

ダイナミックな環境変化を考慮したデータ転送制御を可能とする 低トラフィック型分散処理システムの提案

佐々木靖彦^{†1} 寺西裕一^{†1} 村田正幸^{†2}

従来に比較して圧倒的に大量・多量なモノ、センサー、ヒトがつながれる Internet of Things(IoT)の時代に向けて、著しく増加するトラフィックに起因する種々の問題を解決できる新しいネットワークシステムが求められている。これに対し、我々は、アプリケーションセマンティクスを考慮して“変化”や“違い”を柔軟かつ論理的に定義・表現・処理し、必要な情報のみを厳選して転送可能とすることでシステム中を流れる総トラフィックが抑制できるような分散処理システム技術（デルタエンベデッドネットワークシステム、 Δ NS）を開発してきた。本稿では、このような従来の Δ NSに対する機能拡張として、データ転送の選別を行う際にアプリケーションの重要度に応じた優先度付けを与えられるようにする方式について提案する。ダイナミックな環境変化をも考慮することが可能なきめ細かいトラフィック制御を行うために、特権 Δ DDとユーザ Δ DDよりなる階層化 Δ DDを用いたデータ転送制御手法を活用する。提案する分散処理システムの効果を確認するために行った実験では、バッテリー駆動の処理ノードにおける残存電力（ノード固有状況）が時間とともに変化する仮想ノード環境において、提案手法を用いた処理ノードが適切にトラフィック転送制御を行うことによって、重要度の高いアプリケーションにおいて従来よりも長時間の動作を可能とすることをシミュレーションにより機能確認する。

A proposal of low-traffic distributed processing system that enables data transfer control based on dynamic environmental changes

YASUHIKO SASAKI^{†1} YUICHI TERANISHI^{†1} MASAYUKI MURATA^{†2}

1. はじめに

有無線ネットワーク技術の進展と普及により、従来は個々独立に存在していた小さなモノやセンサーや人が常時ネットワークに繋がるようになり（Internet of Things, IoT）、近い将来、その数は兆単位に及ぶ膨大なものになるといった予測がある[1]。また、膨大な数の情報が取得される場所は、地理的にもモノやセンサーが配備され得る地球上のあらゆる範囲に及ぶようになり、システム構成も、より地理分散性の高いノード集合的な処理形態（pervasive computing）へと発展していくことが予想される。

しかしながら、上述したような膨大な数のモノ、センサー、人のそれぞれが情報を生成するようになれば、それらの情報がシステム中に流入することで膨大なトラフィックの発生要因となる。さらに、地理的に分散して取得/制御される情報を転送/集約して処理する場合には、必然的に広い範囲の情報の流通が発生し、これがトラフィックを著しく増加させる要因となる。

このような大規模な情報流通時代を支えるネットワークシステム基盤の構築にあたっては、トラフィックの収容性や信頼性を高めるために幾つかのアプローチが存在し得

るが、（１）システムの対応能力（キャパシティ）を強化するアプローチ、あるいは、（２）既存システムと同等の機能を提供しながら、より少ないトラフィックでそれを実現するアプローチ、等が考えられる。

例えば、（１）としては、スイッチ、ルータ等の性能や搭載リソース、あるいは、それらの物量を強化することによってシステムの対応能力を強化することが可能である。一方で、トラフィック収容性や信頼性を高めるためのもう一つのアプローチ（２）として、分散する処理ノード間のトラフィックをなるべく発生させないように保ちながらも、本来システムが目的としていた機能と同水準のものを実現することができるようにするといった方法も存在する。

上述の分類において、我々は、特に後者（２）に該当するアプローチとして、“デルタエンベデッドネットワークシステム（ Δ NS）”と呼ばれる技術を開発してきた[2]。 Δ NSでは、個々のアプリケーション開発者の設計意図（アプリケーションセマンティクス）に合わせて“変化”や“違い”を柔軟かつ論理的に定義・表現・処理しながら活用することで、アプリケーションの目的とする結果の生成に本当に必要な情報のみを厳選して転送することが可能となる。これにより、システム中を流れる総トラフィックが抑制されるような低トラフィック型分散処理アプリケーションを構築することが可能となる。

同技術では、個々のアプリケーションが持つセマンティ

^{†1}（独）情報通信研究機構ネットワーク研究本部
National Institute of Information and Communications Technologies
佐々木の現所属は Hitachi America, Ltd.

