

UNP におけるネットワーク自動構築機能

加藤 淳[†] 藤内 俊一[†] 諸隈 立志[†] 坂村 健^{†‡}

YRP ユビキタス・ネットワーキング研究所[†] 東京大学大学院情報学環[‡]

E-mail: {atsushi, tounai, moro, ken}@ubin.jp

我々は、身のまわりのいたるところにコンピュータが組込まれるユビキタスコンピューティングにおいて、次世代の制御系ネットワークプロトコルとして UNP: Ubiquitous Networking Protocol を提案する。UNP は、ユビキタスコンピューティングでの制御ネットワーク技術の要件となる 1. リアルタイム性, 2. セキュア性, 3. 運用容易性という 3 つの特徴を持つ。本稿では、運用容易性を実現する UNP のネットワーク自動構築機能について、評価を行う。そして、評価を通して、UNP の運用容易性が実用レベルであることを確認する。

The Network Auto Configuration Mechanism in UNP

Atsushi Kato [†], Shunichi Tonai [†], Tatsushi Morokuma [†], and Ken Sakamura^{†‡}

YRP Ubiquitous Networking Laboratory. [†]

Interfaculty Initiative in Information Studies, Graduate School of University of Tokyo. [‡]

We propose UNP: Ubiquitous Networking Protocol, which is the next generation control networking protocol for the ubiquitous computing. UNP has three characteristics, 1. realtime response, 2. security, 3. ease of operation, to be satisfied with the control network requirements for the ubiquitous computing. In this paper, we evaluate the network auto configuration mechanism to realize an ease of operation in UNP. We show that UNP can achieve an ease of operation for the real-world application.

1 はじめに

近年、回路技術の進歩に伴う LSI の小型化・低価格化を背景に、ユビキタスコンピューティングが社会に浸透しつつある。ユビキタスコンピューティングとは、身の周りのいたるところにコンピュータを組込むことにより、人間生活を高度にサポートし、人々に快適性を提供する技術の総称である。一方、あらゆる分野において、環境保全に対応し、資源・エネルギーを無駄なく有効に活用する社会として、持続型・循環型経済社会の実現が求められている。今、ユビキタスコンピューティングを適用することによって、「快適性」と「環境負荷の低減」の両立を図る 21 世紀型社会モデルの構築に期待が高まっている。その実現には、コンピュータが組込まれた機器に対するきめ細かい制御や、機器同士が協調して動作することを可能にする制御ネットワーク技術が不可欠となる。

ユビキタスコンピューティングにおける制御ネットワークは、1. 制御ネットワークであるため、厳密なリアルタイム性を保証すること, 2. 生命や財産に深く関わるネットワークであるため、高いセキュリティを保証すること, 3. 制御対象となる機器の数が膨大に増えるため、運用容易性を持つこと, という 3 つの要件を満たす必要がある。

ここで、既存の制御ネットワーク技術を見渡すと、その中には既にデファクトといえる技術も存在する。ビルなどにおいて設備機器を制御する LonWorks[1]

や、自動車の車載ネットワークとして、パワートレイン系やボディ系で採用される CAN[3] である。また最近では、情報家電の登場によってホームネットワークが再び脚光を浴びており、国内では ECHONET[2] が注目を集めている。しかし、これらの制御ネットワーク技術は、前述の 3 つの要件を満たしているとは言い難く、来たるユビキタス社会のニーズに対応していない。

そこで、我々は、ユビキタスコンピューティングにおける制御ネットワーク技術に必要な要件すべてを満たす次世代の制御系ネットワークプロトコルとして、UNP: Ubiquitous Networking Protocol を提案する。UNP は、1. 改良型トークンパッシング方式による厳密なリアルタイム性の保証, 2. eTRON[4] アーキテクチャによる高いセキュリティの保証, 3. ネットワーク自動構築機能による運用容易性の実現, という特徴を持つ。本稿では、運用容易性を実現する UNP のネットワーク自動構築機能について、その詳細を説明し、評価を行った。その結果、運用容易性が実用レベルであることを確認した。

本稿の構成は、次の通りである。第 2 章では、関連研究について説明する。第 3 章では、UNP のネットワークトポロジ及びプロトコル階層構造、そしてネットワーク自動構築機能について説明する。第 4 章では、ネットワーク自動構築機能について評価する。そして最後に、本稿についてまとめる。

2 関連研究

既存の制御ネットワーク技術に、LonWorks、CAN、ECHONETがある。本章では、ユビキタスコンピューティングにおける制御ネットワーク技術の要件という観点から、LonWorks、CAN、ECHONETの説明を行う。

