

## 第9部

# 衛星通信によるネットワーク構築実験



## 第1章 概要

本年度、WISH では Open Shortest Path First(OSPF) を拡張し、片方向の衛星回線を含むネットワークにおいて動作する改変を行なった。

インターネットにおいて片方向の衛星回線を含むネットワークを構築する場合、動的な経路制御プロトコルが正しく動作しないことが問題となる。その原因は、既存の経路制御プロトコルが通信路の双方向性を前提としているため、片方向の通信路を通信に利用できる状態と判断しないからである。

この問題の解決法として仮想ブロードキャストリンクを用いた手法が提案されている。この手法では、衛星回線の受信局から送信局へパケットをトンネリングし、受信局による衛星回線への送信を送信局に代行させることで、衛星回線上に仮想的なブロードキャストリンクを構築する。これにより、既存の経路制御プロトコルをそのまま衛星回線に適用できる。

しかし、衛星回線には少数の送信局と非常に多くの受信局が接続することが一般的なのに対し、既存の経路制御技術はこのようなネットワークトポロジを想定していない。そのため、多数の受信局が衛星回線に接続した場合、既存の経路制御プロトコルでは効率や性能が問題になる。

あらゆる規模の受信局が接続する衛星回線をインターネットにおいて利用するためには、衛星回線を含むネットワークトポロジに適応した経路制御プロトコルが必要である。

評価として、本研究で改変した OSPF と仮想ブロードキャストリンク機構を利用した既存の OSPF を、テストベッド上で動作させた。その結果、仮想ブロードキャストリンクを用いて OSPF を動作させた場合、いくつかのネットワークトポロジでは正しく経路制御を行なえないと分かった。また、本研究で改変した OSPF が、片方向のリンクを含むあらゆるネットワークトポロジで動作すると分かった。

本研究により、片方向のリンクを含むネットワークにおいて正しく経路制御を行なうルーティングプロトコルが構築された。これにより、従来の手法が持つ問題点を解決した。今後、あらゆる規模の衛星回線をトポロジの制限なくインターネットの一部として利用できるようになる。

## 第2章 UDL を含むネットワークで動作する OSPF の設計

### 2.1 用語とトポロジ

#### 2.1.1 UDL

衛星回線は、通常の通信回線にはない様々な特性を持つ。それらの特性のうち経路制御上で問題になるのは、衛星回線を片方向に用いた場合の単一方向性である。本稿では衛星回線を片方向リンク (UDL) として抽象化し議論する。また、衛星回線を UDL と呼ぶのに対し、双方向性を持つ通常の通信回線を BDL(Bi-Directional-Link) と呼ぶ。UDL には以下に示す 2 種類のノードが接続する。

**Feeder:** UDL に対し送受信を行なうノード

**Receiver:** UDL を受信専用で用いるノード

衛星回線において、同時に送信を行なえる Feeder は 1 つである。また、送信の衝突を検出する機構が考えられていないため、1 つの衛星回線に存在する Feeder は一般的に 1 つ (または少数) である。これに対し、衛星回線から同時にデータを受信するノードの数には制限がなく、あらゆる規模の Receiver が同一の回線上に接続すると考えられる。したがって、本稿で指す UDL には、“1 つ (または少数) の Feeder と、あらゆる規模の Receiver が接続する” と定義する。

また、衛星回線の物理層において Receiver は回線に流れる全てのデータを受信する。そのため、各ノードのインタフェースにデータリンクアドレスを付与することで、データリンクによるユニキャスト、マルチキャスト、ブロードキャストが実現される。また、インタフェースにアドレスを設けず、Receiver が全てのパケットを受信し、IP 層による宛先アドレスの判別によってこれらの送信形態を実現することもできる。本稿では“Feeder は UDL を用いてユニキャスト、マルチキャスト、ブロードキャストを行なえる” と定義する。

#### 2.1.2 Feeder、Receiver における通信環境

衛星回線をインターネット上で用いる場合、

Feeder、及び Receiver は他のネットワークと通信するために BDL を併用する。Receiver はデータの送信を行なうため BDL を用いる必要がある。また Feeder は、他に Feeder が存在する場合を除き衛星回線からデータを受信することがない。そのため、Feeder が1つしか存在しないトポロジでは受信用に BDL が必要である。また、Feeder が複数存在するトポロジにおいても、少なくとも1つの Feeder が BDL を持つ必要がある。本論文では、UDL における Feeder、及び Receiver は必ず1つ以上の BDL を持つと定義する。本節で述べた通信環境を図 2.1 に示す。

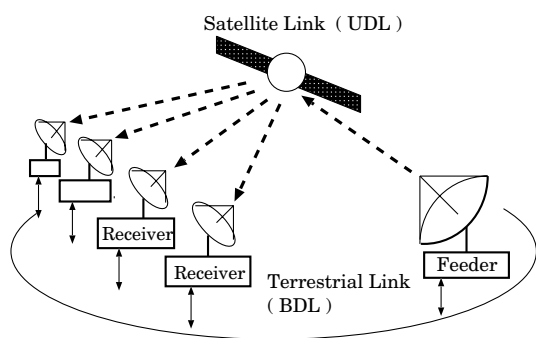


図 2.1 衛星回線を含むネットワークにおける通信環境

## 2.2 設計概要

Feeder と Receiver が UDL による接続性を検知するため、UDL Hello プロトコルを用いる。UDL Hello は Hello プロトコルを片方向の通信路上で動作するように変更したもので、Receiver は BDL を用いて Hello Packet を Feeder に送信する。

