

IP over DVB-RCS の設計と実装

慶應義塾大学 環境情報学部

山本 聡

soh@sfc.wide.ad.jp

指導教員

村井 純

徳田 英幸

楠本 博之

中村 修

南 政樹

平成 15 年 12 月 29 日

概要

本研究では、DVB-RCS(Digital Video Broadcasting Return Channel via Satellite)[1] をインターネットで利用するための機構を設計し、実装した。DVB-RCS は衛星回線を用いて受信局から送信局への狭帯域な戻り回線を提供する通信規格である。DVB-RCS を用いれば、地上回線網のない地域でも衛星回線の同報性を活かしたブロードキャスト型のネットワークに接続できる。

DVB-RCS は、一つのHUB(親局)と複数のTerminal(子局)からなるトポロジ上で、HUBから全てのTerminalへの単一のブロードキャスト型の片方向リンクと、TerminalからHUBへの複数のPoint-to-Point型の片方向リンクにより構成される。DVB-RCSを含むネットワークは複数の片方向リンクの集合である。既存のインターネット技術は、片方向リンクにより構成されたネットワークを意図して作られていない。このため、DVB-RCSを含むネットワーク上では、既存のインターネット技術が正常に動作せず、データリンクアドレス解決や経路制御など通信の基盤となる技術が使えない。

本研究では、DVB-RCSを、一つのブロードキャスト型マルチアクセスリンク(以下、ブロードキャストリンク)として抽象化した。ブロードキャストリンクでは、あるノードが送信したデータは、全てのノードが直接受信できる必要がある。DVB-RCSを仮想的に一つのブロードキャストリンクとして扱うためには、HUBとTerminal間が片方向リンクで接続しているため双方向に通信が行えない問題がある。また、TerminalはHUBにしか接続性を持たないため、Terminal同士の通信、Terminalからのブロードキャストが行えないという問題がある。この二つの問題を解決する必要がある。

本研究では、HUBからTerminalへの片方向リンクとTerminalからHUBへの片方向リンクを結合して一つの双方向リンクを作成する機構を提案した。また、HUBがTerminalからの通信を代行することで、Terminalが送信したデータを全てのノードが直接受信できるようにする機構を提案した。この二つの機構は、データリンクフレームの改変やインターネットの通信技術の改変を行わずにブロードキャストリンクの抽象化を行う。この二つの機構を導入したシステムの設計と実装を行った。

また、DVB-RCSを含むネットワークをエミュレートしたテストベッドを構築し、本システムの評価を行った。その結果、既存のインターネット技術が正常に動作し、本システムが有効であることを検証した。

本システムにより、DVB-RCSを既存の回線と同じようにインターネットに透過的に導入できる。

Abstract

This thesis describes mechanisms for using DVB-RCS (Digital Video Broadcasting Return Channel via Satellite) on the Internet. DVB-RCS is a telecommunication standard for the return link from a terminal to a HUB using a satellite link. DVB-RCS is expected to provide the bidirectional connectivity to a broadcast network in any place without a terrestrial link.

On DVB-RCS, both a forward link and a return link are unidirectional. The forward link is a broadcast link. The HUB broadcasts data to all terminals by the forward link. On the other hand, the return link is a point-to-point link. Each terminal sends data to the HUB, but can not broadcast it to the other terminal. The Internet architecture assumes the bidirectional connectivity of its data link. Thus, the technologies such as the address resolution or routing protocols do not work correctly on the DVB-RCS, because the network is composed of unidirectional links.

In this research, we treated DVB-RCS as a bidirectional broadcast link. In order for every node to be capable of receiving data from the other nodes, there were two problems. One is that the HUB and a terminal is connected by two different unidirectional links. The other is that a terminal can not directly communicate with the other terminals.

To solve these problems, we proposed a mechanism which combines the two unidirectional links as a single bidirectional link on a virtual interface. Additionally, we proposed another mechanism on which the HUB forwards the data from a terminal to the broadcast link by proxy. Both mechanisms are designed and implemented without violating the existing data link format and the layering model of the Internet.

We evaluated the mechanisms on the testbed network which emulates a network including DVB-RCS as its data link. As a result, the bidirectional communication was established correctly on the proposed data link topology. And then, we confirmed the effectiveness of those mechanisms. By this research, DVB-RCS can be utilized efficiently as well as the other data links on the Internet.

目次

第 1 章	序論	1
1.1	背景	1
1.2	目的	2
1.3	本論文の構成	2
第 2 章	問題点	3
2.1	DVB-RCS の概要	3
2.1.1	DVB-RCS のトポロジー	3
2.1.2	通信方式	3
2.1.3	データリンクフレームのフォーマット	4
2.2	DVB-RCS をインターネットで用いる際の問題点	4
2.2.1	データリンクアドレス解決の問題	5
2.2.2	経路制御の問題	6
第 3 章	関連研究	7
3.1	衛星回線の接続形態	7
3.2	UDLR	7
3.3	Satellite Link Configuration Protocol	8
3.4	既存の DVB-RCS の実装	9
3.5	解決手法の比較	11
第 4 章	解決手法	12
4.1	解決へのアプローチ	12
4.2	既存のインターネット技術を改変する手法	12
4.3	複数の論理的な Point to Point リンクに抽象化する手法	12
4.4	単一の論理的なブロードキャストリンクに抽象化する手法	13
4.5	本研究でのアプローチ	13
4.6	手法の位置付け	14
第 5 章	設計	15
5.1	設計要件	15
5.2	LED	16
5.3	仮想インターフェース	17
5.3.1	仮想インターフェースの概要	17
5.3.2	仮想インターフェースの設計	17

5.4	HUBによる通信の代行機構	18
5.4.1	HUBによる通信の代行機構の概要	18
5.4.2	HUBによる通信の代行機構の設計	18
5.4.3	HUBによる通信の代行機構の動作	19
5.4.4	HUBによる通信の代行機構を実装した際に起きる問題	19
5.4.5	LEDヘッダによる送信 Terminal の識別	20
第6章	実装	22
6.1	実装環境	22
6.2	HUBのデータリンクフレーム処理の実装	22
6.2.1	仮想インターフェースの実装	22
6.2.2	HUBによる通信代行機構の実装	23
6.3	Terminalのデータリンクフレーム処理の実装	24
6.3.1	仮想インターフェースの実装	24
6.3.2	LEDヘッダの実装	25
第7章	評価	26
7.1	実験環境	26
7.2	定性的評価	27
7.3	ネットワーク内での基本動作確認	27
7.3.1	TerminalのステーションからHUBのステーションへの到達性	29
7.3.2	Terminalのステーションから他のTerminalのステーションへの到達性	31
7.3.3	Terminalのステーション同士の到達性	34
7.3.4	ネットワーク外との基本動作確認	36
第8章	結論	38
8.1	まとめ	38
8.2	今後の課題	38

目次

2.1	HUB と Terminal の構成	4
2.2	通信方式	5
2.3	ARP	6
3.1	UDLR の送信局と受信局が通信する概念図	8
3.2	SLCP のモデル	9
3.3	EMS による IP over DVB-RCS の実装	10
3.4	インターネットから見た EMS の実装	10
4.1	Point to Point リンク	13
4.2	ブロードキャストリンク	13
4.3	DVB-RCS の HUB と Terminal がルータやブリッジであるトポロジ	14
4.4	ブロードキャストリンクに抽象化する手法の階層の位置付け	14
5.1	一つの双方向インターフェースに抽象化	15
5.2	Terminal 同士が論理的に直接通信	16
5.3	LED の概念図	16
5.4	仮想インターフェースの概念図	17
5.5	HUB による通信の代行機構	18
5.6	HUB による通信の代行機構の動作	19
5.7	ノード A、ノード B の双方向通信	19
5.8	LED ヘッダによる送信 Terminal の識別	21
5.9	カプセル化されたデータリンクフレームのフォーマット	21
6.1	HUB のデータリンクフレーム処理の実装	22
6.2	Terminal のデータリンクフレーム処理の実装	24
7.1	DVB-RCS テストベットネットワーク・トポロジ	26
7.2	ネットワーク内評価トポロジ	27
7.3	ネットワーク外評価トポロジ	36

表 目 次

3.1	関連研究と要求事項の関係	11
7.1	HUB、Terminal のインターフェース情報	27
7.2	各 PC の IP アドレスと MAC アドレス	28
7.3	データリンク情報取得プログラム	28
7.4	PC-B から PC-A への ping	29
7.5	HUB のデータリンク情報取得プログラム	29
7.6	Terminal-2 のデータリンク情報取得プログラム	30
7.7	PC-B から PC-C への ping	31
7.8	HUB のデータリンク情報取得プログラム	32
7.9	Terminal-2 のデータリンク情報取得プログラム	33
7.10	PC-B から PC-D への ping	34
7.11	Terminal-1 のデータリンク情報取得プログラム	34
7.12	HUB のデータリンク情報取得プログラム	35
7.13	実験後の PC-B の ARP Table	35
7.14	各 PC の IP アドレス	36
7.15	HUB、Terminal のインターフェース情報	37
7.16	要求事項	37

第1章 序論

1.1 背景

インターネットの利用形態は多様化し、様々なメディアを取り込みつつある。それに伴って、インターネットでは、様々な通信媒体を用途に応じて活用することが望まれている。近年、地上回線が整備されていない離島や山間部などの地域でも広帯域にインターネット接続できる通信媒体への要求が高まっている。また、音声や映像などの大容量のデータを多数に同報的に配信するマルチキャスト型のアプリケーションと、それを支える情報基盤の必要性が高まっている。

衛星回線は広域性、同報性、地理不遍性という地上回線にないメリットがある。このような特性をもつ衛星回線をインターネットの通信媒体として利用する研究が広く行われている。衛星回線をインターネットで利用することにより、狭帯域な地上回線しか整備されていない地域に広帯域なデータ配信ができ、マルチキャスト型のサービスの提供が容易にできる。

衛星回線をインターネットで利用する場合、既存の手法では、双方向回線として利用する場合と、送信局から受信局へデータを伝送する片方向回線として利用する場合の2通りがある。

2つの送受信局を双方向で接続すると、地上回線が整備されていない地域でもインターネット環境を構築できる。この手法では、1つの局ごとに4つのチャンネルを割り当てるため、地球局の数が増えるにしたがって、必要なチャンネル数が増え、周波数資源を消費する。また、双方向回線では、2つの地球局が1対1で接続するため、衛星回線の特性である広域性、同報性を生かせない。また、双方向回線を運用するには、各地球局に無線機器を操作する管理者が必要であり、電波の送信設備や運用にかかるコストが大きい。

一方、衛星回線を片方向回線として利用する場合、同一の情報を多数の受信者に対して同時に配信できる。一般的に送信局に比べ受信局の方が設備費用や運用、受信局が衛星回線に接続する数は制限がないなどの利点がある。しかし、インターネットのサービスやプロトコルは回線の双方向性を前提としている。このため、片方向回線をインターネットで利用すると、経路制御プロトコルをはじめとする基盤技術が正しく動作しない問題が生じる。この問題を解決するため、UDLR (Unidirectional Link Routing) [2] 技術が標準化されている。UDLRでは、地上回線等によって送信局、受信局間の双方向性が整備されていることを前提としている。このため、地上回線が整備されていない離島や、アジア地域内陸部などの地域では、片方向回線をインターネット上で使えない。