LonWorksは、ビルなどにおいて設備機器を制御する制御ネットワーク技術である。伝送媒体としてツイストペア線を使用する場合、通信速度は最大1.25Mbps、伝送距離は最大300mとなる。最大接続ノード数は約32,000ノードで、データリンク層レベルでのデータ長は最大228byteである。また、LonWorksは、BA(Building Automation)の分野では、圧倒的なシェアを占め、既にデファクトという地位を築いている。しかし、ユビキタスコンピューティングにおける制御ネットワーク技術の要件という観点からLonWorksを検証すると、その問題点が見えてくる。第1に、アクセス方式として予測CSMA方式を採用している。CSMA方式とは、データの送信を行う場合、はじめに伝送媒体が使用されているかどうかを調べ、使用されている場合は、使用されなくなるまでデータの送信を待機する方式である。制御系におけるリアルタイム性とは、最大遅延時間あるいは最大待ち時間の保証を意味する。CSMA方式は、伝送媒体の使用率が高くなるにつれ、各ノードでデータ送信に要する時間が予測不可能となる欠点があり、データを送信するまでにかかる最大遅延時間を算出することができない。よって、LonWorksは、リアルタイム性を保証しているとは言い難い。第2に、LonWorksは、そのプロトコル仕様として、セキュリティが規定されていない。第3に、LonWorksのネットワークにおいて機器の増設を行う場合、特定のツールを用いてその論理アドレスを静的に設定する必要がある。また、あるノードが不意にネットワークから減設された場合、システムはそれを検知することができない。これらの制約から、LonWorksは運用容易性を有していない。

CANは、自動車の車載ネットワークとして採用される制御ネットワーク技術である。伝送媒体としてツイストペア線を使用する場合、通信速度は最大1Mbps、伝送距離は最大40mとなる。接続ノード数の仕様の制限はなく、データリンク層レベルでのデータ長は8バイト固定である。CANは特に、エンジン/ブレーキ制御などのパワートレイン系やドア/ミラー制御などのボディ系の車載ネットワークのデファクトである。しかし、CANについても、ユビキタスコンピューティングにおける制御ネットワーク技術の要件という観点から検証すると、いくつかの問題点が見つかる。第1に、アクセス方式として優先度ベースのCSMA方式を採用している点である。前述の様に、CSMA方式ではリアルタイム性を保証しているとは言い難い。第2に、LonWorksと同様に、そのプロトコル仕様として、セキュリティが規定されていない。第3に、

ノードの不意な減設が発生した場合でも、システムはそれを検知することができず、運用容易性に課題が残る。

ECHONETは、ホームネットワークにおいて情報家電などを対象とする制御ネットワーク技術である。伝送媒体として電灯線を使用する場合、通信速度は最大9.6Kbps、伝送距離は減衰に依存する。接続ノード数は約64,000ノードで、データリンク層レベルでのデータ長は最大255バイトである。ホームネットワークが脚光を浴びる中、国内の大手家電メーカーからECHONET対応製品が販売されるなど、多くの注目を集めている。しかし、ECHONETについても、ユビキタスコンピューティングにおける制御ネットワーク技術の要件という観点から検証すると、いくつかの問題点が見つかる。第1に、LonWorks、CANと同様に、アクセス方式としてCSMA方式を採用している点である。前述のように、リアルタイム性を保証しているとは言い難い。第2に、ECHONETは、プロトコル仕様としてセキュリティに関する規定はあるが、シンプルなセキュリティモデルであるため、ユビキタスコンピューティングで要求されるセキュリティとしては不十分である。第3に、ノードの不意な減設が発生した場合でも、システムはそれを検知することができず、運用容易性に課題が残る。

以上のように、要件となるリアルタイム性、セキュリティ、運用容易性の3点すべてを満足する制御ネットワーク技術は、現状存在しない。これが、UNPを提案する背景となっている。

UNPのアクセス方式は、改良型のトークン・パッシング方式を採用している。トークン・パッシング方式とは、1つのネットワーク単位内でトークンという送信権をノード間で巡回させ、トークンを持つノードだけがデータを送信することができる方式である。この方式の場合、ネットワークに接続されているノードの数から最大遅延時間を算出することが可能である。つまり、UNPはネットワークの効率性よりもむしろ、決められた時間内で必ずパケットを送信することができるという信頼性に重きを置いたプロトコルである。また、UNPはeTRONアーキテクチャを採用したセキュリティ層を規定している。これにより、eTRONが持つ耐タンパ性によって、暗号処理および暗号鍵を厳重に制御・管理することが可能となり、高いセキュリティを実現している。そして、運用容易性を実現するネットワーク自動構築という機能を有する。ネットワーク自動構築機能とは、機器をネットワークに増設する場合に、ネットワークの他のノードが協調し、新しく接続されるノードの論理アドレスを動的に決定したり、機器がネットワークから減設される場合、ネットワークの他のノードが自動的にそれを検出し、特定のパケットを送信することで減設が発生したことをシステムに通知する機能である。この機能は、制御対象が多いほど、その効果を発揮する。