^{†2} 大阪大学大学院情報科学研究科情報ネットワーク学専攻
Osaka University

クスに依存してトラフィックが制御され、それに応じてトラフィック量も削減される。アプリケーションセマンティクスを用いる方法は、ネットワーク層が関知し得ないアプリケーション層のみが知る固有のロジックや情報をも活用することが可能であるため、トラフィックの削減効果を大きく高めることができるという利点がある（1/100～1/1000クラス[2]）。しかしながら、 Δ NS 自体にはアプリケーションが自発的にトラフィックを制御する仕組みがあるのみで、依然、インフラ側の都合を考慮したトラフィックの制御を行う仕組みは存在しない。

一方で、インフラ側によるトラフィック制御が必要とされる場面も多々ある。例えば、同一インフラ上で動作する複数のアプリケーション間でネットワーク利用度の公平性を保ちたい場合、あるいは、インフラ全体の最適化を進めたい場合などがそれに相当する。トラフィックの優先度制御が行える仕組みが存在すれば、このような目的を達成することができるが、実際、DiffServ[3]に代表されるようなフロー制御の仕組みでは、トラフィックの種類をクラス分けした上でデータ転送に優先度付けを行うことができるようにして、インフラ側の都合を考慮したトラフィック制御が行えるようになっている。

上記2つの視点をまとめれば、1) アプリケーション目的にあわせたトラフィック制御、および、2) インフラ目的にあわせたトラフィック制御、の双方が相互協調的に行えることで、複数のアプリケーションが載るインフラ全体はより望ましい形で構築されるようになると言えるが、将来の分散処理システムでは、さらなるきめ細かい制御が必要となるケースが考えられる。

例えば、将来の先進的な分散処理システムを搭載するインフラは、従来一般的であった少数の管理者に制御されるインフラとは異なり、無数の様々な参加者（処理ノード提供者）により構成される共有的なインフラとして構築されるようになる可能性がある。Internet of Things (IoT) の時代においては、膨大な数のあらゆるものがネットワークに繋がるのが想定されるが、そのようにして繋がれるものの全体像は、まさに多数かつ多様な参加者による共有インフラとなり得るものである。このような共有インフラを考える場合には、多数かつ多様な参加者の都合を考慮することが可能なトラフィック制御の仕組みが必要となる。しかも、IoT の時代には、あらゆる場所に配置されうるモノまでもが接続対象となるため、その設置環境は場所に依存すると同時にダイナミックに変化することが普通である。したがって、このようなダイナミックな環境変化を考慮することができるきめ細かいトラフィック制御の仕組みが求められることとなる。

本稿では、従来の Δ NS に対する機能拡張として、処理ノード固有の状況としてのダイナミックな環境変化を考慮した上でデータ転送制御を行うことを可能とする低トラフ

ック型分散処理システムの提案を行う。

2. デルタエンベデッドネットワークシステム

2.1 デルタエンベデッドネットワークシステムの概要

デルタエンベデッドネットワークシステム（ Δ NS）は、膨大な数のモノ、センサー、ヒトがつながる大規模情報流通時代の到来に備え、劇的なトラフィック急増やそれに起因する多くの問題を解決できる新しいネットワークシステムアーキテクチャを目指している。同技術の全体的な構成については[2]に示されているため、本稿では主にその機能拡張部分について説明する。ただし、その説明に必要な基本部分について、以下に再掲する。

Δ NS における基本的な考えは、システムが膨大な情報に対応できるようにするために、データの処理ノードが処理の対象とするデータを適切かつ効率的に取捨選択する仕組みを導入できるようにすることにある。その際、取捨選択における判断基準として、対象物（対象データ）における“変化”や“違い”に着目することに特徴がある。本稿では、このような変化や違いを総称して、“デルタ（または、 Δ ）”と呼ぶ。

