

第 16 部

衛星通信によるネットワーク構築実験

第 1 章

はじめに

1.1 WISH タスクフォース

WISH タスクフォースでは、衛星通信ネットワークと従来の地上網を融合しインターネットの新しい課題の解決を目的とするネットワーク構築実験を行なっている [140]。

インターネットは大きく成長を遂げ、次世代の情報通信基盤として注目を集めている。接続地域が広がるにつれ、その規模性はより一層重要となっており、地理的普遍性をもつ衛星通信はこれを満たすインターネットメディアとして期待できる。また、衛星通信には回線設定の柔軟性や通信局の設置の簡易性といった利点や、片道約 250ms の遅延といった特徴がある。WISH タスクフォースではこれらの特性を考慮した WISH ネットワークの構築実験や運用を行なっている。

本稿では、1995 年度における以下のような研究・実験を報告する。

- 衛星通信ネットワークと地上ネットワークを統合した動的なネットワーク形成に関する研究
- 衛星通信ネットワークに対応したトランスポートプロトコルの研究
- 衛星通信ネットワークを利用したマルチキャスト配送実験

なお、WISH タスクフォースにおける衛星通信は、「デジタル衛星通信の大学間高度共同利用研究協議会」の実験可搬局免許を用いた (株) 日本サテライトシステムズの衛星通信回線、および (株) 日本サテライトシステムズとの随時分割契約による衛星通信専用サービスを用いて行なわれている。

1.2 WISH における衛星通信環境

WISH のネットワーク環境では、超小型地球局 VSAT による衛星通信地球局が日本全国 9 地点に 12 局設置されている (図 1.1)。

VSAT はパラボラアンテナ、屋外装置 ODU、屋内装置 IDU から構成される (図 1.2)。データの送受信のための計算機との接続は、双方のシリアルインターフェースを通じて最

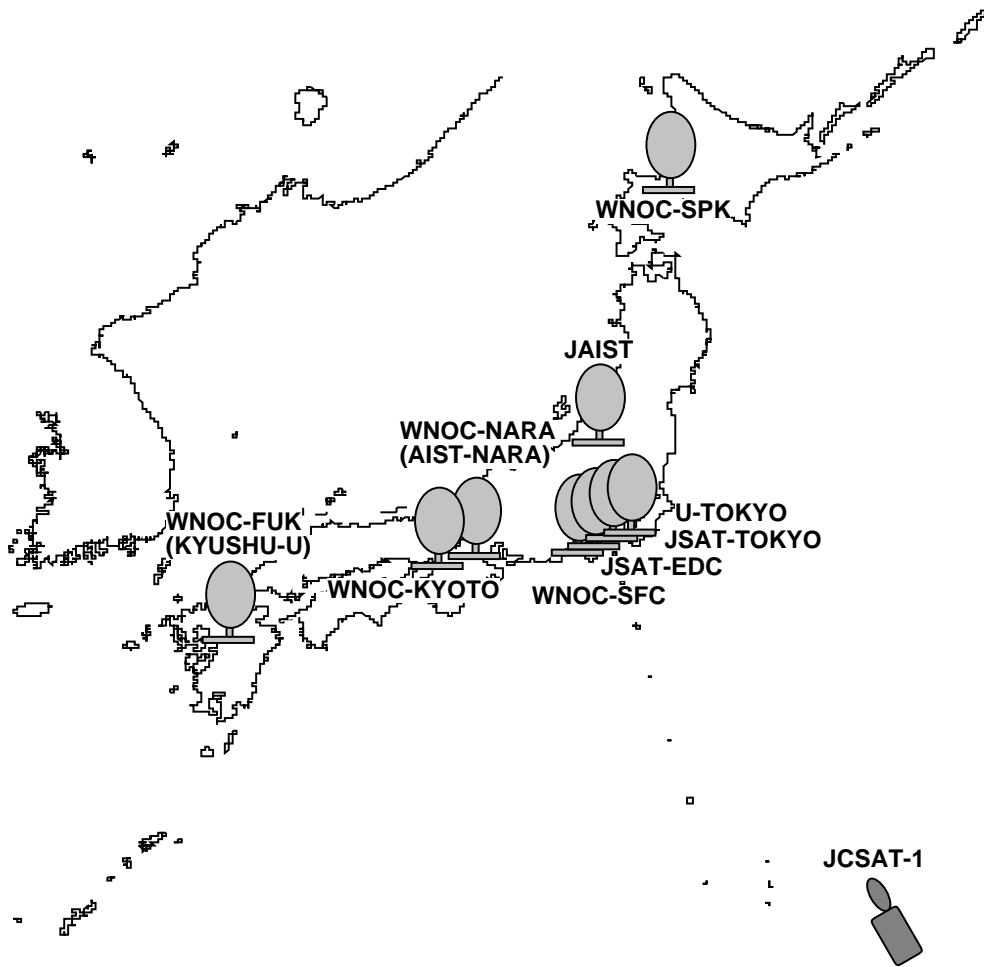


図 1.1: WISH ネットワーク

大 2Mbps の通信速度で IDU と接続される [141]。また、通信の各種設定も IDU によって行なわれる。

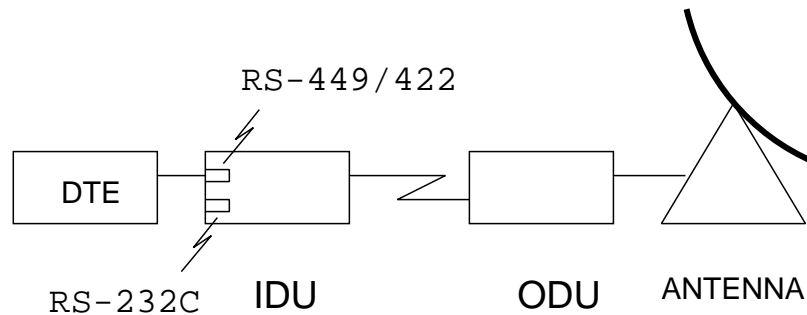


図 1.2: VSAT の構成

VSAT における通信は、送信局の送信チャンネルと受信局の受信チャンネルを合致させて行う。ここでいうチャンネルとは、衛星へ送信するアップリンク周波数と衛星から受信するダウンリンク周波数の組を意味する。

また、本研究で使用した地球局では、TDMA (Time Division Multiple Access) の機能はなく、同時に複数局から通信を受信することは出来ない。

第 2 章

インターネットのための衛星通信制御機構の構築

世界的規模へと成長したインターネットでは、接続地域が広がるにつれ、その規模性はより一層重要となっている。また、インターネットの利用形態は多様化し、動的に自由にトポロジを形成する通信媒体に対する要求も高い。このような状況における通信媒体として、衛星通信の重要性が高まっている。

これらの要求を満たす衛星通信と従来の地上ネットワークを統合したインターネットの構築を目的に、衛星通信ネットワークにおけるトポロジモデルを考案した。このモデルを実現するための衛星通信ネットワーク制御機構の設計・実装を行なった [142][143]。

また、本機構を用いた実験を、実際の衛星通信ネットワーク環境や衛星通信ネットワークをシミュレートした環境の上で行ない、実用性を実証した。

本機構で用いた機構により衛星通信ネットワークを統合したインターネットの構築が実現することで、今後同報型マルチメディア通信の効率化、地理的条件に因われない接続性を得ることが可能になる。また、衛星通信局の設置簡易性から非常時におけるバックアップ回線としての利用も実現できる。

2.1 研究課題

2.1.1 多地点間の相互通信の仕組みに関する課題

1 つ以上の送信周波数を複数の衛星通信局でいかに効率良く共有しながら、多地点間の相互通信を実現することが出来るかを考えなければならない。

解決方法の 1 つとして、ALOHANET[144] のようなデータリンク層の通信プロトコルによる方法がある。この方法は、複数の通信局で単一の周波数を共有し衝突の検出と再送を行なうため、回線利用率が高い場合の通信効率が低下する。

