

# 特別研究報告

題目

アプリケーションレベルマルチキャストの一方式の  
提案とその性能評価

指導教官

東野 輝夫

報告者

中村 嘉隆

平成 14 年 2 月 19 日

大阪大学 基礎工学部 情報科学科

アプリケーションレベルマルチキャストの一方式の  
提案とその性能評価

中村 嘉隆

内容梗概

アプリケーションレベルマルチキャストは、エンドホスト間のユニキャスト通信を仮想的なリンクとみなした論理ネットワーク上でのマルチキャストであり、従来のマルチキャストと比べ既存の通信環境での実現が容易で、ユニキャストトランスポートプロトコルの制御機能も利用可能であるといった利点がある。

従来の研究では、ネットワーク資源の利用コストやエンド間の遅延を考慮した経路木の生成、エンドホストの非安定性への対応を課題として扱ってきた。しかし、電子会議等に利用するには、エンドホスト付近の帯域制約や受信したいメディアの偏りといった問題がある。

本研究が提案するプロトコルでは、エンド間ユニキャストから構成される論理ネットワーク上に、それぞれ遅延をメトリックとしてメディアタイプ別の経路木を構築する。また、各エンドホストが要求するメディアの偏りをプリファレンスとして数値化し、帯域制約や端末の処理能力を満足しつつアプリケーション全体のプリファレンス値の総和がなるべく大きくなるように、各エンドホストにおいてデータ転送動作を制御する。提案プロトコルにより、帯域制約やプリファレンスの問題を解決し、受信者の要求を十分満足させるような配送ができるため、電子会議のようなコミュニケーションシステムへの利用が可能となる。

また、提案プロトコルと従来のマルチキャストプロトコルの性能を比較するため、ネットワークシミュレータ ns-2 を用いてシミュレーションによる性能評価を行なう。

主な用語

マルチキャスト, アプリケーションレベルマルチキャスト, Overlay ネットワーク, ユーザプリファレンス, 帯域制約, 転送制御

## 目次

1	まえがき	3
2	対象とするアプリケーションとネットワーク	5
2.1	アプリケーション	5
2.2	Overlay ネットワーク	6
3	提案するアプリケーションレベルマルチキャストのプロトコル記述	9
3.1	チャンネルの確立	9
3.2	周期時間によるノード間の同期	10
3.3	タイプ A のメディア用経路木構築と維持	10
3.4	タイプ V のメディア用経路木構築と維持	11
3.4.1	MEDIA/Keep メッセージによるプリファレンス値の集約	12
3.4.2	MEDIA/Join メッセージによる受信要求集約と配送可能性計算	13
3.4.3	MEDIA/Accept による配送メディア変更	17
4	考察	19
5	シミュレーション実験	21
5.1	シナリオ設定	21
5.2	シミュレーション結果	21
5.3	今後の測定予定	22
6	あとがき	23
	謝辞	24
	参考文献	25

## 1 まえがき

近年の通信環境の発展に伴い、ネットワークを介した比較的小規模なグループによるグループコミュニケーションの需要が増加すると考えられる。しかし、そのような小規模コミュニケーショングループが多数確立されるようになった場合、各ユーザからのデータパケットを他のユーザに転送するコネクション処理を限られたサーバで行なうことは、サーバ側の帯域制限およびコネクション数制限等の問題より現実的でない。この問題に対し、解決法としてサーバ機能をグループのいずれかのエンドホストに移動させる方式が挙げられるが、通信の安定性の多くがサーバとなるエンドホストの処理性能や帯域に依存し、かつそのエンドホストがセッションに参加し続けなければならないなど問題点も多い。

サーバを用いない方法としては、ユニキャストコネクションをエンドホスト間で直接生成する方法もある。この場合、グループメンバーの増加に伴うエンドホストでのコネクション数の増加が問題となる。一方、ネイティブマルチキャストはこれら全ての問題を解決できるが、この方法でもルータのマルチキャストへの対応やグローバルマルチキャストアドレスの割り当て、あるいはセキュリティやスケーラビリティ等の問題があり、グループコミュニケーションにそのまま適用することは容易ではない。

アプリケーションレベルマルチキャストは、エンドホスト間のユニキャストチャネルを仮想的なリンクとみなした Overlay ネットワーク上でのマルチキャストであり、現在の通信基盤での実現が容易であること、信頼性保証、セキュリティ、フロー制御、レート制御等のユニキャストトランスポートプロトコルの制御機能が利用可能であることなど利点が多く、近年グループコミュニケーションを実現する方法として注目されている。この分野の関連研究として、文献 [1, 2, 3, 4, 5, 6, 7, 9, 10] などが挙げられる。

アプリケーションレベルマルチキャストを実際にグループコミュニケーションに利用するには、実リンク上でのパケット重複、エンドホストの転送負荷、エンドホストのセッションからの離脱による Overlay ネットワークの流動性、およびエンドホスト付近での帯域制約、受信したいメディアの偏りといった問題を考慮したプロトコルが要求される。アプリケーションレベルマルチキャストに関する従来の研究は、マルチキャスト経路木構築時に、パケット重複や転送負荷等の経路木のコストが妥当であること、ネットワークの流動性に適応するこ

とを目標にしたものがほとんどである．文献 [1] で設計されているプロトコル Narada では遅延のみをメトリックにしてソースノードからの最短経路木を構築している．文献 [4] でも遅延のみをメトリックにし，スパニングツリーで経路木を構築している．また，文献 [2] では Narada [1] を電子会議アプリケーション向けに設計しているが，メトリックが帯域である最短経路木を基本に遅延も考慮した最短経路木が構築されるのみである．これらで提案されているプロトコルでは，各メディアに対するユーザの嗜好の偏りを考慮していないので，エンド間の帯域制約等により受信者の要求するメディアを配送することができない場合があり，会議アプリケーション等に用いた場合，受信者の要求を満足させることができない可能性がある．

本研究が提案するプロトコルでは，Overlay ネットワーク上にオーディオタイプのメディア向けには共有木を，ビデオタイプのメディア向けには最短経路木をそれぞれ遅延をメトリックとして構築している．また，各エンドホストのメディアに対する嗜好の偏りをプリファレンスとして数値化し，帯域制約や各エンドホストの処理能力を満足しつつ，アプリケーション全体におけるプリファレンス値の総和がなるべく大きくなるように各エンドホストにおいてデータ転送動作を制御する．これにより，メディアを実時間交換するようなコミュニケーションシステムに利用することが可能となる．