次章より、運用容易性を実現するUNPのネットワーク自動構築機能について、説明する。

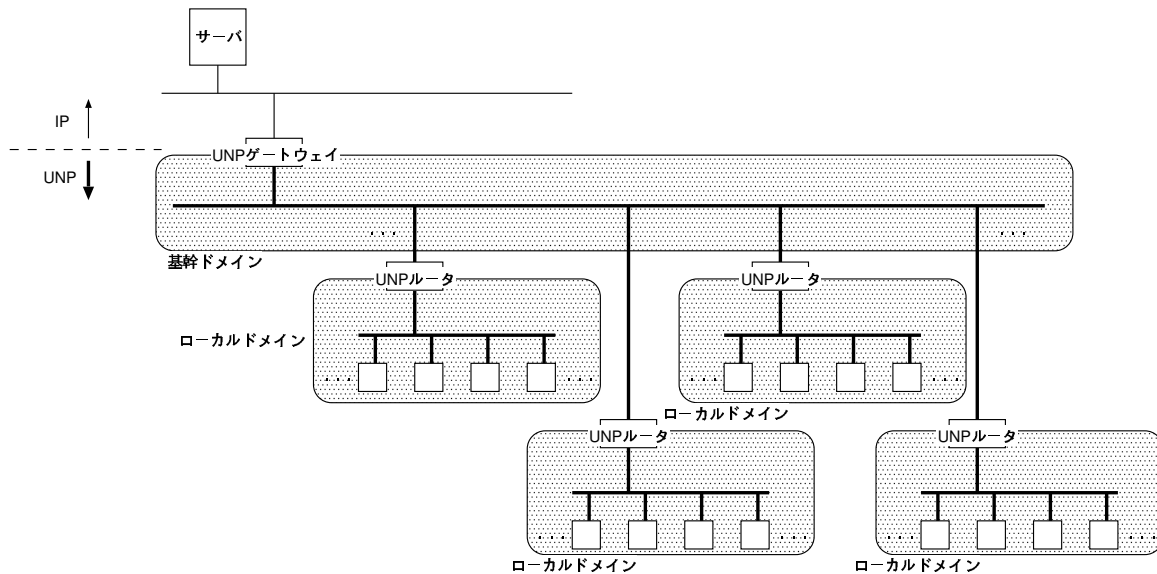


図 1: UNP ネットワークトポロジ

3 UNP のネットワーク自動構築

本章では、UNP のネットワークトポロジ、プロトコル階層構造及び UNP ネットワークのすべてのノードが持つ ucode について説明する。その後、ネットワーク自動構築機能について説明する。

3.1 ネットワークトポロジ

図 1 に、UNP のネットワークトポロジを示す。トークンが巡回するネットワーク単位をドメインと呼ぶ。そして、制御対象となる機器（ノード）を接続するドメインをローカルドメインという。ローカルドメイン内において、ノードは、1 バイトのアドレス空間を持つノード ID で識別される。1 ローカルドメインあたり、最大で 254 ノードが接続可能である。また、システムの拡張性を考慮し、UNP は階層型のネットワークトポロジを持つ。ローカルドメインは UNP ルータによって区切られ、更に基幹ドメインに收容される。ドメインは、1 バイトのアドレス空間を持つドメイン ID で識別される。基幹ドメインには、最大で 254 台の UNP ルータが接続可能である。つまり、UNP は、制御対象ノード数が数万点という大規模施設への適用や、プロトコルとしての処理性能・妥当性を考慮して、2 バイトのアドレス空間を有し、システム全体として約 64,000 ノードを接続することができる。そして、ノードは UNP システム内で、ドメイン ID とノード ID の組で一意に識別される。また、汎用 PC をサーバとして用い、UNP が制御する機器の構成管理などを行うことを想定し、UNP ネットワークは、UNP ゲートウェイを介して、IP ネットワークに接続することが可能である。

3.2 プロトコル階層構造

図 2 に、UNP のプロトコル階層構造を示す。

物理層 UNP としては、特に規定しない。試作版では、伝送媒体として、ツイストペアケーブルを使用した。また、電気的な規格として、EIA-485 を採用した。

データリンク層 アクセス方式として、改良型トークンパッシング方式を採用し、厳密なリアルタイム性を実現している。また、ネットワーク自動構築機能を規定している。

セキュリティ層 高セキュリティを提供する eTRON アーキテクチャを適用し、データの暗号化/復号やメッセージの認証を行う。eTRON は、暗号処理用のチップを搭載した耐タンパ性を有するハードウェアである。

トランスポート層 UNP のデータリンク層レベルで扱うデータの最大長は、247 バイトである。ただし、運用容易性を考慮し、ネットワーク経由でプログラムを更新することを想定している。よって、247 バイト以上のデータについても送信/受信が可能ないように、パケット分割/組立に関する仕様を規定している。

アプリケーション層 機器の協調動作に関する機能などを担当する。

3.3 ucode

ucode[5] とは、実世界の様々な「モノ」に一意に付与される 128 ビットの ID である。UNP は ucode を強く意識したプロトコル仕様となっており、ucode をネットワーク自動構築時に使用する。