そこで本研究では、データリンクプロトコルに依らず地上ネットワークを併用することによって送受信局間で相互通信が出来る機構の構築を行なう。

2.1.2 経路制御に関する課題

巨大なインターネット上の全てのネットワークに対する経路情報を静的に設定することは非効率的であることは明らかである。そのため、経路を動的に制御するためのプロトコルが規定されており、これに基づく実装もいくつか存在する。しかし、そのどれもが隣接ノードを通して経路情報の交換を行なうことによって経路の設定を行なう。しかし、衛星通信ネットワークを単方向の同報型ネットワークとして構築する場合既存の経路制御技術の利用はできない。このような点から衛星通信ネットワーク上で経路制御プロトコルが動作可能な環境の構築を図る。

2.2 設計

以下のような方針に基づいて衛星通信制御システムの設計を行なった。

- 形成する衛星通信ネットワークは 1 つにする
- 切替可能な同報型ネットワークトポロジを形成する

一対多型トポロジを形成することが出来れば、一対一型通信に対応することも出来るため、この中で送信者の切り替えを制御するものとする。

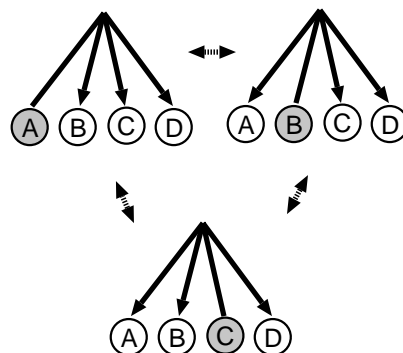


図 2.1: トポロジ切り替え

また、受信者間の通信、および受信者から送信者への通信は地上ネットワークを併用する。

- トポロジの制御は 1 台の計算機で集中制御する
- 使用する通信周波数チャンネル、データの通信帯域は一定とする

これらの方針のもと、図 2.1 で示すような疑似多対多型衛星通信ネットワークを構築する。

2.3 システム構成

本システムは以下のモジュールから構成される。

- 衛星通信ネットワーク対応データリンク層

VSAT におけるデータ通信には IDU のシリアルインターフェースが用いられる。一般的にシリアルインターフェースを用いたネットワークは一对一型の接続以外考慮されていない。しかし衛星通信の同報性を利用するためには、データリンクのアーキテクチャとして一对多型の接続に基づかななくてはならない。そこで、一对多型接続が可能なシリアルインターフェースを用いたネットワークドライバが必要となる [145]。

- 衛星通信機器制御部

本研究で用いる衛星通信は、各所に設置された IDU の設定を通して行なわれる。そのため IDU の制御を通じて計算機から実際の衛星通信に関する設定を操作するモジュールが必要となる。

- トポロジ制御部

オペレータの判断に合わせた衛星通信ネットワークのトポロジを形成するためには、全ての地球局で統一的な協調作業が必要となる。このために各所の IDU を制御する計算機をさらに統合管理する機構が必要となる。

- ネットワーク構成部

次章にて詳述。

2.4 ネットワーク構成部概要

衛星通信ネットワークが任意のホスト間の通信経路に含まれるようになるためには、動的に経路情報を制御するプロトコルを動作させる必要がある。しかし、衛星通信ネットワークを単方向型ネットワークとして構築するため既存の経路制御技術をそのまま使用することは出来ない。また、送信衛星通信ホストの切り替えによる他ネットワークへの影響は大きい。このような問題を考慮に入れ衛星通信ネットワーク上で経路制御プロトコルが動作する環境を設計する。

2.4.1 仮想リンク完全結合方式

送信衛星ホストと受信衛星ホスト間において地上側ネットワークインターフェースを用いた仮想リンクを設定し、更に衛星側インターフェースと仮想リンクインターフェースを統合した仮想インターフェースを用意する (図 2.2)。これにより、送信衛星ホストが任意の受信衛星ホストとの双方向通信を行なうのに往路には衛星通信ネットワークを使用でき、復路には仮想リンクネットワークを使用できる。

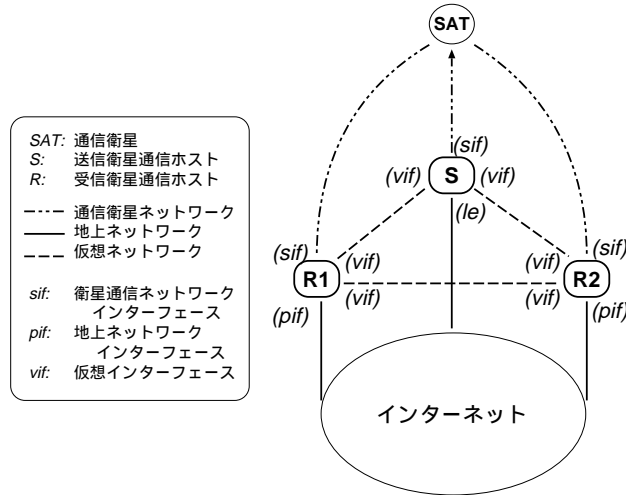


図 2.2: 仮想リンク完全結合方式

衛星ホストの数を n とすると送信衛星ホストが切り替わることを考えた場合、全ての衛星ホスト間で完全結合グラフを形成するためには $n \times (n-1)/2$ 個の仮想リンクを設定する必要が生じ、 n に対する規模性は乏しい。

2.4.2 仮想ゲートウェイ方式

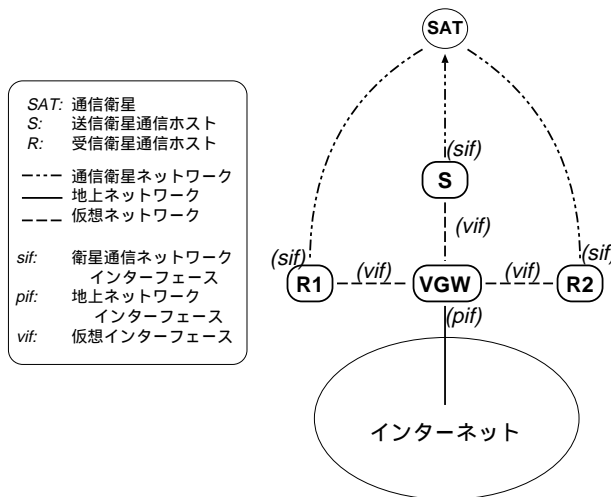


図 2.3: 仮想ゲートウェイ方式

衛星通信ネットワークへの仮想ゲートウェイを作り、仮想ゲートウェイと送信衛星ホスト間で仮想リンクを設定する(図 2.3)。受信衛星ホストから送信衛星ホスト、受信衛星ホス

ト間の通信についても仮想ゲートウェイマシンへの仮想リンクを通して行なえる。

経路制御に関しては、仮想リンク完全結合方式と同様に衛星側インターフェースと仮想リンクインターフェースを統合した仮想インターフェースを仮想ゲートウェイと各衛星ホスト間に設ける。

送信衛星ホストがほぼ固定され近隣に仮想ゲートウェイが配置されている場合には有効な方式となるが、送信衛星ホストの切替が頻雑に起こったり、地上ネットワークにおける仮想ゲートウェイと送信衛星ホスト間の経路コストが大きい場合には冗長な経路が生ずる可能性がある。

仮想リンクに関する設定は VGW と衛星通信ホスト間で必要であり、その個数は n である。

2.4.3 仮想ゲートウェイ仮想リンク完全結合併用方式

仮想リンク完全結合方式と仮想ゲートウェイ方式を組み合わせた方式である(図 2.4)。近隣の衛星ホストでは仮想ゲートウェイを共有することで、両方式のそれぞれの長所を取り入れることが出来る。