また，今回提案したプロトコルに対してネットワークシミュレータ ns-2 [11] を用いた評価実験を行ない，制御メッセージ送信からメディア受信までにかかる時間，必要となる制御メッセージ量を測定し，性能評価を行なう．

以下，研究成果の報告を行なう．2章では提案プロトコルの対象とするアプリケーションとネットワークについて説明する．3章では本研究が提案するアプリケーションレベルマルチキャストのプロトコルの内容を説明する．4章では提案プロトコルに対する考察を述べる．5章では ns-2 を用いたシミュレーション実験の概要を述べる．6章では本論文のまとめを行なう．

## 2 対象とするアプリケーションとネットワーク

### 2.1 アプリケーション

本研究では、ビデオ/オーディオ会議やリモートレクチャ遠隔会議等、複数人のコミュニケーションシステムの特性を考慮し、各エンドホストが他のエンドホストに対し配信するメディアを、低遅延かつ低ロス率を要するメディアと、高帯域かつ中程度以下の遅延を要するメディアの2種類に大きく分類する。

低遅延かつ低ロス率を要するメディア 音声やサムネイル画像、共有ホワイトボードへの書き込みデータなど、排他的、あるいは非継続的な特性により、全体として利用する帯域が比較的小さいが、遅延やロスには敏感なメディアがこのタイプのメディアである。以下では、このタイプのメディアをタイプ  $A$  のメディアとよぶ。

高帯域かつ中程度以下の遅延を要するメディア 電子会議における各参加者の映像やマルチサイト会議の各会場の中継画像など、遅延よりは帯域を要するメディア [2] とする。以下、このタイプのメディアをタイプ  $V$  のメディアとよぶ。

エンド間の帯域やエンドホストの処理能力上、各エンドホストから発信されるタイプ  $V$  のメディアを全ホストに配信することは、ある程度の数のエンドホストが存在するもとは不可能であると考えられる。そこで、それらの制約のなかで、個々のユーザ満足度を数値化し評価するため、エンドホスト  $i$  のエンドホスト  $s$  をソースとするタイプ  $V$  のメディア  $V_s$  に対する嗜好を表すプリファレンス値  $p_s(i)$  を導入する。プリファレンス値は一種の優先度を表す。主に受信したいと考えるメディアはより高いプリファレンスを持つ、すなわちそのようなメディアに対してはより大きいプリファレンス値を与えることとする。

以下では、エンドホスト  $i$  は  $p_s(i)$  を非負整数値で指定するとし、ホスト  $i$  が受信中の各メディアに対するプリファレンス値総和をホスト  $i$  の満足度、全ホストの満足度総和がアプリケーション全体の満足度と定義する。本研究が提案するアプリケーションレベルマルチキャストプロトコル Emma ( End-user Multicast for Multi-party Applications ) はアプ

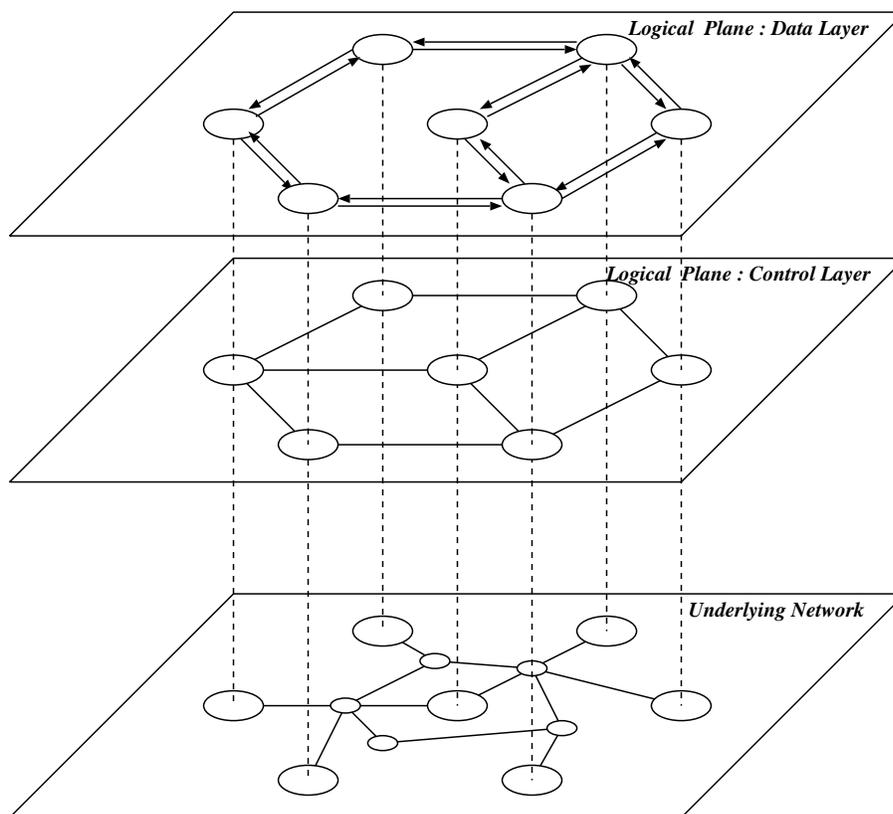


図 1: Overlay ネットワーク

リケーション全体の満足度をなるべく大きくするよう、各エンドホストの振舞いを分散制御する。

## 2.2 Overlay ネットワーク

エンドホスト間の、トランスポート層レベル通信が提供可能なネットワーク層を Underlying ネットワークとよぶ。提案プロトコル Emma では Underlying ネットワーク上のエンド間ユニキャストチャンネルを双方向リンクとみなし、それらとエンドホストから以下のように定義される論理ネットワークを Overlay ネットワークとよぶ。

- Overlay ネットワークは制御レイヤとデータレイヤの2つの論理プレーンからなる。
- 制御レイヤは、エンドホスト間の双方向ユニキャストチャンネルの集合体である。この双方向ユニキャストチャンネルを制御チャンネルとよび、エンドホスト  $i$ 、エンドホスト  $j$  間の制

御チャンネルを  $(i, j)$  で表す．以下ではエンドホスト  $i$  をノード  $i$  とよぶ．

- データレイヤは，制御チャンネルと並行に存在する論理的に有向なチャンネルの集合体である．この有向チャンネルをデータチャンネルとよび，ノード  $i$  からノード  $j$  へのデータチャンネルを  $(i, \vec{j})$  で表す．また，データチャンネルの成立には制御チャンネル  $(i, j)$  の存在が必要である．なお， $(i, \vec{j})$ ， $(j, \vec{i})$  は独立したチャンネルとして扱う．