デルタは、数・テキスト・画像・音・時間・空間といった多様な種類の情報における変化や違いを意味する。また、 Δ NS は、ネットワーク中の処理ノード群にデルタを取り扱うための種々の基本機能が埋め込まれることを特徴とする分散処理システムの総称である。各ノードが分散的にデルタの有/無を複合的に判断しながら、ネットワーク全体で、あるアプリケーションとしての機能を達成する。

ここで、 Δ NS は、従来から多くの研究がなされてきた“差分の”情報だけを送ることによりデータ転送量を抑制するような技術（例えば、音声データ転送における ADPCM や画像データ転送における MPEG といった技術）とは根本的に違うものであることに注意する必要がある。また、データに内在する時間的な、あるいは、空間的な相関を利用してデータ量を削減するような技術とも異なる。

これらの技術とは異なり、 Δ NS が転送するデータ量を削減するために用いる基本的な考えは、アプリケーションが持つ（知る）高いレベルのセマンティクスを最大限に活用して、アプリケーション処理における結果の生成にとって不要なデータを検知・判断した上でそれらが転送されないように抑制することにある。また、アプリケーション内はもちろん、アプリケーション間に跨って、できる限り多くのケースでデータ転送の共有化を行うことによって、さらなるトラフィック量の削減を実現しようとする。 Δ NS では、そのような転送に関わる判断・抑制・共有を行うための仕組みが各分散処理ノードに用意される。後述するように、デルタノード（ Δ -node）とデルタデシジョンダイア

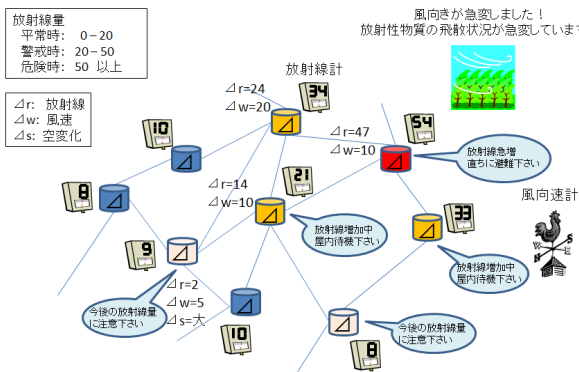


図1 環境モニタリングにおける Δ NSの活用例

グラム (Δ DD) は、それを実現するための主要な基本概念となる。

Δ NS は汎用的なシステムであるため、決して特定の応用に特化してのみ利用可能なものではない。ただし、典型的な適用シーンは存在する。例えば、図1のような災害・事故関連の応用に向けた適用シーンを想定した場合、センシングデータにおける平時とは異なる“変化・違い”を認識して、それに合わせた処理を行うようなシステムが必要となる。 Δ NS は、このような、アプリケーションが固有に認識する変化・違いを識別して、より少ないトラフィックで処理を行うようなシステムの開発を可能とする。

2.2 Δ NSの基本構成

Δ NS は、汎用 OS 上にアプリケーションを構築するためのデルタに関する種々の機能（ライブラリモジュール）を予め搭載した開発支援基盤である。ユーザは、デルタに関して、ユーザ固有の処理対象（データ）、処理方法、 Δ 有/無判断の基準条件、条件成立時のデータ転送手順を指定することにより、“複合的データに対する変化や違い”に着目したアプリケーションを簡便に開発できるようになる。 Δ NS が提供する中心的な機能は以下の2つである。

- (1) デルタの定義・表現： デルタを柔軟かつ論理的に定義・表現するための方法
- (2) デルタの処理： デルタを効率的に処理するための方法

Δ NS では、分散配置された各処理ノードに Δ DD マネージャ（複合的データ集合に対する変化・違いの表現手段である Δ DD を取り扱うモジュール：詳細は後述）が配備される（図2）。 Δ DD マネージャによって生成された Δ DD は各アプリケーションからのデータ出力が転送されるべき条件を満たすかどうかを表現・保持しており、その値が真となるときのみデータ転送モジュールから出力が許可され、