しかし、仮想ゲートウェイの配置方法に対する最適な解を見付けることが難しい。

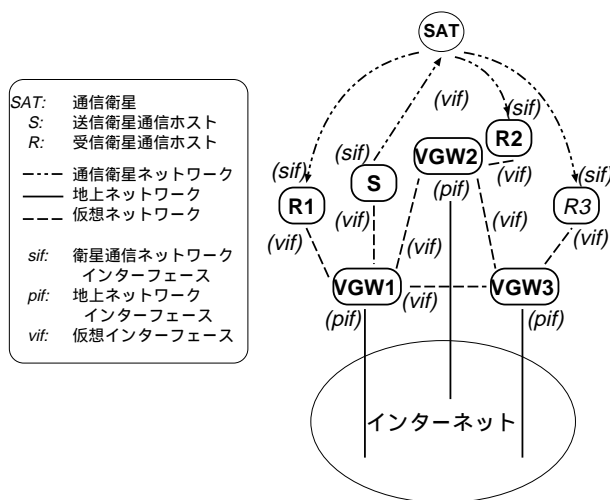


図 2.4: 仮想ゲートウェイ仮想リンク完全結合併用方式

2.4.4 ネットワーク構成部のまとめ

ネットワーク構成部として 3 つの方式を提示した。これらの方式の特徴を表 2.1まとめると、

表 2.1: ネットワーク構成方式のまとめ

	大規模性	経路の冗長性
仮想リンク完全結合方式	×	
仮想ゲートウェイ方式		×
仮想ゲートウェイ仮想リンク完全結合併用方式		

広域ネットワークにおいては仮想ゲートウェイ仮想リンク完全結合併用方式は最も適した方式である。しかし、経路に関する設定の複雑さと本研究の現在の環境を考慮し、仮想リンク完全結合併用方式の実装を行った。

2.5 評価

2.5.1 トポロジ制御部の評価

衛星ネットワーク上で送信衛星ホストの切替えが起こった場合のオーバーヘッドについて評価を行なった。

衛星通信ネットワーク上にない計算機から任意の衛星ホストへ 64 バイトの ICMP エコー要求を送り続けながら、トポロジ制御ホスト (TC) を通じて衛星ネットワークの送信者を切替えることを行なった。この時の ICMP エコー要求パケットのシーケンス番号と RTT 時間の変化を図 2.5 に示す。

現送信衛星ホスト (S) への送信停止命令から、次の送信衛星ホスト (S') の送信開始までは数秒の時間がかかり、その期間中は衛星通信ネットワークへの通信が行なえなくなる。図 2.5 から 2 度目の送信衛星ホスト変更においてシーケンス番号 59 のパケットが 1 つ到達していないことが分かる。

また、送信衛星ホスト切替えに要する時間を計測した (10 回当たりの平均)。これは IDU 制御に関する時間、すなわち IDU 内での処理時間と IDU 制御計算機との通信時間の合計を一定とすると、TC-S 間、TC-R1 間の往復時間 (RTT: Round Trip Time) に左右される。そこで、トポロジ制御計算機から各 IDU 制御計算機間の通信状況からトポロジ変更に要する時間についても計測し、RTT と合わせて表 2.2 に示す。

表 2.2 の切替え所要時間には、TC-S、TC-R1 間の通信のオーバーヘッドなどが含まれている。この間の RTT がより大きい衛星通信ホスト間において同様な送信者切替えを行なった場合にはより多くの所要時間が費される。特に本実装では、TCP を用いたために 3 ウェイハンドシェイクなど通信のオーバーヘッドが大きく、UDP を用いるなどして改善する余地がある。しかし、会議システムにおいて発言者に合わせてトポロジを切替える場合等には十分短い時間である。

```

% ping 133.4.120.1
PING 133.4.120.1 (133.4.120.1):
icmp_seq=10 time=4.142 ms
icmp_seq=11 time=4.307 ms    <- この後、切替
icmp_seq=12 time=339.609 ms
icmp_seq=13 time=421.229 ms
...
icmp_seq=57 time=4.793 ms
icmp_seq=58 time=3.877 ms    <- この後、切替
icmp_seq=60 time=292.877 ms
^C
--- 133.4.120.1 ping statistics ---
60 packets transmitted, 54 packets received,
                                9% packet loss
round-trip min/avg/max = 3.748/117.971/421.229 ms

```

図 2.5: トポロジ変更の様子

表 2.2: 送信者切替所要時間

IDU 処理	TC-S 間	TC-R1 間	切替所要
419 ms	3 ms	110 ms	1450 ms

2.5.2 経路制御部の評価

送信衛星ホストの切替えに応じて、衛星通信ネットワークで経路制御プロトコルが正しく動作するか評価を行なった。ここでの評価は実際の衛星通信ネットワーク上ではなく疑似的な環境のもとで行なった。図 2.6にこの環境を示す。

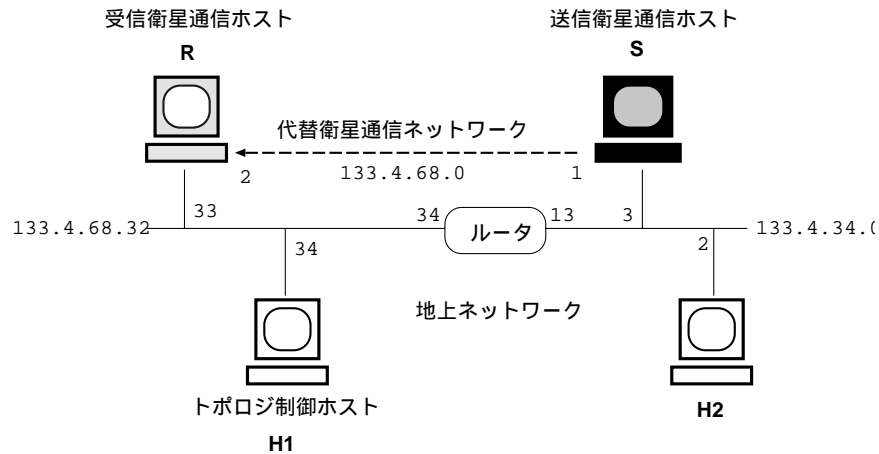


図 2.6: 中継ネットワークとしての評価環境

高速シリアルインターフェース間を回線エミュレータを用い 64Kbps の通信速度で設定し、衛星通信ネットワークとして想定する。S を送信衛星通信ホスト、R を受信衛星通信ホストとして想定した。

図 2.6における H1、H2 の経路表 (一部省略、以下同様) をそれぞれ図 2.7、図 2.8に示す。

```

$ netstat -rn
Routing tables
Internet:
Destination      Gateway           Flags   Interface
133.4.34/27      link#1           UC      we0
133.4.68/27      133.4.34.3      UG      we0
133.4.68.32/27  133.4.34.3      UG      we0
    
```

図 2.7: H1 における経路表

これにより、ホスト H1 からネットワーク 133.4.68.32 への経路は代替衛星通信ネットワークを経由することが分かる。また、逆にネットワーク 133.4.68.32 上のホスト H2 からネッ

```
$ netstat -rn
Routing tables
Internet:
Destination      Gateway          Flags   Interface
133.4.34/27      133.4.68.34    UG      ne0
133.4.68/27      133.4.68.33    UG      ne0
133.4.68.32/27  link#5          UC      ne0
```

図 2.8: H2 における経路表

トワーク 133.4.34.0 への経路は代替衛星通信ネットワークを経由せず地上ネットワークを経由することが分かる。

ここで、送信衛星通信ホストを S から R へ切り替えトポロジを変更する。経路の再設定がなされ安定した後の H1、H2 の経路表をそれぞれ図 2.9、図 2.10に示す。ホスト H1 が

```
$ netstat -rn
Routing tables
Internet:
Destination      Gateway          Flags   Interface
133.4.34/27      link#1           UC      we0
133.4.68/27      133.4.34.3      UG      we0
133.4.68.32/27  133.4.34.13     UG      we0
```