DVB-RCSは衛星回線を用いて受信局から送信局への狭帯域な戻り回線を提供し、複数の受信局が衛星経由で送信局にアクセスする通信規格である。DVB-RCSでは、受信局からの戻り回線の利用帯域を使用状況に応じて動的に変更できることや、受信局の設定を送

信局で制御できるため、受信局のユーザは衛星利用免許が不要である。DVB-RCS をインターネットで利用することにより、地上回線が整備されていない地域において片方向回線の特性を生かしたネットワークを手軽に構築できる。しかし、DVB-RCS では既存の通信媒体にはない特殊な接続性を持つため、現段階では DVB-RCS をインターネットにおいて透過的に利用できない。

1.2 目的

本研究では、DVB-RCS 上の接続性をモデル化し、それに基づいたインタフェースとデータリンク層を構築する手法と具体的な設計を示す。また、DVB-RCS のエミュレーション環境上において実装を行い、本機構の正当性と規模性を評価する。

1.3 本論文の構成

第 2 章では、DVB-RCS の概要と DVB-RCS における経路制御上の問題を述べる。第 3 章では、既存の解決法と、それらの対象と目的の違いを述べる。第 4 章では、第 2 章で述べた問題の解決手法を述べる。第 5 章では、仮想インタフェース機構と仮想ブロードキャストリンク機構を用いる解決法の詳細を述べ、本研究による設計を示す。第 6 章では、本機構の実装を述べる。第 7 章では、本研究で実装した 2 つの機構について評価を行う。第 8 章で以上の議論をまとめる。

第2章 問題点

本章では、DVB-RCSの概要を述べる。また、DVB-RCSをインターネットで利用する上での問題点を述べる。

2.1 DVB-RCSの概要

DVB-RCSは、ETSI(European Telecommunications Standard Institute)[3]が定義した、家庭やオフィスなどで広く用いられている放送用の衛星回線に、子局からのフィードバック用戻り通信を提供する通信規格である。DVB-RCSでは、子局に広帯域でデータを配信する親局をHUBと呼び、HUBに狭帯域な回線で接続する子局をTerminalと呼ぶ。また、HUBからTerminalへの回線をForward Linkと呼び、TerminalからHUBへの回線をReturn Linkと呼ぶ。

2.1.1 DVB-RCSのトポロジー

DVB-RCSは片方向回線と同じネットワークトポロジーにおいて、TerminalからHUBへのReturn Linkを衛星回線経由で提供する。DVB-RCSをインターネットで利用することにより、衛星回線の広域性、同報性を生かした衛星回線経由の双方向通信が行える。

HUBとTerminalの構成を図2.1に示す。このトポロジーには、以下のような特徴がある。

- (1) Forward Linkに同報型通信が可能
同一の情報を、広範囲に同時に送信できる、同報性に優れたトポロジーである。
- (2) 地上回線に依存しないネットワークを構築可能
Return Linkを衛星経由で行うため、双方向衛星回線と同じように地上回線に依存せずにネットワークを構築できる。
- (3) HUBがTerminalを制御可能
Return Linkの使用状況に応じてTerminalの使用帯域を動的に変更したり、Terminalの設定をHUB側でコントロールできる。

2.1.2 通信方式

HUBからTerminalに向かうチャンネルはTDM(Time Division Multiple Access)、TerminalからHUBに向かうチャンネルはMF-TDMA(Multi Frequency Time Division Multiple Access)を用いてデータを送信する。MF-TDMAとは、一本の回線を時間と周波

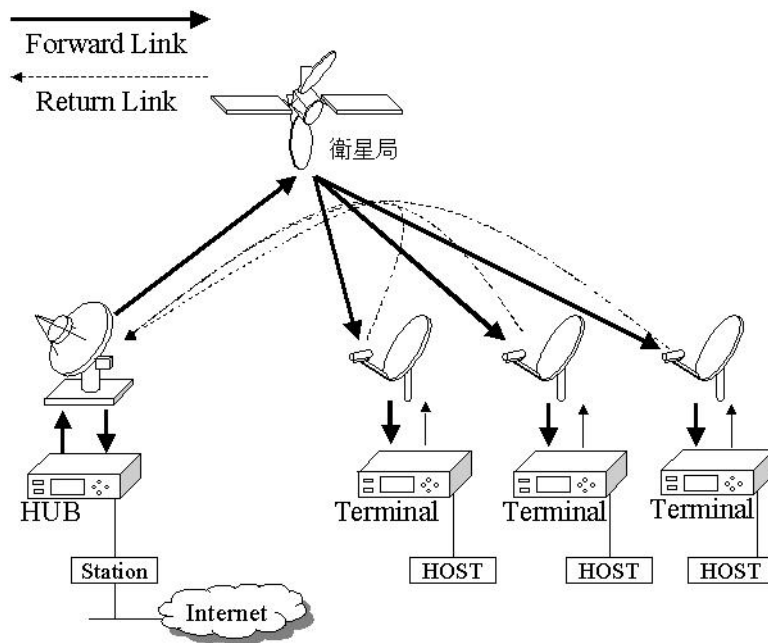


図 2.1: HUB と Terminal の構成

数によって分割し、多重通信を行う技術である。HUB は各 Terminal に周波数帯域と時間を割り当て、Terminal はこの割り当てに従い HUB に向けて電波を送信する。この方式を用いて複数の Terminal から送られる信号は衛星局で一つの回線にまとめられる。HUB は信号を復調し、全 Terminal からのデータを受信する。図 2.2 に MF-TDMA の概念図を示す。図の斜線部分は、ある一つの Terminal の信号の位置である。

2.1.3 データリンクフレームのフォーマット

DVB-RCS は、衛星放送を目的に開発された。HUB から Terminal へのデータ伝送に MPEG2-TS を用いる。また、Terminal から HUB へのデータ伝送に ATM 規格を利用する。IP パケットや Ethernet などのデータリンクフレームを MPEG2-TS で配送する方式は IETF において議論されており、現在標準化作業が行われている。

2.2 DVB-RCS をインターネットで用いる際の問題点

DVB-RCS 規格では、Forward Link と Return Link が別々のデータリンクであり、使用するデータリンクプロトコルも異なる。既存のインターネット技術の多くは、このような片方向にブロードキャスト型、もう片方向に Point-to-Point 型で、送受信に別々のデータリンクプロトコルを用いるネットワークを考慮していない。また、通信インターフェースは双方向に通信できると仮定して実装されている。このため、以下のような問題が起こる。

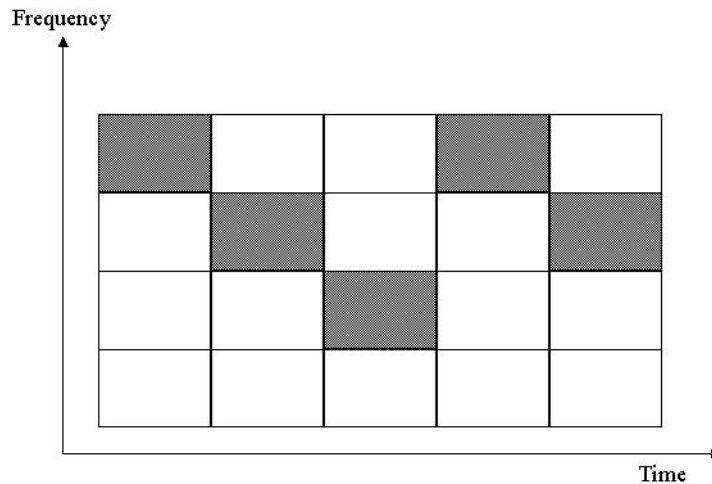


図 2.2: 通信方式

2.2.1 データリンクアドレス解決の問題

データリンクアドレス解決では主にマルチアクセス型のアドレス解決プロトコルを用いるか、固定で設定する手法がある。DVB-RCS 上では、以下の理由によりアドレス解決を行うのが困難である。

< 動的にアドレス解決を行う場合 >

Ethernet 等、IP アドレスと物理アドレスの対応が変化するネットワークでは、マルチアクセス型のアドレス解決プロトコルが用いられる。マルチアクセス型のアドレス解決プロトコルでは、例えば ARP などがある。ARP の動作を下図に示す。

A が宛先 IP アドレスを持つ B に対してパケットを送る際に、A,B ともに Ethernet アドレスを知らなかったとする。この場合、ARP は以下のように動作する。

- (1) A はパケットの宛先 IP アドレスであるノードの MAC アドレスを知るために、MAC アドレスを尋ねるメッセージ (ARP request) をマルチアクセスリンクにブロードキャストする。
- (2) B は自分の MAC アドレスを A に送信する。
- (3) A は B に対してデータを送ることができるようになる。
- (4) IP と MAC アドレスの対応表は一定時間立つと削除される。また、IP と MAC アドレスの対応に変化がある場合は、ARP キャッシュをクリアする。

これを、図 2.3 に示す。

DVB-RCS のネットワーク上で ARP を動作させた場合について考える。DVB-RCS は片方向リンクなので、ARP request がブロードキャストされるリンクと、ARP reply を返すリンクが異なる。DVB-RCS のステーション同士が ARP 要求をした場合、DVB-RCS は片方向リンクによって接続しているため、双方向に通信できず ARP が正常に動作しない。ま

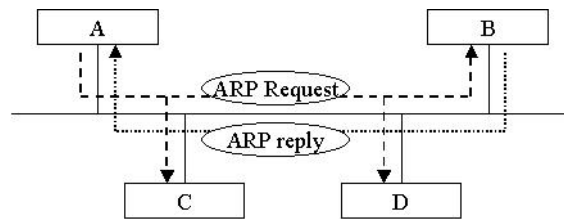


図 2.3: ARP

た、DVB-RCS では HUB は各 Terminal に接続性があるが Terminal は HUB にしか接続性がない。このため、Terminal からの ARP 要求がブロードキャストされず各ノードに到達しないため、ARP が正常に動作しない。

< 固定でアドレス解決を行う場合 >

Point to Point 接続した ATM ネットワークのような、決められた固定の IP アドレスにむけてパケットを送信する場合、アドレス解決を一度行えばそれ以降アドレス解決の必要はない。DVB-RCS 上で固定アクセス型のアドレス解決を行う場合、Terminal に繋がるノードが増えるに従ってアドレス解決の負担が増える。

2.2.2 経路制御の問題

DVB-RCS を用いて Terminal にホストとしてのインターネット接続だけでなく、ネットワークを提供しようとした場合や、マルチキャストによる転送を動的に行おうとした場合、DVB-RCS 上で OSPF や IGMP などの経路制御プロトコルが正常に動作する必要がある。

RIP や RIP2 などの距離ベクトル型の経路制御プロトコルでは、到達可能メッセージを一定時間毎にリンクに繋がるルータへ各インターフェースから送信する。メッセージを受信することで、送信元を経由して到達できるネットワークと、到達するまでのホップ数を知る。

OSPF などのリンクステート型の経路制御プロトコルでは、隣接するルータ同士が接続性を監視し、リンク情報とコストをドメイン全体にフラッティングする。これにより、経路情報を交換する。

IGMP は、マルチキャストルータが直接接続するインターフェース上に、Membership Query をマルチキャストホスト宛てに送信する。ホストから Membership Report を受信すると、メンバーシップリストにそのグループを追加する。また、Leave Group メッセージを受信するとそのグループをメンバーシップリストから削除する。

