムがある．そのデータベースには，過去に起った故障の徴候や故障原因とそれに対する修復手順が蓄積されている．また新しい故障の徴候，原因，そして修復手順が発見された場合には，そのデータベースに追加される．故障の徴候あるいは故障が検出された時，診断データベースを参照して，その症状を呈すると考えられる故障原因と修復手順を見つけ，保守を行う．

（4） 診断後の処置

コンピュータ・ネットワークにおいては，独立なシステム以上に自動診断機能が充実していなければならない．前述のような各種の方法により，あるスイッチング・ノードが隣接ノードの故障を検出した場合，その故障を NCC に通知し，NCC は論理的にそのスイッチング・ノードをネットワークから一時切り離す．故障回復後，NCC はそのスイッチング・ノードを論理的に結合状態に組み入れる．

3.2 修復・回復機能

スイッチング・ノードが故障を起し，修復された場合，メモリ中に入っているプログラムは通常破壊(消)されている．スイッチング・ノードはローカルな補助記憶装置を持たぬ場合が多く，そのため NCC または

隣接ノードからプログラム・リロードができると便利である．これはダウン・ライン・ローディング(Down-line Loading)と呼ばれる．これを可能にするためには，スイッチング・ノード内にブートストラップ部分が消えずに保存（通常，ROM (Read-Only-Memory) に格納されている）されているか，あるいは通信インタフェース内に特別なブートストラップ用のハードウェア機構を内蔵させておく必要がある．スイッチング・ノード内にブートストラップが恒久的に組み込まれていない場合は，NCC または隣接ノードが通信インタフェースに向けてブートストラップ・ルーチンを含むメッセージを送る．これを受け取った通信インタフェースは，スイッチング・ノードのメモリ中に強制的にブートストラップ・ルーチンを書き込み，ブートストラップの最初の命令に制御を渡す．この場合，NCC からの受信相手がスイッチング・ノードではなく，通信インタフェース自体であるのでリンク・レベル・プロトコルで特別な配慮をしておく必要がある．リローディングの残りの手順は，システム・プログラムを一連のメッセージとしてスイッチング・ノードに向けて送り，ブートストラップがそれを受信し，メモリにロードすることにより完了する．スイッチング・ノードの構成，すなわち接続回線数，端末の種類および個数，ホスト・コンピュータの接続の有無などのそのスイッチング・ノード固有の情報は，NCC から送ってもらう必要がある．たとえば，DEC 社の DNA では，各スイッチング・ノードの ROM にはブートストラップが格納されており，隣接ノードから完全なプロトコル・ソフトウェアのコアー・イメージを MAINT (Maintenance) メッセージとして送っている．

4. おわりに

以上，本稿ではコンピュータ・ネットワークで採用されている高信頼化技術の内，紙面の関係上，状態監視機能および保守機能のみしか言及できなかったが，これら以外の高信頼化技術については，たとえば，参考文献 6) に載っている参考文献表の各参考文献を参照されたい．

最後に，本稿がコンピュータ・ネットワークの高信頼化技術の設計・理解の一助となれば幸いである．

参考文献

1) 鍛治勝三：JIPNET の NCP とロジカル・リンク，日経エレクトロニクス，No. 136, pp. 76-95 (1976年6月14日号)．
2) McKenzie, A. A.: The ARPA Network Control Center, Proc. ACM/IEEE Fourth Data Communications Symposium, pp. 5-1 to 5-6 (Oct. 1975).
3) Sullivan, N.: TYMNET-Maintenance Consideration in a Very Large Network, Proc. ACM/IEEE Fifth Data Communications Symposium, pp. 3-1 to 3-3 (Sept. 1977).
4) Yau, S. S. and Cheung, R. C.: Design of Self-Checking Software, Proc. 1975 International Conference on Reliable Software, pp. 450-457 (Apr. 1975).
5) Digital Equipment Corp.: DECNET Digital Network Architecture Network Services Protocol Specification, p. 87 (Nov. 1976).
6) Morgan, D. E. and Taylor, D. J.: A Survey of Methods for Improving Computer Network Reliability and Availability, IEEE Computer, Vol. 10, No. 11, pp. 42-50 (Nov. 1977).

(昭和56年12月7日受付)