図 2.9: トポロジ変更後の H1 における経路表

らネットワーク 133.4.68.32 への経路は代替衛星通信ネットワークを経由せず、逆に H2 からネットワーク 133.4.34.0 への経路は経路するよう変更されたことが分かる。

トポロジ変更の前後双方共に、送信衛星ホストから受信衛星ホストへは中継ネットワークとして機能し、逆の経路は仮想リンクを経由せずに地上ネットワークを用いるように設定され、適切な経路設定がなされていることが確認できた。

```
$ netstat -rn
Routing tables
Internet:
Destination      Gateway          Flags    Interface
133.4.34/27      133.4.68.34    UG       ne0
133.4.68/27      133.4.68.33    UG       ne0
133.4.68.32/27  link#5          UC       ne0
```

図 2.10: トポロジ変更後の H2 における経路表

2.6 まとめ

既存の地上ネットワーク上では実現出来なかった動的トポロジ形成の仕組みを衛星通信を用いることで実現した。衛星通信の特徴である同報性、接続形態の柔軟性、地理的な普遍性を活かすだけでなく、通信形態や目的などの要求に応じてネットワークトポロジの管理・制御をすることが可能となった。

本機構での特徴は、次の通りである。

- 衛星通信ネットワークの接続形態を必要に応じて自由に設定・変更できる点
- 接続形態を変更した場合他のネットワークにあたえる影響を抑制する点
- 双方型通信媒体を基本とするインターネットにおいて同報型単方向の衛星通信ネットワークを統合した点

また、衛星通信ネットワークを用いた実験及びシミュレーションによって評価を行ない衛星通信を用いたネットワークがインターネットの1つとして十分利用可能であることを示した。

今後の課題は、実際に衛星通信ネットワークをインターネットの一部として運用することである。また、形成するネットワークの個数やトポロジを限定しないより複雑なトポロジの形成や、ネットワークの状態を地上ネットワークと合わせて監視し最もトラフィックの多い地点間へ接続を切り替えるといった自動的なトポロジ切替え機能の研究が必要である。

第 3 章

衛星回線における TCP の研究

WISH のこれまでの実験結果より、従来の TCP では、衛星回線のような広帯域高遅延の環境において、ウィンドウサイズの限界により、帯域を有効に利用することができないことがわかった [146][147]。

Van Jacobson は、RFC1323[148] においてギガビットネットワークに対する TCP の改善案を提案した。RFC1323 による提案の中には、TCP の新しいオプションとしてウィンドウスケールオプションを追加することが含まれている。ウィンドウスケールオプションを用いることにより、TCP では、 $2^{30} - 1$ までのウィンドウサイズを指定することができる。

本節では、RFC1323 を用いた場合の衛星回線上における効果をシミュレーションなどから解析し、その問題点と解決策の考察を行なった [149]。

3.1 ネットワーク構成

図 3.1 に、本シミュレーションで使用したネットワークの構成を示す。ノード 1、2 は、それぞれ Ethernet に接続されている。Ethernet の帯域は、10Mbps、伝搬遅延は 50nsec に設定した。ルータ 3、4 の間は衛星回線に接続されている。衛星回線の帯域は、2Mbps、伝搬遅延 500msec に設定した。ルータのバッファサイズは 20Kbyte とした。

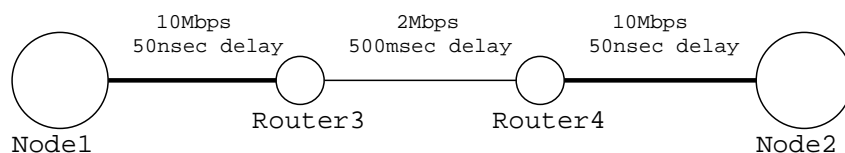


図 3.1: ネットワーク構成

3.2 シミュレーション結果

ある経路上の通信における最適なウィンドウサイズは、以下の計算式で求めることができる。

$$\text{最適なウィンドウサイズ} = \frac{\text{経路上の一番細いリンクの帯域}}{\text{経路全体の伝搬遅延}} \quad (3.1)$$

この式を用いて、図のネットワーク構成の場合の最適なウィンドウサイズを求めると、

$$2Mbps/8 \times 500msec = 125Kbyte$$

となる。この構成において、ウィンドウサイズを変えてノード 1 からノード 2 に向けて、60Mbyte のデータ転送を行なった。

図 3.2 に 従来の TCP のウィンドウサイズの最大値 (65535byte) で通信を行なった場合の結果を示す。転送開始後、最初の数秒でウィンドウサイズが 65535 バイトに達し、以後変化がないことがわかる。図 3.3 に帯域の利用率を示す。50%程度で帯域利用率が安定していることがわかる。

この場合は、ウィンドウが最大値の 65535 バイトまで達しているが、ネットワークの帯域に対して、ウィンドウサイズが小さすぎるために、帯域の 50%程度しか利用できていない。

図 3.4 に、計算で求めた最適なウィンドウサイズ (125Kbyte) で通信を行なった場合の結果を示す。図より、最初の数秒でウィンドウサイズが最大値の 125Kbyte に達し、以後変化がないことがわかる。図 3.5 にこの時の帯域利用率を示す。帯域利用率がほぼ 100%に達していることがわかる。

この場合、ネットワークの帯域に対しウィンドウサイズが最適なため、ネットワーク資源を有効に利用できている。

図 3.6 に、最適なウィンドウサイズの 2 倍の大きさ (250Kbyte) で通信を行なった場合の結果を示す。この場合、ウィンドウが最大値まで開き切る前に、輻輳が起これ、再度ウィンドウが開き始め、また輻輳するというウィンドウサイズの振動現象が観測されている。

図 3.7 にこの場合の帯域利用率を示す。帯域利用率が頻繁に変化し、平均帯域利用率が低くなっていることがわかる。この場合、ネットワークの帯域に対してウィンドウサイズが大きすぎるため、輻輳が発生し、ネットワーク資源を有効に利用することができない。

3.3 シミュレーションによる考察

以上の結果により、TCP による通信では、帯域を有効に利用するためには最適なウィンドウサイズを指定する必要があることがわかった。ウィンドウサイズが小さくても大きくても帯域を有効に利用することができない。

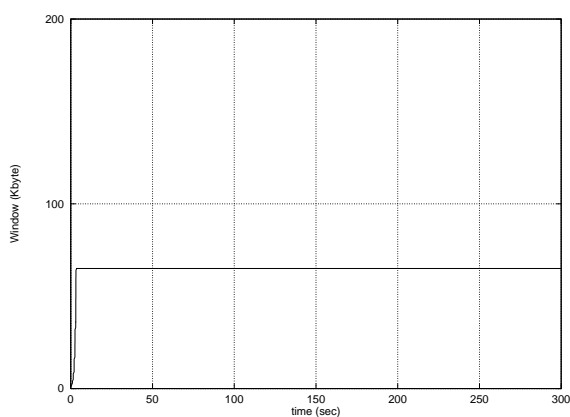


図 3.2: 最大ウィンドウサイズ 65535byte のデータ転送時のウィンドウサイズの変化

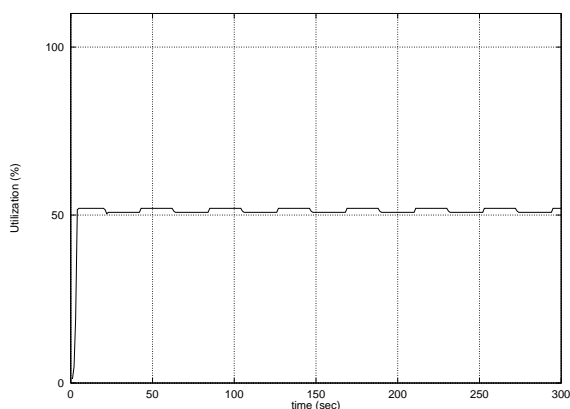


図 3.3: 最大ウィンドウサイズ 65535byte のデータ転送時の帯域占有率の変化

現在の TCP では、`setsockopt()` システムコールを用いてウィンドウサイズの指定を行なう。しかし最適なウィンドウサイズは、通信を行なうノード間の経路によって異なる。このため、`setsockopt` を用いて最適なウィンドウを指定する方法は以下のような方法に限定される。