なお，各有向チャンネル  $(i, \vec{j})$  に対し，始点ノード  $i$  はそのチャンネルに配信可能なメディア数，終点ノード  $j$  は受信可能なメディア数を指定することで，アプリケーションが利用可能な帯域を制限できる．また，ノード  $i$  は  $(i, \vec{j})$  の帯域予測を RTT 測定等によって行なう．

図 1 に Overlay ネットワークの概念図を示す．また，表 1 にこれ以降で用いる記号の表記を示す．

表 1: 表記

記号	意味
$(i, j)$	ノード $i, j$ 間の制御チャンネル
$(i, \vec{j})$	ノード $i$ から $j$ へのデータチャンネル
$d(i, j)$	$(i, \vec{j})$ の計測などによって得られた遅延
$D(v)$	$v$ からのエンド間遅延の最大
$T_s$	$s$ を根とする経路木
$T_s(i)$	ノード $i$ を根とする, $T_s$ の部分木 ( $T_s(s) = T_s$ )
$\uparrow_s(i)$	$T_s$ について, ノード $i$ の上流ノード
$\downarrow_s(i)$	$T_s$ について, ノード $i$ の下流ノード集合
$S^{[R]}$	メディア $V_s$ について “受信状態 (R)” であるような, ノード集合 $S$ の部分集合
$S^{[Q]}$	メディア $V_s$ について “要求状態 (Q)” であるような, ノード集合 $S$ の部分集合
$S^{[N]}$	メディア $V_s$ について “未受信状態 (N)” であるような, ノード集合 $S$ の部分集合
$T^\perp$	木 $T$ の葉ノード集合
$p_s(i)$	メディア $V_s$ に対するノード $i$ のプリファレンス値

### 3 提案するアプリケーションレベルマルチキャストの Protokol 記述

提案 Protokol Emma では、制御メッセージとして用いるメッセージを CTRL , DATA , MEDIA の 3 タイプに分類し、さらに各タイプをいくつかのサブタイプに分類する。これ以降、各メッセージのタイプを、“タイプ/サブタイプ”で分類する。また、特にメッセージ内容を明示したい場合には“タイプ/サブタイプ { メッセージ内容 }”と表す場合もある。メッセージタイプの一覧を表 2 に示す。

表 2: ノード  $u$  から  $v$  への制御メッセージ

タイプ	サブタイプ	レイヤ	内容 *	意味
CTRL	Join	Underlying	“Accept” or “Reject”	制御チャネル $(u, v)$ の構築要求
	Join-Reply	Underlying		CTRL/Join の返答
	Alive	Control	セッション参加中	
	Leave	Control	セッションから離脱	
DATA	Join	Control	“Accept” or “Reject”	制御チャネル $(u, v)$ 上にデータチャネル $(v, \vec{u})$ と $(u, \vec{v})$ の構築を要求
	Join-Reply	Control		DATA/Join の返答
MEDIA	Join	Control	$s, P_s(v)$	$V_s$ の受信要求 (集約された preference $P_s(v)$ を含む)
	Join-Reply	Control	“Accept” or “Reject”	Media/join の返答
	Keep	Control	$s, P_s(u)$	$(v, \vec{u})$ における $V_s$ の継続受信通知 (集約された preference $P_s(u)$ を含む)
	Leave	Control	$s$	$(v, \vec{u})$ における $V_s$ の受信停止要求

\* 送受信ノードアドレスはデフォルトで含まれるとする

#### 3.1 チャネルの確立

Emma では、ノード  $u$  はセッション参加時に、ロビーサーバの役割を持つサーバに自身の IP アドレスを登録し、そのサーバにすでに登録しているノード、すなわち、すでにセッション

ンに参加しているノードの IP アドレスリストを入手する．そして，それらのノードとのエンド間遅延及び帯域を計測し，帯域に余裕のあるいくつかのノード（各ノードを  $v$  とする）に対し，制御チャンネルの構築を要求する CTRL/Join メッセージを送出する．送出先のノード  $v$  に受け入れられ，CTRL/Join-ReplyAccept が返ってきた場合は制御チャンネル  $(u, v)$  を構築する．構築した制御チャンネル  $(u, v)$  にはデータチャンネル構築を要求する DATA/Join メッセージを送出する．DATA/Join を受け取ったノード  $v$  は自身の入出力次数差がある閾値以内となるように受け入れ可否を決定し，DATA/Join-Reply を用いた返答を行なう．返答が DATA/JoinAccept だった場合，すなわちノード  $u$  の  $v$  に対するデータチャンネル構築要求が受け入れられた場合はデータチャンネル  $(v, \vec{u})$  が生成される．

### 3.2 周期時間によるノード間の同期

各ノードはシーケンス番号を管理し，定められた時間周期ごとにそれを 1 ずつ増加させる．各ノードは制御チャンネル確立時に，接続先ノードから現在のシーケンス番号と次の時間周期までのオフセットを受けとり，以降その周期に従いシーケンス番号を増加させる．これにより，各ノードがおおよそ同じタイミングでシーケンス番号を同じ値に更新することになる．あるノードでシーケンス番号が  $i$  に更新される時刻を  $TS_i$  で表す． $TS_i$  から  $TS_{i+1}$  までの時間  $TS_{i+1} - TS_i$  をタイムスロットとよぶ．Emma では，各メディアの継続受信通知や受信要求を，制御チャンネルで隣接したどのノードから受け取ったかという情報のみが必要であり，ノードごとに状態を管理する必要はない．したがって，継続受信通知や受信要求を送出するタイミングをタイムスロットにより管理し，経路木の根に向かい転送されるメッセージをタイムスロットごとに集約することでオーバーヘッドを削減する．

### 3.3 タイプ $A$ のメディア用経路木構築と維持

ノード  $u$  は，データチャンネルが存在している，あるノード  $v$  について， $v$  が保持するエンド間最大遅延の値  $D_v$  と  $u v$  間の遅延  $d(u, v)$  の和があらかじめ定められたセッション許容遅延以下であれば，以下のような処理を行なう．

1.  $v$  をタイプ  $A$  のメディア用の送受信ノードとして登録

2. 自身のエンド間最大遅延  $D_u$  を  $D_v + d(u, v)$  に設定
3.  $v$  を親ノードとし,  $v$  の親ノードにも制御チャネル, データチャネルを生成
4.  $v$  の親ノードを自身の親ノード  $v$  との接続が切れた場合の代替経路に設定

これにより, タイプ  $A$  のメディアの経路木は, エンド間最大遅延がセッション許容遅延以下である共有木となる. なお, セッション許容遅延を満足できないノードは, 入出力次数制限の範囲で他のノードと新しく制御チャネルを生成する.