現在インターネットで利用されているこれらの経路制御プロトコルのアルゴリズムでは、隣接するルータと双方向でメッセージを交換し、経路情報を取得する。このため、隣接するルータ同士はリンク上のインターフェース間で双方向に通信可能であることが前提となっている。このため、DVB-RCS のように、片方向インターフェースで双方向に繋がるリンク上では、既存の経路制御プロトコルが正常に動作しない。このため、DVB-RCS 上では、動的な経路の割り当てを行えない。

第3章 関連研究

本章では、片方向リンクをインターネット上で用いる手法について関連研究として取り上げる。インターネットは双方向通信なので、多くの場合は片方向リンクと双方向リンクを組み合わせて用いる。

3.1 衛星回線の接続形態

片方向リンクと双方向リンクを用いてインターネット回線の一部として利用する技術は、次の二つに分類される。

- ・ ホストを接続する技術
この技術は、片方向リンクの衛星回線を用いて、インターネットの末端のノードを提供する。経路制御を衛星回線上で行うことを想定しない。
- ・ ネットワークを接続する技術
この技術は、片方向リンクの衛星回線を、既存のネットワークに透過的に利用可能とする。データリンクアドレス解決技術や経路制御技術などのネットワーク技術を動作させる。

以降に、ネットワークを接続する技術の技術として UDLR を、ホストを接続する技術として SLCP[4] と EMS による既存の DVB-RCS の実装 [5] について説明する。

3.2 UDLR

UDLR は、片方向リンク上に仮想的にブロードキャストリンクの機能を実現し、上位層に対してデータリンクの実際の機能をブロードキャストリンクにみせることで多くの経路制御プロトコルを動作させる。

UDLR では、受信局から送信局へのパケット送信にトンネリングを使用する。受信局は送信局の片方向インタフェースへ送信するデータリンクフレーム全体をペイロードとする IP パケットを作成し、双方向インタフェースから送信局の双方向インタフェース宛に送信する。これをカプセル化という。送信局は、双方向インタフェースで受け取ったパケットが受信局でカプセル化されたものである場合、ペイロードとして格納されている元のデータリンクフレームを取り出す。これを脱カプセル化と定義する。データリンクフレーム内部の IP ヘッダに記述されている宛先 IP アドレスを判別し、自分宛のデータリンクフレームを片方向インタフェースから受信したパケットとして扱う。それ以外のパケットはそのまま片方向リンクに送信する。これにより、次の通信路を実現する。

UDLR では、受信局の数が多くなった場合、トンネルの設定の軽減や、UDL の状態検知のために、動的なトンネル設定プロトコルが提案されている。これを、DTCP(Dynamic Tunnel Configuration Protocol) と呼ぶ。

片方向リンクに多数の受信局が接続した場合、大多数のノードが接続するブロードキャストリンクが構築される。しかし、RIP や OSPF などの経路制御プロトコルは衛星回線の大規模性を想定していない。このようなネットワークポロジには既存のインターネットプロトコルが対応できないと考えられる。

また、トンネリング技術は、ネットワークの階層化構造に違反しており、他の技術と競合が起こる可能性がある。また、送信局と受信局ではカプセル化の処理が必要である。

UDLR の送信局と受信局が通信する概念図を図 3.1 に示す。

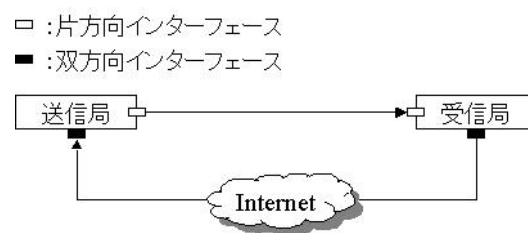


図 3.1: UDLR の送信局と受信局が通信する概念図

UDLR のカプセル化は、地上回線による戻りの経路がインターネットを用いる必要があるためである。DVB-RCS は片方向リンクで接続されるが、HUB から Terminal の経路と Terminal から HUB への経路は 1 ホップで到達性がある。このため、UDLR のように戻りの経路にインターネットを用いる必要はない。このため、カプセル化の処理を必要としない。

3.3 Satellite Link Configuration Protocol

Satellite Link Configuration Protocol(以下、SLCP) は、片方向リンクを受信するホストに対し、インターネットで利用されるアプリケーションに変更や制限を課すことなくサービスを提供可能とする技術である。SLCP は、片方向リンクの受信局に対し IP アドレスを割り当て、そのアドレスに対する経路制御により、効率的なパケット転送方式の実現を可能とする。

SLCP サーバは、受信局から IP アドレス割り当ての要求がくると、データベースモジュールから受信局に割り当てる IP アドレスを決定し、その IP アドレスと受信局の片方向リンクの MAC アドレスの組を受信局に渡す。受信局が片方向リンクを利用中、片方向リンクの接続性を監視する。また、片方向リンクの利用終了時に、ARP テーブルから該当エントリを削除する要求を送る。

SLCP クライアントは、認証とセキュリティ上で必要な処理を行い、IP アドレスの割り当て要求を SLCP サーバに送信する。アドレスの割り当てを受けた後、片方向インターフェースに該当 IP アドレスを付与する。片方向インターフェースの設定が終了した後、受信局から双方向インターフェースを介して送出される IP パケットは、片方向インターフェースの IP アドレスを送信元 IP アドレスとして持つようになる。

また、例外的処置として、SLCP サーバがダウンした場合、SLCP クライアントがダウンした場合、片方向リンクの接続性が断たれた場合、双方向リンクの接続性が断たれた場合の対応が実装されている。SLCP サーバは片方向インターフェースに定期的に SLCP HELLO メッセージを流す。SLCP クライアントは独自のタイマーを持ち、定期的に SLCP HELLO を受信することで、SLCP サーバのダウンの検知と、片方向リンクの接続性の検知を行う。また、SLCP クライアントから定期的に keep alive メッセージを SLCP サーバに送信する。SLCP サーバは独自のタイマーを持ち、定期的に keep alive メッセージを受信することで、SLCP クライアントのダウンの検知と、双方向リンクの接続性の検知を行う。

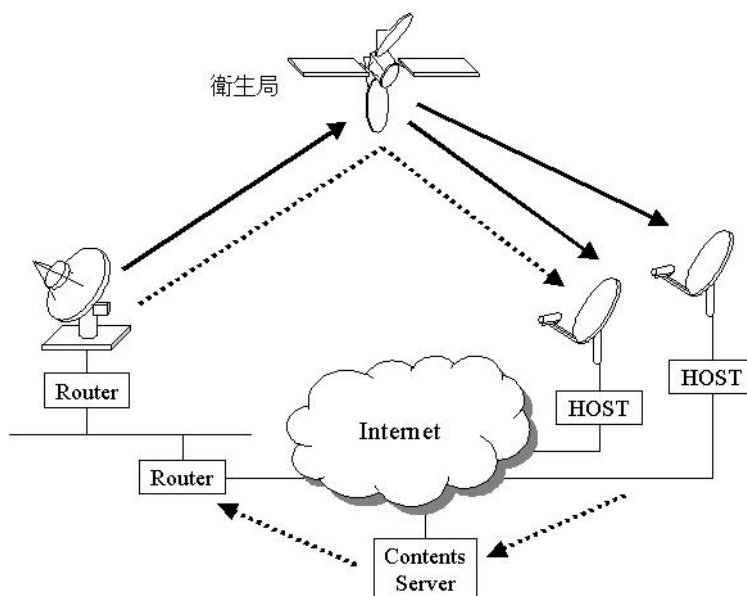


図 3.2: SLCP のモデル

SLCP はホストを接続するモデルを対象としている。衛星回線を用いて経路交換を行うことを想定していない。このため、対象とする目的が異なる。

3.4 既存の DVB-RCS の実装

ここで、EMS Satellite Network による既存の DVB-RCS の実装を示す。

HUB のノードから Terminal のノードへの送信の場合、Terminal に向けられたデータは HUB に入る。HUB はパケットを OPAL(データリンクフレームを MPEG2-TS にカプセル化するルータ)に向けて送信する。OPAL は HUB から受け取ったパケットを MPEG2-TS に変換し、宛先 Terminal の MAC アドレスに向けてデータリンクフレームを送信する。また、HOST から Router への送信の場合、Terminal が HOST から受け取ったパケットは ATM のインターフェースから直接 HUB に送信される。

EMS による DVB-RCS の実装は、Terminal にホストとしての機能を与えることを目的に設計されている。このため、Terminal は、IP アドレスによるレイヤー 3 の機能を持ちながら経路情報を持たないという実装になっており、Terminal のホスト以下にセグメントを作

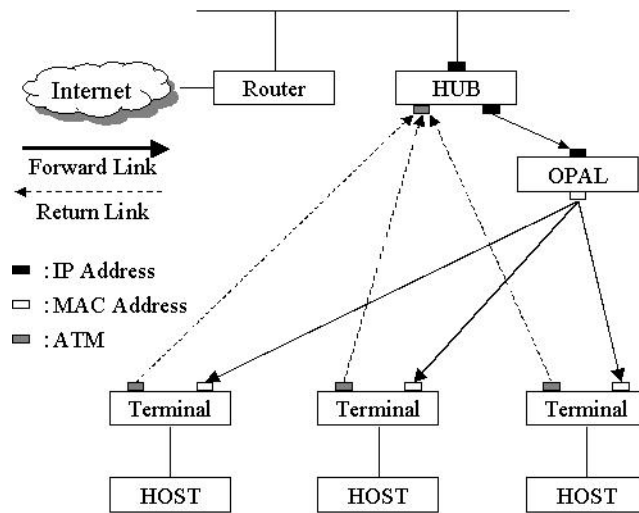


図 3.3: EMS による IP over DVB-RCS の実装

ることができない。このため、IP トンネリングなどの手法を用いない限り、Terminal 以下のノードをインターネットの通信に組み込むことができない。また、HUB と Terminal の間で、データリンクの異なる片方向インターフェースや、ブロードキャストと Point-to-Point の混在するネットワークトポロジ上で IP の伝送を行っているため、2.4 章で述べたように、ARP 等のデータリンクアドレス解決プロトコルが正常に動作せず、HUB の管理者が手動でデータリンクアドレス解決を行っている。この実装を含むネットワーク上では、RIP や OSPF などの経路制御プロトコルが正常に動作しない。

この実装をインターネットから見た場合、HUB インターフェースと Terminal インターフェースの間でしか通信できない一つの大きなルータのようになる。また、このルータは経路情報を持つことができない。これを図 3.4 に示す。

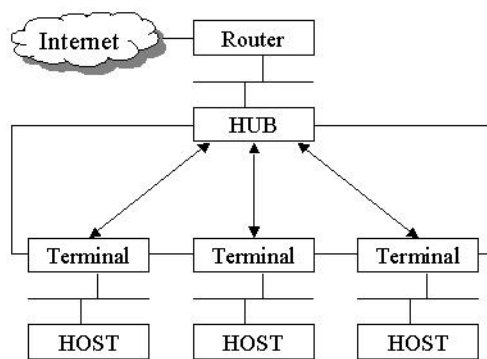


図 3.4: インターネットから見た EMS の実装

3.5 解決手法の比較

本章では、片方向回線をインターネットで用いるための既存の技術を述べた。UDLR によるブロードキャストリンクの構築手法は、地上回線経由でインターネットを通じて戻りの回線を提供することを前提とし、トンネリングの手法を用いている。SLCP は、ホスト接続型の技術であるため、ネットワークを提供する役割を果たさず、目的が異なる。EMS による DVB-RCS の実装は、Terminal にホストしか繋がらないと仮定した場合に有効な実装である。しかし、DVB-RCS を用いて地上回線の利用できない地域にネットワークを構築することや、動的なマルチキャスト配信を行うことを想定して作られていない。