1. 接続されるネットワーク毎にアプリケーションを用意する。
例えば、衛星回線につながるインターフェースを持つノード上でのアプリケーションは、予め大きめのウィンドウサイズで通信を行なうように設定しておく。
2. アプリケーションの起動時にウィンドウサイズを設定する。
アプリケーションの引数にウィンドウサイズを指定するように設計する。

しかし、1の方法では、2つ以上のネットワークインターフェースを持つノードや、ネットワークの構成変更に対応することができない。2の方法では、アプリケーションの利用

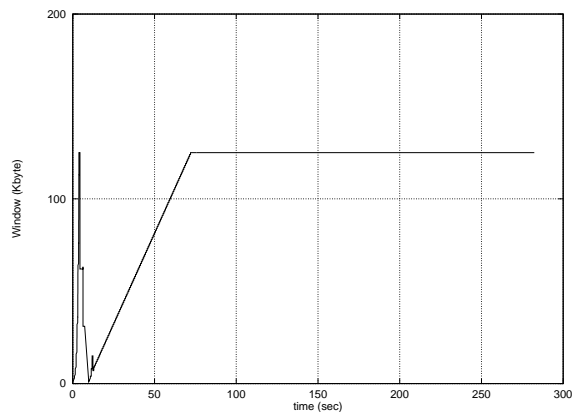


図 3.4: 最大ウィンドウサイズ 125Kbyte のデータ転送時のウィンドウサイズの変化

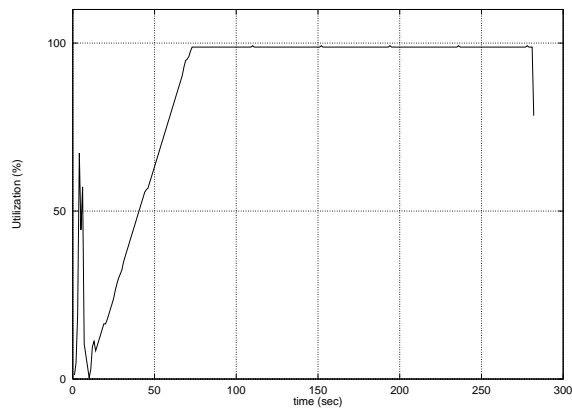


図 3.5: 最大ウィンドウサイズ 125Kbyte のデータ転送時の帯域占有率の変化

者があらかじめ最適なウィンドウサイズを知っている必要がある。いずれの方法も柔軟性に欠け、良い解決策ではない。

この問題に対処するためには、`setsockopt()` システムコールによる固定的なウィンドウサイズ指定の仕組みではなく、より柔軟なウィンドウサイズの指定の仕組みが必要である。

3.4 新しいフロー制御機構

これまでの考察から、現在の TCP では最大のスループットを実現するためには最適なウィンドウサイズを指定する必要があることが分かった。しかし、最適なウィンドウサイズは、パケットが通過する経路ごとに異なり、コネクション毎に異なる値を決定する必要がある。

これらのことから、各コネクションがデータ転送を行ないながら最適なウィンドウサイ

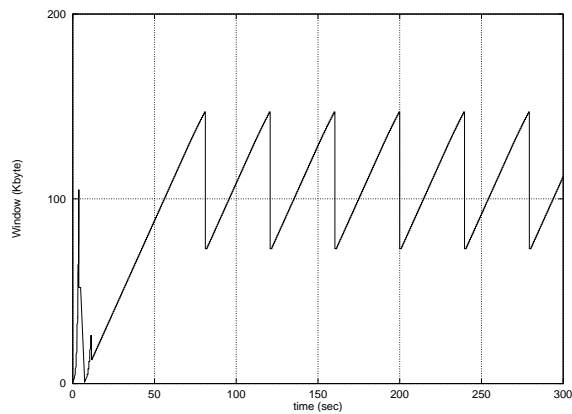


図 3.6: 最大ウィンドウサイズ 250Kbyte のデータ転送時のウィンドウサイズの変化

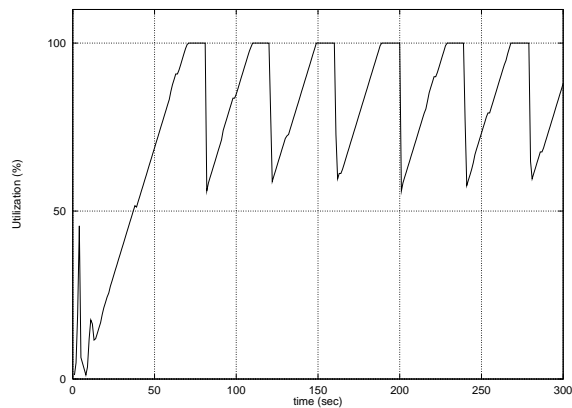


図 3.7: 最大ウィンドウサイズ 250Kbyte のデータ転送時の帯域占有率の変化

ズを推測するアルゴリズムの設計を行なった。

3.4.1 基本設計方針

新しいフロー制御機構は、以下の設計方針に基づいて行なった。

- 広帯域、高遅延ネットワークにおいて高いスループットを実現する。
- 狭帯域や低遅延のネットワークにおいても従来の TCP に劣らないスループットを実現する。
- TCP のウィンドウ更新アルゴリズムを変更する。
- スロースタート、輻輳回避アルゴリズムには変更を加えない

- CPU の負荷が大きくなりすぎないように、できるだけ簡単なアルゴリズムで実現する。

3.4.2 ウインドウサイズ決定アルゴリズム

最適なウインドウサイズを決定するためには、ネットワークからのフィードバックを利用する必要がある。しかし、現在のインターネットのアーキテクチャでは、様々な通信媒体の接続を可能にするため、ネットワークの状態をフィードバックする機構が最小限に抑えられている。また中継ノードがフローを制御する、始点抑制制御メッセージも定義されているがすべてのノードに実装されているわけではない。

本研究では、データを中継するルータに特別な機能を組み込まずに得られる情報を用いて、ネットワークの状態を推測するアルゴリズムを設計した。

インターネットのアーキテクチャでネットワークから得られる情報には以下のものがあげられる。

- ラウンドトリップタイム

ネットワークの帯域の利用率が上がると、中継ノードは、転送しきれないパケットを一旦待ち行列に加えて、処理に余裕ができるまで待機させる。このため、ネットワークの利用率が增加すると、ラウンドトリップタイムが増加していく。

- スループット

パケットが通過する経路上のネットワークは、少なくともコネクションが現在実現しているスループットを許容できる。

以上のパラメータに着目し、REAL シミュレータを用いて図 3.1 のネットワーク構成で、60M バイトのファイル転送を行なった。図 3.8 にラウンドトリップタイムの変化を、図 3.9 に帯域の利用率の変化を、図 3.10 にルータのバッファの大きさの変化を示す。

図 3.8 からラウンドトリップタイムが定期的に変化していることがわかる。図 3.9 と図 3.10 と比較と、帯域の利用率が 100% に近づいた時点からルータのバッファに転送待ちのパケットが溜り始め、ラウンドトリップタイムの遅延が大きくなり、輻輳の発生後にウインドウを閉じた時点で遅延が小さくなることが分かる。

図 3.11 にスループットの変化を示す。図 3.9 と比較することにより、帯域の使用率が 100% になった時点でスループットがあがらなくなることが観測される。