各ノードは適当なタイムスロット間隔で制御チャネルに対しフラッディングを行ない, 自身が保持するエンド間最大遅延値を更新する.

### 3.4 タイプ $V$ のメディア用経路木構築と維持

タイプ  $V$  のメディアの経路木は送信者を根とし, 最大遅延がセッション許容遅延以下であるような最短経路木とする. この経路木を生成するため, 各ノードは DVMRP ( Distance Vector Multicast Routing Protocol ) のように定期的に各ノードからのタイプ  $V$  のパケットをフラッディングする.

以下では, ノード  $s$  が送信するタイプ  $V$  のメディアを  $V_s$  で表す.  $V_s$  に対し, ノード  $v$  は  $V_s$  へのプリファレンス値  $p_s(v)$  を指定しておくとする.  $V_s$  の経路木を  $T_s$ ,  $T_s$  について  $v$  の上流ノード, すなわち, 親ノードを  $\uparrow_s(v)$ , 下流ノード集合, すなわち, 子ノード集合を  $\downarrow_s(v)$  で表す.

タイプ  $A$  のメディアとは異なり, 各ノードは経路木に基づくパケット転送テーブルを保持し, それに基づき経路木の各子ノードにパケットを転送するか否かを決定する. また, 与えられたプリファレンス値に基づき, パケット転送テーブルの内容を決定する. 以下ではその方法を述べる.

なお, 提案プロトコル Emma では, 各ノードにおいてメディア  $V_s$  に対し, “受信 (R)” , “要求 (Q)” または “未受信 (N)” のいずれか状態を示すフラグを保持することとする.

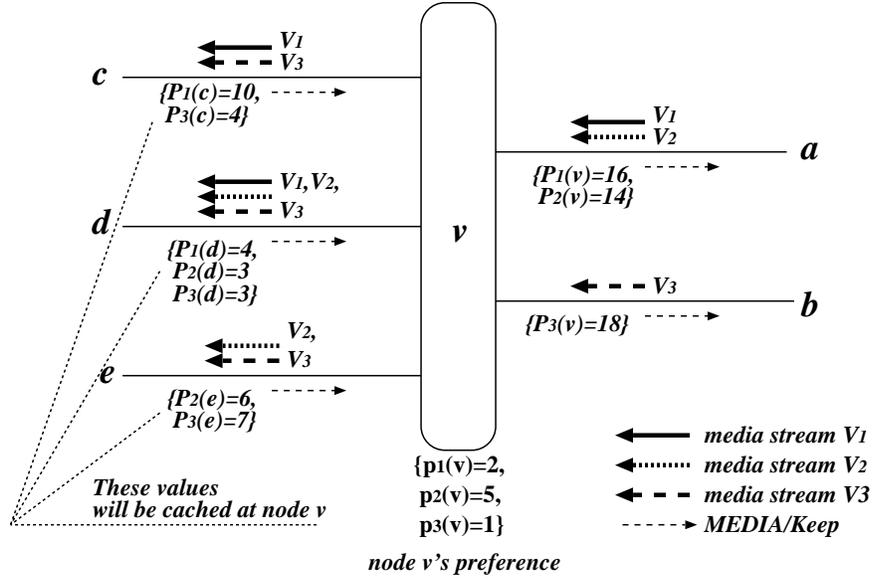


図 2: MEDIA/Keep による プリファレンス集約

### 3.4.1 MEDIA/Keep メッセージによるプリファレンス値の集約

各ノード  $v$  は受信中のメディア  $V_s$  に対し,  $p_s(v)$  を含む MEDIA/Keep メッセージをタイムスロットごとに, 上流ノード  $u = \uparrow_s(v)$  に送出する. なお,  $v$  が同じ親ノードから複数のメディアを受信している場合, メッセージ数削減のため, それらに関する MEDIA/Keep メッセージは 1 つのメッセージとして送出する. ただし,  $v$  が 1 つ以上の子ノードから  $V_s$  に関する MEDIA/Keep を受け取っている場合は送出する MEDIA/Keep の内容を以下のように決定する.

以下,  $v$  を根とする  $V_s$  の部分経路木  $T_s(v)$  上にあり, かつ  $V_s$  について受信状態であるノードの,  $V_s$  に対するプリファレンス総和を  $P_s(v)$  で表すと,

$$P_s(v) = p_s(v) + \sum_{w \in \downarrow_s(v)^{[R]}} P_s(w) \quad (1)$$

が成り立つ. 各ノード  $v$  が, 送出する MEDIA/Keep メッセージに  $P_s(v)$  を含めることで, MEDIA/Keep を受け取るノードは, 自身の下流ノードで  $V_s$  を受信しているノードのプリファレンスを知ることができる. ノード  $v$  は各子ノード  $w$  から受け取った MEDIA/Keep に含まれる  $P_s(w)$  を子ノード別に保持しておく.

なお，MEDIA/Keep はタイムスロットごとに送出され，ある子ノード  $w$  から一定のタイムスロット数 MEDIA/Keep を受けなかった場合は， $v$  は  $w$  への配送を停止し，パケット転送テーブルを書き換える．

図 2 に MEDIA/Keep メッセージによるプリファレンス集約の例を示す．

ノード  $v$  は親ノード  $a$  から  $V_1, V_2$  を， $b$  から  $V_3$  を受信し， $c$  には  $V_1, V_3$  を， $d$  には  $V_1, V_2, V_3$  を， $e$  には  $V_2, V_3$  をそれぞれ送信している． $v$  はそれらの子ノードから受け取る MEDIA/Keep に含まれる，集約されたプリファレンスに自身のプリファレンスを合わせて  $P_1(v), P_2(v)$  及び  $P_3(v)$  を計算する． $v$  は  $a$  への MEDIA/Keep には  $P_1(v), P_2(v)$  を， $b$  への MEDIA/Keep には  $P_3(v)$  を含めて送信する．

