

分散を指向したネットワーク アーキテクチャとその電力への応用

荻 宏美 島田 寿正 田岡 久雄 西田 正吾 坂口 敏明
東京電力(株) 技術研究所 三菱電機(株) 中央研究所

近年の情報化社会の進展に伴い、電力の信頼性への要求が一段と強くなっている。また、電源の多様化が進む中で、電力システムの運用制御システムは一層複雑になりつつある。一方、生物システムは非常に高度で複雑な機能を実現しているが、その一つの鍵は自律分散に基づくシステム構築にあるといわれている。

本論文では、自律機能を備えた将来の電力システム運用制御システムを、ネットワークの構築手法のレベルから検討し、耐故障性・柔軟性・進化性・拡張性・分割性等を備えた新しいネットワークアーキテクチャを提案する。また、本ネットワーク上で、自律機能を実現するための分散型ソフトウェア構築の指針を述べ、その検証を、並列処理システム用のプロセッサであるトランスピュータを用いて構築した計算機ネットワーク上で行なった結果について、報告する。

AUTONOMOUS DISTRIBUTED NETWORK ARCHITECTURE AND ITS APPLICATION TO POWER SYSTEM

Hiromi OGI, Toshimasa SHIMADA, Hisao TAOKA, Shogo Nishida and Toshiaki Sakaguchi

Engineering Research Center,
The Tokyo Electric Power Co., Inc.
2-4-1, Nishitsutsujigaoka,
Chofu, Tokyo, 182 Japan

Central Research Laboratory,
Mitsubishi Electric Corporation
8-1-1, Tsukaguchi-honmachi,
Amagasaki, Hyogo, 661 Japan

With the progress of information society, high reliability of electrical power system is required. Biological system, one of large scale system with very high level and complex functions, has autonomous function in it. Autonomy is said to be one of the most important key to configure large scale system.

We propose the autonomous distributed network architecture, which has the autonomous feature that consist of fault tolerance, flexibility, evolutionability, extensibility and partitionability. This paper describes the detail of the proposed network architecture, the concept of building distributed system's software to get autonomous feature, and the evaluation result in computer network composed of transputers.

1. まえがき

近年の情報化社会の進展に伴い、電力の信頼性への要求が一段と強くなっている。また、電源の多様化が進む中で、電力系統の運用制御システムは一層複雑になりつつある。そこで、将来の運用制御システムを考えると、信頼性を向上させるための、耐故障性・拡張性・分割性といったいわゆる自律機能を備えたシステムが今まで以上に要求されてくると思われる。

また、需給制御のような集中型のシステムの他、VQ制御・保護リレーに見られるローカルなシステム、あるいは復旧制御のような階層型等種々の形態が混在しており、その形態も一定でなく徐々に変化している。一箇所で異なる部所のオペレーションができるといった柔軟性も、非常時や今後のシステムのあり方を考える上で重要な要素となろう。

一方、生物システムは非常に高度で複雑な機能を実現しているが、その一つの鍵は自律分散に基づくシステム構築にあるといわれている²⁾³⁾。

筆者らは、このような自律機能を備えた将来の電力系統運用制御システムをネットワークの構築手法のレベルから検討し、耐故障性・柔軟性・進化性・拡張性・分割性等を備えたネットワークアーキテクチャを提案してきた⁴⁾。本論文では、新しいネットワークアーキテクチャの諸機能を述べるとともに、本ネットワークアーキテクチャを実際のシステムに適用する際の分散型ソフトウェアの構築のための指針を提案し、並列処理システム向きのプロセッサであるトランスピュータを用いて構築した計算機ネットワーク上で検証した結果について報告する。

2. 自律機能と新しいネットワークアーキテクチャ

電力系統の分野を含めて、分散型の制御システムは一般に図1のようにとらえることができる。即ち、ある制御対象があり、それに対して制御システムが制御のアルゴリズムに従って制御を行なっている。その下位には、他の制御装置との情報の授受や、システムの故障時の制御機能の代替・バイパス等の対策を司る情報システムがある。さらに下位にプロトコル等の通信を司る通信システムがある。システムの自律性は、通信システム等、一つの階層のみで実現することはできない。上記3つの階層を総合的に築き上げることによって、システム全体に自律性を持たすことが必要である。