また、OSPF はマルチアクセスネットワークにおいてリンク情報の増加を防ぐため、各ルータ間の接続性をネットワークとルータとのリンクによって表現する。本設計もこれと同様に、UDL を用いた Feeder と Receiver の接続性を各ノードとネットワークとのリンクによって表現する。ネットワーク LSA を生成するため、UDL 上において DR と BDR を選出する。DR、BDR に選出されるのは Feeder だけである。Feeder が複数存在する場合、priority が高い Feeder が優先される。DR が生成する Network LSA は、Attached Router フィールドに全ての Feeder と全ての Receiver のルータ ID を含む。

表 2.1 受信だけのリンク情報

Link Type	そのリンクの種類
1	ポイントツーポイントのリンク
2	通過できるマルチアクセスネットワークへのリンク
3	スタブネットワークへのリンク
4	仮想リンク
5	受信だけを行なえるリンク

UDL には多数の Receiver が接続するため、全てのノード間、または DR と全てのノード間でデータベースを同期させることが困難である。本設計では、DR と Feeder の間、または DR と Receiver のうち指定されたルータ (以下 Receiver\_Type1) の間でだけデータベースを同期させる。UDL に接続するルータの種類を次に示す。

- DR、BDR に選出された Feeder
- DR、BDR に選出されなかった Feeder
- DR とデータベースを同期させる Receiver (Receiver\_Type1)
- DR とデータベースを同期させない Receiver (以下 Receiver\_Type2)

Receiver\_Type2 が DR とデータベースを同期させない理由は、UDL に接続するノードは BDL を用いてエリア内の他のルータとデータベースを同期させる可能性が高いからである。この場合、DR と Receiver は直接データベースを同期させることなく同じデータベースを保持できる。これに当てはまらないネットワークトポロジでは、Receiver\_Type1 を用いる。

Feeder は、Router LSA において、UDL への接続性を通過できるマルチアクセスネットワークへのリンク (Link Type 2) として広告する。Receiver は、Router LSA において、UDL における受信だけの接続性を広告するため新しいリンク情報を用いる。これを表 2.1 に示す。

link information における各フィールドの値を表 2.2 に示す。

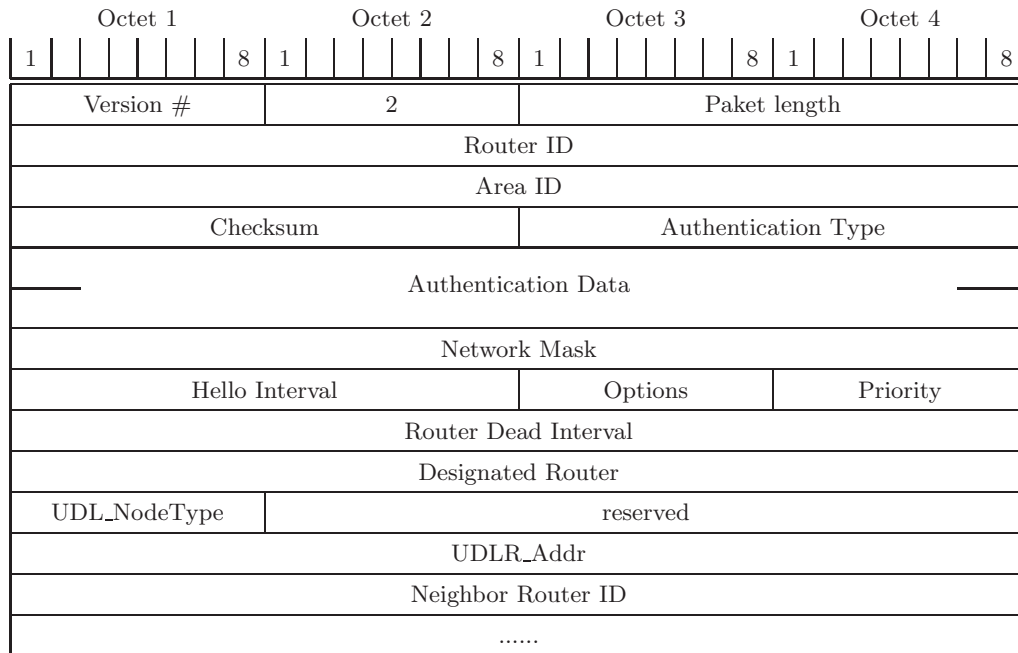


図 2.2 UDL\_Hello Packet のフォーマット

表 2.2 Link ID, Link Data の値

フィールド	値
Link ID	DR の IP アドレス
Link Data	Receiver が持つ UDL インタフェースの IP アドレス

表 2.3 UDL\_NodeType と UDLR\_Addr の値

ノードの種類	UDL_Node Type	UDLR_Addr
Feeder	1	Feeder の BDL インタフェースの IP アドレス
Receiver	2	Receiver の UDL インタフェースの IP アドレス

### 2.3 UDL\_Hello プロトコル

Feeder から Receiver への到達性を検知するため、UDL\_Hello プロトコルを設計した。UDL\_Hello パケット (以下 UDL\_Hello) のフォーマットを図 2.2 に示す。

網かけ部分が既存の Hello パケットに付加したフィールドである。

UDL\_NodeType パケットを送信したノードの種類。

UDLR\_Addr ノードの種類によって値が異なる。

UDL\_NodeType と UDLR\_Addr の値を表 2.3 に示す。

Feeder 間において、UDL\_Hello プロトコルは既存の Hello プロトコルと同じ動作をする。Feeder と Receiver 間における動作を次に示す。

