ノード間通信インターフェースに依存しない 汎用フラッディング機構の実現

Versatile Flooding Database System using Adjacent Node Virtual Communication Interface

今泉英明 [1,2]
Hideaki Imaizumi
hiddy@sfc.wide.ad.jp

杉浦一徳 [2,1]
Kazunori Sugiura
uhyo@sfc.wide.ad.jp

中村 修 [3]
Osamu Nakamura
osamu@sfc.wide.ad.jp

[1] 慶應義塾大学大学院 政策・メディア研究科
Graduate School of Media and Governance, Keio University

[2] 通信総合研究所
Communications Research Lab.

[3] 慶應義塾大学 環境情報学部
Faculty of Environmental Infomation, Keio University

概 要

本論文では，様々なアプリケーションが汎用に利用できるフラッディング機構を提案する．また，隣接ノード間の通信方式を仮想的な通信インターフェースに抽象化する手法を提案，本機構に適用することによって，特定のネットワーク層プロトコルやデータリンク技術に全く依存しないフラッディング機構を実現した．
他のフラッディング機能を提供する機構との比較を行った結果，本機構は他の比較対象と比べ有用であることが分かった．また，仮想的な隣接ノード間通信インターフェースのプロトコル・オーバヘッドの測定評価を行った結果，フラッディング機能を提供する上で障害にはならないことが分かった．

1 はじめに

インターネット格闘ゲームや実時間動画転送中継等のリアルタイムアプリケーションの出現は，インターネットの提供するベストエフォート型のサービスで全ての要求を満たすことが困難であることを明らかにした．

このような問題を解決するために，MPLS(Multi Protocol Label Switching) 技術 [1] や DiffServ (Differentiated Service) 技術 [2] 等の通信品質の保証を行う新しい技術が登場した．これらの技術は，その技術が適用されたルータ群から構成されるネットワーク内において，通信品質を保証する．

このような技術において，ネットワークの状態を検知・把握することは重要である．特にネットワークのトポロジと資源の利用状況が把握できれば，特定の通信品質要件を満たすパスを算出できる．また，これらの技術が適用されたネットワークと適用されていないネットワークの境界ルータを検出することで，新たに効率的なネットワーク管理手法を実現できる．

ネットワークの状態を検知・把握する代表的な手法として，リンクステート型経路制御機構が用いるフラッディング (氾濫) 法が挙げられる [3]．フラッディング法は，ネットワークを構成する各ノードが持つ断片的な情報をネットワーク全体に流し，各ノードが全てのノードの情報をデータベースとして共有することによって，ネットワークの状態とその変化を検知・把握する手法である．リンクステート型経路制御機構において各ルータは，内部に持つフラッディング機能を用いてネットワーク全体のトポロジと経路決定に必要となる情報を取得し，その情報から経路を決定する．IP ネットワークにおけるリンクステート型経路制御プロトコルとしては OSPF(Open Shortest Path First)[4] や IS-IS(Intermediate System-Intermediate System)[6, 7]，ATM ネットワークでは PNNI(Private Network-Network Interface)[9]，IPv6 では OSPF version 3[5] や IS-IS[8] が挙げられる．

MPLS 技術では，特定の通信品質を満たすパスを算出するために，様々なネットワークの状態情報をリンクステート型経路制御機構内のフラッディング機能を用いて行う [11]．MPLS 技術や DiffServ 技術を包含し，より統合的なトラフィック制御機構を議論・開発している IETF TEWG(Traffic Engineering Working Group) においても，ネットワークの状態情報を取得するために，リンクステート型経路制御機構内のフラッディング機能の利用を前提としている [12]．

しかしフラッディング機能は，特定の経路制御を目的としたものであり，他の利用は考えられていなかった．そのため PNNI を除く全てのリンクステート型経路制御プロトコルでは，他のアプリケーションがフラッディング機能を利用するための API を十分に定義していない．しかし，インターネット技術の変化によって，フラッディング法は経路制御以外の目的でも用いられつつある．

特に MPLS 技術では，トラフィック制御，必要ラベル数の削減，および境界ルータの検知にフラッディング

法が用いられている．MPLS 技術によるトラフィック制御機構を Linux 上で実装した例では，経路制御ソフトウェア GateD(Gateway Daemon)[13] を改変し，OSPF 内のフラッディング機能の API を独自に構築し，同機能を利用した [14]．同様に筆者らも，GateD を改変し，OSPF 内のリンクステート・データベースを参照してトポロジ情報を把握することによって，MPLS ネットワークの必要ラベル数を大幅に削減する手法を実現した [15, 16]．また，MPLS ネットワークの境界ルータを，フラッディング法を用いて動的に検知することで，より効率的に MPLS ネットワークを管理できることが分かった [16]．この境界ルータの検知は，DiffServ ネットワークの管理にも有益な手法である．

フラッディング法を利用するためには既存の経路制御ソフトウェアを改変する必要があり，容易に利用できない．また，ネットワークの種類によって異なるリンクステート型経路制御プロトコルを利用する必要があるため，プロトコル毎にその詳細を理解し，適切な経路制御ソフトウェアを改変する必要がある．さらに，フラッディング機能を利用するためには，リンクステート型経路制御プロトコルによって経路制御が行われている必要がある．したがって経路制御運用方針へ制限を与えることになる．

以上を踏まえ，本研究ではフラッディング機能をアプリケーションが容易に利用可能な機構を提案, 実現した．また，本研究では，IP/IPv6 等のネットワーク層プロトコルや ATM/Ethernet 等のデータリンク技術といった隣接ノード間の通信インターフェースを，Ethernet に似た仮想的なネットワーク・インターフェースに抽象化する手法を提案，本機構に適用した．これにより本機構は，特定のネットワーク層プロトコルやデータリンク技術に全くしない．