また、一般にダイナミックに変動するシステムに要求される自律機能の主なものとして、(1)システムの部分的な故障に際して、情報の授受や制御機能がシステム全体として正常に働くための耐故障性、(2)システムの変化に対応できるような柔軟性・進化性、(3)大規模化に伴うシステムの拡張や、保守・故障時の分割に対応できるような拡張性・分割性がある。ネットワークのアーキテクチャは、n-cube・メッシュ・スター・多重バス等、種々の方式が提案されている⁵⁾が、電力系統の運用制御システムへの適用を考えると、上記の機能を備えたものが望ましい。

その一つとして、Pradhanにより、マルチプロセッサシステムへの適用を目的に、ノード数nのネットワークにおいて、 $i = 2j \pmod{n}$ 、 $i = 2j + 1 \pmod{n}$ の関係を満たすノードi, j間を接続する方法が提案されている⁶⁾。

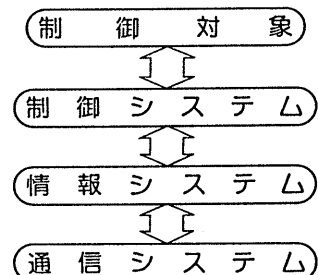


図1 システムの構造

このネットワークアーキテクチャは、以下のような特長を備えている。

- (1) 各ノードから出るリンクの数は、高々4本である。
- (2) リニアアレイ型・ハイアラキー型双方のネットワークが構築できる。
- (3) ルーティングが容易であり、経由リンク数は高々 $\log_2 n$ である。
- (4) 単一故障に対し耐故障性を備えている。
- (5) 故障時のルーティングでの経由リンク数は高々 $2 \log_2 n + 1$ である。
- (6) 上記の性質を満足しながら拡張・分割が容易にできる。

これに対し、我々の提案するネットワークアーキテクチャは、Pradhanの方法を一部改良したもので、前述の関係を満たすノード i 、 j 間に加えて、ノード 0 、 $(n-1)$ 間を接続する。この改良により、故障時のルーティングの複雑さを取り除き、分割時のノード番号に対する制約が少なくなる。

3. 新しいネットワークアーキテクチャの諸機能

<3.1> ネットワークの柔軟性・進化性

提案するネットワークアーキテクチャの、16ノードの場合の構築例が図2である。また、図3に各ノード間(ノード 0 、 $(n-1)$ 間を除く)を示している。図でわかるように、本ネットワークアーキテクチャは、ノードの接続リンク数は高々4であり、総リンク数は高々 $2n-2$ と、冗長度を2倍に抑えた構成となっている。また、リニアアレイ型とハイアラキー型のネットワークが、図4・図5に示すように物理的に構築可能であり、通信の効率化が図られるとともに、進化性・柔軟性を備えたものとなっている⁴⁾。

<3.2> ネットワークの耐故障性

リニアアレイ型ネットワークでは最大 $n/2$ 、ハイアラキー型ネットワークでは最大 $2 \log_2 n$ のリンクを経由すれば、任意のノード間でルーティングを行なうことができる。しかし、ネットワークの構造にこだわらなければ、ノード i を基点にしたツリー構造のネットワークを図6のように構築できることから、ルーティングは、 $\log_2 n$ 以下のリンクを経由するだけで可能となる。

ネットワークの構造は、正常時はリニアアレイ型・ハイアラキー型ともにフィジカルに保証されているが、故障時はロジカルに構造を維持する必要がある。そこで故障時のルーティングは、リニアアレイ型・ハイアラキー型の構造にかかわらず行なうことになる⁴⁾。本ネットワークアーキテクチャは、単一故障に対しては情報の授受を完全に保証している。

ノードあるいはリンクが故障した場合は、その隣接ノードのみが故障の事実を知ればよく、ルーティングは隣接ノードに情報が来た時点で、故障ノードをバイパスして終点ノードに転送することで可能となる。このことは、本システムが自律性を備えていることにほかならない。