1. Feeder は、一定時間おきに OSPF\_ALLSPFRouters マルチキャストアドレス宛に UDL\_Hello を送信する。
2. Receiver は、UDL\_Hello を受信し、UDLR\_Addr フィールドから Feeder の BDL インタフェースの IP アドレスを知る。
3. Receiver は、UDLR\_Addr フィールドに示された IP アドレス宛にユニキャストで UDL\_Hello を送信する。この時、Neighbor Router ID フィールドに過去一定時間に UDL\_Hello を受けた先の Feeder の Router ID を入れる。
4. Feeder は、Receiver から UDL\_Hello を受信し UDLR\_Addr フィールドから Receiver の UDL ネットワーク上の IP アドレスを知る。この時、Neighbor Router ID フィールドに自分のルー

タ ID を発見すると、UDL において Feeder から Receiver へ到達性があると認識する。

5. Feeder は、次に UDL\_Hello を送信する時、Neighbor RouterID フィールドに Receiver のルータ ID を入れる。
6. Receiver は受けとった UDL\_Hello に自分のルータ ID を発見すると、Receiver から Feeder に対し BDL を用いた接続性があると認識する。

## 2.4 LSA の広告

OSPF ルータは各隣接ルータとの関係をイベントによって遷移させる。これを、図 2.3 に示す。

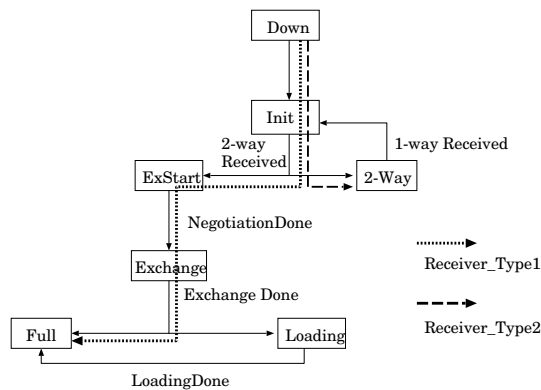


図 2.3 隣接ルータの状態遷移

### Receiver\_Type1

Receiver\_Type1 は、Feeder と UDL\_Hello を交換し、UDL と BDL を併用した双方向の到達性を確認すると 2-Way の状態に進む。Feeder が DR,BDR の場合、トポロジデータベースを同期させ Full の状態に進む。DR と Full の状態になると、Router LSA に“受信だけを行なえるリンク情報”(Link Type 5) を含め、Link State Update パケットによって隣接ルータに広告する。

OSPF では、トポロジデータベースを同期させたり新しい LSA を広告するため、次の 4 種類のパケットを用いる。

- Database Description Packet
- Link State Request Packet
- Link State Update Packet

- Link State Acknowledgement Packet

Receiver\_Type1 は次の方法でこれらのパケットを送信する。

- 宛先が DR(BDR) へのユニキャストである場合  
BD(BDR) の BDL インタフェースの IP アドレス宛に送信する。
- 宛先が OSPF\_ALLDRouters マルチキャストアドレス (224.0.0.6) である場合  
DR の BDL インタフェースの IP アドレス宛に送信する。またパケットの複製を BDR の BDL インタフェースの IP アドレス宛に送信する。

### Receiver\_Type2

Receiver\_Type2 は、Feeder と UDL\_Hello を交換し 2-Way の状態に進む。Feeder が DR,BDR の場合もこの先の状態へは進まない。UDL\_Hello によって DR の IP アドレスを得ると、Router LSA に“受信だけを行なえるリンク情報”(Link Type 5) を含め、BDL 上の隣接ルータに広告する。Feeder と Receiver がエリア内で双方向の接続性があるトポロジでは、Receiver\_Type2 の Router LSA は flooding され各ルータを経由して Feeder に届く。

### DR

DR は、UDL\_Hello によって隣接ルータの種類を知る。隣接ルータが Feeder,Receiver\_Type1 の場合、これらとトポロジデータベースを同期させ、Full の状態に進む。Receiver\_Type2 の場合、2-Way 以降の状態へは進まない。

DR が Network LSA を生成する際、Attached Router フィールドには以下のルータ ID を含める。

- Full の状態にある Feeder,Receiver\_Type1
- 2-Way の状態にある Receiver\_Type2

## 2.5 最短経路の計算

UDL を含むエリア内の全てのルータにおいて、Shortest Path Tree を作成する機構に次の機能を追加する。

- 木構造に追加した Router LSA のリンク情報を調べる時、受信のみのリンク情報は無視する。
  - Receiver を経由して UDL に送信されるパスが計算されるのを防ぐ。
- 木構造に Network LSA を追加し、Attached Router フィールドからそのネットワークを経由したリンクをたどる場合、リンク先の Router LSA がそのネットワークに対し“通過できるマルチアクセスネットワークへのリンク”(Link Type 2)が“受信だけを行なえるリンク)のリンク情報”(Link Type 5)を持っていることを確認する。持っていない場合、そのリンクを無視する。
  - OSPF は、LSA で広告されたリンクを Shortest Path Tree に追加する場合、逆方向のリンクを確認する。これはトポロジデータベースの整合性を確かめるためである。Network LSA を追加した場合、リンク先の Router LSA もネットワークへのリンク情報を含む必要がある。

## 2.6 Receiver の使い分け

2 種類の Receiver には以下の特徴がある。

- Receiver\_Type1  
Receiver と DR が直接エリア内のデータベースを同期させるため、各ノードがエリア内に他の隣接ルータを持たない場合も動作する。そのため、ネットワークトポロジに依存せず経路制御が行なえる。しかし、Receiver の数が増えた場合、全ての Receiver がこの手法を用いると DR に負荷がかかる。
- Receiver\_Type2  
Receiver と DR はそれぞれの隣接ルータとトポロジデータベースを交換し、同一のデータベースを得る。そのため、エリア内において各ノードが別のルータと接続している必要がある。このように動作するトポロジが限定されるが、Receiver の増加に DR の負荷が影響されない。