表 3.1: 関連研究と要求事項の関係

	接続形態	使用する回線
UDLR	ネットワークを接続する技術	地上回線と併用
SLCP	ホストを接続する技術	地上回線と併用
EMS の実装	ホストを接続する技術	衛星回線のみ
本研究の要求事項	ネットワークを接続する技術	衛星回線のみ

本研究では、DVB-RCS を用いて、衛星回線のみを用いてネットワークを提供する技術を要求事項とする。関連研究で述べた既存の衛星インターネット技術では、この要求事項を満たすものは存在しない。そこで、本研究では、DVB-RCS を用いてネットワークを設計する上で最も適したモデルについて考察し、その議論に基づいて設計を行う。

第4章 解決手法

本章では、前章まで述べてきた、DVB-RCS をインターネットに用いる際の、問題点の解決手法について述べる。

4.1 解決へのアプローチ

DVB-RCS のもつ特殊なトポロジ上で、データリンクアドレス解決と経路情報交換を正常に動作させる為のアプローチとして、以下があげられる。

- ・ 既存のインターネット技術を改変する手法
- ・ 複数の論理的な Point to Point リンクに抽象化する手法
- ・ ネットワークを一つのブロードキャストリンクに抽象化する手法

4.2 既存のインターネット技術を改変する手法

これは、既存のデータリンクアドレス解決プロトコルや経路制御プロトコルを直接改変し、DVB-RCS のネットワークトポロジに適應させる手法である。このため、プロトコルごとに DVB-RCS のネットワークトポロジに適應するように設計をし直さなければならない。また、前提とする既存のインターネット機構に改変を加えるため、新たな技術に対応できなったり、既に運用されている他のネットワークとの間で競合が起こる可能性がある。したがって、この手法を用いてインターネット上に早期に DVB-RCS を利用することは難しい。

4.3 複数の論理的な Point to Point リンクに抽象化する手法

HUB は全 Terminal に接続性があるが、Terminal は HUB にしか接続性を持たない。このため、HUB と各 Terminal に個別の Point-to-Point リンクが存在すると仮定する手法が考えられる。

このネットワークモデルを図 4.1 に示す。

このモデルでは、論理的に HUB から各 Terminal への接続性が複数に分割される。このため、HUB から複数の Terminal にマルチキャストやブロードキャストによってデータを配信する場合、論理的に定義された複数の Point-to-Point リンクに重複してデータが流れる問題が生じる。また、Point to Point リンクでは、HUB から Terminal の接続性が論理的に分断されるため、衛星のもつ同報性を生かしくくなるという欠点がある。

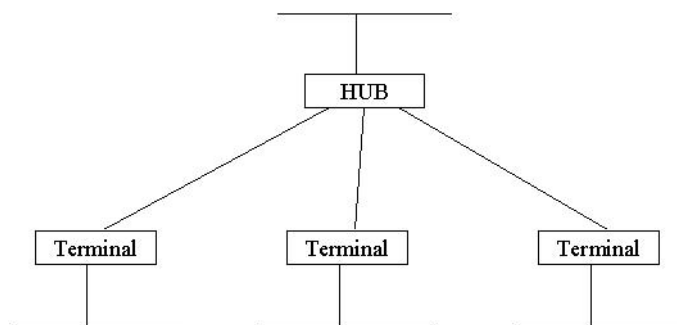


図 4.1: Point to Point リンク

4.4 単一の論理的なブロードキャストリンクに抽象化する手法

HUB の全 Terminal への接続性を生かし、HUB と各 Terminal に単一のブロードキャストリンクが存在すると仮定する手法が考えられる。

このネットワークモデルを図 4.2 に示す。

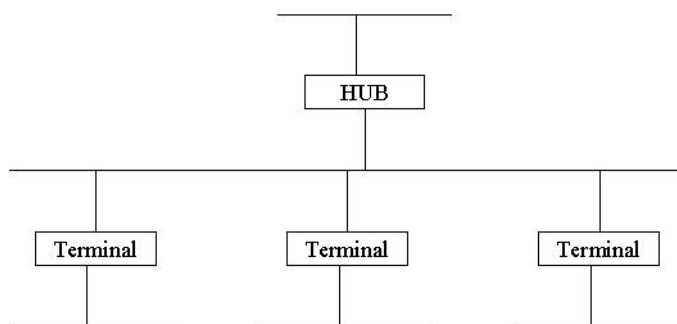


図 4.2: ブロードキャストリンク

ブロードキャストリンクでは、HUB からの同報型トラフィックの特性が生かせる。また、ブロードキャストリンクは Ethernet などの既存のインターネット通信との親和性が高い

ブロードキャストリンクとして用いる場合、DVB-RCS の構造上ターミナル同士の通信は HUB を介さない限り行えない。このため、ブロードキャストリンクとして利用する場合、Terminal 間のユニキャスト、Terminal からのマルチキャスト、ブロードキャストが動作しない。HUB、Terminal 間を一つのブロードキャストリンクとして定義するには、Terminal 間の双方向通信を仮想的に実現する必要がある。

4.5 本研究でのアプローチ

本研究でのアプローチでは、DVB-RCS のネットワークを単一のブロードキャストリンクに抽象化する手法を採用する。その理由として以下が挙げられる。

- (1) ブロードキャストリンクに抽象化させることで、HUB からの同報性を生かせる。

- (2) DVB-RCS の特殊なトポロジが通常のリンク機能を持つため、ARP や RIP、OSPF などの既存のインターネットプロトコルに改変を加えることなく正常に動作させられる。
- (3) HUB から各 Terminal に流れるデータは実際には同一のリンクを通るため、帯域の面では、リンクを分けるメリットがない。

4.6 手法の位置付け

より汎用的なシステムを構築するためには、HUB、Terminal がルータであった場合でも、ブリッジであった場合でも透過的にブロードキャストリンクを構築できる必要がある。図 4.3 に、DVB-RCS の HUB と Terminal がルータやブリッジであるトポロジを示す。

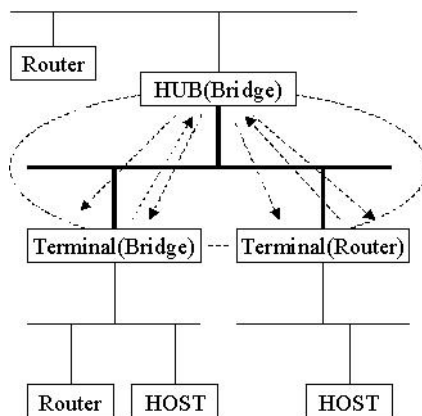


図 4.3: DVB-RCS の HUB と Terminal がルータやブリッジであるトポロジ

このようなトポロジでブロードキャストリンクを作成する場合、ネットワーク層以上でシステムを解決しようとする、ブリッジの機能やルータの機能にさらにブロードキャストリンクを作成するための機構を導入しなければならない。

HUB や Terminal がブリッジでもルータでもブロードキャストリンクを構築するためには、データリンク層においてすでに解決されている必要がある。図 4.4 に、ブロードキャストリンクに抽象化する手法を導入するべき階層の位置付けを示す。

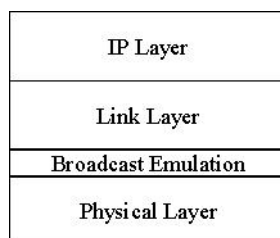


図 4.4: ブロードキャストリンクに抽象化する手法の階層の位置付け

第5章 設計

本章では、第4章で述べたモデルに従い、DVB-RCS 上での仮想ブロードキャストリンクを作成するための設計を述べる。システムを設計する上での要件を述べ、それに基づき、仮想ブロードキャストリンクを構築する機構 LED(LAN Emulation for DVB-RCS) について説明する。

5.1 設計要件

DVB-RCS のネットワーク上でブロードキャストリンクを構築するための条件について述べる。

- ・ 片方向インターフェースから双方向インターフェースへの抽象化
 - ブロードキャストリンクでは、リンク上の各ノードは、双方向通信が可能な単一のインターフェースで接続している。DVB-RCS 上のノードは、それぞれ送信インターフェースと受信インターフェースを別々に持つ。DVB-RCS をブロードキャストリンクとして用いるには、複数の片方向インターフェースを一つの双方向インターフェースに抽象化する必要がある。

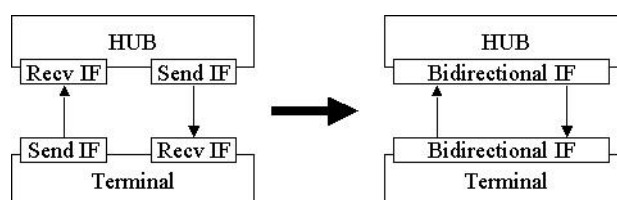


図 5.1: 一つの双方向インターフェースに抽象化

- ・ 各ノード間の直接的な通信
 - ブロードキャストリンクでは、あるノードが送信したデータは、全てのノードが直接受信できる。DVB-RCS では、HUB は全ての Terminal に直接通信を行えるが、Terminal は HUB を介さない限り他の Terminal と通信できない。したがって、DVB-RCS のネットワークをブロードキャストリンクに抽象化するには、Terminal 同士が論理的に直接通信できるようにする必要がある。

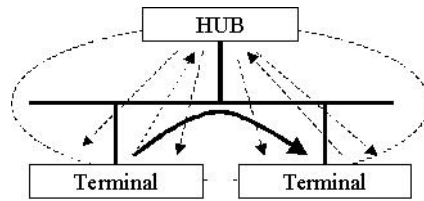


図 5.2: Terminal 同士が論理的に直接通信

5.2 LED

本研究では 5.1 の設計要件を解決するためのシステムを設計する。このシステムを LED と呼ぶ。

LED は、設計要件を解決するために次の二つの機構をもつ。

- ・ 仮想インターフェース
 - 片方向インターフェースから双方向インターフェースへの抽象化を行う機構
- ・ HUB による通信の代行機構
 - 各ノード間の直接的な通信を行う機構

LED の概念図を図 5.3 に示す。

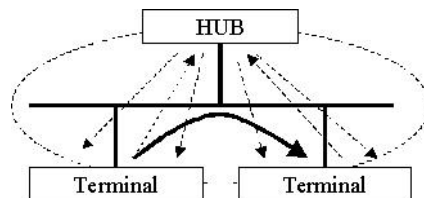


図 5.3: LED の概念図

LED を設計する上で、以下の設計条件を挙げる。

- ・ 送受信データに改変を加えない
 - 仮想インターフェースからトラフィックを送受信する際に、IP パケットやデータリンクフレームに改変を加えないものとする。改変を加えると、通信を行うノード同士が意図していないトラフィックを送受信することになり、障害のもととなるからである。
- ・ データリンク層内で解決する
 - 本来別リンクのものをブロードキャストリンクに抽象化することを考慮した場合、抽象化する機能は単一の階層で実装されている必要がある。これは、イン