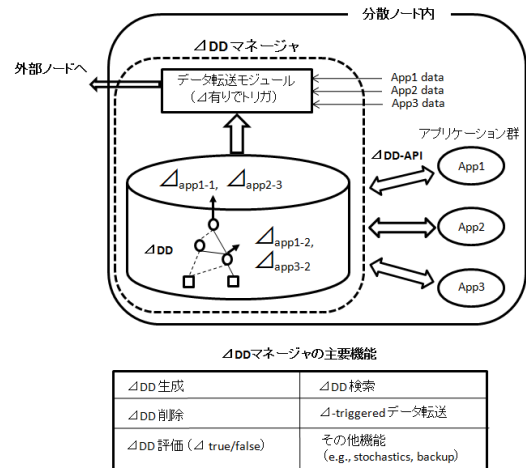


図2 分散処理ノードの構成

それ以外のケースでは出力が抑制される。

Δ DD マネージャは、 Δ DD-API を介して各アプリケーションから呼び出され、 Δ DD に関する各種機能を提供する。主たる機能は、 Δ DD の生成・更新、評価、消去、 Δ 駆動データ転送などの基本的機能と、各種管理を行うための追加的機能である。 Δ DD-API を呼ぶためのモジュールはスタブモジュールとしてライブラリ提供され、ユーザは各アプリケーション内に同モジュールをインクルードすることにより、 Δ DD マネージャの各機能呼び出す。

3. デルタの定義・表現

3.1 デルタの定義・表現

個々のアプリケーションが持つセマンティクスとして“変化”や“違い”を定義・表現するために、以下の2つ表現手法を用いる。

- (a) デルタノード (Δ ノード)
- (b) デルタデシジョンダイアグラム (Δ DD)

デルタノード（あるいは、 Δ ノード）は、“一つの”着目点に関する変化や違いを表現するための要素的表現手法である。この要素を複数結合させることにより、後述するデルタデシジョンダイアグラム (Δ DD) が構成される。

Δ ノードを図3に示す。 Δ ノードは、変化や違いを判断する対象となる2つのデータへのポインタ、比較プロセスへのポインタ、および、判断基準条件へのポインタよりなる（これら4つをあわせて“四つ組、quartet”と呼ぶ）。そして、着目点に関するデルタの有/無を決定する。

四つ組は、間接表現を用いることにより、データ対象の種類を限定しないようにしている。すなわち、データの種類に関わらず（例えば、観測された数値データの場合であっても、撮影された画像データの場合であっても同様に）、

△の有無 = 比較プロセス(データ1, データ2, 基準)

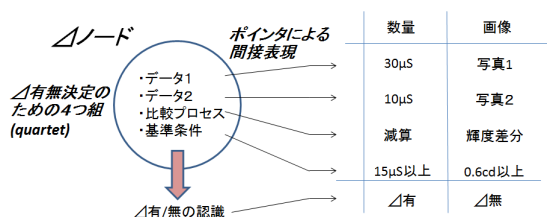


図3 デルタノード (△ノード)

データ, 比較のプロセス, 比較の結果に対する判断基準条件のいずれもが間接的に与えられるようにして, いかなる種類の対象でも本フレームワークに適合させることができるようになっている. 図3では, 数量データと画像データの2種データの例を示しているが, いずれにおいても△ノードがデルタの“有/無”を判断して論理値 (true/false) の評価結果を生成するという意味では同様の振る舞いをする.

3.2 デルタデシジョンダイアグラム (△DD)

△ノードは, 変化や違いを柔軟に表現することを目的に導入されたものだが, あくまで“一つの”着目点に関するものである. 他方, 様々な種類のデータを複数集めた“複合的な”着目点のセットに対して変化や違いを定義・表現することができればより便利である.

図4は, そのような複合的データに対する変化・違いの利用例を示したものである. 図1の適用シーンにおける警報システムにおいて, 警報発令の条件を△ノードを組み合わせ表現する場合を例示している.

△NS では, このような複合的なデータに対する△を定義・表現できるようにするために, “デルタデシジョンダイアグラム (delta decision diagram, 以下, △DD)” というダイアグラム表現を使用する (図5). すなわち, △ノードを複数結合させることにより, 複数の異種データをまとめた集合データに対するデルタの有/無を判断できるようにする.