2 フラッディング法

フラッディング法は，ノードの持つ断片的な情報をネットワークを構成する各ノードが広告し，全く同じ情報の集合を保持する手法である．

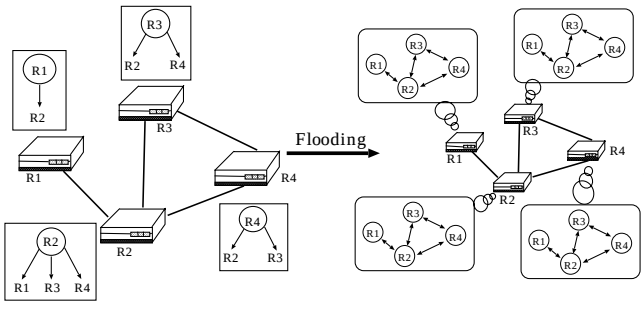

図 1: フラッディング法

リンクステート型経路制御機構によって呼び名は異なるが，本論文では一様に，この交換される単位である各情報を**ノード情報広告 (Node Information Advertisement, 以後 NIA)** と呼び，NIA の集合を**ノード情報広告・データベース (以後 NIADB)**，フラッディングが行われるネットワークの範囲を**フラッディング・ドメイン (以後ドメイン)**，フラッディングを実現するプロトコルを**フラッディング・プロトコル**と呼ぶ．

フラッディング・プロトコルは，隣接ノードと通信を行い，それぞれの持っていない NIA を交換することで，ドメイン上の NIADB を構築する．

本論文では，ドメイン上の各ノードが必ず同一の NIADB を保持することを保証する，信頼性のあるフラッディング法のみを対象とする．

各ノードが，それぞれの持つリンクステート情報を NIA としてフラッディングした場合，NIADB はドメインに属するノード全てのリンクステート情報を含む．したがって各ノードは，この NIADB からドメイン全体のトポロジを把握できる．その様子を図 1 に示す．

リンクステート型経路制御機構は，リンクステート情報に加えて経路を決定するために必要な情報を加えて NIA として広告を行う．各ノード上のリンクステート型経路制御機構は，全く等価な NIADB から経路を計算するため，ネットワーク全体を通して正しく経路制御される．

2.1 用途

ネットワークにはホーム・ネットワークのような小さいものから，WAN(Wide Area Network) のように大きいものまで，多様な規模が考えられる．フラッディング法は，その規模によって様々な役割・用途がある．

本節では，ホーム・ネットワークと MAN(Metropolitan Area Network)/WAN の二つのネットワークにおいてフラッディング法を用いる具体的な例をあげ，フラッディング法の用途を示す．

2.1.1 ホーム・ネットワーク

テレビやビデオといった身の回りにある様々な機器が小型化・機能化されて，用途によって様々な種類のインターフェースを持つようになった．したがってホーム・ネットワークでは，これらのインターフェースが相互に接続し，ネットワークを構成することが予想される．

これらの機器の中には，他の機器と協調しなければサービスを提供できないものがある．例えばビデオや DVD プレーヤは，モニタ (テレビ) やスピーカ (ステレオ，ヘッドホン) という媒体を必要とし，それらに対して動画・音声情報を送信する．また，風呂沸し機などは，お風呂が湧いたという状態変化をスピーカやモニタ等の媒体を通して通知する．したがってこれらの機器は，ネットワーク上に点在する媒体となる機器の特性を何らかの方法で知る必要がある．

ネットワークを構成するノードの特性といった属性情報を検知する方法として，フラッディング法を利用する方法 [17] と，ノードの属性情報を管理するディレクトリ・サーバを利用する方法 [18, 19] の二つがある．

これらの様子を図 2 に示す．フラッディング法はネッ

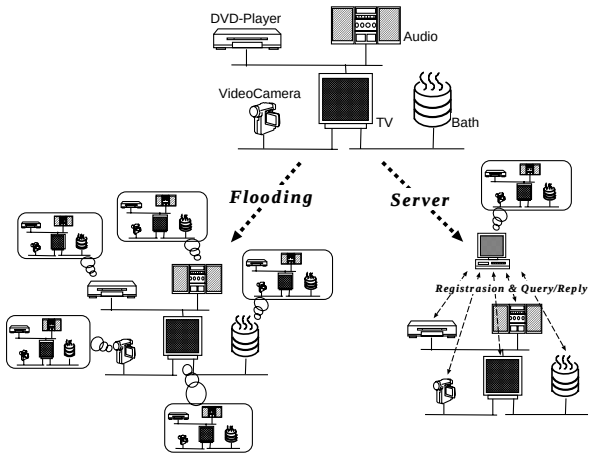

図 2: ホーム・ネットワークの機器情報管理イメージ

トワーク上の各ノードが，他のノードの属性情報全てを保持するため，ネットワークに応じたメモリ容量が要求される．しかし，サーバという特殊な機器を利用しないため耐故障性に優れている．一方サーバを用いる方法では各ノードに対して必要以上のメモリ容量を要求しないが，サーバを管理する必要があり，またサーバが故障した場合機器同士の協調作業に大きな支障が起きる．

以上に示したようにフラッディング法は，これからのホーム・ネットワークを支援する技術を構成する要素である．

2.1.2 MAN/WAN

近年 MAN や WAN を支えるバックボーン・ネットワークにおいて，通信品質の保証を行う様々な技術が登場している．これらの通信品質保証技術は一般的に，経路制御を用いる方法と，パケットの通過するパス上のルータ群の出力キューに特殊なキューイング技術を適用する方法が組み合わされて実現される．

経路制御によって通信品質の保証を行う場合，各ノードのリンクステート情報として転送遅延や最大帯域，予約帯域，パケット損失率などの様々なネットワーク状態情報を，ルータが把握する必要がある．これらの情報は，フラッディング法を用いて広告されるのが一般的である．

一方キューイング技術を用いて通信品質の保証を行う場合，通過する各ルータの入力/出力キューでは統一されたポリシによってトラッフィックを制御しなければならない．このようなポリシは，特定のマネージャが管理し，各ルータに反映させる [20, 21]．したがって，このマネージャは新たなルータの追加/削除等のネットワークの変化を検知し，ポリシを伝達する必要がある．また同時に，通信品質の保証を行う経路制御を運用している場合，ネットワークは非常に複雑に動作するため，経路制御に利用されているリンクステート情報の変化を，経路制御を行わないマネージャも監視することでより細かくネットワークの状態を管理できる．これらは，フラッディング法を用いることによって達成できる．その様子を図 3 に示す．