3.4 データリンク層パケットフォーマット

図 3 に、データリンク層のパケットフォーマットを示す。また、パケットの各フィールドについて、表 1 に示す。ここで、ドメイン ID 及びノード ID

アプリケーション	機器制御アプリケーション
アプリケーション層	・協調動作 など
トランスポート層	・パケット分割/組立
セキュリティ層	・暗号化/復号 ・メッセージ認証
データリンク層	・ネットワーク自動構築機構 ・改良型トークンパッシング方式
物理層	特に規定しない

図 2: UNP プロトコル階層構造

DFI	DDID	DNID	SDID	SNID	PT	OPC	LEN	DATA	CRC
-----	------	------	------	------	----	-----	-----	------	-----

図 3: データリンク層パケットフォーマット

について、0x00 は未定義、0xFF はブロードキャストと定義する。表 2 に、パケット送信時の DDID 及び DNID についてまとめる。

3.5 ネットワーク自動構築のシーケンス

UNP ネットワークに接続されるノードは、ネットワーク自動構築を行うために、表 3 に示す各ノード ID を保持する。

3.5.1 初期状態

ネットワークにノード alpha が存在したとする。alpha は、ある一定の間隔で、ネットワークにパケットが流れているかどうかを監視している。このタイムをアイドル状態検出タイムと呼ぶ。また、プローブパケットとは、プローブパケットに対する応答があるかどうかで、ネットワーク内の自分以外のノードの存在を確認するパケットである(図 4)。アイドル状態検出タイムのタイムアウトが 7 回連続で発生した場合、alpha はプローブパケットを送信する(図 5)。

プローブパケットの送信後に、プローブ応答パケットを受信しなかった場合、alpha は、再びアイドル状態検出タイムを起動する。アイドル状態検出タイムが 3 回連続でタイムアウトした場合に、再度プローブパケットを送信する(図 6)。

図 6 のプロセスは、alpha がプローブ応答パケットを受信するまで繰り返される。また、以前使用していたノード ID がない場合、alpha のノード ID は 0x01 とする。

3.5.2 2 台目接続

ここで、ノード bravo がネットワークに接続されたとする。bravo は、アイドル状態検出タイムのタイムアウト回数の違いのため、必ず alpha が送信したプローブパケットを受信する(図 7)。

表 1: データリンク層ヘッダ

略称	サイズ (byte)	説明
DFI	1	送信先識別子 ネットワーク自動構築機能では、0x01 を固定で使用する。
DDID	1	送信先ノードのドメイン ID
DNID	1	送信先ノードのノード ID
SDID	1	送信元ノードのドメイン ID
SNID	1	送信元ノードのノード ID
PT	1	パケットタイプ 0x00: トークン 0x01: プローブ 0x02: 勧誘 0x03: 勧誘結果通知 0x06: next 更新指示 0x12: 減設通知
OPC	1	オペレーションコード 0x00: 通知 0x10: 要求 (Read) 0x11: 要求 (Write) 0x20: 応答 (肯定) 0x2?: 応答 (否定) ?=1~6(要因別)
LEN	1	DATA 部の長さ
DATA	0~247	データリンク層のデータ部
CRC	2	DFI から DATA に対する CRC

表 2: パケット送信時の DDID 及び DNID

DDID	DNID	説明
0x00	0x??	自ドメインの特定ノードに対するユニキャスト
0x00	0xFF	自ドメインの全ノードに対するブロードキャスト
0x??	0x??	特定ドメイン/特定ノードに対するユニキャスト
0x??	0xFF	特定ドメインの全ノードに対するブロードキャスト
0xFF	0x??	全ドメインの特定ノードに対するユニキャスト
0xFF	0xFF	全ドメインの全ノードに対するブロードキャスト

0x?? : 0x01 - 0xFE

表 3: 保持するノード ID

ノード ID	説明
前段ノード ID	自ノードにトークンを渡すノード (前段ノード) のノード ID
次段ノード ID	自ノードがトークンを渡すノード (次段ノード) のノード ID
次々段ノード ID	次段ノードがトークンを渡すノード (次々段ノード) のノード ID

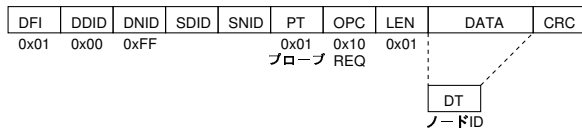


図 4: プローブパケット

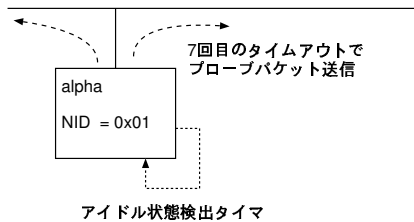


図 5: プローブパケット送信 (1 回目)