ターネットの階層構造における機能の役割分担が複数に分断されることを防ぐためである。また、インターネットのデータリンクの抽象化を行うため、データリンク層の範囲内での実装を行う。データリンク層内でシステムを構築することにより、IP 層以上からは DVB-RCS のネットワークトポロジを通常のブロードキャストリンクとして認識させることができる。

5.3 仮想インターフェース

5.3.1 仮想インターフェースの概要

仮想インターフェースは、複数の片方向インターフェースを論理的に単一の双方向インターフェースに集約する機構である。仮想インターフェースを用いることで、DVB-RCS 上の HUB、Terminal を、双方向の接続性を持つ単一のインターフェースで論理的に接続しているように抽象化する。

仮想インターフェースの概念図を図 5.4 に示す。

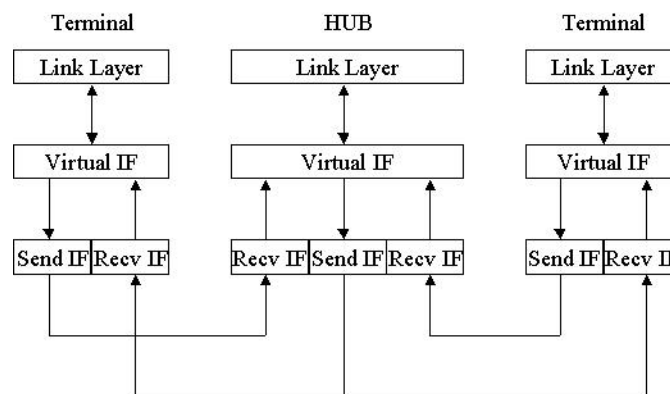


図 5.4: 仮想インターフェースの概念図

5.3.2 仮想インターフェースの設計

DVB-RCS では、Forward と Return の経路は別リンクだが、1 ホップでノード間が通信可能であるため、送信データに改変を加える必要はない。したがって、階層構造に違反せず、カプセル化の処理を必要としないインターフェースの抽象化が行える。本設計では、HUB、Terminal で送信インターフェースと受信インターフェースを一つのインターフェースに抽象化する。図 5.1 に HUB と Terminal における仮想インターフェースを用いた動作を次に示す。データリンクフレームを受信した場合、仮想インターフェースで受信を行ったようにみせる。データリンクフレームを送信する際は、仮想インターフェースより送信を行ったようにみせる。

仮想インターフェースは、独自の MAC アドレスを持つ。

5.4 HUBによる通信の代行機構

5.4.1 HUBによる通信の代行機構の概要

HUBによる通信の代行機構は、TerminalからTerminalへの通信、Terminalからリンク上へのブロードキャストを、HUBが代行することでブロードキャストリンクを構築する機構である。HUBによる通信の代行機構の概念図を図5.5に示す。

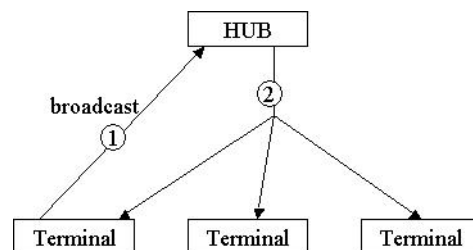


図 5.5: HUBによる通信の代行機構

5.4.2 HUBによる通信の代行機構の設計

HUBによる通信の代行機構は、データリンクフレームの送信元MACアドレスをHUBが学習し、MACアドレスのキャッシュテーブルを作成する。キャッシュテーブルの情報をを用いてTerminalからの通信を代行するかどうか決める。以下にHUBによる通信の代行機構の設計を示す。

- ・ HUBが仮想インターフェースからデータリンクフレームを受信すると、送信元MACアドレスをキャッシュテーブルに格納する。HUBは受信したパケットの宛先MACアドレスを、キャッシュテーブルから探索する。
 - (a) 宛先MACアドレスがキャッシュテーブルにあった場合
Terminal-Terminal間通信であるため、Forward Linkに同じデータもう一度転送する。
 - (b) 宛先MACアドレスがブロードキャスト、マルチキャストであった場合
Terminalからのブロードキャストであるため、受け取ったデータを仮想インターフェースに同じデータもう一度転送する。
 - (c) 宛先MACアドレスがキャッシュテーブルになかった場合、またはHUB自身のMACアドレスであった場合
パケットの宛先はブロードキャストリンクにはないため、Forward Linkへの転送を行わない。

これにより、Terminalからのブロードキャストができる。

5.4.3 HUB による通信の代行機構の動作

図 5.6 に、HUB による通信の代行機構の動作を示す。

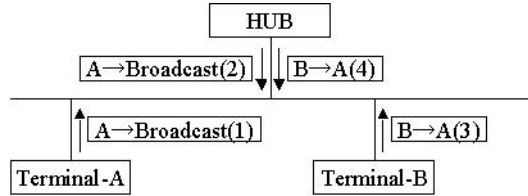


図 5.6: HUB による通信の代行機構の動作

- (1) Terminal-A がブロードキャストのデータリンクフレームを HUB に対して送信する。
- (2) Terminal-A からブロードキャストのデータリンクフレームを HUB が受信すると、HUB はデータリンクフレームの送信元 MAC アドレスをキャッシュテーブルに格納する。また、ブロードキャストなので、HUB は仮想インターフェースにデータリンクフレームを送信する。
- (3) Terminal-B が Terminal-A に送信を行う。
- (4) データリンクフレームを HUB が受信する。HUB は Terminal-A が仮想インターフェースに繋がっていることを知っているため、仮想インターフェースに同じデータリンクフレームを転送する。

5.4.4 HUB による通信の代行機構を実装した際に起きる問題

HUB による通信の代行機構を実装した場合、通信に問題が起きる可能性がある。

Terminal がブリッジだったとする。図 5.7 のノード A、ノード B が双方向通信した場合について考える。

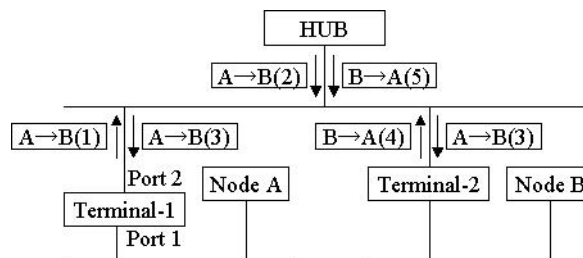


図 5.7: ノード A、ノード B の双方向通信

- (1) ノード A が宛先ノード B データリンクフレームを Terminal-1 に渡す。Terminal-1 はブリッジであるため、ノード A が port1 のリンク上にあると認識する。

- (2) HUB による通信の代行機構により、同じデータリンクフレームが Forward Link に流される。
- (3) Terminal-1 と Terminal-2 が仮想インターフェース経由で受信する。このとき、Terminal-1 は送信元がノード A のデータリンクフレームを仮想インターフェースで受信するため、キャッシュ情報が更新され、ノード A が仮想インターフェースに繋がったと認識する。
- (4) Terminal-2 からノード A 行きのデータリンクフレーム送信される。
- (5) HUB を経由して Terminal-1 に入った際、Terminal-1 は A が仮想インターフェースポートに繋がっていると認識しているため、port1 からデータを流さずに破棄してしまう。このため、ノード B からノード A の接続が確立されない。

5.4.5 LED ヘッダによる送信 Terminal の識別

HUB による通信の代行機構を実装した際に起きる問題は、Terminal が仮想インターフェースから送信したデータリンクフレームを、再び仮想インターフェースで受信することによって発生する。

LED ヘッダによる送信 Terminal の識別は、受信したデータリンクフレームの送信元を識別し、送信元が Terminal 自身ならばそのデータリンクフレームを破棄する。これにより、上記の問題を解決する。LED ヘッダによる送信 Terminal の識別は以下のように動作する。

- (1) HUB は Forward Link に送信するデータリンクフレームを、仮想インターフェースの MAC アドレスが記入されたヘッダでカプセル化する。また、Terminal は Return Link に送信するデータリンクフレームを、仮想インターフェースの MAC アドレスが記入されたヘッダでカプセル化する。このヘッダを LED ヘッダと呼ぶ。
- (2) LED の機能により、HUB が Terminal からのデータリンクフレームをもう一度 Forward Link に転送する必要があった場合、HUB は LED ヘッダを付けたまま Forward Link に転送する。この際、HUB は LED ヘッダのカプセル化を行わない。
- (3) Terminal がデータリンクフレームを受信した時、LED ヘッダを参照することで、HUB が、どの Terminal が送信したデータリンクフレームであるかを知る。
 - (a) 受信した Terminal の MAC アドレスと、LED ヘッダの MAC アドレスが異なる場合
HUB が送信したデータリンクフレームか、他 Terminal が送信したデータリンクフレームが HUB によって Forward Link に転送されたものである。この場合、LED ヘッダを取り除いたデータリンクフレームを受信する。
 - (b) 受信した Terminal の MAC アドレスと、LED ヘッダの MAC アドレスが同じ場合
Terminal 自身が送信したデータリンクフレームが、HUB によって Forward Link に転送されたものである。この場合、データリンクフレームを破棄する。

LED ヘッダによる送信 Terminal の識別を図 5.8 に示す。この図は、ノード A からのブロードキャストを、Terminal が再び受信しないことを表す。

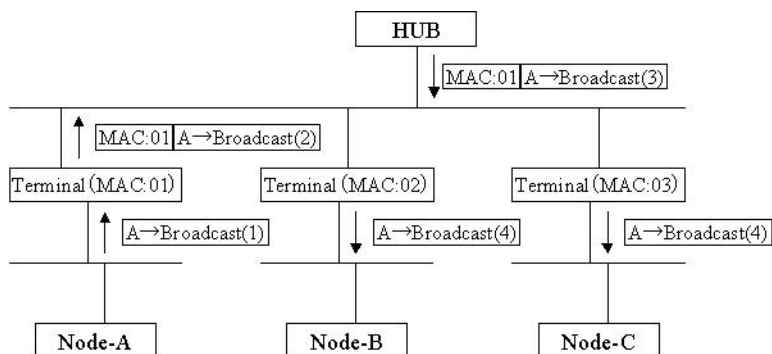


図 5.8: LED ヘッダによる送信 Terminal の識別

LED ヘッダによる送信 Terminal の識別により、Terminal 自身が送信したデータリンクフレームは、Terminal は受信しない。これにより HUB による通信の代行機構を実装した際に起きる問題を解決する。

LED ヘッダによってカプセル化されたデータリンクフレームのフォーマットを図 5.9 に示す。

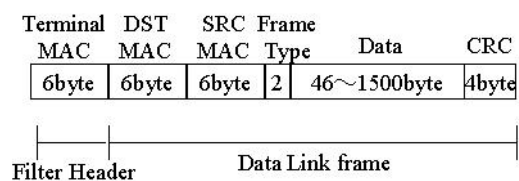


図 5.9: カプセル化されたデータリンクフレームのフォーマット

LED ヘッダによるカプセル化により、ブロードキャストリンク上のデータリンクフレームは 6byte 長くなる。

第6章 実装

6.1 実装環境

実装は、FreeBSD 4.8-RELEASE[6] 上で行った。実装では HUB と Terminal をブリッジとして作成した。

また、

6.2 HUB のデータリンクフレーム処理の実装

図 7.1 に LED を実装した HUB のデータリンクフレームの処理を行う実装概要図を示す。図中の実線は Terminal のステーションから HUB のステーションへの転送を示す。また、点線は Terminal のステーションから Terminal のステーションへの転送を示す。片方向の受信インターフェースを Recv IF、片方向の送信インターフェースを Send IF、双方向インターフェースを Normal IF とする。HUB は、HUB による通信代行機構を示す。