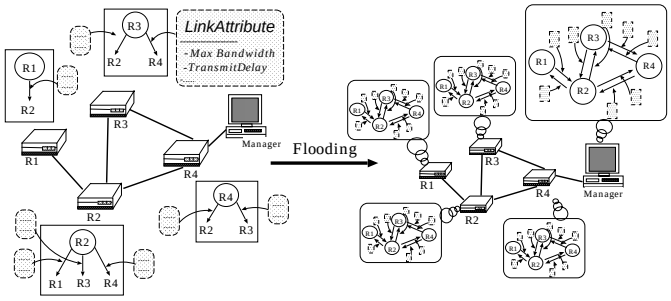

図 3: 通信品質保証技術が適用されたネットワークの管理

以上に示したように，通信品質の保証を行う新たなネットワーク技術において，フラッディング法は不可欠である．

2.2 ノード情報広告 (NIA) に必要な情報

NIA は，必ず以下の四つの情報を含んでいなければならない．

1. NIA 広告ノード ID
2. NIA ID
3. NIA シーケンス番号
4. NIA 有効時間

NIA 広告ノード ID は，NIA を広告したノードを識別するための識別子である．NIA ID は，NIA が保持する情報を識別するための識別子である．NIA を広告したノードは，NIA を更新する場合 NIA シーケンス番号を増やして広告する．NIA 有効時間は，NIA が有効な情報だと判断できる残り時間である．NIA 有効時間を過ぎた NIA は無効とされ，NIADB から削除される．

2.3 フラッディング機構に必要な API

本節では，経路制御と利用目的を限定せずに，フラッディング機構を利用するために必要である最小限の API を示す．フラッディング機構を利用するアプリケーションを利用者と表現する．利用者には経路制御機構も含まれる．

フラッディング機構を利用する上で必要な操作は大きく以下の三つに分類できる．

1. NIA 参照インターフェース
2. NIA 変化通知インターフェース
3. NIA 広告/削除インターフェース

NIA 参照インターフェースは，利用者がフラッディング機構の持つ NIADB 内の NIA を参照するためのインターフェースである．

NIA 変化通知インターフェースは，フラッディング機構が利用者に対して NIADB に変化があったことを

伝えるインターフェースである．NIADB の変化通知とは，**新規 NIA 通知**，**無効 NIA 通知**の二つに分けられる．新規 NIA 通知は，新しい NIA が NIADB に追加されたことを利用者に通知し，無効 NIA 通知は，ある NIA が無効になり NIADB から消されたことを利用者に通知する．

NIA 広告/削除インターフェースは，利用者がフラッディング機構に対して，NIA をドメインに広告，または広告した NIA をドメインから削除するインターフェースである．ドメインから NIA を削除する方法が NIA 有効時間を待つ方法以外に無い場合は，このインターフェースは省略される．その他の場合は，特殊な NIA をドメインに広告することによって行われる．

3 現状の問題点と本研究のアプローチ

アプリケーションが信頼性のあるフラッディング法を利用するには，自分で新たな実装を行うかフラッディング機能を持つ既存の機構を利用するかの二通りしか解法はない．信頼性のあるフラッディング法を実現するには複雑なプロトコルが必要であり，実装を行うことは容易ではない．したがって，現状ではフラッディング法を利用したいアプリケーションは，リンクステート型経路制御機構に含まれる同機能を利用することが一般的である．

アプリケーションが既存のリンクステート型経路制御機構の持つフラッディング機能を使う場合の問題点として，1)API の不足，2) ネットワークの種類への依存，3) 経路制御運用方針への制限の 3 点を挙げることができる．

本節では，それぞれの問題点の詳細を示す．

3.1 API の不足

本質的に経路制御が目的であるため，リンクステート型経路制御機構は，フラッディング機能を利用するための API をアプリケーションに提供しない．アプリケーションが現状で改変すること無くフラッディング機構を利用可能な方法を，第 2 章で示した API の分類に当てはめ，表 1 に示す．PNNI のみが， NIA 広告を行うための API として Proxy-PAR(PNNI Argumented Routing) を提供している．これは PNNI の持つフラッディング機能を，PNNI を直接利用できない隣接ノードに対する API である．

表 1: 現状で提供されている方法

	NIA 広告	NIA 変更通知	NIA 参照
IS-IS	×	×	SNMP(標準化中 [25])
OSPFv2	×	×	SNMP[23]
OSPFv3	×	×	SNMP(標準化中 [24])
PNNI	Proxy-PAR[10]	×	SNMP,Proxy-PAR

PNNI の Proxy-PAR では，定期的なポーリングによって NIADB の変更を検知するが，基本的には，NIA 変更通知の API を持つ機構は無い．

表 1 に示した全ての機構は，他のアプリケーションに対して Simple Network Management Protocol (SNMP) [22] を通して，NIA を参照できる．しかし，特定の条件に合致する NIA の検索を行うには不十分である．

以上の示した通り，リンクステート型経路制御機構の持つフラッディング機能を全て利用するには同機構に対する改変が必要となる．アプリケーション開発者は，開発するアプリケーション以外の非本質的なことに注力しなくてはならなくなり，また， アプリケーション開発者ごとに異なる改変を助長し，それらを共有性が失われる．

したがって汎用的な機構と，様々なアプリケーションから利用できる API が必要である．

3.2 ネットワークの種類への依存

アプリケーションが，フラッディング機能を利用するためには，適用したいネットワークの種類から適切なリンクステート型経路制御プロトコルを導き，そのプロトコルが実装されたソフトウェアを利用する．

したがってフラッディング機能を利用するアプリケーションは，ネットワークの種類に応じて，様々なリンクステート型経路制御プロトコルの詳細を知り，プロトコルや API の相違点を吸収しなければならない．

また，複数のデータリンク技術によって構成されるネットワークにおいてフラッディング機能を利用する場合，プロトコルの改変が必要となる．

本研究では，データリンク/ネットワーク層技術を仮想通信インターフェースに抽象化し，このインターフェースを利用し汎用なフラッディング機構を構築する．アプリケーション開発者は本機構の API を利用することで，ネットワークの種類に依存しないアプリケーションを開発できる．また，経路制御機構も一つのアプリケーションとして応用できる．