本設計では、それぞれの Receiver の特性を考慮し、ネットワークトポロジによってこれらを使い分ける。UDL を OSPF で経路制御するネットワークで用いる場合、想定されるトポロジを以下に示す。

### ケース 1

Feeder,Receiver 間にエリア内で双方向の接続性がある。このトポロジを図 2.4 に示す。この場合、Feeder 及び Receiver はそれぞれ隣接ルータと UDL を含むエリアのトポロジデータベースを同期させる。そのため、DR と Receiver の間でデータベースを同期させる必要がない。このケースでは、Receiver\_Type2 を用いる。

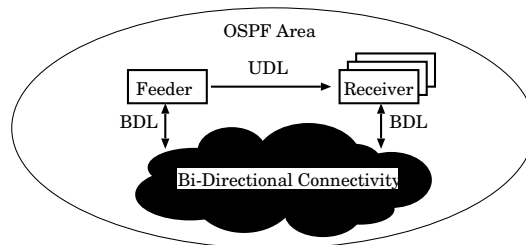


図 2.4 Feeder と Receiver 間にエリア内で双方向の接続性があるトポロジ

### ケース 2

Feeder と Receiver 間にエリア内で双方向の接続性がない。Feeder はエリア境界ルータ (Area Border Router, 以下 ABR) であり、エリア A に UDL を持ち、エリア B に BDL を持つ。このトポロジを図 2.5 に示す。

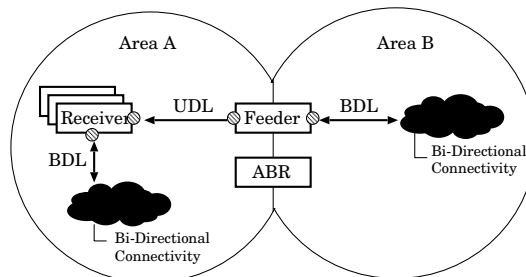


図 2.5 Feeder が異なるエリアに BDL を持っているトポロジ

このようなネットワークでは、エリア A とエリア B が双方向に接続するため、Feeder の他に両エリアをまたがる別の ABR が必要である。その理由は、各



エリア間の通信は必ずバックボーンを経由するため、隣接するエリア間にトラフィックが流れる場合、片方のエリアはバックボーンだからである。バックボーンと片方向にだけ接続されるエリアは、外部のネットワークと双方向の接続性を持たないため、2.1.2で述べた通信環境を充たさない。

このトポロジでは、Receiver はエリア A の隣接ルータを持つが、Feeder はエリア A のトポロジデータベースを同期させる隣接ルータを他に持たない。そのため、Receiver のどれか 1 つを Receiver\_Type1 として動作させ、Feeder にエリア A のデータベースを伝搬する必要がある。

ケース 3

Feeder と Receiver 間にエリア内で双方向の接続性がない。Receiver は ABR であり、異なるエリアに BDL を持っている。このトポロジを図 2.6 に示す。

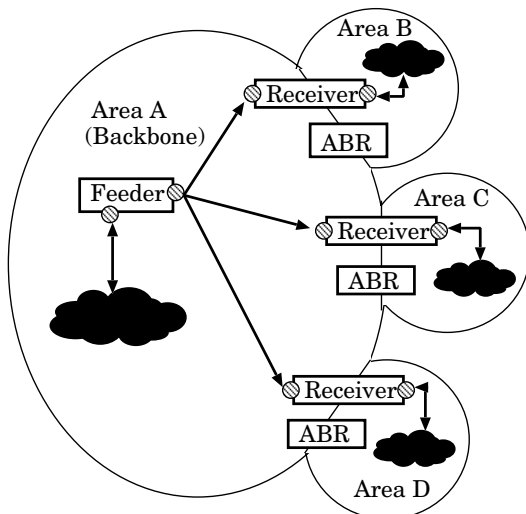


図 2.6 Receiver が異なるエリアに BDL を持っているトポロジ

このようなトポロジは、衛星回線を用いて離れたネットワークを接続する場合に有効である。図ではエリア A がバックボーンである。この場合、ケース 2 と同様の理由により、エリア A と各エリアが双方向に接続するため Receiver の他に両エリアをまたがる別の ABR が必要である。このトポロジでは、Feeder はエリア A に隣接ルータを持つが、Receiver はエリア A のトポロジデータベースを同期させる隣接ルータを他に持たない。そのため、各 Receiver が Receiver\_Type1

として動作し、Feeder に直接 Router LSA を広告する必要がある。また、Receiver が BDL を持つエリアがバックボーンではない場合、次の手法が考えられる。

1. Receiver が仮想リンクを用いて ABR の隣接ルータになる。
2. バーチャルリンクによって、Receiver を UDL を含むエリア (バックボーン) 接続する。
3. Receiver は ABR とバックボーンのトポロジデータベースを交換し、Receiver\_Type2 として動作する。

ケース 4

Feeder と Receiver 間にエリア内で双方向の接続性がない。Feeder, Receiver 共に ABR ではなくエリア A に BDL を持っているが、エリア A は UDL で分割されたトポロジになっている。そのため、Feeder と Receiver 間にエリア内で双方向の接続性がない。これを図 2.7 に示す。バックボーンエリア内は、全