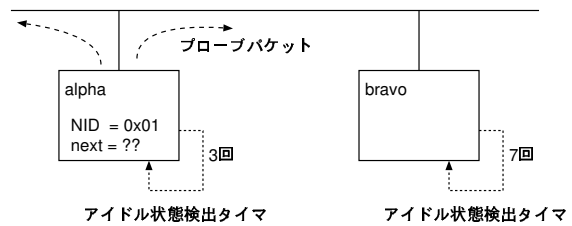


図 7: 2 台目接続



図 8: プローブ応答パケット

プローブパケットを受信した bravo は、送信元の alpha に対して、プローブ応答パケット (図 8) を送信する。bravo は、プローブ応答パケットを送信する時点で、以前使用していたノード ID がある場合は、そのノード ID を使用する。以前使用していたノード ID がない場合、bravo のノード ID は 0x02 となる。また、安定状態になると、ノードはトークン (図 9) を巡回させることになる。その場合、それぞれのノードは、自分がトークンを送信するノードを把握している必要がある。よって、bravo は、プローブパケットから、自分が次にトークンを送信する alpha のノード ID (次段ノード ID) を保持する (図 10)。

alpha は、bravo からプローブ応答パケットを受信すると、受信したパケットの SNID から、次段ノード ID として、bravo のノード ID を保持する (図 11)。

3.5.3 安定状態

alpha は、トークンを渡すノード、つまり bravo に対してトークンを送信する。同様に、トークンを受信した bravo も、自分がトークンを渡すノード alpha に対して、トークンを送信する。この時点でネットワークが安定状態となる (図 12)。ここで各ノードは、受け取るトークンから、前段ノード ID を保持する。また、ネットワークに最初に接続された alpha は、安定状態においてノードの増設を行うドメインマスタとなる。

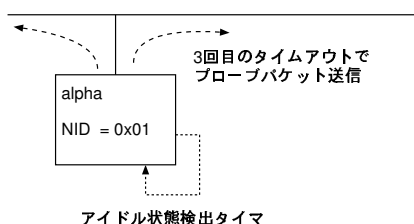


図 6: プローブパケット送信 (2 回目以降)

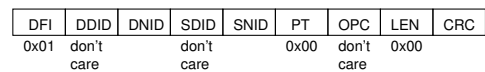


図 9: トークンパケット

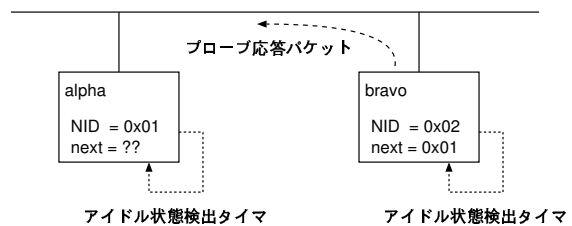


図 10: プローブ応答パケット送信

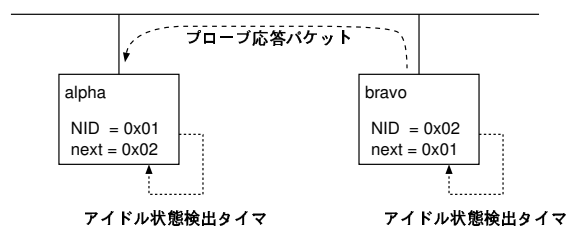


図 11: プローブ応答パケット受信

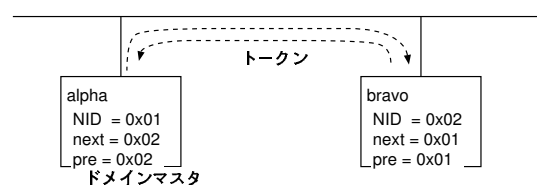


図 12: 安定状態

3.5.4 ノードの増設

alpha と bravo で構成される安定状態のネットワークに、新しいノード charlie を増設する。charlie もアイドル状態検出タイマを起動させるが、すでにネットワークが安定状態に移行し、定期的にトークンが流れているため、charlie のアイドル状態検出タイマがタイムアウトすることはない。ドメインマスタである alpha は、ネットワークに定期的に勧誘パケット (図 14) をブロードキャストして、ネットワークに参加するノードがあるかどうかを検査している。図 13 に、増設シーケンスを示す。

1. ドメインマスタ (alpha) は、一定周期毎に勧誘パケットをドメイン内にブロードキャストする。この時、既にネットワークに接続されている bravo は、このパケットを無視する。
2. 参加待ち状態の charlie は、実世界で一意となる ucode を元に算出したバックオフ時間の後、勧誘応答パケット (図 15) により保持しているノード ID をドメインマスタ (alpha) に申告する。保持しているノード ID がない場合は、未定義値 (0x00) を申告する。
3. 勧誘応答パケット受信後、ドメインマスタ (alpha) はトークンを送信する。その後、申告されたノード ID の重複検査を行う。重複していない場合は、そのノード ID を割り当てる。重複している場合、もしくは以前使用していたノード ID がない場合は、未使用で最若番のノード ID を割り当てる。
4. ドメインマスタ (alpha) は、次のトークンを受信後、勧誘結果通知パケット (図 16) により、勧誘結果をドメイン内にブロードキャストする。参加待ち状態であったノード charlie は、勧誘結果通知パケット中の ucode と自身の ucode を比較し、自身の値と同じ場合は、格納されているノード ID と取り込み、自分がトークンを渡すノードのノード ID (ドメインマスタである alpha のノード ID) を、次段ノード ID として保持する。
5. ドメインマスタ (alpha) は、charlie からの勧誘結果通知応答パケットを受信後、トークンを送信する。
6. 再度トークンを受信後、ドメインマスタ (alpha) は自身の前段ノード ID を元に、前段の bravo に対して next 更新指示パケット (図 17) を送信し、bravo がトークンを渡すノードを変更する。
7. ドメインマスタ (alpha) は、bravo からの next 更新指示応答パケットを受信後、トークンを送信する。
8. bravo は、charlie にトークンを送信する。
9. charlie は、alpha にトークンを送信する。