### 3.4.2 MEDIA/Join メッセージによる受信要求集約と配送可能性計算

ノード  $w$  が未受信メディア  $V_s$  を受信したい場合， $v = \uparrow_s(w)$  へ MEDIA/Join メッセージを送信する．MEDIA/Join メッセージは，他ノードが送信した  $V_s$  についての MEDIA/Join メッセージと集約されて上流ノードへと転送される．MEDIA/Join メッセージにはシーケンス番号を含めておき，タイムスロット間隔ごとに送信する．

$V_s$  が既に配送されているノード  $s'$  に MEDIA/Join が到着した場合， $s'$  は， $V_s$  の受信を要求するノードにより形成される  $T_s(s')$  の部分木，すなわち  $V_s$  の要求木上で既に配送されている他のメディアをいくつか部分的に停止させることによるプリファレンスの損失総和の最小値と， $V_s$  に対し要求状態にあるノードのプリファレンス総和を比較し，後者が上回る場合， $V_s$  は受け入れ可能と判断する．

$P_s(v)^{[Q]}$  で， $T_s(v)$  上のノードのうち， $V_s$  の要求状態にあるノードのプリファレンス総和を表すとすれば， $P_s(v)^{[Q]}$  は，

$$P_s(v)^{[Q]} = p_s(v) + \sum_{w \in \downarrow_s(v)^{[Q]}} P_s(w)^{[Q]} \quad (2)$$

で得られる．

これ以降， $loss_s(v)$  で， $V_s$  の要求状態にあるノードからなる  $T_s(v)$  の部分木上に  $V_s$  を配送するために既存のメディア配送を適当に中止する場合の最小のプリファレンス損失総和

を表すとする．ここで， $loss_s(v)$  には必ず  $(u, \vec{v})$  上で配送されるメディアを含むことから， $loss_s(v)$  は以下のように定義することができる．なお， $V(u, \vec{v})$  で， $(u, \vec{v})$  上に配送されているメディアの集合を表す．

$$\begin{aligned}
loss_s(v) &= \min_{V_t \in V(u, \vec{v})} \left\{ p_t(v) + \sum_{w \in \downarrow_s(v)^{[Q]}} P_t(w) \right. \\
&\quad \left. + \sum_{i \in \{T_s(v)^{[Q]} \times T_t(v)\}^\perp} \sum_{j \in \downarrow_s(i)^{[Q]}} loss_s(j) \right\} \quad (3)
\end{aligned}$$

上式で  $loss_s(v)$  の min の引数はそれぞれ以下の内容を表している．

- 第1項 ノード  $v$  の  $V_t$  に対する プリファレンス
- 第2項  $V_t$  の配送が  $(u, \vec{v})$  で停止された場合に損失するプリファレンス総和
- 第3項  $V_s$  の要求状態にあるノードからなる  $T_s(v)$  の部分木における， $V_t$  の終端ノードとなっている各ノード  $i$  において， $i$  の子ノード以下で別のメディアの配送を停止することにより損失するプリファレンス総和

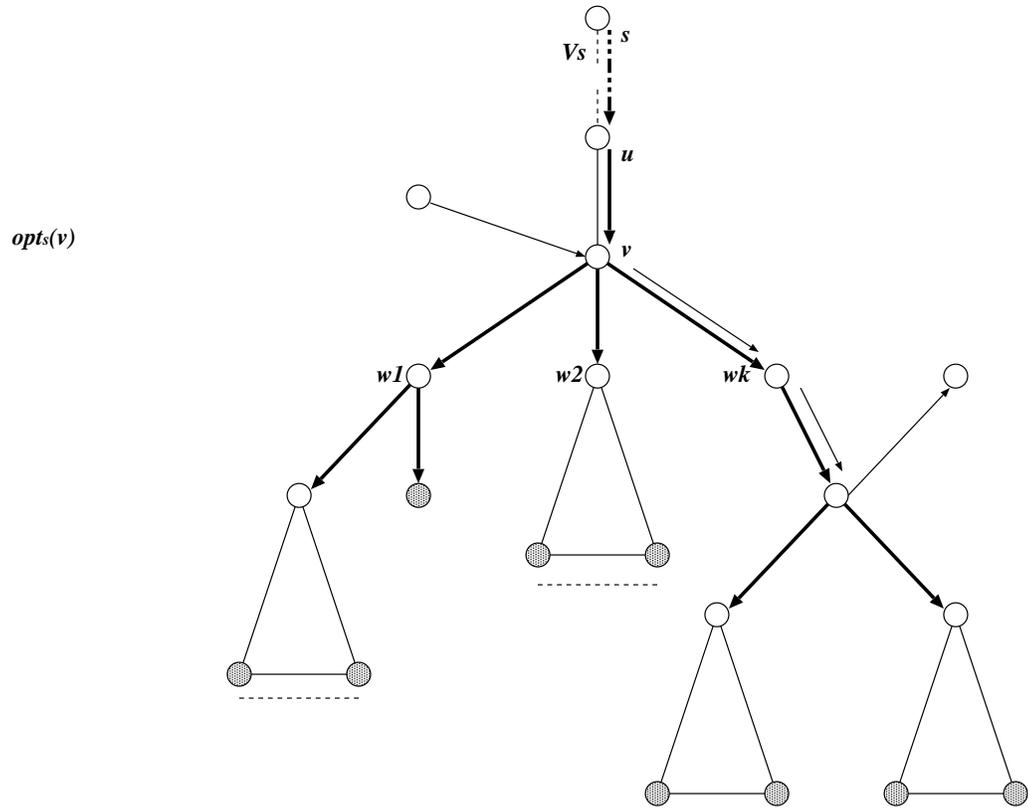
したがって，min の引数全体は，ノード  $u$  から  $v$  を通るメディア  $V_t$  と，その葉ノード以下で別のメディアの配送を停止して， $V_s$  の要求状態にあるノードから構成される  $T_s(v)$  の部分木上に  $V_s$  を配送するための空き容量を確保するために損失するプリファレンス総和を表す．

以下では， $loss_s(v)$  を実現するメディア配送停止の仕方を，配送停止メディアと停止される部分木の根ノードの組の集合  $opt_s(v)$  で表すとする．

このとき， $opt_s(v)$  は  $loss_s(v)$  の定義に基づき，以下で定義できる．

$$\begin{aligned}
opt_s(v) &= \{(t, v)\} \cup \bigcup_{i \in \{T_s(v)^{[Q]} \times T_t(v)\}^\perp, j \in \downarrow_s(i)^{[Q]}} opt_s(j) \quad (4)
\end{aligned}$$