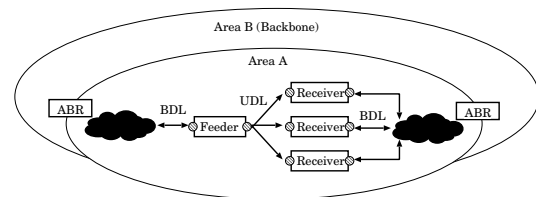


図 2.7 エリアが UDL で分割されているトポロジ

てのルータが必ず双方向の接続性を持つ必要がある。そのため、UDL で分割されたエリアはバックボーンとして動作しない。図では、エリア A が UDL によって分割されているので、これと隣接するエリア B はバックボーンである。このようなトポロジでは、分割されたエリアの区分それぞれが ABR を持つ必要がある。ABR を持たない区分は UDL だけでインターネットに接続されているため、2.1.2 で述べた通信環境を充たさない。

この場合、UDL を利用するためには Feeder 側の区分のルータが Receiver 側の区分のトポロジデータベースを保つ必要がある。そのため、Receiver のうち 1 つを Receiver\_Type1 として動作させる。Receiver 側の区分のルータは、Shortest Path Tree を作成する場合、Receiver における UDL への接続性を無視する。そのため、Feeder 側の区分へエリア内の経路を



持たず、バックボーンを経由して到達する。Feeder 側の区分のルータは UDL を経由し、Receiver 側の区分へ到達する。

### 第 3 章 評価

本章では、仮想ブロードキャストリンク機構を用いて OSPF を用いた場合と、本稿で述べた OSPF の動作を比較する。テストベッドネットワークは Ethernet により構成した。また、UDL 環境は Receiver の Ethernet デバイスドライバを変更して実現した。評価トポロジは、2.6 で述べた 4 つのトポロジを用いた。

#### 3.1 ケース 1 トポロジでの評価

Feeder と Receiver にエリア内で双方向の接続性があるトポロジ上での動作を評価した。図 3.1 に示すネットワークにおいて各機構を動作させた。

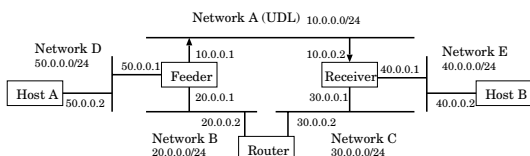


図 3.1 ケース 1 の評価トポロジ

図 3.4 では、Network A が UDL である。Network B、C、D、E は BDL である。Feeder、Receiver、Router では OSPF を動作させる。Host A では 50.0.0.1 にデフォルト経路をスタティックに設定した。Host B では 40.0.0.1 にデフォルト経路をスタティックに設定した。全てのノードはエリア 0(バックボーン)に属する。各ノードがネットワークに接続するコストを表 3.1 に示す。

表 3.1 各リンクのコスト

ノード	Net-work A	Net-work B	Net-work C	Net-work D	Net-work E
Feeder	1	1	-	1	-
Receiver	65535	-	1	-	1
Router	-	1	100	-	-

#### 3.1.1 仮想ブロードキャストリンクにおける OSPF の動作

DTCP を用いて、Receiver から 20.0.0.1(Feeder) へのトンネルを設定した。経路が収束すると、OSPF を動作させたルータ全てがエリア内の経路を保持した。この時の Feeder、Receiver、Router における経路表を図 3.2、図 3.3、図 3.4 に示す。

Destination	Gateway	Flags	Refs	Use	Netif
10/24	link#2	UC	0	0	ed1
20/24	link#3	UC	0	0	vx0
30/24	10.0.0.2	UG12c	0	0	ed1
40/24	10.0.0.2	UG12c	1	32	ed1
50/24	link#4	UC	0	0	ed0

図 3.2 Feeder における経路表

Destination	Gateway	Flags	Refs	Use	Netif
10/24	link#1	UC	0	0	ed1
20/24	30.0.0.2	UG12c	42	42	ed2
30/24	link#2	UC	0	0	ed2
40/24	link#3	UC	0	0	ed3
50/24	30.0.0.2	UG12c	0	35	ed2

図 3.3 Receiver における経路表

Destination	Gateway	Flags	Refs	Use	Netif
10/24	20.0.0.1	UG12c	0	0	ed1
20/24	link#1	UC	0	0	ed1
30/24	link#2	UC	0	0	de0
40/24	30.0.0.1	UG12c	0	8	de0
50/24	20.0.0.1	UG12c	0	29	ed1

図 3.4 Router における経路表

経路が収束した時点で、traceroute によって Host A と Host B の間で通信する場合に使用する経路を調べた。Host A から Host B に向かうパケットは、Feeder、Receiver の順にルータに中継され、UDL を用いて Host B に到達した。Host B から Host A に向かうパケットは、Receiver、Router、Feeder の順にルータに中継され、BDL だけを用いて Host A に到達した。Host A から Host B に traceroute を行なった結果を図 3.5 に示す。Host B から Host A に traceroute を行なった結果を図 3.6 に示す。

1	50.0.0.1	0.629 ms	0.522 ms	0.493 ms
2	10.0.0.2	1.281 ms	1.014 ms	1.027 ms
3	40.0.0.2	1.255 ms	1.255 ms	1.215 ms

図 3.5 Host A から Host B への traceroute の結果

1	40.0.0.1	0.372 ms	0.277 ms	0.276 ms
2	30.0.0.2	0.921 ms	0.648 ms	0.646 ms
3	20.0.0.1	0.823 ms	0.706 ms	0.692 ms
4	50.0.0.2	1.167 ms	1.145 ms	1.125 ms

図 3.6 Host B から Host A への traceroute の結果

### 3.1.2 本研究で設計した OSPF の動作

Receiver\_Type1 と Receiver\_Type2 の両方を動作させた。その結果、各ルータにおいて、仮想ブロードキャストリンク上で既存の OSPF を動作させた場合と同じ経路表が作成された。また、Host A、Host B 間における traceroute でも、同じ結果が得られた。