増設された charlie は、論理リング的にドメインマスタである alpha の前段に挿入される。また、各ノードは次段ノードがどのノードにトークンを送信しているかを監視し、次々段ノード ID を保持する。4 台目以降も、同様のシーケンスにて増設を行う。

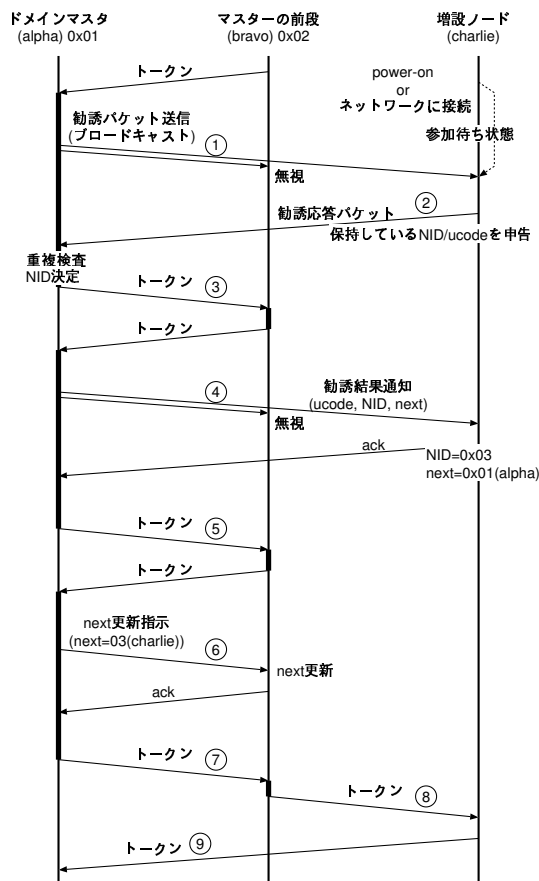


図 13: 増設シーケンス

DFI	DDID	DNID	SDID	SNID	PT	OPC	LEN	CRC
0x01	0x00	0xFF			0x02 勧誘	0x10 REQ	0x00	

図 14: 勧誘パケット

DFI	DDID	DNID	SDID	SNID	PT	OPC	LEN	DATA	CRC
0x01	0x00				0x02 勧誘 送信ノードのNID	0x20 ACK	0x10	DT ucode	

図 15: 勧誘応答パケット

DFI	DDID	DNID	SDID	SNID	PT	OPC	LEN	DATA	CRC
0x01	0x00	0xFF			0x03 勧誘 結果通知	0x11 REQ	0x12	16 ucode値 1 NID 1 next	

図 16: 勧誘結果通知パケット

DFI	DDID	DNID	SDID	SNID	PT	OPC	LEN	DATA	CRC
0x01					0x06 next 更新指示	0x11 REQ (set)	0x01	DT nextのNID値	

図 17: next 更新指示パケット

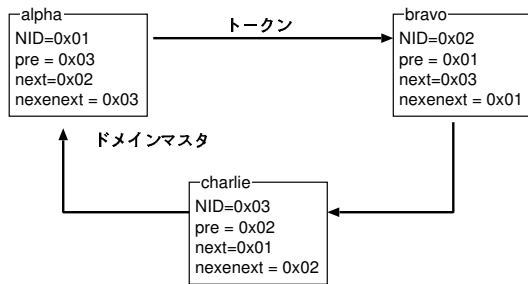


図 18: 増設位置

3.5.5 ノードの減設

alpha, bravo, charlie, そしてノード delta が安定状態にあるネットワークから, 不意にノード charlie が減設される場合を示す (図 19)。

1. 次段監視タイマとは, 次段ノードがパケットを送信するかどうかを監視するタイマである。各ノードはトークンを送信後, 次段監視タイマを起動して, 次段のノードを監視している。安定状態では, 常にトークンがネットワークを流れているので, 次段監視タイマがタイムアウトすることはない。
2. 不意に, charlie の減設が発生したとする。それを知らない bravo は, charlie に対してトークンを送信する。
3. bravo の次段監視タイマがタイムアウトする。この時, bravo は, charlie のドメイン ID 及びノード ID で, charlie に成り変わり減設通知パケット (図 20) を送信する。この減設通知パケットにより, システムは charlie が減設したことを検知することができる。
4. bravo は, 次々段ノード ID を元に, 次々段の delta に対して, トークンを送信する。
5. delta は, bravo が送信したトークンを受信する。bravo は, delta が alpha に対してトークンを送信した時点で, delta を自身の次段とし, delta のノード ID を次段ノード ID とする。また, delta が送信したトークンの SNID から, alpha を自身の次々段とし, alpha のノード ID を次々段ノード ID とする。
6. 次回, bravo が送信するトークンは, delta 宛となる。