式(4)に基づき，ノード  $v$  がノード  $u = \uparrow_s(v)$  に送る MEDIA/Join メッセージに，を保持させることで，ノード  $v$  は  $P_s(v)$ ， $loss_s(v)$  及び  $opt_s(v)$  の計算を行なうことができる．



⊠ 3:  $opt_s(v)$

- 要求するメディア  $V_s$  に対するノード  $v$  におけるプリファレンス総和  $P_s(v)$
- $v$  を根とする要求木  $T_s(v)^{[Q]}$  上の各ノード  $z$  について  $loss_s(z)$
- $v$  を根とする要求木  $T_s(v)^{[Q]}$  上の各ノード  $z$  について  $opt_s(z)$
- $T_s(v)^{[Q]}$  の葉ノード集合  $\{T_s(v)^{[Q]}\}^\perp$

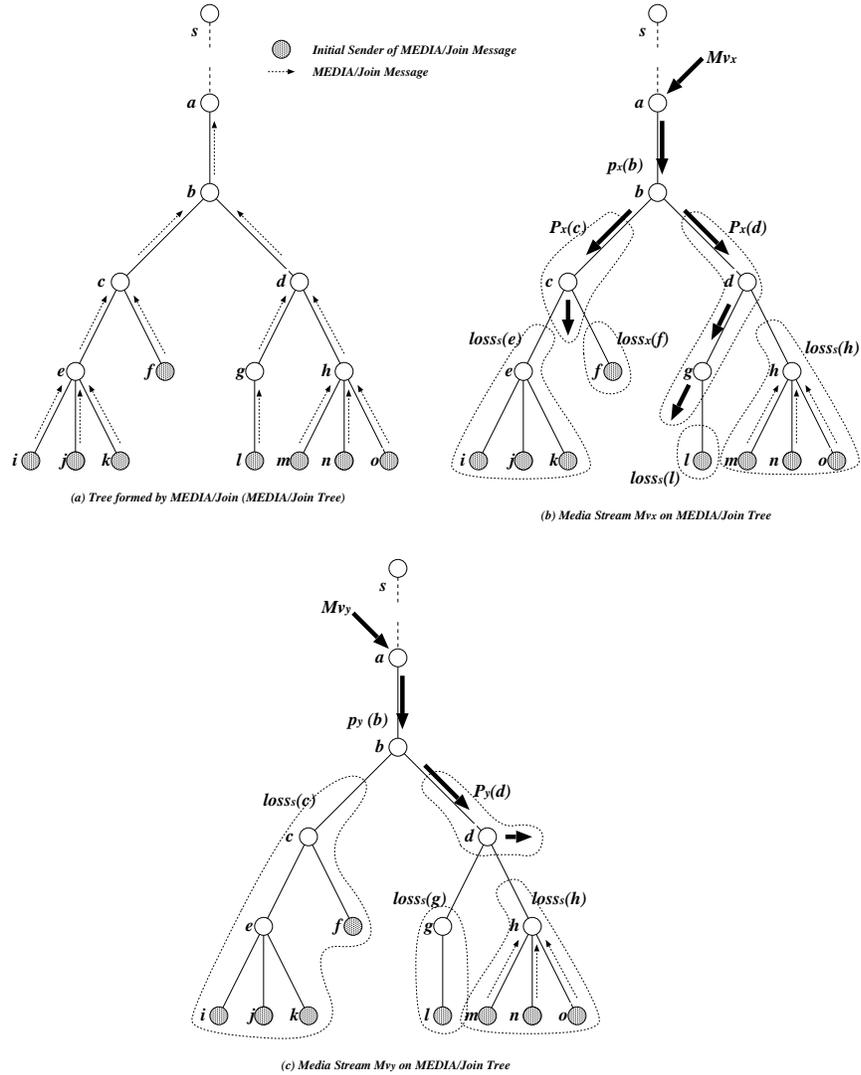


図 4: MEDIA/Join による配送要求と配送可能性計算

図 4 に例を示す．図 4 (a) は  $V_s$  に対する MEDIA/Join により形成される木を示している．ノード  $b$  は 2 つの子ノード  $c, d$  からそれぞれ MEDIA/Join を受け取り， $b$  にお

るプリファレンス総和  $P_s(b)$  を計算すると共に、経路木  $T_s$  における上流ノードである  $a$  からのデータチャネル  $(a, \vec{b})$  上で配送されているメディア、すなわちここでは  $V_x$ 、 $V_y$  の2つに対し  $loss_s(b)$  及び  $opt_s(b)$  を計算する。

メディア  $V_x$  が図 4 (b) のように  $V_s$  の要求木の  $(a, \vec{b})$ 、 $(b, \vec{c})$ 、 $(b, \vec{d})$ 、 $(d, \vec{g})$  上で配送されているとする。 $(a, \vec{b})$  で  $V_x$  の配送が停止された場合のプリファレンス損失は、 $b$  のプリファレンスと、 $c$  以下及び  $d$  以下のプリファレンス総和である  $p_x(b) + P_x(c) + P_x(d)$  として得られる。このとき、この要求木上で  $V_s$  を配送するためには、さらに  $(c, \vec{e})$ 、 $(e, \vec{j})$ 、 $(e, \vec{k})$ 、 $(e, \vec{l})$ 、 $(c, \vec{f})$ 、 $(g, \vec{l})$ 、 $(d, \vec{h})$ 、 $(h, \vec{m})$ 、 $(h, \vec{n})$ 、 $(h, \vec{o})$  上に空き帯域を作るためにその他のメディア配送を停止しなければならないが、この停止による損失の最小値は、 $loss_s(v)$  の定義より、 $loss_s(e) + loss_x(f) + loss_x(l) + loss_x(h)$  として得られる。したがって、 $V_x$  を  $b$  以下で配送停止とする際のプリファレンス損失の最小値は、

$$p_x(b) + P_x(c) + P_x(d) \\ + loss_s(e) + loss_s(f) + loss_s(l) + loss_s(h)$$

で得られる。

同様に、図 4 (c) の  $V_y$  については、

$$p_y(b) + P_y(d) + loss_s(c) + loss_s(g) + loss_s(h)$$

で得られる。

これらの値を比較し、より小さい値を  $loss_s(b)$  とする。 $loss_s(b)$  がメディア  $Mv_y$  の停止により得られるとすれば、このときの  $opt_s(b)$  は、各  $loss_s(q)$  に対応する  $opt_s(q)$  を用いて、

$$\{(y, b)\} \cup opt_s(c) \cup opt_s(g) \cup opt_s(h)$$

で得られる。

### 3.4.3 MEDIA/Accept による配送メディア変更

MEDIA/Join が  $V_s$  がすでに配送されているノード  $s'$  ( $s' = s$  の場合も含む) に到着した場合、以下の作業を行なう。