<3.3> ネットワークの拡張性・分割性

ネットワークの拡張は、Pradhanの方法⁵⁾に依らず、複数のネットワークを図7のように、一方のノード 0 と他方のノード $(n-1)$ を接続するリンクを付加することで、規模を拡大していく。拡張されたネットワークは、耐故障性・柔軟性等、本来のネットワークが備えていた諸機能をすべて備えたものとなっている。当然、再びノード 0 、 $(n-1)$ 間で分割も可能である。また、拡張前のネットワークで一部を分割する場合は、ノード 0 とノード $(n-1)$ 間のリンクを加えたことにより、連続する番号を持つノードの集合であれば、分割可能となる。

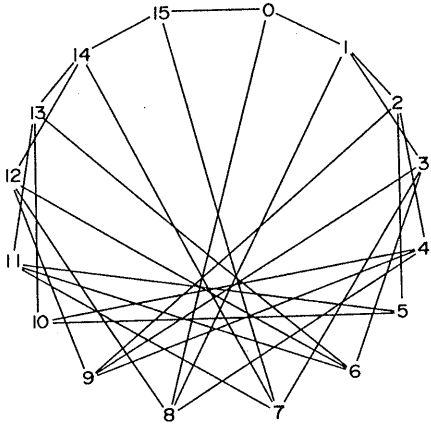


図 2 16ノードネットワークの例

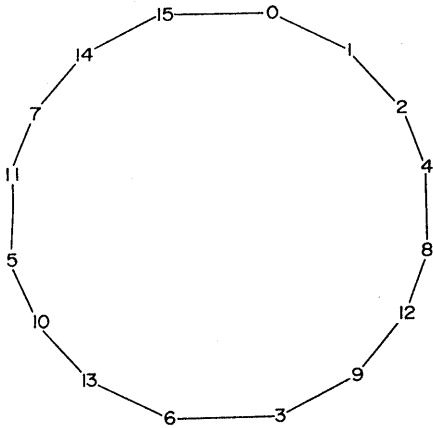


図 4 リニアアレイ型ネットワーク

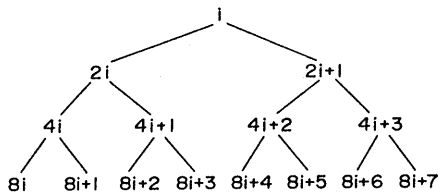


図 6 ハイアラキー型ネットワークの構築方法

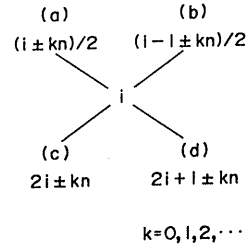
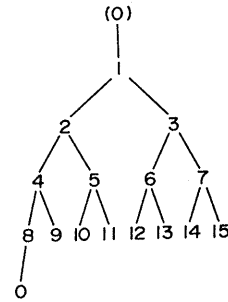
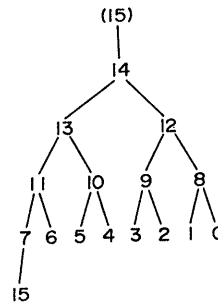


図 3 ノード間の接続方法



(a) 基点ノード 0 の場合



(b) 基点ノード (n - 1) の場合

図 5 ハイアラキー型ネットワーク

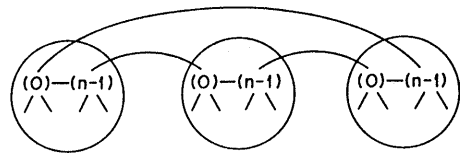


図 7 ネットワークの拡張方法

4. 分散型ソフトウェアの構築

分散型ソフトウェアの構築に際しては、以下の基本的な方針で行なう。

(1) ソフトウェアを、アプリケーションソフトウェアとネットワークソフトウェアに分け、図8に示すように、それぞれ制御部と通信部で並列に走らせる。これにより、アプリケーションの変更があっても、アプリケーションソフトウェアの変更のみで、ネットワークの枠組みは変更する必要がなく、柔軟性・進化性を持たせることができる。