## 3.2 ケース 2 トポロジ上での評価

Feeder と Receiver にエリア内で双方向の接続生がないトポロジ上での動作を評価した。Feeder は ABR で、UDL が存在するエリアとは別エリアに BDL を持っている。図 3.1 に示すネットワークにおいて各機構を動作させた。

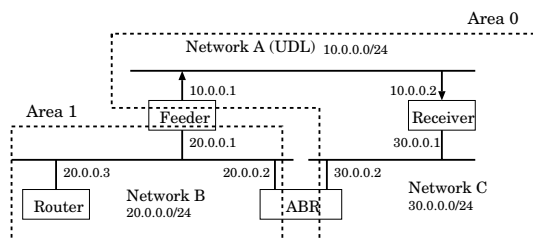


図 3.7 ケース 2 の評価トポロジ

図 3.7 では、Network A が UDL である。Network B、C は BDL である。Network A、C はエリア 0 に属し、Network B はエリア 1 に属する。エリア 0、エリア 1 間のエリア境界ルータは、Feeder、ABR の 2 つである。各ルータがネットワークに接続するコストを表 3.2 に示す。

表 3.2 各リンクのコスト

ノード	Network A	Network B	Network C
Feeder	1	1	-
Receiver	65535	-	1
ABR	-	1	100
Router	-	1	-

### 3.2.1 仮想ブロードキャストリンクにおける OSPF の動作

DTCP を用いて、Receiver から 20.0.0.1(Feeder) へのトンネルを設定した。経路が収束すると、各ルータは以下の経路を保持した。

- Feeder
  - Network A、C をエリア 0 のエリア内経路として保持
  - Network B をエリア 1 のエリア内経路として保持。
- Receiver
  - Network A、C をエリア 0 のエリア内経路として保持
  - Network C を ABR を経由したエリア外経路として保持
- ABR
  - Network A、C をエリア 0 のエリア内経路として保持
  - Network B をエリア 1 のエリア内経路として保持。
- Router
  - Network B をエリア 1 のエリア内経路として保持
  - Network A、C を Feeder を経由したエリア外経路として保持

Router から 10.0.0.2 のアドレスへ traceroute した結果を図 3.8 に示す。また、Router から 30.0.0.2 のアドレスへ traceroute した結果を図 3.9 に示す。どちらの場合も、Feeder をエリアの出口とし、UDL を経由して中継され、各 IP アドレスに到達している。

```

1 20.0.0.1 0.299 ms 0.234 ms 0.231 ms
2 10.0.0.2 0.916 ms 0.565 ms 0.549 ms
    
```

図 3.8 Router から 10.0.0.2 への traceroute

```

1 20.0.0.1 0.306 ms 0.234 ms 0.231 ms
2 10.0.0.2 0.926 ms 0.562 ms 0.552 ms
3 30.0.0.2 0.731 ms 0.617 ms 0.691 ms
    
```

図 3.9 Router から 30.0.0.2 への traceroute

### 3.2.2 本研究で設計した OSPF の動作

Feeder と Receiver の間にエリア内で双方向の接続性がないトポロジでは、Receiver\_Type2 は動作しない。そのため、Receiver\_Type1 だけを動作させた。その結果、各ルータにおいて、仮想ブロードキャストリンク上で既存の OSPF を動作させた場合と同じ経路表が作成された。また、Feeder、Receiver、ABR、Router において、ospfd が持ったエリア内経路とエリア外経路も同一だった。また、Router から 10.0.0.2、30.0.0.2 への traceroute も同じ結果だった。

### 3.3 ケース 3 トポロジ上での評価

Feeder と Receiver にエリア内で双方向の接続性がないトポロジ上での動作を評価した。Receiver は ABR で、UDL が存在するとは別のエリアに BDL を持っている。図 3.10 に示すネットワークにおいて各機構を動作させた。

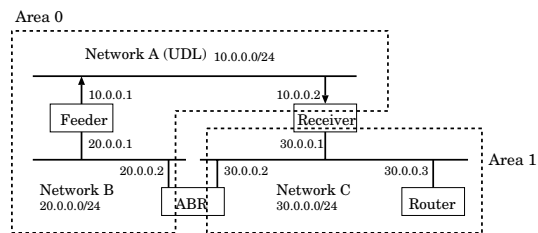


図 3.10 ケース 3 の評価トポロジ

図 3.10 では、Network A が UDL である。Network B、C は BDL である。Network A、B はエリア 0 に属し、Network C はエリア 1 に属する。エリア 0、エリア 1 間のエリア境界ルータは、Receiver、ABR の 2 つである。各ルータがネットワークに接続するコストを表 3.3 に示す。

表 3.3 各リンクのコスト

ノード	Network A	Network B	Network C
Feeder	1	1	-
Receiver	65535	-	1
ABR	-	1	100
Router	-	-	1

### 3.3.1 仮想ブロードキャストリンクにおける OSPF の動作

DTCP を用いて、Receiver から 20.0.0.1(Feeder) へのトンネルを設定した。しかし、一旦経路が収束した後、UDL が経路として利用できなくなり、Receiver の経路制御デーモンが動作しないなどの障害が現れた。これは、次の問題が生じたためである。