図5は, 3つのデータセットに対する論理的結合でグループ (塊) としてのデルタの有/無を表現した例を示している. 例えば, 最上部出力 (△_{g1}) は, △₁が無, △₂が有, △₃が有のとき, グループとして△_{g1}が有であることを表現している. すなわち, ダイアグラムの上から下へ向かって, 各ノードにおける△ノードの判断値 (有/無) に従って二つの枝のうち対応する経路を選択していけば, 最終的に0か1のいずれかのノードに到達する. 0ノードに到達すればグループとしてのデルタは無であり, 1ノードに到達すればグループとしてのデルタが有であると判断される.

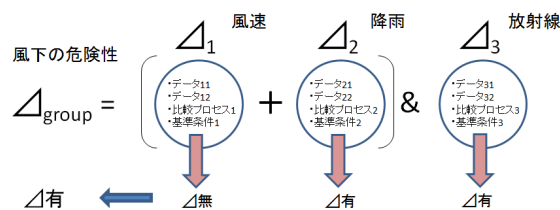


図4 デルタノード (△ノード) の論理的結合

4. デルタの処理と評価

4.1 デルタの処理方法

△ノードと△DDは, 異種の複合的データに対する変化や違いを定義・表現し, その評価を通してデルタの有/無を判断するものであるが, そこでの処理は, △ノード単独の処理と△DDの処理に分けられる.

まず, △ノード単独の処理としての△の有/無は, 個々のアプリケーション開発者が用意する比較プロセスを用いて決定される. 2つのデータに対する比較プロセスと比較基準とを用いて△の有/無があるかどうかを決定する.

一方, △DDの処理においては, 個々の△ノードを集めた複合論理に対応したダイアグラムを構成し, さらに, △ノードそれぞれの評価結果を算出した後に決定される複合的データに対する△の有/無を評価結果として決定する. △DDの生成・更新は, シャノン展開 (Shannon expansion, 図6 (a)) をベースとしている. すなわち, 任意の論理は, ある一つの変数に関してtrue/falseそれぞれの場合の部分論理の結合として表現することが可能であることを利用する. また, 任意の論理は, 2つの論理間の演算を繰り返すことにより構築されるが, そのような演算はシャノン展開として分解される部分論理間の演算を再帰的に繰り返し適用することで構築することができるという性質を利用する (図

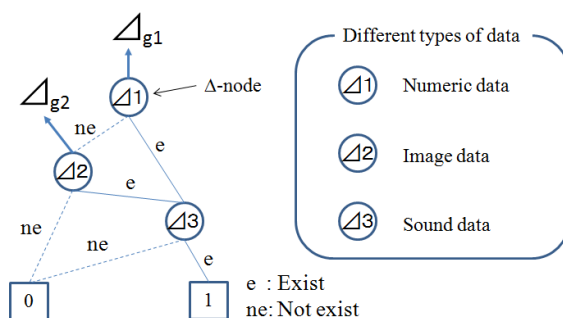


図5 デルタデシジョンダイアグラム (△DD)

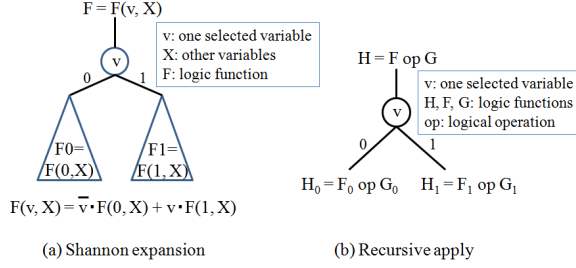


図6 シャノン展開と再帰的演算

6 (b)). 詳細に関しては[2]を参照されたい.

4.2 デルタの評価方法

いったん ΔDD が生成された後は、それを構成する各 Δ ノードの4つ組におけるその時々々のデータ1, データ2に応じて ΔDD の評価が行われる. まず, 評価時点において, 各 Δ ノードにおけるデータが比較プロセスを用いて比較され, Δ の有/無が判断される. 次に, ΔDD 中の複数の出力それぞれに関して, 出力から最下部の 0/1 ノードの方向に向かってパスがスキャンされる. パスのスキャンでは, 各 Δ ノードの二枝のうち上記の Δ ノードの判断結果に応じた枝が選択され, 0/1 いずれのノードに到達するかを見ることにより複合的データに対する Δ の有/無が判断される.