(2) ネットワークソフトウェアに、リンク故障時のバイパス機能やノード故障時の代替機能等の機能を組み込み、耐故障性を持たせる。

(3) ネットワークソフトウェアは、均質性を持たせる。かつ、各ノードがローカルな情報のみで判断し処理を行なっているにもかかわらず、システム全体である機能を果たすものとする。これにより、各ノードは自律的な処理が可能となり、メンテナンスビリティが向上し、拡張・分割が容易になる。

<4.1> ネットワークソフトウェア

ネットワークソフトウェアは、基本的に以下の機能を実現するものとする。

(1) 故障検出機能 ——— 故障検出は、(A)相手が情報を受け取らない、(B)相手からの情報が来ない、のいずれかを検知して行ない、制御部のみの故障は通信部が、他は隣接ノードが故障を検出する。隣接ノードは、送信先ノードが受け取らなければまずリンク故障と考え、別ルートで送り直す。2度別ルートで送っても、送信先が情報を受け取らない時にノード故障と判断する。処理実行中にノードが故障した場合は、相手からデータが来ないことで故障を検出する。

(2) 代替機能 ——— ノード故障時の対応として、(A)故障ノードの機能を他ノードで代替する、(B)故障ノードの機能を停止する、(C)故障ノードをシステムから分離して単独で制御を行なう、の3つの方法がある。基本的には(A)の方法を採り、隣接ノードの一つがその機能を代替することにする。なお、(B)、(C)の場合は、代替ノードへの情報の転送を行なわなくてもよい。代替機能を実現する方法としては、アプリケーションの状態変数を常に代替ノードおよび自ノードの通信部に保持しておく。そして、ある単位の処理が終了するたびに、情報の伝送と同時に、状態変数を代替ノードおよび自ノードの通信部に転送する。

(3) 故障回復機能 ——— 故障中のルーチングは、送り出し時は正常時のルーチングのアルゴリズムを採用し、隣接ノードで故障の有無を確認した後、故障が継続していれば、再度故障時のルーチングを行なう。このことにより、故障の回復時の対応を早くすることができるのと同時に、故障の事実を隣接ノードが知るのみでよくなる。

ネットワークソフトウェアの一例を示したのが、表1および図9である。表1に示す各プロセスを作り、各機能単位にプロセスを割り当て、それらを図9に示すように、各ノード内で並列に走らせる。各動作モードでのノード内の情報の流れが、図10である。

<4.2> アプリケーションソフトウェア

アプリケーションソフトウェアには、以下のような機能を持たせる。

(1) 入力力で起動、出力力で終了する処理を単位にしたプロセスで構成する。

(2) 1つの処理が終了した時点で、出力情報を送信先に送ると同時に、状態変数を代替ノードおよび自ノードの通信部に送る。

表 1 各プロセスの機能

プロセス名	機能
タグ作成プロセス	・送信情報にルーチングのためのタグを付加する。
タグ修正プロセス	・タグを故障時のルーチング用のタグに修正する。
ルーチングプロセス	・情報の送信先と使用リンクをタグより求め、タグの処理を行なう。
入力プロセス	・他ノードより情報を受け、自ノード宛情報はタグを外して受信バッファプロセスへ渡す。宛情報はそのままルーチングプロセスへ渡す。
故障検出プロセス	・情報をリンクに出力する。送信先ノードが情報を受け取れば正常、一定時間内に受け取られなければ故障と判断し、タグをタグ修正プロセスに渡し、タグを修正の後、再度リンクに出力する。
受信バッファプロセス	・自ノードおよび代替ノードの状態変数を蓄え、入力に必要な状態変数を付加して制御部に渡す。
送信バッファプロセス	・情報を送信先ノードへ状態変数を送る。代替ノードおよび受信バッファプロセスへ送る。
制御部故障検出プロセス	・情報を制御部に出力する。制御部が情報を受け取れば正常、一定時間内に受け取られなければ故障と判断し、情報を代替ノードに転送する。