これらのパラメータに着目した結果、ラウンドトリップタイムの遅延が大きくなり始めた時点、及びスループットがあがらなくなる時点のウインドウサイズが最適なウインドウサイズになることがわかった。

以上のことから、ラウンドトリップタイムとスループットの比が最大になる点のウインドウサイズを最適なウインドウサイズと判断するアルゴリズムを設計した。図 3.12 にラウンドトリップタイムとスループットの比の変化を示す。

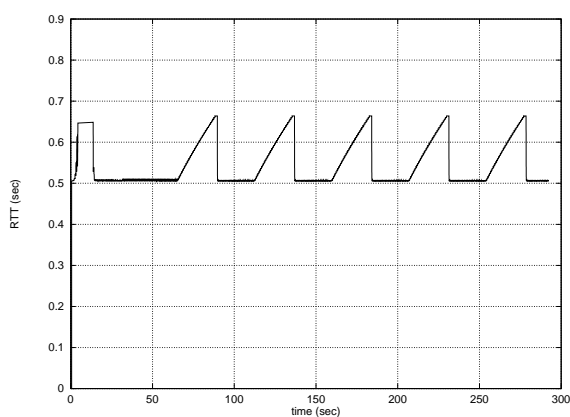


図 3.8: ラウンドトリップタイムの変化

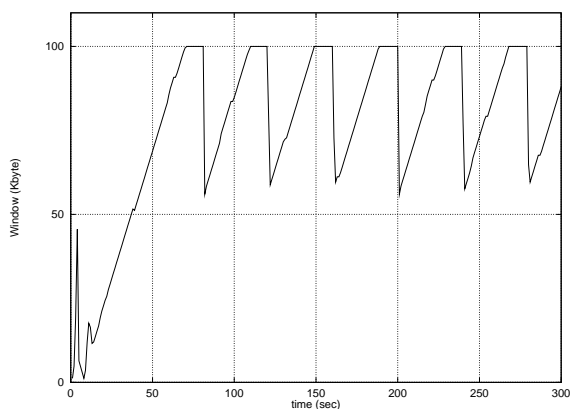


図 3.9: 帯域占有率の変化

3.4.3 ウィンドウ更新アルゴリズム

ネットワークの状態は頻繁に変化する。ある時点の最適なウィンドウサイズは、時間が経てば大きくなり過ぎたり、小さくなり過ぎる場合がある。現在のウィンドウサイズが最適なウィンドウサイズよりも大きい場合は、ラウンドトリップタイムやスループットの変化から判断することができるが、小さい場合は、これらの値があまり変化しない可能性があり、判別が難しい。

このため、一定の間隔で常にラウンドトリップタイムとスループットの比を観測し、ラウンドトリップタイムとスループットの比が増加している間は、ウィンドウサイズを増加させ、ラウンドトリップタイムとスループットの比が減少した場合、ウィンドウサイズを減少させる。

このアルゴリズムにより、ウィンドウサイズは、最適なウィンドウサイズの間を中心に

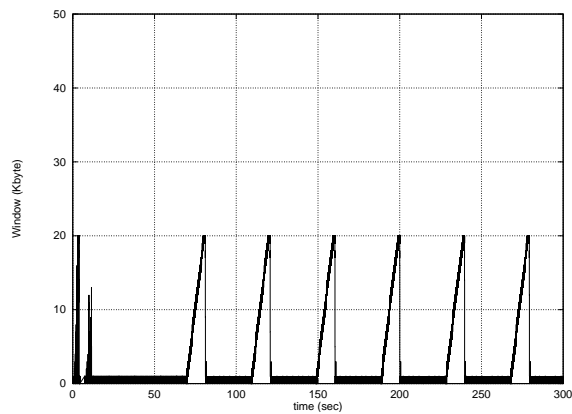


図 3.10: ルータの待ち行列の変化

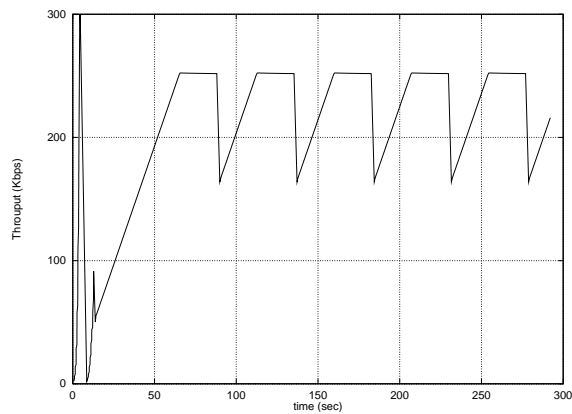


図 3.11: スループットの変化

振動する。ウィンドウサイズの増加、減少の割合を適切に設定することで、輻輳を起こさずに有効な帯域の利用ができる。

3.4.4 公平性

ネットワーク資源を公平に利用するためには、複数のユーザがネットワークを共有し、データ転送を行なう場合に、全てのユーザが同じ量の帯域を共有できなければならない。このような機能を実現するためには、

- ウィンドウサイズが小さい場合、ウィンドウサイズの増加の割合を大きくする。
- ウィンドウサイズが大きい場合、ウィンドウサイズの減少の割合を大きくする。

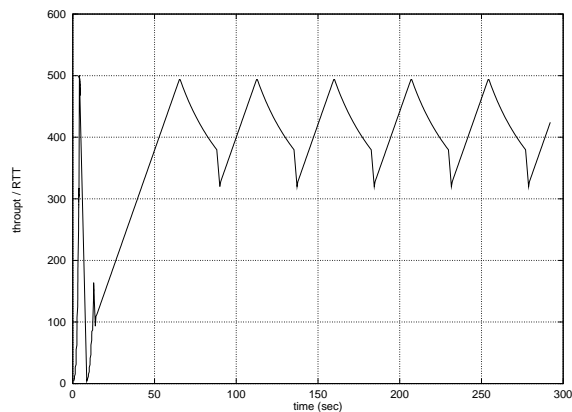


図 3.12: ラウンドトリップタイムとスループットの比の変化

というアルゴリズムが必要である。実装の負荷を考慮し、以下のような比較的単純なアルゴリズムでこの機能を実現した。

- ウィンドウサイズを増加する場合
 $\text{ウィンドウサイズ} = \text{ウィンドウサイズ} +$
- ウィンドウサイズを減少する場合
 $\text{ウィンドウサイズ} = \text{ウィンドウサイズ} \times (\quad < 1)$

3.5 REAL への実装

CPU やネットワークインターフェースの処理能力の制限を除き、アルゴリズムのみの評価を行なうために REAL[150] シミュレータ内に実装を行なった。

REAL シミュレータのフロー制御アルゴリズムは、アルゴリズム毎に一つのファイルにまとめられている。例えば、TCP Tahoe は `jk_tahoe.c`、TCP Reno は、`jk_reno.c` というファイルに C 言語の形で実装されている。今回の実装は、TCP Reno を原型とし、`jk_reno.c` を元にウィンドウ更新アルゴリズムに変更を加えたファイルを作成した。

図 3.13 に実装したウィンドウ更新アルゴリズムのフローチャートを示す。

ウィンドウサイズを下げた直後の X の値は、スループットが減少するため値が 1 つ前の X の値 `Last_X` よりも小さくなる可能性が高い。この場合、輻輳状態にないにもかかわらずウィンドウサイズを連続的に減少させてしまう。この状態を避けるために、ウィンドウサイズを下げた直後は、`Down_Flag` 変数を 1 にセットし、必ず一度ウィンドウサイズを増加させている。また、今回の実装では、設計の節で示したウィンドウサイズを増加する際に加算する量 α を回線の最大セグメント長に、振動の幅を小さくするため、ウィンドウサイズを減少させる際に乗じる β を 0.875 に設定した。浮動小数点演算を用いずに計算を行