1. Receiver において、20.0.0.1(Feeder) へのトンネルが設定される。この時、Receiver から 20.0.0.1 への経路がまだ存在していない場合、2 に進む。Receiver から 20.0.0.1 への経路が存在した場合、3 に進む。
2. Network B、C が各ルータに広告され、Receiver と Feeder 間において、ABR を経由した経路が作成される。
3. Receiver から 20.0.0.1 へ ABR を経由して到達する。
4. Receiver と Feeder が隣接ルータとして互いを認識し、Network A(UDL) の存在が各ルータに広告される。
5. Receiver は、20.0.0.1 への経路として UDL を利用する設定に変わる。その理由は、OSPF ではコストにかかわらずエリア内の経路がエリア外の経路に優先するからである。この場合、Receiver は ABR を経由するエリア外の経路よりも、Network A(UDL) を用いたエリア内の経路を優先してしまう。
6. Receiver はカプセル化したパケットを UDL に送信しようとし、失敗する。
7. Feeder と Receiver の間の経路が失われる。

### 3.3.2 本研究で設計した OSPF の動作

Feeder と Receiver の間にエリア内で双方向の接続性がないトポロジでは、Receiver\_Type2 は動作し

ない。そのため、Receiver\_Type1 を動作させた。この機構では、仮想ブロードキャストリンク上で OSPF を動作させた時に生じた問題は起こらなかった。経路が収束すると、各ルータは次の経路を保持した。

- Feeder
  - Network A、B をエリア 0 のエリア内経路として保持
  - Network C を Receiver を経由したエリア外経路として保持
- Receiver
  - Network A、B をエリア 0 のエリア内経路として保持
  - Network C をエリア 1 のエリア内経路として保持
- ABR
  - Network A、B をエリア 0 のエリア内経路として保持
  - Network C をエリア 1 のエリア内経路として保持
- Router
  - Network C をエリア 1 のエリア内経路として保持
  - Network A、B を ABR を経由したエリア外経路として保持

Feeder から 30.0.0.3 の IP アドレスへ traceroute した結果を図 3.11 に示す。Feeder から送信されたパケットは、UDL を経由し、Receiver に中継され、30.0.0.3 に到達している。Router から 10.0.0.1 の IP アドレスへ traceroute した結果を図 3.12 に示す。Router から送信されたパケットは、ABR を経由してエリア 0 に中継され、Feeder(10.0.0.1) に到達している。

1	10.0.0.2	0.447 ms	0.337 ms	0.316 ms
2	30.0.0.3	0.572 ms	0.550 ms	0.549 ms

図 3.11 Feeder から 30.0.0.3 への traceroute

1	30.0.0.2	0.650 ms	0.369 ms	0.369 ms
2	10.0.0.1	0.760 ms	0.593 ms	0.561 ms

図 3.12 Router から 10.0.0.1 への traceroute

### 3.4 ケース 4

UDL によってエリアが分割されているトポロジ。Feeder、Receiver 共に ABR ではないが、エリア内に Feeder から Receiver への接続性だけが存在している。Feeder 側の区分と Receiver 側の区分には、それぞれ ABR が存在する。図 3.13 に示すネットワークにおいて、各機構を動作させた。

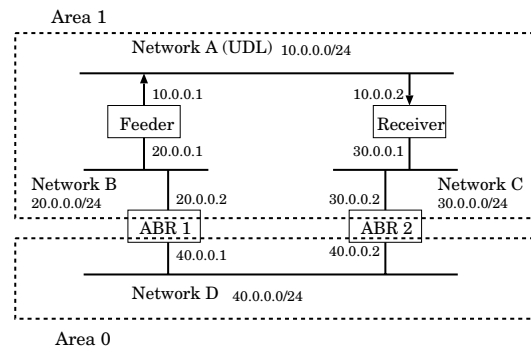


図 3.13 ケース 4 の評価トポロジ

図 3.4 では、Network A が UDL である。Network B、C、D、は BDL である。Network A、B、C はエリア 1 に属し、Network D はエリア 0(バックボーン)に属している。エリア 0、エリア 1 間には、ABR1、ABR2 という 2 つのエリア境界ルータが存在している。各ルータがネットワークに接続するコストを表 3.4 に示す。

表 3.4 各リンクのコスト

ノード	Net-work A	Net-work B	Net-work C	Net-work D
Feeder	1	1	-	-
Receiver	65535	-	1	-
ABR1	-	1	-	1
ABR2	-	-	1	1

### 3.4.1 仮想ブロードキャストリンクにおける OSPF の動作

DTCP を用いて、Receiver から 20.0.0.1(Feeder) へのトンネルを設定した。しかし、UDL を利用しない形で経路が収束した後、UDL を利用した Feeder から Receiver への経路と UDL を逆流しようとする Receiver から Feeder への経路が設定された。その後、再び UDL を利用しない形で経路が設定されるといった繰り返しが生じた。これは、次の問題が生じたためである。

1. Receiver において、20.0.0.1(Feeder) へのトンネルが設定される。この時、Receiver から 20.0.0.1 への経路が存在していない場合、2 に進む。Receiver から 20.0.0.1 への経路が存在した場合、4 に進む。
2. エリア 1 は Feeder 側の区分 (Feeder、ABR1、Network B) と Receiver 側の区分 (Receiver、ABR2、Network C) に別れ、独立したエリアとしてエリア内の経路が計算される。また、エリア 0 でもエリア内の経路が計算される。
3. ABR1、ABR2 は summary LSA によってそれぞれのエリアの経路を別エリアに広告する。

