

組込みシステムのIPネットワークにおける自律的な経路制御

森幸克†

松本倫子†

吉田紀彦†

† 埼玉大学大学院理工学研究科

1 はじめに

近年、組込みシステムの大規模化、複雑化とともにシステムを分散化し、ネットワークを通じて協調動作させるケースが増加しつつある。そのような分散型の組込みシステムでは、ノードの追加、交換、取り外し等によるネットワークの構成の変更は手作業で行われることが多く、頻繁に構成を変更することが困難な現状にある。また、システムが大規模化していけばネットワークも大規模化し、ネットワークを構成すること自体が多大な負担になると考えられ、ネットワークを自動構成することへの要望が高まってきている。ネットワークを自動構成するための要素技術には、AutoIP [1], Multicast DNS [2] 等、標準化されたものもあるが、それらは単一のサブネット内での利用を前提としており、階層化したネットワークでは直接使うことが出来ないと考えられる。

本研究では、サブネットを越えた通信を実現するため、まず日本信号株式会社との共同研究として、名前による経路制御を提案する。さらに実際のシステム構築のための第一段階として、設計の工程にあたるモデル化を行う。

2 提案手法

本研究ではネットワークを構成する際、名前の割り当てを行い、その名前に基づいた経路制御を行う手法を提案する。このとき、ネットワークは階層構造(木構造)であることを前提とする。これは、階層構造を利用することで、性能面で厳しい制約を持つ組み込みシステムにおいて、実装しやすい軽量のシステムとするためである。

また、ここで提案する手法はJavaベースのシミュレーション実験を行い有用性を確認した。

2.1 名前の構造

ノードに割り当てられる名前はドメインに似た構造をしており、KK1.KB2.DS1のような形式をとる。ここで、KK, KB, DS は提供されるサービスの種類を示している。また、親ノードの名前を含んでおり、この場

合はKB2.DS1が親ノードである。説明の便宜上、名前(フルネーム)の最初の“.”より左側の文字列をローカルネーム、右側の文字列をペアレントネームと呼ぶことにする。

ローカルネームは兄弟ノード間で重複しないように決定され、次の形式をとる

$$\left(\begin{array}{c} \text{ノードが提供する} \\ \text{サービス名} \end{array} \right) \left(\begin{array}{c} \text{同種のサービス名を持つ} \\ \text{兄弟ノードの値と重複} \\ \text{しない最小の自然数} \end{array} \right)$$

ペアレントネームは親ノードの名前そのものである。

2.2 Auto Naming Protocol

Auto Naming Protocol(ANP)は、本研究で提案する名前割り当て手法である。ANPの手順の概要は次のようになる。ここで、AutoIPによるIPアドレスが割り当て済みであることを前提とする。

1. IPアドレス割り当て後、一定時間の間、親ノード・兄弟ノードの存在を探查する。
2. ローカルネームの候補を決定し、一定時間の間、兄弟ノードと衝突の有無を確認する。衝突した場合はIPアドレスの値が最も小さいノードを優先して決定する。
3. 親ノードの名前が決定次第、自ノードの名前も決定し、それを子ノードに通知する。
4. 自ノードの名前が確定した後も、一定の間隔で親ノードの存在を探查する。

2.3 Name Based Routing

ANPで設定された名前にはネットワークにおける位置情報が含まれており、これを利用して上位アプリケーションにおけるメッセージ等のルーティングを行う。そのルーティング方式が、本研究で提案するName Based Routing(NBR)である。NBRにおけるルーティングの手順は次のようになる。

1. 宛先が自ノードの名前と一致する場合、そのメッセージは自ノードで処理する。

Autonomous Routing in IP-based Embedded System Network
†Yoshikatsu MORI †Noriko MATSUMOTO †Norihiko YOSHIDA
†Graduate School of Science and Engineering, Saitama University

2. 宛先が子ノード以下，親ノード及びその子ノード以下，それ以外のいずれかに該当するか判定する．これは宛先に自ノードおよび親ノードの名前が後方一致で含まれるかどうかで判定される．
3. 宛先に応じて適切なノードへ転送する．実際に転送するに当たっては各サブネット内で名前解決を行う必要があるが，これには Multicast DNS を利用することとする．