5. ダイナミックな環境変化を考慮したデータ転送制御

5.1 階層化 ΔDD (ユーザ ΔDD と特権 ΔDD)

第1章で述べたような多数かつ多様な参加者の都合を考慮しながら, ダイナミックな環境変化をも考慮することが可能なトラフィック制御を可能とするために, 本稿では, 階層化 ΔDD を用いた ΔNS を導入する. 本システムでは, ΔNS が元来活用している ΔDD の特徴を生かし, 通常の ΔDD (ユーザ ΔDD) によって実現される個々のアプリケーションが持つトラフィック制御の機構の上位的な位置づけの部分に, さらにもう一つの ΔDD (特権 ΔDD) を付加することによって, 処理ノード固有に時々刻々と変わる状況を考慮した上で複数のアプリケーション間でのデータ転送の優先度付けを制御できるようにする.

階層化 ΔDD におけるユーザ ΔDD と特権 ΔDD の関係を図7に示す. ユーザ ΔDD は, 従来 ΔNS で用いられていた ΔDD と基本的に変わりない. ただし, その出力がデータ転送条件の判断にそのまま使われるわけではなく, 条件判断の一部 (第1次条件判断) となる点が従来の ΔDD の使われ方と異なる.

一方, 特権 ΔDD は, 時々刻々と変わる処理ノード固有の状況を反映したデータを入力として受け取り, そのノー

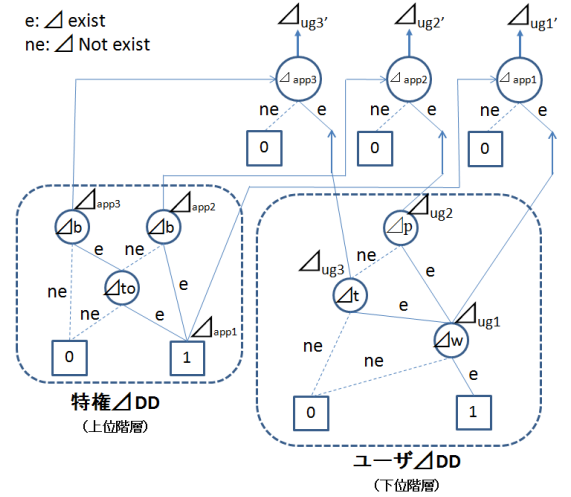


図7 階層化 ΔDD (ユーザ ΔDD と特権 ΔDD)

ドにとって最適なトラフィック処理が行えるようにアプリケーション間でのデータ転送に関する優先度付けを行うべく条件判断 (第2次条件判断) を行う. 特権 ΔDD による第2次条件判断は, ユーザ ΔDD による第1次条件判断よりもより高位に位置付けられ, より強制力の高い制御性を持つこととなる.

5.2 階層化 ΔDD によるトラフィックの制御例

階層化 ΔDD を用いたトラフィックの制御例を図7, および, 表1, 表2を用いて説明する. 本例は, 広域分散配置された環境モニタリング用のセンサーノード群における使用例を表している. 本例において, 各センサーノードでは, 降水量, 風速, 温度といった環境センシングデータが計測され必要に応じて送出される. その際, 各センサーノード固有の条件としてバッテリー残存量とバッテリー消費に影響を与える外界温度とが考慮された上で, データ転送を行うかどうかの最終決定がなされる.

ユーザ ΔDD では, アプリケーションの目的にあわせた気象監視条件を満足したデータだけが厳選される (Δp , Δt , Δw). 一方特権 ΔDD では, センサーノードの固有の条件であるバッテリー残存量と外界温度に関する規定条件を満足させるデータだけが厳選される (Δb , Δto). すなわち, これら二つの ΔDD によって, アプリケーション側の都合とインフラ側の都合の双方がデータ転送において考慮される. ただし, インフラ側の都合によりデータ転送が行われないように決定される場合には, そのことをアプリケーション側からは判断できない状況が生じる. したがって, インフラ側の都合によってデータ転送がなされない状態に入った際に, システムがそのことをアプリケーションに対して通知することで, アプリケーション側では適切な対処を行うよう振舞うことが可能となる.