— ABR1 は Feeder 側の区分の経路をエリア 0 に広告する。

— また、ABR1 はエリア 0 の経路を Feeder 側の区分に広告する。

— ABR2 は Receiver 側の区分の経路をエリア 0 に広告する。

— また、ABR2 はエリア 0 の経路を Receiver 側の区分に広告する。

— その後、ABR1 は ABR2 が広告した Receiver 側の区分の経路を Feeder 側の区分に広告する。

— ABR2 は ABR1 が広告した Feeder 側の区分の経路を Receiver 側の区分に広告する。

以上によって、Feeder 側の区分と Receiver 側の区分はバックボーンを経由し、独立したエリア同士として通信ができる状態になる。

4. Receiver から 20.0.0.1(Feeder) へバックボーンを経由して到達する。

5. Receiver と Feeder が隣接ルータとして互いを認識し、Network A(UDL) の存在が各ルータに広告される。

6. Receiver は、20.0.0.1 への経路として UDL を利用する設定に変わる。また、Receiver 側の区分のルータは Feeder 側の区分へ到達するために UDL を逆流する経路を利用しようとする。その理由は、ケース 3 と同様に、OSPF ではコストにかかわらずエリア内の経路がエリア外の経路に優先するからである。Network A(UDL) の存在が広告された時点で、Feeder 側の区分と Receiver 側の区分は 1 つのエリアとして経路計算される。Receiver 側の全てのルータは Feeder 側の区分へ到達するため、UDL を逆流する経路を利用しようとする。

7. Receiver 側の区分から Feeder 側の区分への到達生が失われる。Receiver は BDL を利用してトンネリングするはずのケットを再び UDL に送信しようとし失敗する。Feeder と Receiver 間では Hello が片方向にしか流れなくなるので、RouterDeadInterval が経過すると Feeder と Receiver は通信の障害を検知する。

8. Feeder と Receiver は、UDL をリンク情報に含めない新しい Router LSA を作成し、それぞれの区分に広告する。

9. エリア 1 内の全てのルータが、UDL を利用しない形で経路を再計算する。(2 に戻る。)

つまり、Receiver 側がバックボーンを経由して 20.0.0.1(Feeder) に到達している間は Feeder と Receiver が経路制御メッセージを交換できる。しかし、Feeder と Receiver が必要な情報を交換し終え新しい経路が計算されると、Receiver は 20.0.0.1(Feeder) に到達できなくなる。そのため、一定時間後に元の経路に戻る。

### 3.4.2 本研究で設計した OSPF の動作

Feeder と Receiver の間にエリア内で双方向の接続性がないトポロジでは、Receiver\_Type2 は動作しない。そのため、Receiver\_Type1 を動作させた。この場合、仮想ブロードキャストリンク上で OSPF を動作させた時に生じた問題は起こらなかった。経路が収束すると、各ルータは次の経路を保持した。

- Feeder
  - Network A、B、C をエリア 1 のエリア内部経路として保持
  - Network D を ABR1 を経由したエリア外経路として保持
- Receiver
  - Network C をエリア 1 のエリア内経路として保持
  - Network A、B、D を ABR2 を経由したエリア外経路として保持
- ABR1
  - Network A、B、C をエリア 1 のエリア内経路として保持
  - Network D をエリア 0 のエリア内経路として保持
- ABR2
  - Network C をエリア 1 のエリア内経路として保持
  - Network D をエリア 0 のエリア内経路として保持
  - Network A、B を ABR1 を経由したエリア外経路として保持

ABR1 から 30.0.0.2 の IP アドレスへ traceroute した結果を図 3.14 に示す。ABR1 から送信されたパケットは、UDL を経由し Receiver に中継され 30.0.0.2 に到達する。

1	20.0.0.1	0.479ms	0.520ms	0.225ms
2	10.0.0.2	0.917ms	0.589ms	0.561ms
3	30.0.0.2	1.078ms	0.798ms	0.921ms

図 3.14 ABR1 から 30.0.0.2 への traceroute

Receiver から 20.0.0.1 の IP アドレスへ traceroute した結果を図 3.15 に示す。Receiver から送信されたパケットは、ABR2、ABR1 の順に中継され 20.0.0.1 に到達する。

1	30.0.0.2(30.0.0.2)	0.635ms	0.451ms	0.413ms
2	40.0.0.1(40.0.0.1)	0.959ms	0.600ms	0.573ms
3	20.0.0.1(20.0.0.1)	0.585ms	0.584ms	1.029ms

図 3.15 Receiver から 20.0.0.1 への traceroute

#### 第4章 おわりに

WISH WG では、早くから UDL を含むネットワークにおける経路制御の問題に着目し、これまでいくつかの解決法を提案した。また、IETF での議論に積極的に参加してきた。仮想ブロードキャストリンクを用いた解決法は、比較的簡単な実装で多くの経路制御プロトコルに対応できる。そのため、短期的な解決法として IETF において標準化が進められている。

本年度の活動では、Open Shortest Path First (OSPF) を拡張し、片方向の衛星回線を含むネットワークにおいて動作する変更を行なった。それと共に、既存の手法を用いて OSPF を動作させた場合について詳細な研究を行なった。その結果、仮想ブロードキャストリンクを用いて OSPF を動作させた場合、いくつかのネットワークポロジでは正しく経路制御を行なえないと分かった。本研究で変更した OSPF は、UDL を含むあらゆるネットワークポロジで動作し、このような問題点を解決した。また、本研究は将来的に UDL を視野に入れた新しい経路制御プロトコルを設計するための前段階の研究としても意義がある。

本研究で変更した OSPF では、多数の Receiver が Receiver\_Type2 として動作し、少数の Receiver が Receiver\_Type1 として動作することを想定している。どの Receiver がどの種類の動作をするかは、管理者の設定で行なう。しかし、トポロジの変化によって Receiver\_Type2 が Receiver\_Type1 として動作する必要が生じる場合も考えられる。そのため、Receiver の動作を動的に変更する機能が必要である。また、どの程度の Receiver 数まで本機構が動作するのか、ネットワークシミュレータなどを用いて調査する予定である。