3.3 経路制御運用方針への制限

アプリケーションがリンクステート型経路制御機構の持つフラッディング機能を利用するためには，適用したいネットワークを特定のリンクステート型経路制御プロトコルによって経路制御を行う必要がある．経路制御プロトコルとして特定のものを利用しなければならないため，ネットワークの設計や経路制御の自由度が低下する．

経路制御は行わずに，そのリンクステート型経路制御プロトコルを動作させる方法もあるが，経路計算や経路計算に必要な情報の広告など，明らかに非効率的である．

本研究では，経路制御などの特定の目的を持たない汎用なフラッディング機構を利用することで，経路制御運用方針へ影響を無くした．したがってアプリケー

ションは，経路制御とは全く無関係にフラッディング機能を利用できる．

4　汎用フラッディング機構の概要

本機構は，ネットワークを構成する各ノード上で動作し，様々なアプリケーションが，ネットワークの詳細を知ること無く，汎用にフラッディング機能を利用するための機構である．

本節では，本機構の全体の概要と構成要素を示し，それぞれの構成要素について概要を示す．

4.1　汎用フラッディング機構の概要

本機構は，大きく以下の二つの要素から構成される．

1. 隣接ノード間仮想通信インターフェース (Adjacency Node Communication Interface, 以後 ANCIF)

2. 汎用フラッディング・データベース機構 (Versatile Flooding Database System，以後 VFDBS)

これらの関係を図 4 に示す．

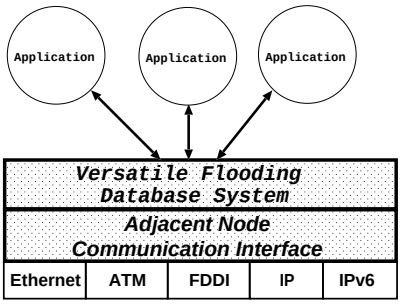

図 4: 本機構全体の構成要素

ANCIF は，隣接ノードとの通信インターフェースを仮想の通信インターフェースに抽象化することによって，上位のアプリケーションに対して単一の API を提供する．ANCIF を利用するアプリケーションは，実際の通信インターフェースの詳細を意識せずに隣接ノードと通信できる．図の例では，上位の VFDBS は ANCIF の API や詳細は知っていても，Ethernet/ATM/IP 等の実際の通信インターフェースに関する詳細には一切関知しない．

一方 VFDBS は，隣接ノードとの通信に ANCIF を用いて，フラッディングを実現する機構である．また，VFDBS は上位のアプリケーションに対し，汎用にフラッディング機能を利用できる API を提供する．

4.2　ANCIF

ANCIF は，Ethernet や ATM のようなデータリンク技術や IP/IPv6 等のネットワーク層プロトコルを通したノードとの通信インターフェースを，Ethernet に似た仮想のデータリンクに抽象化する．またその上で，隣接ノードの発見と双方向の通信を行うための API をアプリケーションに対して提供する．

実際の通信インターフェースを隣接ノード間実通信インターフェース (Practical Communication Interface, 以後 PCIF) と呼ぶ．ANCIF は，PCIF の詳細を隠蔽し，隣接ノードとの通信インターフェースを提供する．

ANCIF は VFDBS に特化したものではなく，汎用性がある．

図 5 に，ANCIF を利用するアプリケーションとの関係を示す．

図 5: ANCIF とアプリケーションの関係

図 5 左は，Ethernet，ATM，FDDI の三つの異なる PCIF をアプリケーションが利用する様子を示している．この場合アプリケーションは各 PCIF の詳細を知る必要があり，それぞれに異なる操作を行う必要がある．一方，図 5 右のように，各 PCIF を単一の ANCIF に抽象化することによって，アプリケーションは ANCIF の操作方法を知るだけで，Ethernet，ATM，FDDI の三つの PCIF を同等に操作できる．

ANCIF と ANCP

ANCIF は，Ethernet に似た仮想的な通信インターフェースであり，Ethernet 同様にマルチアクセス・ネットワークを構成する．ANCIF の下位である PCIF は，その仮想的な通信インターフェースを下位層でエミュレートするために，必要となる情報の定義や機能を持つ．

各ノードには必ずネットワークで一意なノード ID が振られ，各インターフェースにはノードで一意な IFID (Interface ID) が振られている．ANCIF のネットワーク例を図 6 に示す．

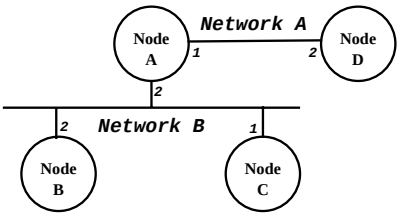

図 6: ANCIF ネットワーク例

ノード A は IFID1，2 で識別される二つの仮想インターフェースを持つ．Network A，B の各ネットワークでは，ノードの各インターフェースは，ノード ID と IFID によって識別される．したがってノード A が Network B に接続するインターフェースは，A/2 で識別される．

ANCP(Adjacent Node Communication Protocol) は，このようにノード ID と IFID の組を用いて各ノードのインターフェースを識別し，ANCIF ネットワーク上で通信を行うためのプロトコルである．ANCP の提供

する機能は，隣接ノードとの信頼性の無いデータグラム型通信である．ANCP はインターフェースを識別する方法が異なる以外，Ethernet とほぼ同じ機能を提供する．

ANCP パケットのヘッダフォーマットを図 7 に示す．ノード ID は EUI-64[26]，IFID には 32 ビットの ID を用

Octet 1	Octet 2	Octet 3	Octet 4
Destination Node ID			
Destination Interface ID			
Source Node ID			
Source Interface ID			
Length		**Protocol ID**	
Header Checksum		unused	

図 7: ANCP ヘッダフォーマット

いる．ノード ID は，コロン区切りで **ff:ff:ff:ff:ff:ff:ff:ff** のように表記する．

Destination Node ID/Interface ID は，宛先のノードのインターフェースを示す．**Source Node ID/Interface ID** は，送り元のノードのインターフェースを示す．**Length** はパケット全体の長さ，**Protocol ID** は上位のプロトコルを識別する識別子，**Header Checksum** は ANCP ヘッダのチェックサムを保持する．ANCP パケットの最大長は 32768 バイトとする．