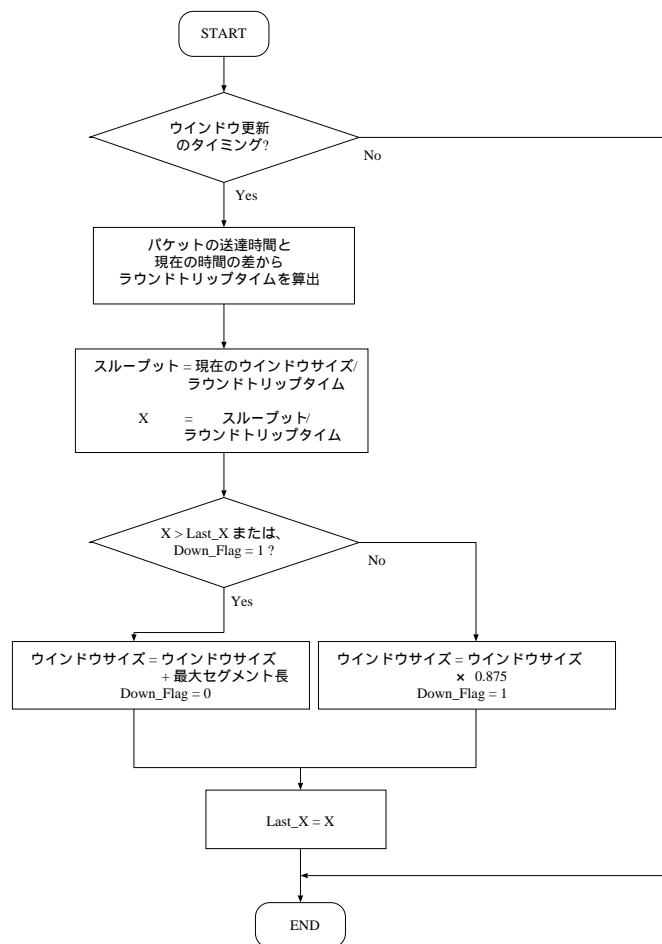


図 3.13: フローチャート

なうため、0.875 の乗算は、3 ビット右シフトした値を元の値から減じることによって実現している。

3.6 評価

3.6.1 広帯域、高遅延ネットワークにおける評価

本論文で設計したアルゴリズムの広帯域、高遅延ネットワークにおける挙動を評価するために、REALシミュレータを用い、図 3.1の構成を用いて 60Mbyte のデータ転送を行なった。図 3.1において、Ethernet の帯域は、10Mbps、伝搬遅延は 50nsec に設定した。ルータ 3、4 の間は衛星回線に接続されている。衛星回線の帯域は、2Mbps、伝搬遅延 500msec に設定した。ルータのバッファサイズは 40960byte とした。

図 3.14 にデータ転送時のウィンドウサイズの変化を示す。図 3.2 に示した従来の TCP のウィンドウサイズの変化と比較してウィンドウサイズが、最適なウィンドウサイズの近辺で小さく振動していることがわかる。

図 3.15 に帯域の利用率を示す。図 3.3 に示した従来の TCP の転送時の帯域の利用率と比較し、効率のよい帯域利用ができていていることが観測できる。

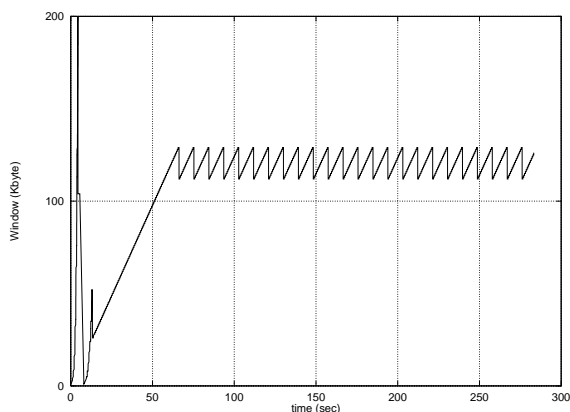


図 3.14: 新しいアルゴリズムのウィンドウサイズの変化

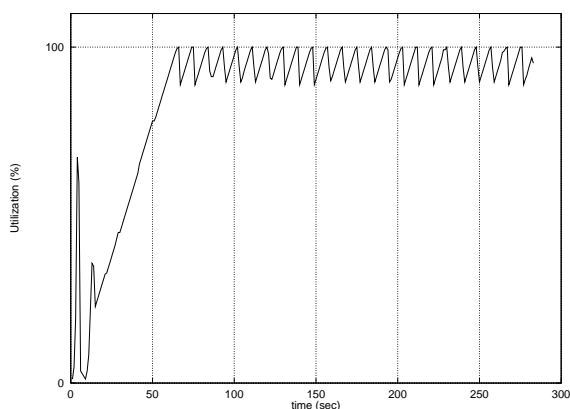


図 3.15: 新しいアルゴリズムの帯域利用率の変化

このシミュレーションでは、従来の TCP の平均帯域利用率が 76%であり、新しいアルゴリズムの平均帯域利用率が 85%であった。

3.6.2 公平性の評価

ネットワーク資源の公平な利用の評価するために、4つのコネクションを同時に開始し、ファイル転送を行なうシミュレーションを図 3.1の設定を用いて行なった。

図 3.16は、4つのコネクションが 50 秒毎に起動した場合のウィンドウサイズの変化を示している。図より各コネクションのウィンドウサイズの差が徐々に縮まり、最終的に同じ量に収束していることがわかる。図 3.17に帯域の利用率を示す。各コネクションが最適なウィンドウサイズへと遷移している状態でも 100%に近い帯域の利用ができていていることがわかる。

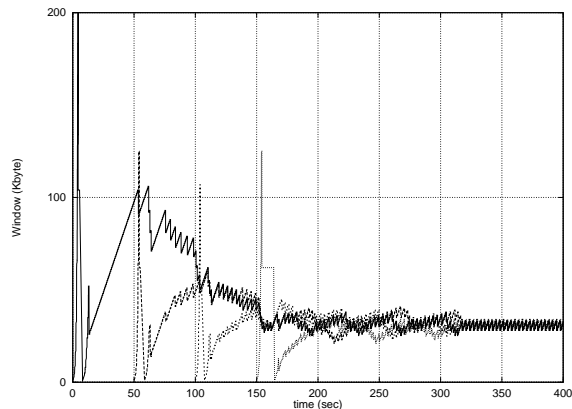


図 3.16: 50 秒毎にコネクションを開始した場合のウィンドウサイズの変化

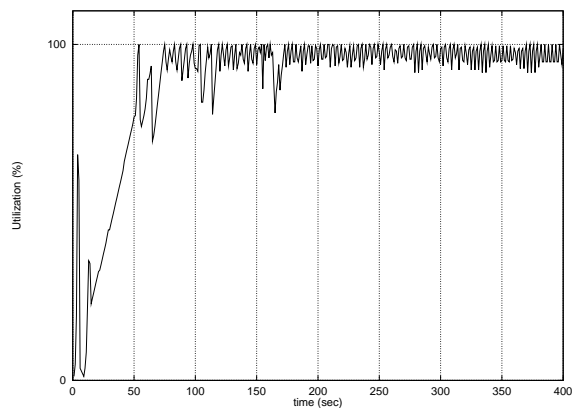


図 3.17: 50 秒毎にコネクションを開始した場合の帯域利用率の変化

3.7 まとめ

本章では、RFC1323 の問題点を考察し、その解決案を提案した。本章のアイデアは単純な構成の広帯域、高遅延ネットワークにおいて有効に機能すると思われるが、広域なネットワークでの振るまいに関してはより詳細な解析が必要である。今後はさらに複雑なトポロジでのシミュレーションや、実際のインターネット上での使用により解析を行ない、TCP