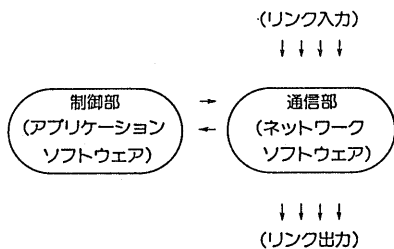
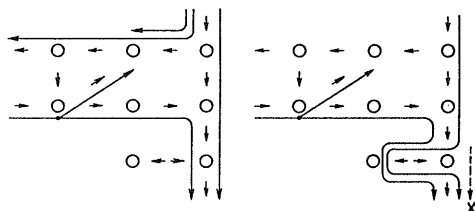


図 8 ノードの機能



(a)

(b)

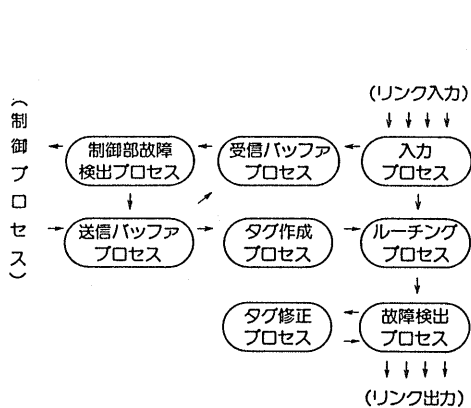
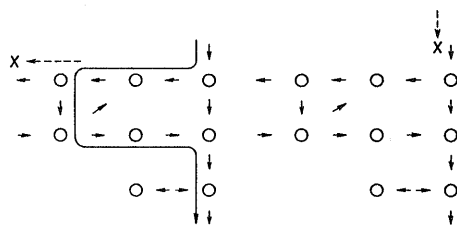


図 9 通信部の機能



(c)

(d)

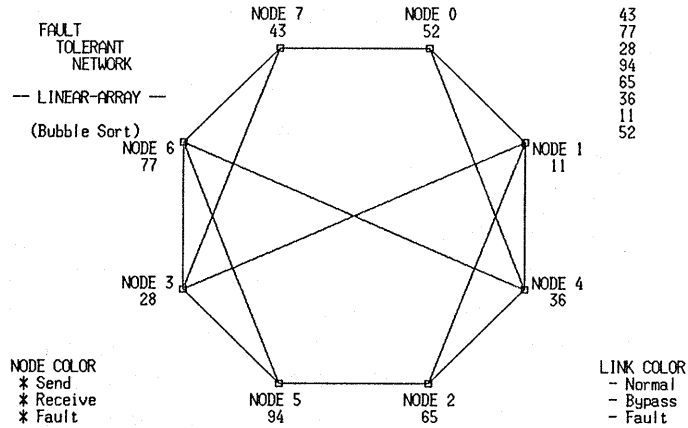
(a) 正常時

(b) リンク故障時

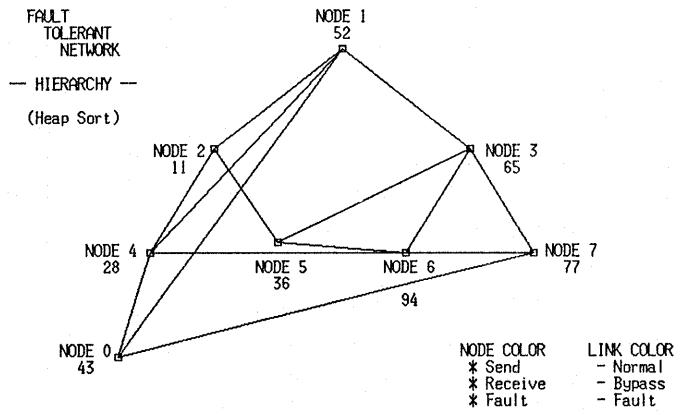
(c) ノード故障 1 (制御部のみ故障)

(d) ノード故障 2 (その他)

図 10 情報の流れ



(a) バブルソート(リニアアレイ型)



(b) ヒープソート(ハイアラキ型)