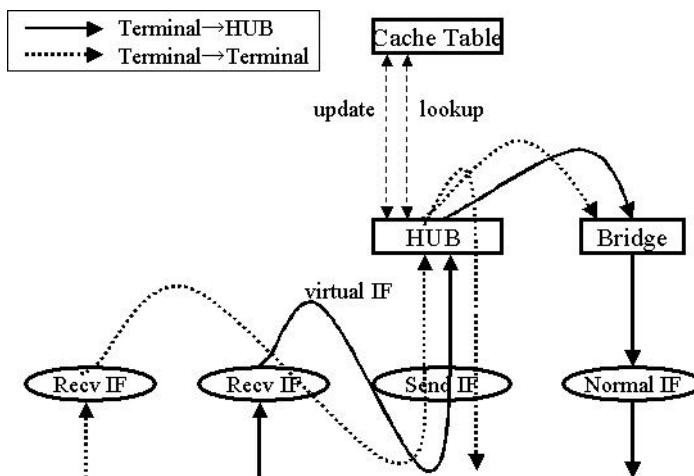


図 6.1: HUB のデータリンクフレーム処理の実装

6.2.1 仮想インターフェースの実装

仮想インターフェースは、カーネル内で実装した。

送信インターフェースを仮想インターフェースとして定義した。カーネルがインターフェースからデータリンクフレームを受け取った際に、送信用のインターフェースから受信した

ように、mbufの中身を変更した。これにより、HUBとTerminalは、送信インターフェースでパケットを送受信しているように見せた。

```
/sys/net/if_ethersubr.c

void
ether_input(struct ifnet *ifp, struct ether_header *eh, struct mbuf *m)
{
    .....
    (void)memcpy(ifname, ifp->if_name, sizeof(ifname));
    if(if_name[0]=='r' && ifp->if_unit==0){
        m->m_pkthdr.rcvif=vifp;
    }
    .....
}
```

6.2.2 HUBによる通信代行機構の実装

HUBによる通信代行機構の実装は、FreeBSDのbridgeのコードに拡張を加えて実装した。送信元が仮想インターフェースに繋がるリンクで、送信先も仮想インターフェースに繋がるリンクであった場合、もう一度転送する必要がある。

FreeBSDのブリッジには、インターフェースに繋がるノードのMACアドレスを学習する関数がある。HUBによる通信代行機構は、この関数を用いて仮想インターフェースに繋がるノードのMACアドレスを取得させた。仮想インターフェースで受け取った宛先MACアドレスが、ブロードキャスト、マルチキャスト、仮想インターフェースに繋がるノードのMACアドレスだった場合、Forward Linkに転送する処理を行わせた。

hubのbridge.cに以下の関数を新たに加えた。

```
/sys/net/bridge.c

static struct mbuf *
bdg_forward(struct mbuf *m0, struct ether_header *const eh, struct ifnet *dst)
{
    ..... /* 関数の宣言、エラー処理等 */
    ifp=TAILQ_FIRST(&ifnet);
    src=m0->m_pkthdr.rcvif;
    rcsbdg=0;
    (void)memcpy(ifname, src->if_name, sizeof(ifname));
    if(ifname[0]=='r' && ifname[1]=='c' && ifname[1]=='s' && src->if_unit>=1){
        /* 仮想IF経由で受信したか否かの判定 */
        rcsbdg=1;
    }
}
```

```

}
if(rcsbdg==1){
    /* データリンクフレームの転送先を取得 */
    dst=bridge_dst_lookup(eh,ifp2sc[real_dst->if_index].cluster);
    if(dst==BDG_MCAST||dst==BDG_MCAST||dst==src){
        /* Forward Linkへの転送 */
        if(!IF_HANDOFF(&ifp->if_snd,m,ifp)){
            BDG_MUTE(ifp);
        }
        BDG_STAT(ifp,BDG_OUT);
    }
}
}
}
}

```

6.3 Terminal のデータリンクフレーム処理の実装

図 7.1 に LED を実装した Terminal のデータリンクフレームの処理を行う実装概要図を示す。図中の実線は HUB のノードから Terminal のノードへの転送を示す。また、点線は Terminal のノードから他の Terminal のノードへの転送を示す。片方向の受信インターフェースを Recv IF、片方向の送信インターフェースを Send IF、双方向インターフェースを Normal IF とする。

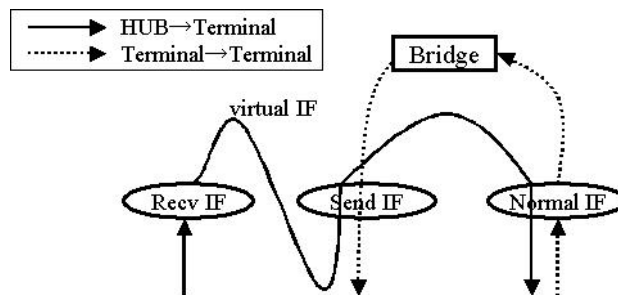


図 6.2: Terminal のデータリンクフレーム処理の実装

6.3.1 仮想インターフェースの実装

Terminal の仮想インターフェースも、HUB と同様に送信用のインターフェースから受信したように mbuf の中身を変更した。

6.3.2 LED ヘッダの実装

LED ヘッダの実装が間に合わなかったため、Terminal が仮想インターフェースで受け取ったデータリンクフレームを、学習せずにそのまま直下のセグメントに流すようにブリッジの機能を変えることで、5.4.3 で述べた問題を短期的に解決した。この場合、Terminal はブリッジとしての機能を完全に果たさなくなり、Terminal に繋がるセグメントは、自分が出したデータリンクフレームと同じものを受信することになる。

Terminal の bridge.c に以下のコードを追加した。

```
static struct mbuf *
bdg_forward(struct mbuf *m0,struct ether_header *const eh,struct ifnet *dst)
{
    ..... /* 関数の宣言、エラー処理等 */
    ifp=TAILQ_FIRST(&ifnet);
    src=m0->m_pkthdr.rcvif;
    rcsbdg=0;
    (void)memcpy(ifname,src->if_name,sizeof(ifname));
    if(ifname[0]=='r'&&ifname[1]=='c'&&ifname[1]=='s'&&src->if_unit>=1){
        /* 仮想 IF 経由で受信したか否かの判定 */
        rcsbdg=1;
    }
    if(rcsbdg==1){
        /* 無条件で Terminal のステーションに転送 */
        if(!IF_HANDOFF(&ifp->if_snd,m,ifp)){
            BDG_MUTE(ifp);
        }
        BDG_STAT(ifp,BDG_OUT);
    }else{
        /* データリンクフレームの転送先を取得 */
        dst=bridge_dst_lookup(eh,ifp2sc[ifp->if_index].cluster);
        if(dst!=src){
            /* Forward Link への転送 */
            if(!IF_HANDOFF(&ifp->if_snd,m,ifp)){
                BDG_MUTE(ifp);
            }
            BDG_STAT(ifp,BDG_OUT);
        }
    }
}
```


第7章 評価

本章では、第5章、第6章で述べた DVB-RCS 上での仮想ブロードキャストリンク機構の評価について述べる。

7.1 実験環境

図 7.1 のネットワークは DVB-RCS のネットワークをエミュレートする実験環境である。HUB と Terminal は FreeBSD 4.8-RELEASE を用いて作成した。

DVB-RCS の Forward Link にブロードキャスト、Return Link に Point-to-Point をエミュレートするため、HUB から Terminal に向けてスイッチを用いてブロードキャストリンクを作成し、Terminal から HUB に向けてはクロスケーブルを用いて接続した。HUB からクロスケーブルを用いて Terminal に送信しないように、カーネル内でデータリンクフレームをフィルタリングした。Terminal からスイッチを用いて HUB に送信しないように、カーネル内でデータリンクフレームをフィルタリングした。HUB と Terminal のインターフェース情報を表 7.1 に示す。

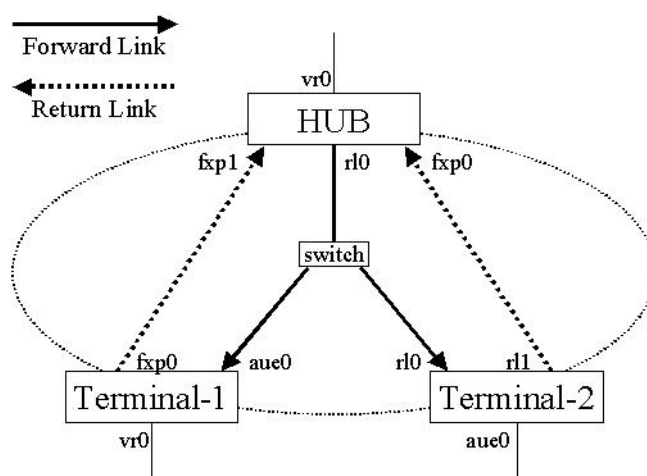


図 7.1: DVB-RCS テストベッドネットワーク・トポロジ

表 7.1: HUB、Terminal のインターフェース情報

マシン名	Normal IF	Send IF	Recv IF
HUB	vr0	rl0	fxp0、 fxp1
Terminal-1	vr0	fxp0	aue0
Terminal-2	aue0	rl1	rl0

7.2 定性的評価

本研究では、定性的評価として、本システムの設計の正当性を検証するために、仮想ブロードキャストリンク機構の動作確認を行った。本システムが既存のインターネット機構に改変を加えることなく正常に動作することを検証することを示す。本システムに求められる要求事項を以下に挙げる。

1. 仮想ブロードキャストリンク間のステーション同士の双方向通信
2. 仮想ブロードキャストリンク間のステーションからのブロードキャスト
3. 静的な別ネットワークとの経路交換
4. 動的な別ネットワークとの経路交換

本項では、1,2 をネットワーク内での基本動作確認、3,4 をネットワーク外との基本動作確認とし、それぞれネットワークトポロジを切り分けて評価を行う。まず、ネットワーク内での基本動作確認について行う。

7.3 ネットワーク内での基本動作確認

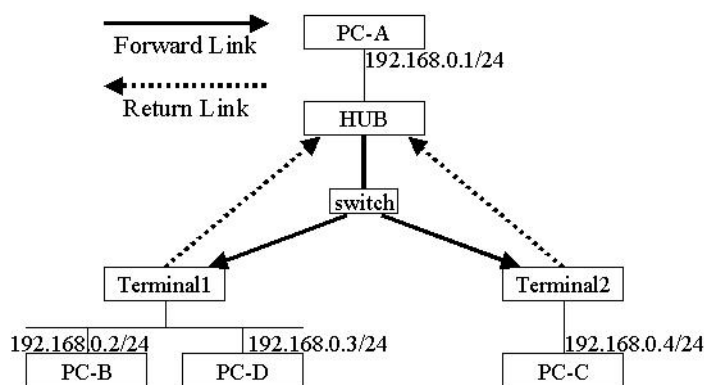


図 7.2: ネットワーク内評価トポロジ

ネットワーク内での基本動作確認を行うために、図 7.2 に示した実験ネットワークトポロジを組み立てた。PC-A、PC-B、PC-C、PC-D は、特別な設定を用いていない通常のルー