Vegas などの新しいアルゴリズムなどとの比較検討を行っていく予定である。

第 4 章

衛星通信ネットワークによるマルチキャスト配送 実験

4.1 はじめに

衛星通信は多くの特徴をもった通信媒体ではあるが、ここでは同報性と広域性に着目する。衛星通信媒体がそのフットプリント、すなわち、通信可能範囲全体にまたがる Ethernet のような同報媒体として使用できる機能を活かし、適当な経路制御とトポロジモデルを前提として地上のインターネットとの役割分担を適正に行なえば、広域にマルチキャストを効率よく転送できる可能性がある。衛星通信を用いたマルチキャスト部分と地上線を用いたユニキャスト部分の分散を全国レベルで実現し、そのモデルを検証した。

4.2 概要

11 月 30 日の「坂本龍一 Tour * 95 D&L with 原田大三郎」の武道館でのコンサートを実験の場として提供頂くことができた [151]。

コンサート会場の日本武道館に衛星通信局を設置し、慶應義塾大学湘南藤沢キャンパス (SFC)、九州大学、奈良先端科学技術大学院大学、(株)日本サテライトシステムズ技術開発センターの 4 地点へ向けて衛星通信ネットワークを通じて中継を試みた。

ツールとしては、nv、vat という従来の Mbone アプリケーションに加え、Stream Works を用いた。Stream Works は、Xing Technology 社による、動画像・音声をネットワーク上に配信するシステムである。ストリームジェネレータと呼ばれるハードウェアを使って画像・音声をエンコードし、ネットワークサーバ、プロパゲーションサーバと呼ばれる中継サーバにデータが送られる。ユーザには中継サーバからデータが配送される。ユーザと中継サーバとの間の通信にはユニキャスト (1 対 1 通信) で通信が行なわれるが、ストリームジェネレータと中継サーバ間、中継サーバどうしで通信を行なう際はユニキャストだけではなく、マルチキャスト (1 対多通信) も利用できる。そこでストリームジェネレータを日本武道館に設置し、上記 4 地点に中継サーバを設置した。

4.3 マルチキャストの経路制御とトポロジ

日本武道館と SFC の地球局との間では片方向 2Mbps のリンクを双方向で設け、それ以外の 3 地点では日本武道館からの受信のみを設定した (図 4.1)。これは、通信機器の制約から (1.2 節) 日本武道館で複数の受信を行なうことはできないからである。インターネットから日本武道館への通信は全て SFC を経由して行なわれる。

このような衛星通信ネットワークを形成することによって、日本武道館から送信されるマルチキャストパケットは全ての地点で受信することができるが、MBone への中継は SFC 経由だけにとどめ、それ以外の地点では MBone への接続は行なわない。よって、Stream Works ストリームジェネレータと各中継サーバの間は地上の MBone 上で結ばれることはない。

このようにネットワークトポロジを形成した理由は、現行のマルチキャスト経路制御の仕組みではパケットを転送する際に複数の経路が存在する場合、ユニキャストにおける逆向き経路 (Reverse Path) を基に判断をするからである。

すなわち SFC 以外の各地点が MBone によって日本武道館と接続されている場合、ユニキャストの逆向き経路は SFC を経由した経路となるため、衛星通信ネットワークからのパケットの転送をゲートウェイが行なわなくなるためである。

4.4 実験結果と考察

MBone での中継は 100 台程度のホストが視聴し、延べ 1 万回以上の Stream Works 中継サーバへの接続が確認された。

また、衛星を使用したマルチキャストバックボーン、そのマルチキャストバックボーンから Stream Works を経由してユニキャストでクライアントにデータを送り込むことは無事に成功した。この成功により、衛星通信ネットワークを用いたマルチキャスト通信の有効性が明らかになった。しかし、現行のマルチキャスト経路制御の方式の制約に合わせたネットワークトポロジの形成を図らなければならなかった。WISH の課題の一つとして、衛星通信ネットワークの特徴を活かすためにも単方向の衛星通信ネットワークでも機能するマルチキャスト経路制御方式の検討が不可欠である。

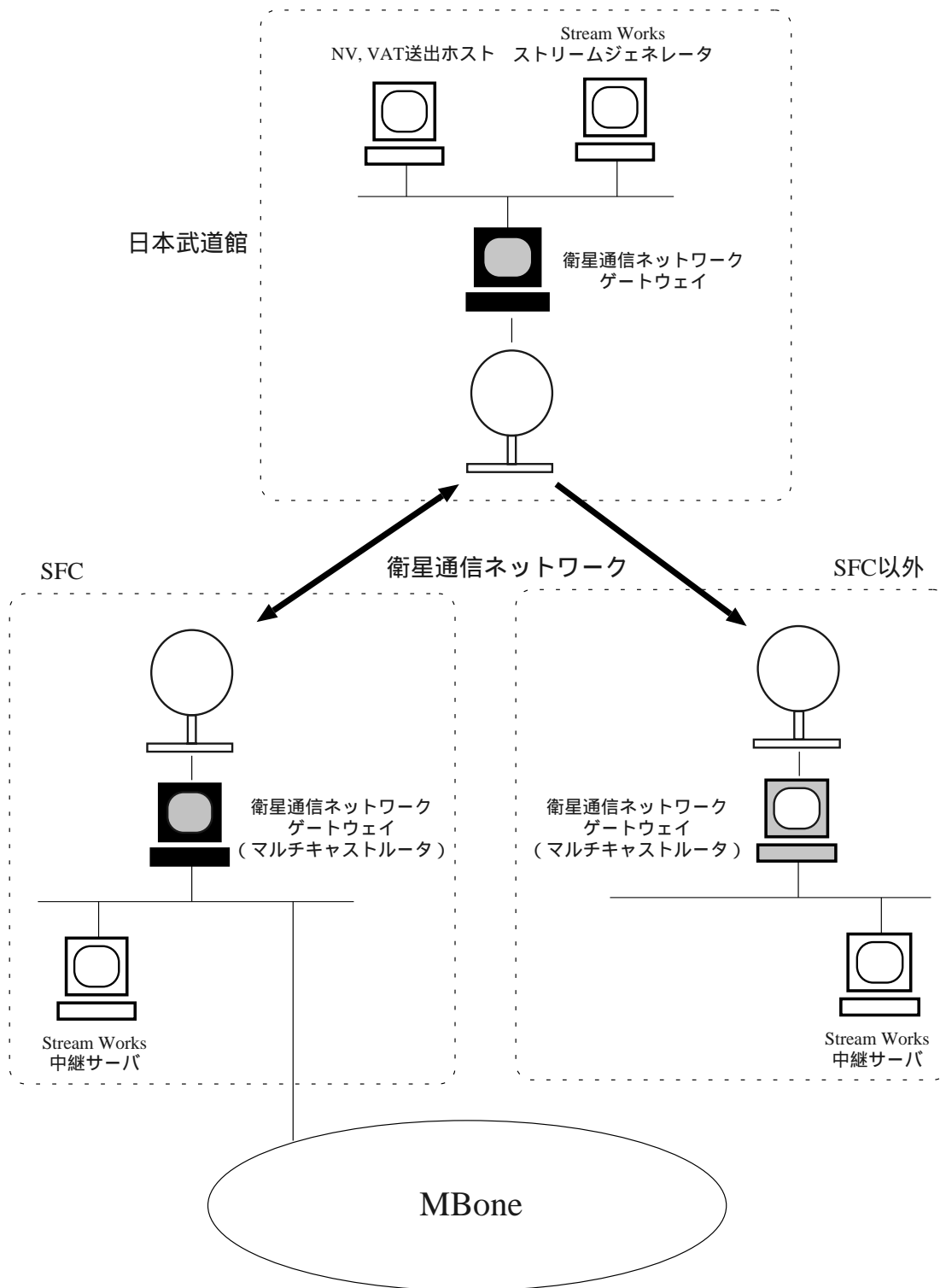


図 4.1: マルチキャスト配送実験ネットワークトポロジ