1. ノード  $s'$  では  $loss_s(s')$  と  $P_s(s')$  を比較
  - $loss_s(s') < P_s(s')$  ならば MEDIA/Join-ReplyAccept を送信
  - そうでなければ MEDIA/Join-Reply{Reject} を送信
2. MEDIA/Join-Reply を ,  $T_s(s')^{[Q]}$  の下流ノードに転送

また , MEDIA/Join-Reply{Accept} を受け取ったノードは , 次の作業を行なう .

1. 指定されたメディアの配送を停止
2.  $V_s$  を受信状態とする
3. MEDIA/Join-Reply{Accept} を下流ノードに転送

これにより , MEDIA/Join-Reply{Accept} を送信した全ノードから  $s'$  までの  $s$  の部分配送木上に ,  $V_s$  を配送するための帯域が空けられる . 一方 , MEDIA/Join-Reply{Reject} は各ノードの  $V_s$  の要求状態をクリアしながら葉ノードへと転送される .

## 4 考察

一般にエンドホスト間でのマルチキャストには、ネットワークの障害及びセッションからの離脱によるエンドホストの不安定性、エンド間の帯域変動などによるエンドチャンネルの不安定性に起因する非安定性問題が考えられる。また、プロトコルの振舞い自体がそのような不安定性を引き起こすことも考えられる。本研究では次のような状況への対策を以下のように検討している。

### - メディア切り替えによる振動

プリファレンスに基づくメディア配送の切り替えが、複数のノードの受信要求やプリファレンスの変更により頻繁に起こることも考えられる。また、複数のメディアが同時に切り替わることも考えられる。これらの振舞いは振動とよばれる現象を生じる可能性もある。この現象への対策としては、(i) メディア切り替えのプリファレンスに一定の閾値を設定し、プリファレンス差が少ない場合はメディア配送切り替えを行わない方法、(ii) タイムスロットにより MEDIA/Join メッセージの送出間隔を調整する方法といったものが考えられる。ただし、それらの方法で用いる閾値や制御メッセージ送出間隔の値と、反応性とは一般にトレードオフの関係にあるため、セッションの規模に応じた適切な値をシミュレーションにより設定する必要がある。

### - 帯域の減少

データチャンネル  $(u, v)$  の帯域減少も考えられるが、これはそのチャンネルを利用する複数メディアの  $v$  での受信レート低下、ロス率増加から  $v$  が判断する。 $v$  はキャッシュされた値から判断可能である。プリファレンス総和  $P_s(v)$  が最も小さいメディア  $V_s$  について、MEDIA/Leave メッセージを  $u$  に送信し、レート調整を行なうことで適応できる。

### - ノードの途中離脱によるネットワーク形状変化

セッションからのノードの途中離脱によるネットワークの不安定性に対しては、次のような方法により各メディアの配送経路木の保持をはかることができる。経路木の構築時に親ノードの親ノードと制御チャンネルを構築しているため、離脱したノードの子ノードであったノードには、受信中の各メディアについて、離脱したノードの親ノードへの制御チャンネルが存在しているはずである。そのノードに対し、もしデータチャンネルが存在しないなら

ば、データチャネルを構築し、MEDIA/Join メッセージの送信を行なう。MEDIA/Join  
メッセージに含まれる要求プリファレンスはそれまでのプリファレンスとする。

## 5 シミュレーション実験

ネットワークシミュレータ ns-2 [11] による Emma のシミュレーション実験を行い、その性能評価を行なう。

### 5.1 シナリオ設定

ある程度のエンドホストを用意し、あるメディアを要求する際の要求メッセージ MEDIA/Join 送出からメディア受信までのレスポンス時間、またネットワークを流れている制御メッセージ量を計測し妥当であるかどうかを評価する。

### 5.2 シミュレーション結果

数ノードのエンドホストによるネットワークにおけるレスポンス時間、制御メッセージ量は以下の表 3 のとおりである。

表 3: シミュレーション結果

	8 ノード	9 ノード	10 ノード	11 ノード
制御メッセージ量 (個)	198	273	373	426
レスポンス時間 (秒)	0.1 ~ 1.5	0.1 ~ 1.6	0.1 ~ 1.6	0.1 ~ 1.6

ここでの制御メッセージ量とは、メディア要求メッセージを送出した時点から配送木が定常状態になった時点までにネットワーク中に流れた制御メッセージの量を求めている。

本実験のタイムスロットは 0.5 秒間隔であり、ネットワークトポロジーは図 5 のとおりである。8 ノード、9 ノード、10 ノードのネットワークは、それぞれノード 8、ノード 9、ノード 10 以降を切ったネットワークである。