表 7.2: 各 PC の IP アドレスと MAC アドレス

マシン名	IP アドレス	MAC アドレス
PC-A	192.168.0.1/24	00:90:99:16:08:21
PC-B	192.168.0.2/24	00:90:27:a5:9a:9a
PC-C	192.168.0.3/24	00:90:99:4c:12:dd
PC-D	192.168.0.4/24	00:90:99:24:fc:05

タである。各 PC の IP アドレスと MAC アドレスを表 7.2 に示す。この実験ネットワークポロジを用いて、HUB のステーションから Terminal のステーション、Terminal のステーションから HUB のステーション、Terminal のステーションから他 Terminal へのステーションの ping プログラムを実行した。

また、ネットワーク上を流れるトラフィックの状態を取得するために、HUB と Terminal-1、Terminal-2 のカーネルにデータリンクパケットの情報を表示する簡易なプログラムを記述した。このプログラムを、データリンク情報取得プログラムと呼ぶ。データリンク情報取得プログラムが表示する情報を表 7.3 に示す。

表 7.3: データリンク情報取得プログラム

in.IF	データリンクフレームを受信したインターフェース
out.IF	データリンクフレームを転送したインターフェース
src	送信元 MAC アドレス
dst	送信先 MAC アドレス

評価は、以下の 3 パターンについて行う。

- パターン 1 Terminal のステーションから HUB のステーションへの到達性
PC-B から PC-A への Ping の動作が設計通り行われたかによって評価する。
- パターン 2 Terminal のステーションから他の Terminal のステーションへの到達性
PC-B から PC-C への Ping の動作が設計通り行われたかによって評価する。
- パターン 3 Terminal のステーション同士の到達性
PC-B から PC-D への Ping の動作が設計通り行われたかによって評価する。

7.3.1 Terminal のステーションから HUB のステーションへの到達性

PC-B から PC-A への Ping が正常に動作するかを確認し、Terminal のステーションから HUB のステーションへのデータリンクフレームの到達性が設計通りかどうかの検証を行う。この際、PC-B からの ARP request が HUB によって Forward Link に転送され、PC-C もこの ARP request を受信できたかどうかの検証も行う。

PC-B から PC-A の ping の結果を表 7.4 に示す。

表 7.4: PC-B から PC-A への ping

```
PING 192.168.0.1 (192.168.0.1): 56 data bytes
64 bytes from 192.168.0.1: icmp_seq=0 ttl=64 time=8.246 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=3.309 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=3.079 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=2.878 ms
```

PC-B から PC-A への ping は正常に動作し、Terminal のステーションから HUB のステーションへのデータリンクフレームの到達性が確認できた。

この時の HUB のデータリンク情報取得プログラムを表 7.5 に示す。

表 7.5: HUB のデータリンク情報取得プログラム

```
/* PC-B からの arp request のブロードキャスト */
in.IF> fxp1->r10 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* virtual IF */
out.IF> vr0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff
out.IF> r10 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* HUB */
/* PC-A からの arp reply */
in.IF> vr0 <src>00:02:b3:27:5c:97 <dst>00:90:27:a5:9a:9a
out.IF> r10 <src>00:02:b3:27:5c:97 <dst>00:90:27:a5:9a:9a
/* PC-B PC-A の ICMP トラフィック */
in.IF> fxp1->r10 <src>00:90:27:a5:9a:9a <dst>00:02:b3:27:5c:97 /* virtual IF */
out.IF> vr0 <src>00:90:27:a5:9a:9a <dst>00:02:b3:27:5c:97
/* PC-A PC-B の ICMP Echo Reply */
in.IF> vr0 <src>00:02:b3:27:5c:97 <dst>00:90:27:a5:9a:9a
out.IF> r10 <src>00:02:b3:27:5c:97 <dst>00:90:27:a5:9a:9a
/* PC-B PC-A の ICMP トラフィック */
in.IF> fxp1->r10 <src>00:90:27:a5:9a:9a <dst>00:02:b3:27:5c:97 /* virtual IF */
out.IF> vr0 <src>00:90:27:a5:9a:9a <dst>00:02:b3:27:5c:97
/* PC-A PC-B の ICMP Echo Reply */
in.IF> vr0 <src>00:02:b3:27:5c:97 <dst>00:90:27:a5:9a:9a
out.IF> r10 <src>00:02:b3:27:5c:97 <dst>00:90:27:a5:9a:9a
```

PC-B から PC-A に ping プログラムを実行した場合、HUB が PC-B からのブロードキャストを Forward Link と自分の直下のセグメントに転送していることが確認できた。また、仮想インターフェースが複数の片方向インターフェースを一つの双方向インターフェースに抽象化しているため、PC-B と PC-A は正常に双方向に直接通信できることが確認できた。また、この時の Terminal-2 のデータリンク情報取得プログラムを表 7.6 以下に示す。

表 7.6: Terminal-2 のデータリンク情報取得プログラム

```
/* PC-B からの arp request のブロードキャスト */
in.IF> r10->r11 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* virtual IF */
out.IF> aue0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff
/* PC-A PC-B の ICMP Echo Reply */
in.IF> r10->r11 <src>00:02:b3:27:5c:97 <dst>00:90:27:a5:9a:9a /* virtual IF */
out.IF> r10 <src>00:02:b3:27:5c:97 <dst>00:90:27:a5:9a:9a
in.IF> r10->r11 <src>00:02:b3:27:5c:97 <dst>00:90:27:a5:9a:9a /* virtual IF */
out.IF> r10 <src>00:02:b3:27:5c:97 <dst>00:90:27:a5:9a:9a
```

HUB が PC-B からのブロードキャストリンクフレームを Forward Link に転送しているため、Terminal-2 は PC-B の ARP Request を受信できたことが確認できた。また、Forward Link は同報型の回線なので、HUB から Terminal-1 へのデータリンクフレームは Terminal-2 も受信している。以上の結果から、Terminal のステーションから HUB のステーションへの双方向通信が設計通りに行われたことが確認できた。

7.3.2 Terminal のステーションから他の Terminal のステーションへの到達性

PC-B から PC-C への Ping が正常に動作するかを確認し、Terminal のステーションから他の Terminal のステーションへのデータリンクフレームの到達性が設計通りかどうかの検証を行う。Terminal-Terminal 間通信では、5.3.2 で述べたように、HUB が仮想インターフェースで受信したデータリンクフレームの宛先を判断して Forward Link に送信する。これが正常に行われたかどうかを確認する。

PC-B から PC-C の ping の結果を表 7.7 に示す。

表 7.7: PC-B から PC-C への ping

PING 192.168.0.3 (192.168.0.3): 56 data bytes
64 bytes from 192.168.0.3: icmp_seq=0 ttl=64 time=10.235 ms
64 bytes from 192.168.0.3: icmp_seq=1 ttl=64 time=4.993 ms
64 bytes from 192.168.0.3: icmp_seq=2 ttl=64 time=4.784 ms
64 bytes from 192.168.0.3: icmp_seq=3 ttl=64 time=4.571 ms

PC-B から PC-C への ping は正常に動作し、Terminal のステーションから他の Terminal のステーションへのデータリンクフレームの到達性が確認できた。

この時の HUB のデータリンク情報取得プログラムを表 7.8 に示す。

表 7.8: HUB のデータリンク情報取得プログラム

```
/* PC-B からの arp request のブロードキャスト */
in.IF> fxp1->r10 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* virtual IF */
out.IF> vr0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff
out.IF> r10 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* hub */
/* PC-A からの arp reply */
in.IF> fxp0->r10 <src>00:90:99:1a:be:d9 <dst>00:90:27:a5:9a:9a /* virtual IF */
out.IF> r10 <src>00:90:99:1a:be:d9 <dst>00:90:27:a5:9a:9a /* rcs bridge */
/* PC-B PC-C の ICMP トラフィック */
in.IF> fxp1->r10 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9 /* virtual IF */
out.IF> r10 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9 /* rcs bridge */
/* PC-C PC-B の ICMP Echo Reply */
in.IF> fxp0->r10 <src> 00:90:99:1a:be:d9 <src> 00:90:27:a5:9a:9a /* virtual IF */
out.IF> r10 <src> 00:90:99:1a:be:d9 <src> 00:90:27:a5:9a:9a /* rcs bridge */
/* PC-B PC-C の ICMP トラフィック */
in.IF> fxp1->r10 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9 /* virtual IF */
out.IF> r10 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9 /* rcs bridge */
/* PC-C PC-B の ICMP Echo Reply */
in.IF> fxp0->r10 <src> 00:90:99:1a:be:d9 <src> 00:90:27:a5:9a:9a /* virtual IF */
out.IF> r10 <src> 00:90:99:1a:be:d9 <src> 00:90:27:a5:9a:9a /* rcs bridge */
```

PC-B から PC-C に ping プログラムを実行した場合、HUB が PC-B からの ARP request のブロードキャストを Forward Link に転送しているため、PC-C は PC-B からの ARP request の受信と ARP reply の送信が可能となった。また、PC-B と PC-C が双方向に通信する際に HUB が仮想インターフェースで受け取ったデータリンクフレームを破棄せずにもういちど転送しているため、HUB を中継して正常に通信が行えた。これにより、5.3.2 の要件を満たしている事が確認できた。

また、この時の Terminal-2 のデータリンク情報取得プログラムを表 7.9 に示す。

表 7.9: Terminal-2 のデータリンク情報取得プログラム

```
/* PC-B からの arp request のブロードキャスト */
in.IF> vr0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* virtual IF */
out.IF> fxp0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff
/* PC-B からの arp request のブロードキャストが HUB から Forward Link に転送 */
in.IF> aue0->fxp0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* virtual IF */
out.IF> vr0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff
/* PC-C からの arp reply */
in.IF> fxp0->r10 <src>00:90:99:1a:be:d9 <dst>00:90:27:a5:9a:9a /* virtual IF */
out.IF> vr0 <src>00:90:99:1a:be:d9 <dst>00:90:27:a5:9a:9a
/* PC-B PC-C の ICMP トラフィック */
in.IF> vr0 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9
out.IF> fxp0 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9
/* PC-B PC-C の ICMP トラフィックが HUB から Forward Link に転送 */
in.IF> aue0->fxp0 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9 /* from HUB */
out.IF> vr0 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9 /* from HUB */
/* PC-C PC-B の ICMP Echo Reply */
in.IF> aue0->fxp0 <src> 00:90:99:1a:be:d9 <src> 00:90:27:a5:9a:9a /* virtual IF */
out.IF> vr0 <src> 00:90:99:1a:be:d9 <src> 00:90:27:a5:9a:9a /* RCS Bridge T */
/* PC-B PC-C の ICMP トラフィック */
in.IF> vr0 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9
out.IF> fxp0 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9
/* PC-B PC-C の ICMP トラフィックが HUB から Forward Link に転送 */
in.IF> aue0->fxp0 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9 /* from HUB */
out.IF> vr0 <src> 00:90:27:a5:9a:9a <src> 00:90:99:1a:be:d9 /* from HUB */
/* PC-C PC-B の ICMP Echo Reply */
in.IF> aue0->fxp0 <src> 00:90:99:1a:be:d9 <src> 00:90:27:a5:9a:9a /* virtual IF */
out.IF> vr0 <src> 00:90:99:1a:be:d9 <src> 00:90:27:a5:9a:9a /* RCS Bridge T */
```