表 1 ユーザ $\angle$ DD の転送条件

項目	転送条件	$\angle$ ノード
降水量	> 20 mm/hour	$\angle$ p
気温	< Normal temperature - 20° C	$\angle$ t
風速	> 15 m/sec	$\angle$ w

表 2 特権 $\angle$ DD の転送条件

項目	転送条件	$\angle$ ノード
バッテリー残量	> 30%	$\angle$ b
外界温度	> 0° C	$\angle$ to

図 7 における階層化 $\angle$ DD では、ユーザ $\angle$ DD が出力として真値を出す場合であっても、実際にデータの転送が行われるか否かは、特権 $\angle$ DD の出力に依存する。バッテリー残存電力が十分に有り、外界温度も通常温度であれば、そのままデータ転送が行われる。しかし、バッテリー残存電力が十分に無いか、あるいは、外界温度が著しく低いなどの 2 つのノード固有条件に関する規定条件を満足させられない状況においては、ユーザ $\angle$ DD の出力がたとえ真値であってもデータ転送は行われないこととなる。

5.3 特権 $\angle$ DD の構成形態

特権 $\angle$ DD の構成形態は、インフラ側とアプリケーション側の取り決めによって、双方にとってより価値の高い最適実現形態を模索しながら決定されることが望ましいが、階層化 $\angle$ DD に基づく $\angle$ NS はそのような最適化に対する高い自由度を有している。

特権 $\angle$ DD に与えられる論理条件は、分散配置される処理ノードに跨って共通であってもよいし、処理ノードごとに異なるものを与えても構わない。後者の構成は最も自由度が高いが、前者の構成であってもある程度の自由度が存在する。すなわち、 $\angle$ DD の構成は処理ノードで共通であっても、 $\angle$ ノードの四つ組における判断基準条件だけを異なるものにすることができるからである。 $\angle$ DD の構成を変えることによってノード固有条件を考慮することもできるし（大きい自由度）、あるいは、四つ組における判断基準条件の違いだけでノード固有条件を考慮することも可能である（小さい自由度）。いずれかの方法を用いて、処理ノードごとに異なるシステム構成や設置条件等を反映するといったことが可能となる。

6. 実装と評価

6.1 実装内容

階層化 $\angle$ DD を用いた $\angle$ NS の一例を構成するため、また、その効果を確認するために、バッテリー駆動される 2 つの処理ノードにおける温度、風速、降水量のセンシングデータ

の無線送信をシミュレーション評価した。処理ノード上に $\angle$ NS 基盤を配備し、その上にセンシングデータを利用するアプリケーションを搭載して動作させた。また、バッテリー電力に関しては、処理ノードの無線伝送量に応じた電力の消費を模擬する仮想デバイスシミュレータを構築し、その残存電力をノード固有データとして取得し、特権 $\angle$ DD への入力とした。

2 つの処理ノードは、機能的には同一であるが、バッテリーの容量が異なっている。第 1 の処理ノード（処理ノード 1）はバッテリー容量が大きい構成（十分な電力供給がなされる環境）となっているのに対し、第 2 の処理ノード（処理ノード 2）はバッテリー容量が小さい構成（電力供給が制限されている環境）となっている。

表 3 処理ノードの無線伝送部における電力仕様

	処理ノード 1	処理ノード 2
無線電力バジェット	4000mAH	800mAH
無線送信電力（active）	295mA	
無線送信電力（standby）	0.5mA	

各処理ノードには、3 つのアプリケーションが搭載されているが、それぞれのアプリケーションの重要度に応じてノード固有の状況（バッテリー残存電力）を考慮した上でデータ送信を行うかどうかが決定的に優先度付けがなされている。その重要度に応じて 3 つのクラス（高い方から A, B, C）に分類されている。