また，ノード ID の下位 24 ビットをグループ ID とし，**ff:00:00:ff:fe:00:00:00** から **ff:00:00:ff:fe:ff:ff:ff** の範囲のノード ID は，**マルチキャスト・ノード ID** として定義される．したがって **Destination Node ID** にマルチキャスト・ノード ID を指定したパケットはマルチキャストされる．グループ ID **00:00:01** には，全てのノードが参加する．ANCIF を利用する上位アプリケーションは，ANCP のマルチキャストを利用して隣接ノードを発見する．

Protocol ID 0 は，ANCP Echo Request/Reply が定義されている．

4.3 VFDBS

VFDBS は，隣接ノード上の VFDBS と NIA の交換を行い，フラッディングを実現する機構である．隣接ノードとの通信には ANCIF を利用する．また，VFDBS は上位のアプリケーションに対し，汎用にフラッディング機能を利用できる API を提供する．

フラッディング・ドメイン

VFDBS は複数のフラッディング・ドメインを自由に構築でき，多重化できる．各フラッディング・ドメインでは**ドメイン ID** という識別子が振られ，ノード内で一意に識別される．

図 8 に，ネットワーク上で複数のフラッディング・ドメインが多重化している様子を示す．図のネットワークは，ノード A，B，C，D，E から構成されおり，各ノードは図のようにリンクを保持している．ノード A は，ドメイン 1，2，4 に所属しており，ノード A 上のアプリケーションが，ドメイン 2 に NIA を広告すると，ノード A，B，C にフラッディングされる．

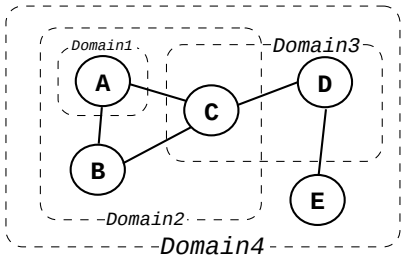

図 8: フラッディング・ドメインの様子

ノード A を中心とした VFDBS の様子を，図 9 に示す．各ノードは所属しているドメインごとに独立した NIADB を持ち，フラッディング・プロトコルを用いて NIADB を隣接ノードと同期させる．ノード A 上の VFDBS は，ドメイン 1，2，4 を管理し，隣接するノード B，C とドメイン 2，4 の NIADB を同期させる．ドメイン 1 は，隣接ノードが所属していないため，ノード A 内で独立したドメインになっている．

ノード ID と NIA

VFDBS が利用するノード ID としては，Ethernet 等の物理アドレスから自動生成可能な 64bit 長の **EUI-64** とし，ANCIF で定義されるノード ID を利用する．

NIA は，図 10 に示すフォーマットのヘッダを持ち，データを保持する．

Age は，NIA が広告されてからの経過時間 (秒) を保持し，**Max Age** と同じ値になったところで，フラッディング・ドメインから削除される．**Checksum** は NIA 全体のチェックサムを保持するフィールドであり，**Length** は NIA 全体の長さを保持している．**NIA ID** は，NIA の持つデータを識別するための 64bit 長の ID である．**Advertise Node ID** は，この NIA を広告したノードのノード ID である．**Sequence Number** は，この NIA の新しさを表す番号である．

VFP

VFP は，OSPF/PNNI を参考にして作られたフラッディングを行うプロトコルである．メッセージの種類は，**Hello**，**Summary**，**Request**，**Update**，**Acknowledge** の 5 種類である．大きな相違点としては，複数のフラッディング・ドメインを扱う機能を持つ点である．各ドメインを識別するドメイン ID は 8bit で表現される．

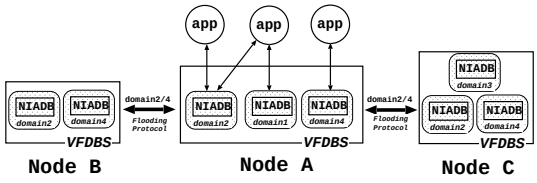

図 9: VFDBS の様子

Octet 1	Octet 2	Octet 3	Octet 4
Age		Max Age	
Checksum		Length	
NIA ID			
Advertise Node ID			
Sequence Number			

図 10: NIA ヘッダフォーマット

5 実装

本節では，ANCIF と VFDBS の実装について説明を行う．

5.1 ANCIF の実装

ANCIF は，C 言語によるライブラリとして NetBSD 1.5 上で実装を行った．PCIF としては，IP/IPv6，ATM をサポートした．IP/IPv6 はそれぞれの RAW ソケットを用い，ATM は NATM(Native ATM) ファミリのソケットを利用した．また，ライブラリの API は，ANCIF がカーネル内に実装された場合にも適用可能なものとした．

5.2 VFDBS の実装

VFDBS は，NetBSD 1.5 上でデーモン・プロセスとして C++言語を用いて実装を行った．アプリケーションに対するインターフェースとして UNIX ドメインソケットを利用した．

その様子を，図 11 に示す．

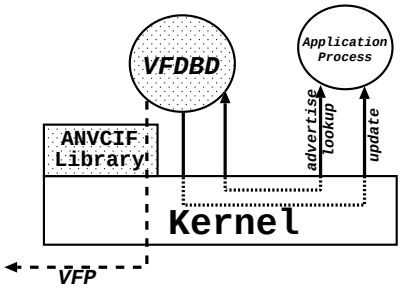

図 11: VFDBD の様子

VFDBS を実装したプロセス vfdbd は，ANCIF を利用して隣接ノードと通信を行い，フラッディング機能を実現する．また，UNIX ドメインソケットを通じて，プロセスに API を提供する．一つのプロセスに対して，広告用と参照用に UNIX ドメインソケットを一つ，変化通知用にもう一つ，計二つを利用している．