HUB による通信の代行機構の機能により、Terminal-1 が Return Link に送信したデータリンクフレームは Forward Link に流される。このため、Terminal-1 は送信した状態と全く同じデータリンクフレームを仮想インターフェースで受信する。仮想インターフェース経由で受信したデータリンクフレームは、ステーションキャッシュを更新せずにそのまま通常のインターフェースに送信しているため、5.3.3 で述べた Terminal を通常の学習ブリッジとした場合に起こるネットワークの問題が起らず、PC-B と PC-C は正常に通信を行えた。以上の結果から、Terminal のステーションから他 Terminal のステーションへの双方向通信が設計通りに行われたことが確認できた。

7.3.3 Terminal のステーション同士の到達性

PC-B から PC-D への Ping が正常に動作するかを確認し、Terminal のステーション同士のデータリンクフレームの到達性が設計通りかどうかの検証を行う。Terminal のステーション同士の通信の場合、Terminal はブロードキャストを除き Return Link への送信を行わない。これが正常に行われたかどうかを確認する。

PC-B から PC-C の ping の結果を表 7.10 に示す。

表 7.10: PC-B から PC-D への ping

```
PING 192.168.0.4 (192.168.0.3): 56 data bytes
64 bytes from 192.168.0.4: icmp_seq=0 ttl=64 time=0.491 ms
64 bytes from 192.168.0.4: icmp_seq=1 ttl=64 time=0.176 ms
64 bytes from 192.168.0.4: icmp_seq=2 ttl=64 time=0.175 ms
64 bytes from 192.168.0.4: icmp_seq=3 ttl=64 time=0.172 ms
```

PC-B から PC-D への ping は正常に動作し、Terminal のステーション同士のデータリンクフレームの到達性が確認できた。

この時の Terminal-1 のデータリンク情報取得プログラムを表 7.11 に示す。

表 7.11: Terminal-1 のデータリンク情報取得プログラム

```
/* PC-B からの arp request のブロードキャスト */
in.IF> vr0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* virtual IF */
out.IF> fxp0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff
/* PC-D からの arp reply */
in.IF> vr0 <src>00:90:99:24:fc:05 <dst>00:90:27:a5:9a:9a /* virtual IF */
/* PC-B からの arp request のブロードキャストが HUB から Forward Link に転送 */
in.IF> aue0->fxp0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* virtual IF */
out.IF> r10 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff
/* PC-B PC-D の ICMP トラフィック */
in.IF> vr0 <src>00:90:27:a5:9a:9a <dst>00:90:99:24:fc:05 /* virtual IF */
/* PC-D PC-B の ICMP トラフィック */
in.IF> vr0 <src>00:90:99:24:fc:05 <dst>00:90:27:a5:9a:9a /* virtual IF */
```

PC-B から PC-D への送信では、PC-B からの最初の ARP 要求によって、Terminal-1 が PC-B が直下のセグメントにいることを学習するため、PC-D からの APR reply を Forward Link に転送しなくなった。また、PC-D から PC-B にデータを送信する際に、PC-D も直下のセグメントにいることを学習するため、PC-B と PC-D が双方向通信する際のトラフィックを HUB に対して送信していない。

HUB のデータリンク取得情報プログラムは表 7.12 のようになる。

表 7.12: HUB のデータリンク情報取得プログラム

```
/* PC-B からの arp request のブロードキャスト */
in.IF> fxp0->r10 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* virtual IF */—
out.IF> vr0 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff
out.IF> r10 <src>00:90:27:a5:9a:9a <dst>ff:ff:ff:ff:ff:ff /* rcs bridge */
```

HUB では、PC-B からの ARP Request のブロードキャスト以外を受信、転送していない。これにより、5.3.2 の要件を満たしている事が確認できた。

PC-B からの PC-A、PC-C、PC-D との双方向通信とブロードキャストが、設計で述べたものに従って正常に動作した。以上より、ネットワーク内での基本動作が正常に動作することを確認できた。

実験後の PC-B の ARP Table の内容を表 7.13 に示す。

表 7.13: 実験後の PC-B の ARP Table

```
% arp -a
PC-A (192.168.0.1) at 00:90:99:16:08:21 on fxp0 [ethernet]
PC-C (192.168.0.3) at 00:90:99:4c:12:dd on fxp0 [ethernet]
PC-D (192.168.0.4) at 00:90:99:24:fc:05 on fxp0 [ethernet]
```

7.3.4 ネットワーク外との基本動作確認

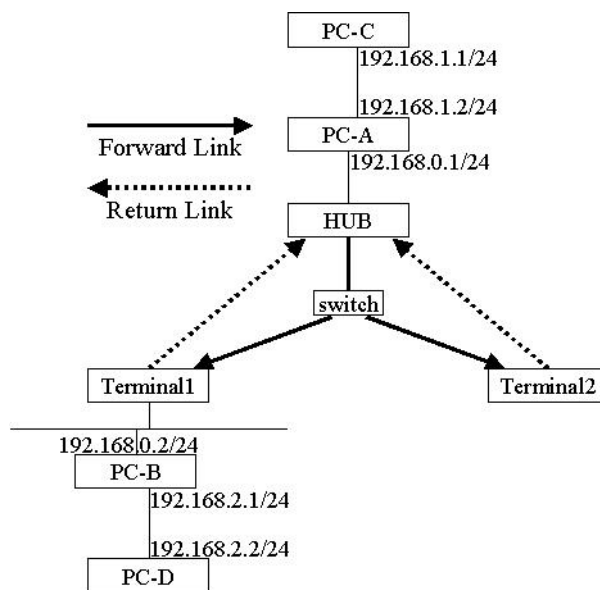


図 7.3: ネットワーク外評価トポロジ

ネットワーク外との基本動作確認を行うために、ネットワークトポロジを図 7.3 に示したように置き換えた。HUB 以下に作られたネットワークと Terminal 以下に作られたネットワークが、DVB-RCS を介して静的または動的に経路交換をできるかどうかについて実験を行った。

動的な経路制御には routed コマンドを用いた。

表 7.14: 各 PC の IP アドレス

マシン名	IP アドレス	IP アドレス
PC-A	fxp0:192.168.1.1/24	
PC-B	fxp0:192.168.1.2/24	vr0 192.168.0.1/24
PC-C	rl0:192.168.0.2/24	vr0 192.168.2.1/24
PC-D	fxp0:192.168.2.2/24	

各 PC に、表 7.15 のように静的に経路情報を記述した。

表 7.15: HUB、Terminal のインターフェース情報

マシン名	Destination	Gateway
PC-A	default	192.168.1.2
PC-B	192.168.2.0/24	192.168.0.2
PC-C	192.168.1.0/24	192.168.0.1
PC-D	default	192.168.2.1

静的に経路情報を各 PC に設定した場合、各 PC 同士は双方向に正常に通信を行うことができた。これは、DVB-RCS のネットワークをデータリンク層でブロードキャストリンクに抽象化しているため、ネットワーク層での通信に影響が出さなかったことが確認できた。

また、静的な経路情報を削除した後、各 PC に `routed` コマンドを用いて RIP を動作させた場合においても、経路情報交換後に各 PC 同士は正常に通信を行えた。

以上の結果から、データリンク層内でブロードキャストリンクを構築することに成功し、ネットワーク内での通信の基本動作と、ネットワーク外との通信の基本動作が正常に動作することが確認できた。これにより、本システムは表 7.16 に示す要求事項をすべて満たした。

表 7.16: 要求事項

要求事項	解決の有無
Terminal 同士の双方向通信	
Terminal からのブロードキャスト	
データリンク層内での解決	

第8章 結論

8.1 まとめ

本研究では、DVB-RCS をインターネットで利用するための機構を設計し、実装した。

DVB-RCS は、一つの HUB と複数の Terminal からなるトポロジ上で、片方向リンクにより構成された。このため、DVB-RCS を含むネットワーク上では、既存のインターネット技術が正常に動作せず、データリンクアドレス解決や経路制御など通信の基盤となる技術が使えなかった。

本研究では、DVB-RCS を、一つのブロードキャストリンクとして抽象化した。一つのブロードキャストリンクとして扱うためには、HUB と Terminal 間が片方向リンクで接続しているため双方向に通信が行えない問題があった。また、Terminal は HUB にしか接続性を持たないため、Terminal 同士の通信、Terminal からのブロードキャストが行えないという問題があった。この二つの問題を解決する必要があった。

本研究では、DVB-RCS を含むネットワークをブロードキャストリンクに構築する機構として、LED を設計した。LED は仮想ブロードキャストリンクと、HUB による通信の代行機構で構成された。LED を導入したシステムの設計と実装を行った。

また、DVB-RCS を含むネットワークをエミュレートしたテストベットを構築し、本システムの評価を行った。その結果、既存のインターネット技術が正常に動作し、本システムが有効であることを検証した。

本システムにより、DVB-RCS を既存の回線と同じようにインターネットに透過的に導入できた。

8.2 今後の課題

今後の課題として、以下の点が挙げられる。

- ・ 規模性

多くの Terminal が HUB に繋がった場合の評価を行えない。このため、ネットワークシミュレーターを用いたシミュレーションを行い、多数の Terminal が繋がる DVB-RCS のネットワークを仮想的に構築し、Terminal が増えた場合でも正常に動作するかどうかの評価を行う必要がある。

- ・ 実装

Terminal の認証機構の構築に間に合わなかったため、Terminal の認証機構を実装し、本機構が正常に動いたことを示す。

- ・ 実用性

通信モデルとエミュレーション環境での動作検証の結果をもとに、実際に DVB-RCS の HUB と Terminal を作成し、実用にフィードバックする。この際、実物の HUB や Terminal には、インターネットの通信を行うだけでなく衛星モデムとしての機能や 2 章で述べた DVB-RCS が本来持つデータリンクの機能を導入する必要がある。今後、J-SAT や DVB Working Group[dvbwork] などの共同研究機関と協力して開発を行い、実用化を進めていく。

謝辞

本研究を進めるにあたり、ご指導をいただきました慶應義塾大学環境情報学部教授の徳田英幸博士と村井純博士に感謝いたします。また、絶えず御助言と御指導をいただきました、同学部の中村修博士と楠本博之博士に深い感謝の意を表します。西田 視磨氏、藤枝 俊輔氏をはじめとする慶應義塾大学政策メディア研究科の方々には、貴重な御助言をいただきました。また、本論文の執筆に当たりまして常に励ましと御協力を下さった慶應義塾大学環境情報学部徳田・村井・楠本・中村研究会の諸兄に感謝いたします。

参考文献

- [1] ETSI Telecom Standards, "ETSI EN 301 790 v1.3.1" European Standard March 2003
- [2] RFC3077:A Link Layer Tunneling Mechanism for Unidirectional Links
- [3] ETSI Telecom Standards ホームページ, <http://www.etsi.org>
- [4] 竹井 淳, "通信衛星を用いたグローバルネットワークにおける識別子と経路制御に関する研究" MAUI Project 博士論文 March 2002
- [5] EMS Satellite Networks ホームページ, <http://emssatnet.com/>
- [6] The FreeBSD Project, <http://www.freebsd.org/>
- [7] DVB Working Project ホームページ, <http://www.dvb.org/>