4 評価

本章では, 運用容易性を実現する UNP のネットワーク自動構築機能に関して, 評価と考察を行う。

4.1 nT-Engine

我々は, UNP が動作する専用の小型ネットワークノードである nT-Engine を試作した。図 21 に, nT-Engine を示す。nT-Engine は, UNP のデータリンク層をハード化した UNP コントローラと, CPU コアを 1 チップ化することで, 小型化に成功している。

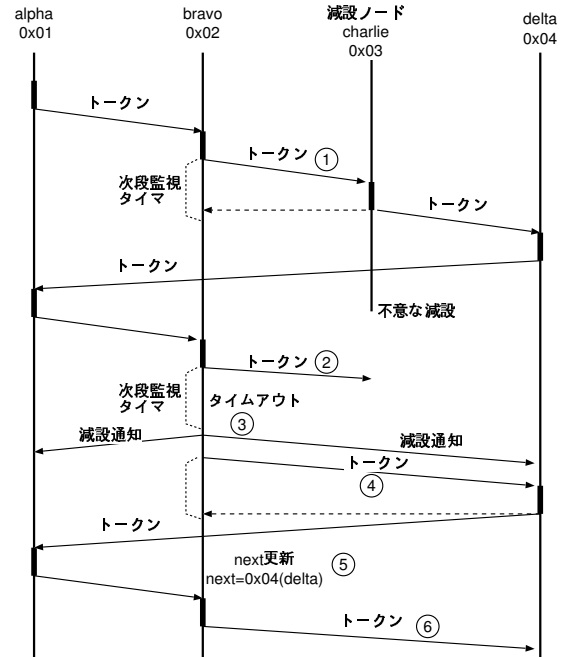


図 19: 減設シーケンス

DFI	DDID	DNID	SDID	SNID	PT	OPC	LEN	CRC
0x01	0xFF	0xFF			0x12	0x00	0x00	

図 20: 減設通知ケット

また, セキュア性を実現する SIM 形状の eTRON を接続するため, SIM ソケットを標準で搭載する。

4.2 ネットワーク自動構築機能の評価

安定状態にあるドメインにおいて, 最もネットワーク的負荷の低い状態は, ノード間でトークンを巡回させているだけの状態である。このケースを best-case と定義する。そして, 最もネットワーク的負荷の高い状態は, ドメイン内の各ノードが常にデータリンク層の最大データ長である 247 バイトのデータを送信している状態である。このケースを worst-case と定義する。また, nT-Engine は, 1 バイトのデータを送信するのに実測で, 5.12 μ s を要する。これを, 1byte 送信時間とする。

最大増設時間に関して, nT-Engine を用いた best-case/worst-case の計算式を以下に示す。

best-case

$$\begin{aligned}
 \text{送信データ (byte)} &= \text{勧誘} + \text{勧誘結果通知} + \text{next 更新} + \\
 &\quad (\text{トークン} * \text{トークン巡回回数} * \text{ノード数}) \\
 &= 177 + (12 * 3 * n) \\
 \text{最大増設時間 } T_{ib}(\text{msec}) &= \text{送信データ} * 1\text{byte 送信時間}/1000 \\
 &= (177 + (12 * 3 * n)) * 5.12/1000
 \end{aligned}$$

worst-case

$$\begin{aligned}
 \text{送信データ (byte)} &= \text{勧誘} + \text{勧誘結果通知} + \text{next 更新} + \\
 &\quad ((\text{最大データ} + \text{トークン}) * \\
 &\quad \text{トークン巡回回数} * \text{ノード数}) \\
 &= 177 + (271 * 3 * n) \\
 \text{最大増設時間 } T_{iw}(\text{msec}) &= \text{送信データ} * 1\text{byte 送信時間}/1000 \\
 &= (177 + (271 * 3 * n)) * 5.12/1000
 \end{aligned}$$