また、シミュレーション結果の値の妥当性については、今後考察する。

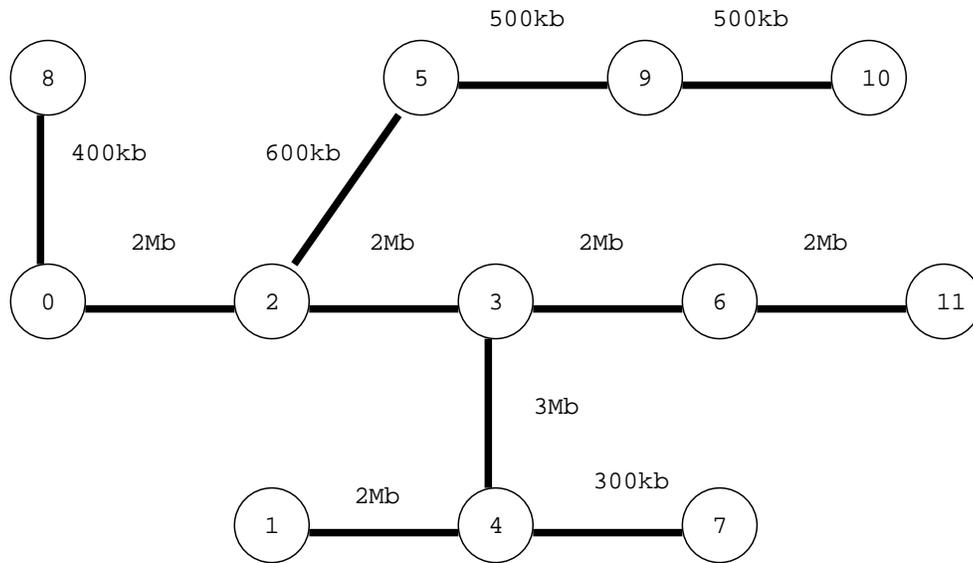


図 5: シミュレーション実験のネットワークポロジ

### 5.3 今後の測定予定

一般にアプリケーションマルチキャストはネイティブマルチキャストと比較してネットワーク利用効率が低下する。したがって、それが十分妥当なレベルであることを示すために実リンク上での重複率も測定する予定である。また、エンドホスト間の実ホップ数およびエンドホストの転送数を計測することで、提案プロトコル Emma が実ホップ数最小でエンドホストの転送負荷が最大であるユニキャストと比較しどの程度の効率向上がはかれているかを調べる。また、ノード離脱に対する十分な適応性があることを示す。

提案プロトコル Emma はプリファレンスを用いた動的転送制御を採用しグループコミュニケーションのアプリケーション向けに設計している。したがって、プリファレンス順に要求し帯域に余裕があれば受け入れ、かつ動的転送制御を行なわない制御方法、また、要求順に順次受け入れる制御方法と比較し、どの程度のプリファレンス向上が達成できるかも測定する予定である。

## 6 あとがき

本研究では、各エンドホストが他のエンドホストにオーディオ及びビデオを配信するコミュニケーションシステム向けに、アプリケーションレベルマルチキャストの一方式を提案した。提案プロトコルではユーザのメディアに対するプリファレンスを考慮したメディア配送制御を行い、全ユーザの満足度ができるべく大きくなるようにした。

また、ネットワークシミュレータ ns-2 [11] を用いたシミュレーション実験により性能評価を行った。

アプリケーションレベルマルチキャストは、マルチキャストインフラストラクチャの普及やグローバルマルチキャストアドレスの割り当て等、ネイティブマルチキャスト利用における諸問題が解決するまでの、グループコミュニケーション向けプロトコルの実用的な解として注目を集めている。

今後の課題としては、シミュレーション実験において未測定的项目を測定し、Emma のアプリケーションレベルマルチキャストとしての妥当性を確認することが挙げられる。また、ネットワークが混在した環境に対応できるよう、フィルタ機能を持つエンドホストやアプリケーションレベルプロキシの存在を仮定し、その場合により適したプロトコルの設計、さらに、会議アプリケーションの実装、ワイヤレスのマルチキャストへの応用などが挙げられる。

## 謝辞

本研究にあたり，理解ある御指導，御鞭撻を賜わり，常に励まして頂きました東野輝夫教授に心から深く感謝致します．

本研究にあたり，適切な御指導，御助言を頂きました北道淳司講師に心から深く感謝致します．

本研究にあたり，的確な御助言を頂きました吉岡敏明助手，中田明夫助手に心から深く感謝致します．

また，本研究にあたり，親切的な御指導，御助言を頂きました谷口研究室の山口弘純助手，博士課程1年生の廣森聡仁氏に心から深く感謝致します．さらに，本研究において，貴重なご意見などを頂きました滋賀大学経済学部の安本慶一先生に心から感謝致します．

最後に，東野研究室の皆様のお心温まる御助言，御協力に感謝致します．

## 参考文献

- [1] Chu,Y.-H., Rao,S.G. and Zhang,H. : “A Case for End System Multicast”, Proc. of ACM SIGMETRICS, 2000.
- [2] Chu,Y.-H., Rao,S.G., Seshan,S. and Zhang,H. : “Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture”, Proc. of ACM SIGCOMM, 2001.
- [3] Andersen,D.G., Balakrishnan,H., Kaashoek,M.F. and Morris,R. : “The Case for Resilient Overlay Networks”, Proc. of 8th Annual Workshop on Hot Topics in Operating Systems(HotOS-VIII), 2001.
- [4] Pendarakis,D., Shi,S., Verma,D. and Waldvogel,M. : “ALMI: An Application Level Multicast Infrastructure”, Proc. of 3rd Usenix Symp. on Internet Technologies & Systems, 2001.
- [5] Roca,V., El-Sayed,A. : “A Host-Based Multicast (HBM) Solution for Group Communications”, Proc. of 1st IEEE Int. Conf. on Networking (ICN’01), 2001.
- [6] Cohen,R. and Kaempfer,G. : “A Unicast-based Approach for Streaming Multicast”, Proc. of IEEE INFOCOM 2001, 2001.
- [7] Ratnasamy,S., Handley,M. Karp,K. and Shenker,S. : “Application-level Multicast using Content-Addressable Networks”, Proc. of 3rd Int. Workshop on Networked Group Communication, 2001.
- [8] Bommaiah,E., Mingyan,L., Mcauley,A. and Talpade,R. : “Amroute: Adhoc Multicast Routing Protocol”, Internet Draft, 1998.
- [9] Chawathe,Y., McCanne,S. and Brewer,E.A. : “RMX: Reliable Multicast for Heterogeneous Networks”, Proc. of IEEE Infocom 2000, 2000.

- [10] Francis,P. : “Yallcast: Extending the Internet Multicast Architecture”,  
<http://www.yallcast.com>.
- [11] Berkeley MASH Research Group, University of California : “The Network Simulator ns-2”, <http://www-mash.ce.berkeley.edu/ns/>, 2000.
- [12] K.L. Calvert, M.B. Doar and E.W. Zegura, “Modeling Internet Topology,” *IEEE Communications Magazine*, pp. 160–163, 1997.
- [13] タネンバウム, A . S . : “コンピュータネットワーク”, 水野忠則, 相田仁, 東野輝夫, 太田賢訳, プレンティスホール出版, 1999.
- [14] 塚越一雄 : “はじめての C++”, 技術評論社, 1999.
- [15] シルト, ハーバート : “STL 標準講座”, 多摩ソフトウェア訳, 翔泳社, 1999.
- [16] シュワルツ, R . L ., クリスチャンセン, T . : “初めての Perl”, 近藤嘉雪訳, オライリー・ジャパン, 1998.