図 1 1 ソーティングの実行例

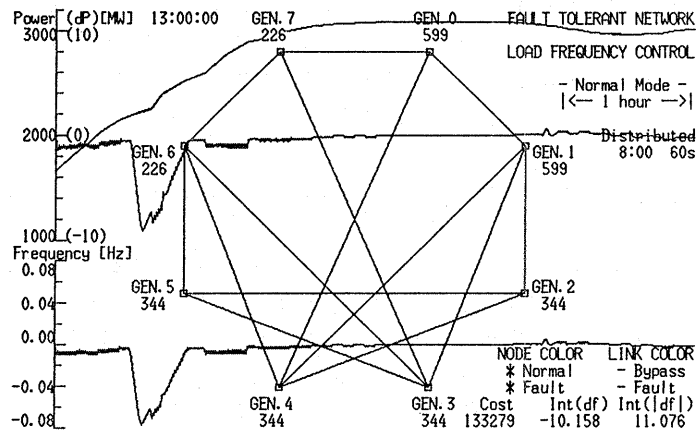


図 1 2 需給制御システムのシミュレーション結果

(3) プロセスは、入力情報とともに通信部内にバッファされた状態変数を受け取って処理を開始する。

これらにより、故障時の代替あるいは故障回復時の立ち上がりを容易にすることができるとともに、システムの拡張・分割を容易にし、ソフトウェアのメインテナビリティを向上させることができる。

5. ネットワークアーキテクチャの検証

トランスピュータは、並列処理機能と通信機能を備えた、最大10MIPSの32ビットのマイクロプロセッサであり、4本のリンクを用いて任意の構造のマルチプロセッサシステムを構築することができる。ソフトウェアは、トランスピュータにより直接実行されるOccamと呼ばれる並列処理記述用言語で記述される。我々は、トランスピュータの持つ並列処理機能と通信機能を利用して、計算機システムを構築し、本ネットワークアーキテクチャの諸機能の検証を行なった。

<5.1> ネットワークアーキテクチャの諸機能の検証

本ネットワークアーキテクチャの情報の授受の様子を見るための例題として、アプリケーションソフトウェア上にソーティングのアルゴリズムを載せた。リニアアレイ型の評価に対してはバブルソートを、ハイアラキー型の評価に対してはヒープソートを実行した。リンク故障は、物理的にリンクをはずすことにより発生させ、ノード故障は、トランスピュータを一時的に停めることで実現した。

各ノードの状態と情報の授受の様子は、モニター用のトランスピュータを、付加し、図11のようにCRT上に表示して確認し、正常時・故障時の各ルーチングの様子、機能の代替等の検証を行なった。

<5.2> 電力への応用

また、本ネットワークアーキテクチャの電力への応用の一例として、需給制御システムを採り上げ、分散型の需給制御アルゴリズムを検討し、その適用を通してネットワークアーキテクチャの諸機能を検証すると共に、分散型システムの特性を評価した。図12にその一例を示す。

シミュレーション結果より、一般に分散型システムは、ローカルな変動に強いことが確かめられた。

6. あとがき

今後、このような自律機能を備えたシステムの実現が、より重要になると思われる。本ネットワークアーキテクチャは、電力システムの運用制御システムのみならず、他の制御システムや分散型・並列処理型計算機システム等の情報ネットワークや通信ネットワーク等、幅広く適用できるものである。

参考文献

- (1) D.P. Siewiorek: "Architecture of Fault-Tolerant Computers," Computer, Vol.17, No.8, pp.9, Aug. 1984.
- (2) 石井「分子生物学とシステム工学」電学誌 102巻 1号 pp.42 (昭57)
- (3) 井原,大島「自律分散システムとその応用」電学誌 104巻 3号 pp.169 (昭59)
- (4) 田中,島田,武藤,田岡,西田,坂口「自律機能を備えた分散形ネットワークアーキテクチャ」電学論C 107巻 4号 pp.349 (昭62)
- (5) 黒川,相磯「並列処理の諸問題:結合方式」情報処理 27巻 9号 pp.1005 (昭61)
- (6) D.K. Pradhan: "Dynamically Restructurable Fault-Tolerant Processor Network Architectures," IEEE Tr.Comput., Vol.C-34, No.5, pp.434, 1985.