表 2 に，アプリケーションが利用する VFDBS への主要な API を示す．

表 2: VFDBS への API

関数宣言	機能
vfdb* **vfdb_open** (void);	VFDBD と接続し，そのハンドラを返す．
void **vfdb_close** (vfdb* *dbif*);	ハンドラを引数に受け，VFDBD との接続を切断し，後処理を行う．
int **vfdb_join** (vfdb* *dbif*, int *domainId*);	指定したドメインに参加する．これによって NIA 変化通知が始まる．
int **vfdb_drop** (vfdb* *dbif*, int *domainId*);	指定したドメインから離脱する．これによって NIA 変化通知が止まる．
int **vfdb_advertise** (vfdb* *dbif*, int *domainId*, NIA* *niap*);	※ NIA 広告用 API 指定した NIA を，domainID で示したドメインへ広告する．
int **vfdb_flush** (vfdb* *dbif*, int *domainId*, NIAID* *niap*, NodeID* *nodep*);	※ NIA 広告用 API 指定した NIA を，domainID で示したドメインから削除する．具体的な動作は，指定された NIA の Age を MaxAge にして広告することで *Premature Aging* を行う．この MaxAge の NIA が NIADB へ導入された時点で，新規 NIA 通知が行われる．その後 NIADB から削除されると無効 NIA 通知が行われる．
int **vfdb_lookup** (vfdb* *dbif*, int *domainId*, NIAID* *id*, NodeID* *adv*, void* *buf* size_t *buflen*);	※ NIA 参照用 API NIA ID と広告したノードのノード ID で識別される NIA を，指定したドメインの NIADB から取得し *buf* へ格納する．無い場合は-1 が返る．
int **vfdb_find** (vfdb* *dbif*, int *domainId*, NIAID* *prfix*, NIAID* *mask*, NodeID* *adv*, void* *buf* size_t *buflen*);	※ NIA 参照用 API 特定のプリフィックスを持った NIA や特定のノードから広告された NIA のヘッダリストを取得し，*buf* に NIAHdr の配列として格納する．返り値として条件にマッチした NIA の数が返される．
int **vfdb_getsock** (vfdb* *dbif*);	NIA 変化通知用ソケット・ディスクリプタを返す．
int **vfdb_recv_update** (vfdb* *dbif*, int* *type* int* *domainId* NIAHdr* *hdr*);	※ NIA 変化通知用 API ドメイン内の変化通知を受ける．*type* には新規/無効を示す値が，*domainId* には変化のあったドメインのドメイン ID が，*hdr* には変化のあった NIA のヘッダが格納される．

6 VFDBS の評価

本節では，以上に示した VFDBS に関して次の二つの評価を行う．

1. VFDBS の定性的評価
2. ANCIF のプロトコル・オーバヘッド測定評価

1 では，VFDBS の持つ性質に関して他の機構と比較し評価を行う．2 では，ANCIF のプロトコル・オーバヘッドを測定し，フラッディング機能を提供する上で問題となるかを評価する．

6.1 VFDBS の定性的評価

現状では，フラッディング機能のみをアプリケーションに提供する機構が無いため，フラッディング機能が他の目的に利用されるリンクステート型経路制御機構を対象とする．

本機構との比較評価を行う対象を以下に示す．

- **IS-IS(Intermediate System-Intermediate System)**
- **OSPFv2(Open Shortest Path First version 2)**
- **OSPFv3(Open Shortest Path First version 3)**
- **PNNI(Private Network-Network Interface)**

本機構と比較対象との比較は，以下の五つの項目について行う．

項目 1. 提供される API
フラッディング機能を必要とするアプリケーションに対して提供される API の種類を比較する．API の種類としては，第 1 節で示した分類を比較の項目とする．

項目 2. ネットワークの種類への依存性
ネットワークの種類に依存しない方が，多様な目的に適用可能であり，柔軟性が高い．

項目 3. 経路制御運用方針への影響
フラッディング機能を利用することが経路制御運用方針に影響を与える場合，特定の経路制御プロトコルが動作するネットワークに限定される．フラッディング機能を利用することが経路制御運用方針に影響を与えない方が，フラッディング機能を利用するアプリケーションにとって，可用範囲が広い．

項目 4. フラッディング・ドメイン構築の自由度
フラッディング・ドメインはフラッディングされる範囲を示す．フラッディングされるべき範囲を越えてフラッディングされたり，その範囲を制御できないことは，必要の無い情報がフラッディングされる可能性があり，非効率的である．一方目的や用途ごとにフラッディング・ドメインを自由に構築/多重できることは，フラッディングされる範囲を限定し，その情報を必要とするアプリケーションにのみ伝達でき，効率的である．

項目 5. NIA の有効時間の制御
NIA には有効時間が設定されており，有効時間を過ぎると無効になり，フラッディング・ドメインから削除される．NIA の有効時間が制御可能であれば，NIA の保持する情報にひとつの属性情報を与える意味があり，フラッディング機能が保持する有用な機能のひとつである．

本研究で提案した機構と比較対象との比較結果を，表 3 に示す．

表 3: 本研究と比較対象との比較評価結果

	項目 1			項目 2	項目 3	項目 4	項目 5
	広告	変化通知	参照				
IS-IS	×	×	○	あり	あり	可	可
OSPFv2	×	×	○	あり	あり	不可	不可
OSPFv3	×	×	○	あり	あり	不可	不可
PNNI	○	×	○	あり	あり	不可	可
本機構	○	○	○	なし	なし	可	可

項目 1 に関して比較を行うと，IS-IS/OSPFv2/OSPFv3 は NIA を参照するための API しか提供していない．PNNI は NIA を広告，参照するための API を提供している．本機構のみが全て種類の API を提供している．**項目 2** に関しては，本機構のみが特定のネットワークへの依存が無い．**項目 3** に関しても，本機構のみが経路制御運用方針への影響が無いことが分かった．**項目 4** に関しては，IS-IS と本機構のみが自由にフラッディング・ドメインを構築可能である．**項目 5** に関しては，IS-IS と PNNI，本機構が有効時間を制御可能であった．

以上に示した通り，本機構は他の比較対象と比べ，有用であることが分かった．

6.2 ANCIF のオーバヘッド測定評価

ANCIF のオーバヘッドを測定するために，PCIF を直接利用してデータ転送行った場合と，ANCIF として利用してデータ転送を行った場合の遅延の差を比較した．