無線伝送用に予め配分されたバッテリー電力バジェットに対して、残存電力が 50% 以上ある場合には全てのクラスのアプリケーションに対してデータ転送が許可される。一方、残存電力がバジェットの 50% を切るとクラス C のアプリケーションはデータ転送が許可されなくなる。さらに、残存電力がバジェットの 30% を切るとクラス B のアプリケーションもデータ転送が許可されなくなる。クラス A のアプリケーションに関しては、常にデータ転送が許可される。このような重要度の違いが特権 $\angle$ DD として表現されている。

なお、3 つのアプリケーションのユーザ $\angle$ DD は、前章 5.2 におけるものと同一のものが共通に用いられているが、これは、純粋に特権 $\angle$ DD の違いに依存した結果の差を見るためである。

6.2 評価結果

階層化 $\angle$ DD を使わない従来の $\angle$ NS（ケース 1）と階層化 $\angle$ DD を使う新しい $\angle$ NS（ケース 2）の 2 つのケースでの評価結果を比較する（図 8、および、表 4）。

ケース 1 では、特権 $\angle$ DD によるデータ転送の制御が行われないため、分散処理ノードの固有状況にかかわらず（処

理ノード1, 処理ノード2ともに), データ転送が3つのアプリケーション全てに対して許可される。したがって, それに応じて無線伝送電力も消費されるために, 短い時間のうちに規定の電力バジェットが消費されてしまうことがわかる。これによって, 重要度の高いアプリケーションであっても低いアプリケーションであっても一律にある時点で必要なデータを転送できなくなり十分な機能提供ができなくなってしまう。

一方, ケース2では, 特権 Δ DDによるデータ転送の制御が行われるため, 処理ノードの残存電力にあわせてデータ転送の制御(優先度付け)がなされる。すなわち, 電力供給に制限のある処理ノード2では, 重要度の低いアプリケーションにおいてはデータ転送が抑制されるが, その分の電力消費も抑制されるため, 電力の消費がケース1よりも遅くなっていることがわかる。これによって, より重要度の高いアプリケーション(no.1)においては, 従来の Δ NSを用いた場合よりも長い時間(従来の1.7倍)にわたって機能提供できていることがわかる。

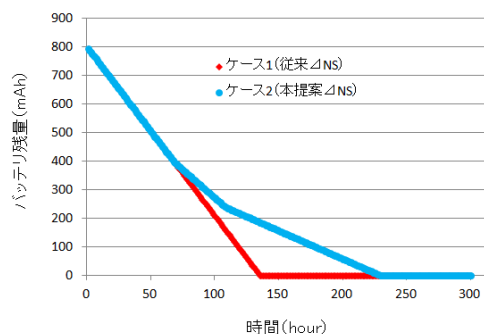


図8 処理ノード2における残存電力の推移

3) K. Nichols, S. Blake, F. Baker, D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” IETF RFC2474, Dec. 1998.

表4 アプリケーション機能時間 (単位: hour)

	ケース1 (従来 Δ NS)		ケース2 (本提案 Δ NS)	
	処理ノード1	処理ノード2	処理ノード1	処理ノード2
アプリ no. 1 (class A)	680	136	1150	230
アプリ no. 2 (class B)	680	136	540	108
アプリ no. 3 (class C)	680	136	340	68

7. さいごに

処理ノード固有のダイナミックな環境変化を考慮したデータ転送制御を可能とする低トラフィック型分散処理システムとして, 階層化 Δ DDを用いた新しいデルタエンベデッドネットワークシステムを提案した。

提案システムの効果を確認するために行った実験では, バッテリ駆動の処理ノードにおける残存電力が時間とともに変化する仮想ノード環境において, 提案手法を用いた処理ノードが適切にトラフィック転送制御を行うことによって, 重要度の高いアプリケーションにおいて従来よりも長時間の動作を可能とすることをシミュレーションにより機能確認した。

参考文献

- 1) Clusters of European Research Projects on IoT (CERP-IoT), "Vision and challenges for realising the Internet of Things," March, 2010.
- 2) 佐々木靖彦, 寺西裕一, “デルタエンベデッドネットワークシステム ～ 大規模情報流通時代に向けた低トラフィック型分散処理システム ～”, 電子情報通信学会 技術研究報告 (IEICE Technical Report), ネットワークシステム, Vol. 111, No. 408, pp5-10, 2012.