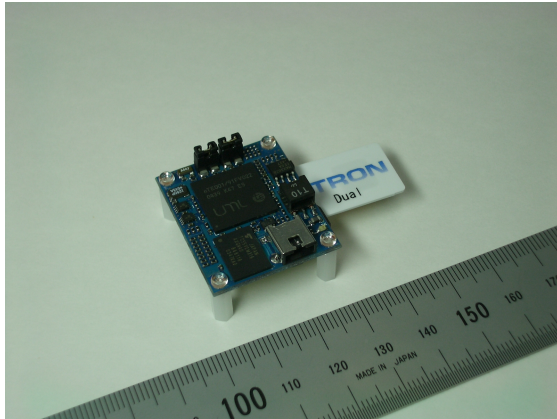


図 21: nT-Engine

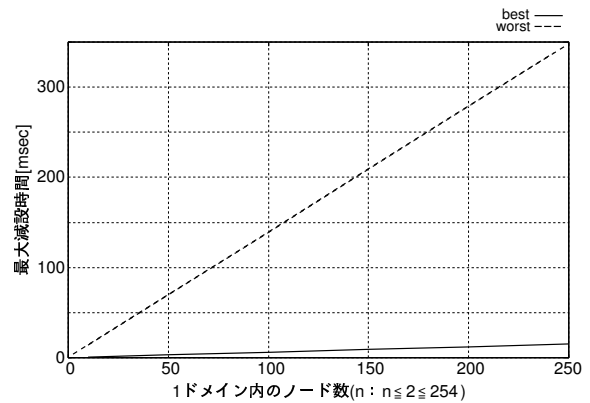


図 23: 最大減設時間とノード数

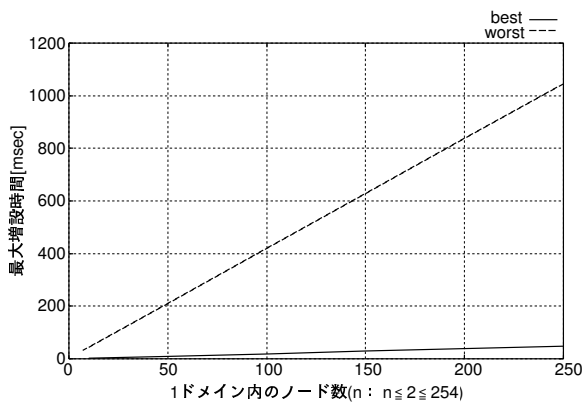


図 22: 最大増設時間とノード数

最大増設時間の結果について、図 22 に示す。この結果は、UNP では 1 ドメインに約 250 台の nT-Engine が接続されている場合でも、1 秒程度で増設が完了することを示している。これにより、nT-Engine をネットワークに接続し、最悪でも 1 秒程度で通信を開始できることがわかる。

また、最大減設時間に関して、nT-Engine を用いた best-case/worst-case の計算式を以下に示す。

best-case

$$\begin{aligned}
 \text{送信データ (byte)} &= \text{preamble} + \text{トークン} + \text{CRC} \\
 &= 12 \\
 \text{トークン送信時間 (usec)} &= \text{送信データ} * 1\text{byte 送信時間} \\
 &= 61.44 \\
 \text{最大遅延時間 } Tdb(\text{msec}) &= \text{トークン送信時間} * \\
 &\quad \text{ノード数}/1000 + \text{次段監視タイマタイマ値} \\
 &= (61.44 * n/1000) + 0.03
 \end{aligned}$$

worst-case

$$\begin{aligned}
 \text{送信データ (byte)} &= \text{preamble} + \text{データ} + \text{CRC} \\
 &= 259 \\
 \text{データ送信時間 (usec)} &= \text{送信データ} * 1\text{byte 送信時間} \\
 &= 1326.08 \\
 \text{最大遅延時間 } Tdw(\text{msec}) &= (\text{データ送信時間} + \text{トークン送信時間}) * \\
 &\quad \text{ノード数}/1000 + \text{次段監視タイマタイマ値} \\
 &= (1387.52 * n/1000) + 0.03
 \end{aligned}$$

最大減設時間の結果について、図 22 に示す。この結果は、UNP では、1 ドメインに約 250 台の nT-Engine が接続されている場合でも、350msec 程度で減設が完了することを示している。これにより、不意に nT-Engine の減設が発生した場合でも、システムは減設通知パケットにより、最悪でも 350msec 程度の遅延で、そのノードの減設を検知できることがわかる。

以上の結果から、UNP は実用的な運用容易性を有していることが確認できる。

5 おわりに

本稿では、運用容易性を実現する UNP のネットワーク自動構築機能について、その詳細を述べ、評価を行った。その結果、実用レベルであることが確認できた。

謝辞

本研究は、情報通信研究機構「コピキタスコンピューティング環境を実現する基盤ネットワークプロトコルの研究開発」の成果の一部が使用されている。

参考文献

- [1] American National Standards Institute, ANSI/EIA 709.1-A-1999 Control Network Protocol Specification, 1999.
- [2] ECHONET CONSORTIUM, “ECHONET Specifications.” <http://www.echonet.gr.jp/>.
- [3] International Organization for Standardization, ISO-11898-1, 1993.
- [4] Ken Sakamura and Noboru Koshizuka, “The eTRON Wide-Area Distributed-System Architecture for E-Commerce,” IEEE Micro, vol.21, no.6, pp.7–13, Dec. 2001.
- [5] Ubiquitous ID Center, “uID Technology.” <http://www.uidcenter.org/index.html>.