IP と IPv6 を PCIF の対象として，以下の測定方法で測定した．

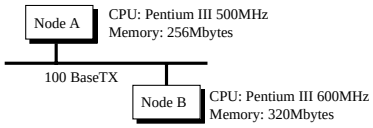

図 12: 測定ネットワーク図

図 12 に示すネットワークにおいて，ノード B からノード A に対して，PCIF を直接利用した場合と ANCIF を利用した場合の RTT(Round Trip Time) を測定した．RTT を測定するために，ANCIF に関しては *ancping* コマンドを利用し，PCIF に関してはそれぞれ *ping / ping6* コマンドを利用し ICMP によって測定した．*ancping* コマンドは PCIF の情報を受けて，マルチキャスト・ノード ID に対して ANCP Echo Request を送信し，ANCP Echo Reply を受信し表示するプログラムである．その様子を図 13 に示す．

```
doobie<netbsd> % ancping -p '<ip/ex0>' ff:00:00:ff:fe:00:00:01/0
ancPING(64=32+8+24 bytes):
    00:00:86:ff:fe:42:57:4c/16777728  -->  ff:00:00:ff:fe:00:00:01/0
32 bytes from 00:00:f4:ff:fe:60:48:d5/33554944: pkt_seq=0  time=0.869 ms
32 bytes from 00:00:f4:ff:fe:60:48:d5/33554944: pkt_seq=1  time=0.867 ms
32 bytes from 00:00:f4:ff:fe:60:48:d5/33554944: pkt_seq=2  time=0.849 ms
^C----ff:00:00:ff:fe:00:00:01/0 ancPING Statistics----
3 packets transmitted, 3 packets received, 0.0 % packet loss
round-trip min/avg/max/stddev = 0.849/0.862/0.869/0.011 ms

doobie<netbsd> % ancping -p '<ipv6/ex0>' ff:00:00:ff:fe:00:00:01/0
ancPING(64=32+8+24 bytes):
    00:00:86:ff:fe:42:57:4c/16783360  -->  ff:00:00:ff:fe:00:00:01/0
32 bytes from 00:00:f4:ff:fe:60:48:d5/33560576: pkt_seq=0  time=3.825 ms
32 bytes from 00:00:f4:ff:fe:60:48:d5/33560576: pkt_seq=1  time=0.849 ms
32 bytes from 00:00:f4:ff:fe:60:48:d5/33560576: pkt_seq=2  time=0.836 ms
^C----ff:00:00:ff:fe:00:00:01/0 ancPING Statistics----
3 packets transmitted, 3 packets received, 0.0 % packet loss
round-trip min/avg/max/stddev = 0.836/1.837/3.825/1.722 ms
```

図 13: ancping の様子

転送するパケットは，PCIF を直接利用した場合も ANCIF を利用した場合も，ペイロード長は同じ長さにした．したがって図 14 に示す通り，ANCP パケットは PCIF のヘッダ分パケット長が長くなる．

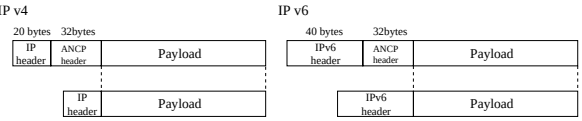

図 14: 測定パケットの様子

測定したペイロード長は，64 バイトから最大の 32736 バイトまでを 12 段階に分けたものであり，それぞれに対して 300 回の RTT 計測を行い，その平均を取った．

PCIF として IP を利用した場合の測定結果を図 15 に示す．ペイロード長が 64 バイトの時に RTT の差が最小で約 0.20 ミリ秒であり，ペイロード長が最大の 32736 バイトの時に，RTT の差が最大の約 0.78 ミリ秒であった．

PCIF として IPv6 を利用した場合の測定結果を図 15 に示す．ペイロード長が 64 バイトの時に RTT の差が最小で約 0.18 ミリ秒であり，ペイロード長が最大の 32736 バイトの時に，RTT の差が最大の約 0.85 ミリ秒であった．

以上に示した通り，最大長のパケットを送信した場合

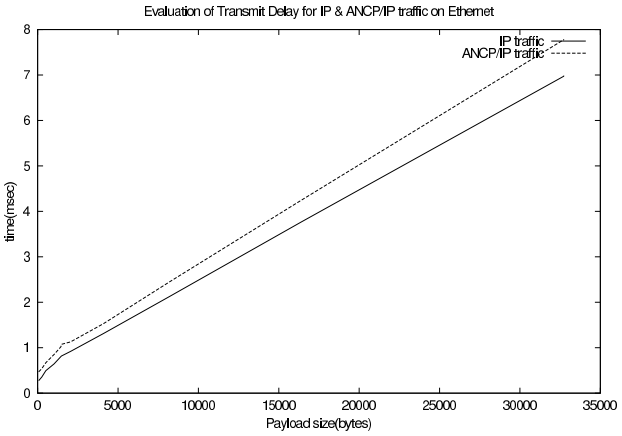

図 15: IP トラフィックと ANCP/IP トラフィックの RTT

でも ANCIF を利用した場合のオーバヘッドは，RTT が 1 ミリ秒弱程度増加するということ分かった．

OSPF では，最小リンク状態到着時間 (*minimum link state arrival time*) が 1 秒であり，新しい NIA は古い NIA が到着してから 1 秒以上経過しない限り，受理されない．また，再転送間隔 (*retransmit interval*) が 5 秒程度であり，パケットが喪失した場合は，再転送間隔時間は同期できない．以上から，先に示した ANCIF のオーバヘッドは，隣接ノードとデータベースの同期に影響を与える程の遅延ではないと考えられる．したがって，ANCIF のオーバヘッドはフラッディング機能を提供する上で障害にならない．

7 まとめ

本論文では，フラッディング法をアプリケーションが容易に利用可能な機構を提案, 実現した．また, IP/IPv6 等のネットワーク層プロトコルや ATM/Ethernet 等のデータリンク技術といった隣接ノード間の通信インターフェースを，Ethernet に似た仮想的なネットワーク・インターフェースに抽象化する手法を提案し，本機構に適用することで，特定のネットワーク技術に依存しないフラッディング機能を提供する機構を実現した．