3 目的

ここまでで名前の割り当てと，その名前に基づく経路制御手法を提案した．しかし，ここまでは手法を提案し，Java ベースのシミュレーション実験を行なった段階である．実際にシステムを実装する場合には，例えばこの部分はハードウェアで，この部分はソフトウェアで実装するというような，詳細な設計が必要となる．

そこでさらに本研究では，実際のシステムの実装に向けモデルベース開発に基づいて，詳細な設計の第一歩としてモデルの作成を行う．

モデルベース開発とは，開発の初期段階からソフトウェア設計の段階までモデルを活用することで，製品のライフサイクル全般に亘った品質向上と開発効率向上を目指した開発手法のことである [3]．ここでいうモデルとは，対象となるシステムの仕様を抽象的に記述したもので，実行可能である．そのモデルを実行し出力を得ることで，エラーの早期検出や，特定の出力結果が明示的になることで設計の詳細化を進めることが可能となる．また，本研究ではモデルの作成には，制御業界で広く用いられている MATLAB/Simulink を用いる．

4 方針

提案手法を最初から全てモデルに織り込むことは困難であるため，段階的に機能を追加し，最終的なモデルを作成する．まず第一段階として，基本的な通信機能，ここではメッセージの送受信を行うモデルを作成した．作成したモデルの一部を図 1 に示す．ここで，一つ一つの四角はノードを，それらを結ぶ線はリンクをそれぞれ表している．

また図 1 のモデルの記述の仕方は回路図に近いものであり，ハードウェア実装をより意識した記述となっている．よりソフトウェア実装を意識した記述として，MATLAB/Simulink の Stateflow を用いることが考えられる．Stateflow とは，状態遷移図と制御フロー図の組み合わせをベースとして，状態の遷移をよりグラフィ

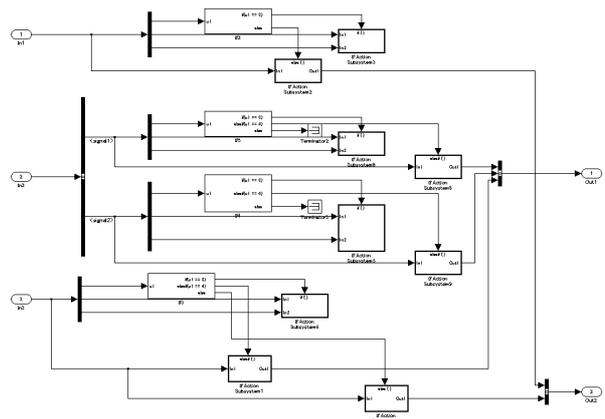


図 1: 作成したモデルの一部

カルに記述できるツールである．一般に，ハードウェアで実装した場合はより高速な処理が可能になり，ソフトウェアで実装した場合にはより拡張性に富むことになるが，リソースに厳しい制約がある組み込みシステムでは，両者の実装の違いはさらに大きな問題となる．このように，どの部分をよりハードウェア実装を意識して，あるいはソフトウェア実装を意識して記述するのかということもモデル化において重要な項目である．

5 まとめ

本研究では，サブネットを越えた通信を実現するために名前による経路制御手法を提案し，さらに実際のシステム実装に向けた第一歩として，モデルの作成を行った．現状は，NBR を取り入れたモデルの作成を行い，モデルを実行することで NBR によるノード間でのメッセージの送受信が行われていることを確認した．今後の課題としては，ANP を取り入れたモデルの作成，提案手法の実機での動作確認などが挙げられる．

参考文献

- [1] S. Cheshire, B. Aboba, and E. Guttman, *Dynamic Configuration of IPv4 Link-Local Addresses*, <http://www.ietf.org/rfc/rfc3927.txt>, May 2005.
- [2] S. Cheshire and M. Krochmal, *Multicast DNS*, <http://files.multicastdns.org/draft-cheshire-dnsext-multicastdns.txt>, December 2011.
- [3] 山田元美, “ 組み込みシステムにおけるモデルベース開発 (MBD) 技術者のスキル標準 ”, SEC journal No. 13, pp. 48-51, 情報処理推進機構, 2008