本論文で提案した機構の持つ性質が有益であることを示すため，他の機構と比較評価を行った．その結果，本論文の機構は他の機構に比べ，有用であった．また，仮想的な隣接ノード間通信インターフェースのプロトコル・オーバヘッドの測定評価を行った．フラッディング機能を提供する上で障害にはならないことが分かった．

今後は，Perlman 等が行っている NIA の正当性を保証する機能 [27, 28] を実装し，アプリケーションが偽の NIA を広告することを防止する機能を追加する．また，上位アプリケーションが VFDBS を設定する API の設計を行う予定である．さらに，ノード間通信インターフェースに依存しないトポロジ情報広告機構等の，本機構を利用したアプリケーションを開発する．

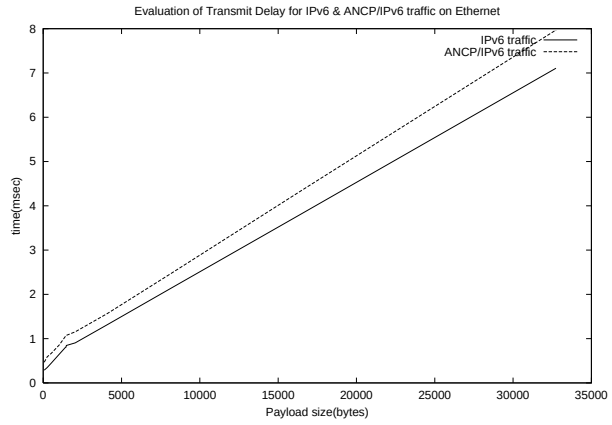

図 16: IPv6 トラフィックと ANCP/IPv6 トラフィックの RTT

参考文献

[1] E. Rosen, A. Viswanathan, R. Callon, “Multiprotocol Label Switching Architecture”, IETF RFC 3031, Jan, 2001

[2] S.Blake, D.Black, M.Carlson, E.Davies, Z.Wang, W.Weiss, “An Architecture for Differentiated Service”, IETF RFC2475, December 1998

[3] McQuillan, J. and et.al., “The New Routing Algorithm for the ARPANET”, IEEE Transactions on Communications, May, 1980

[4] John Moy, “Open Shortest Path First(OSPF) version 2”, IETF RFC2328, Apr 1998

[5] R. Coltun, D. Ferguson, John Moy, “Open Shortest Path First for IPv6”, IETF RFC2740, Dec 1999

[6] “Intermediate System to Intermediate System Intra-Domain Routeing Information Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service ”, ISO 10589

[7] R. Callon, “Use of OSI IS-IS for Routing in TCP/IP and Dual Environments”, IETF RFC1195, Dec, 1990

[8] Christian E. Hopps, “Routing IPv6 with IS-IS”, IETF Internet-draft, Apr, 2001

[9] ATM Forum, “ATM Private Network-Network Interface Specification V1.0”, af-pnni-0055.000, March, 1996

[10] ATM Forum, “ATM PNNI Augmented Routing(PAR) V1.0”, af-ra-0104.000, January, 1999

[11] D. Awduche, J. Malcolm, J. Agogbua, M. O’Dell, J. McManus, “Requirements for Traffic Engineering Over MPLS”, IETF 2702, Sep, 1999

[12] Daniel O. Awduche, Angela Chiu, Anwar Elwalid, Indra Widjaja, XiPeng Xiao, “A Framework for Internet Traffic Engineering”, IETF internet-draft, May 2001

[13] http://www.gated.org (2001 年 8 月現在)

[14] http://www.labs.fujitsu.com/free/te-on-linux/index_jp.html (2001 年 8 月現在)

[15] 永見健一, 今泉英明, 中村修, 江崎浩, “OSPF アグリゲーションを用いたラベルスイッチルータにおけるラベル数の評価”, 情報処理学会分散機構/インターネット運用技術 シンポジウム’99 論文集, Feb, 1999

[16] 今泉英明, 永見健一, 杉浦一徳, 村井純, “ラベルスイッチルータにおける OSPF リンクステートデータベースを用いたフローアグリゲーションの実現”, 電子情報通信学会 コミュニケーションクオリティ研究会 信学技報 25–30, Feb, 2000

[17] Peter Wyckoff, Stephen W. Mclaughry, Tobin J. Lehman, Daniel A. Ford, “T Spaces”, IBM Systems Journal volume 37 number 3, 454–474, Aug, 1998

[18] Sun Microsystems, Inc., “Jini Architecture Specification”, http://www.sun.com/jini/specs/, (2001 年 8 月現在)

[19] Steven Czerwinski, Ben Y. Zhao, Todd D. Hodes, Anthony D. Joseph and Randy H. Katz, “An Architecture for a Secure Service Discovery Service”, Proceedings of The Fifth Annual International Conference on Mobile Computing and Networking (MobiCom’99), Aug, 1999

[20] Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, R., and A. Sastry, ”The COPS (Common Open Policy Service) Protocol” RFC 2748, Jan, 2000

[21] K. Chan, D. Durham, S. Gai, S. Herzog, K. McCloghrie, F. Reichmeyer, J. Seligson, A. Smith, R. Yavatkar, ”COPS Usage for Policy Provisioning,”, RFC 3084, Mar, 2001

[22] M. Fedor, M. Schoffstall, J. Davin, “A Simple Network Management Protocol (SNMP)”, IETF RFC1157, May, 1990

[23] F. Baker, R. Coltun, “OSPF Version 2 Management Infomation Base”, IETF RFC1895, Nov, 1995

[24] draft-ietf-ospf-mib-update-05.txt, IETF Internet-draft

[25] draft-ietf-isis-wg-mib-04.txt, IETF Internet-draft

[26] “GUIDELINES FOR 64-BIT GLOBAL IDENTIFIER(EUI-64) REGISTRATION AUTHORITY”, IEEE, http://standards.ieee.org/regauth/oui/tutorials/EUI64.html(2001年 8 月現在)

[27] R. Perlman, “Interconnections: Bridges and Routers”, Addison-Wesley, 1992

[28] S.L. Murphy, M.R. Badger, ”Digital Signature Protection os the OSPF Routering Protocol”, Proceedings of the Symposium on Network and Distributed System Security(SNDSS’96), pp.93-102, San Diego, California, Feb, 1996