広帯域ストリーム伝送を実現する動的なアプリケーションネットワーク制御方式の提案

岩崎　祐也[1,a)]　丸山　充[1]

（概要）近年ネットワークの高速・広帯域化により，マルチメディアコンテンツ，特に映像などの大容量なストリーミングデータ転送の需要が増加している．クラウドで広帯域ストリームデータを扱う場合には QoS(Quality of Service)が維持できず，ユーザの意図しているサービスを提供できない．このような問題の原因は，クラウドサービスにおいて，アプリケーション側がネットワーク側の状況を意識せずサービスを開始し，逆にネットワークがアプリケーションの状況を把握せずにパケットを転送することによる．そこで，本研究ではネットワークとコンピューティングリソースが協調し，アプリケーションとネットワークが連携することで，網状態に関わらず一定の QoS を確保する広帯域ストリーム伝送を実現するアプリケーションネットワーク制御方式を提案する．

1. 背景

現代のネットワークでは昔に比べ，インターネット利用者が莫大な数になっている．また，タブレットやスマートフォンの普及に伴い利用シーンの多様化が起こっている．従来のネットワークでは会社の書類などを外部に転送したり，取引先で社内の情報を閲覧したりするなど，ファイル転送を中心とした利用シーンが多かった．それに比べて現代では，オンライン会議・中継などのリアルタイム通信や，YouTube やニコニコ動画といったストリーミング転送が増加傾向にある．

このような時代の変化により，データセンターなどのクラウドサービスを提供する施設では，利用者の予測がつかずネットワークエンジニアリングが困難である．その結果，ストリーミング転送ではパケットの欠落が発生し QoS(Quality of Service)が維持できず，安定したストリーミング伝送ができない．

このような問題の原因は，クラウドサービスにおいて，アプリケーション側がネットワーク側の状況を意識せずサービスを開始し，逆にネットワーク側ではアプリケーションの状況を把握できないからである．

そこで，本研究ではネットワークとコンピューティングリソースが協調し，アプリケーションとネットワークが連携することで，網状態に関わらず一定の QoS を確保するアプリケーションネットワーク制御方式を提案する．

また，ネットワークの広帯域化に伴い，高精細のストリームデータを扱うアプリケーションの普及が見込まれている．例えばハイビジョンの 4 倍の解像度の 4K や 16 倍の 8K などがあげられる．本検討ではこうしたコンテンツの放送局品質の素材を対象に overGbps のストリーミング伝送によるネットワーク制御を実現する．

本論文の構成は 2. で提案方式の概要と課題について述べ，3. で提案方式の実現に向けたこれまでの取り組みについて述べ，4. で提案方式を実装した映像伝送システムについて述べ，5. で実験結果について述べ 6. でまとめと今後の課題について述べる．

2. 提案方式概要

QoS を維持する技術は，ネットワークで帯域を予約する方式とアプリケーションで帯域を予約する方式の 2 つに分類できる．

ネットワークで帯域を予約する方式は伝送経路上のルータで帯域を予約する IntServ(Integrated Services) 型（RSVP(Resource reSerVation Protocol)など）と，エッジ側のルータで帯域を予約する DiffServ(Differentiated Services) 型(MPLS(Multi Protocol Label Switching)の EXP ビットを使うなど)に分類される．

IntServ 型はアプリケーションのフローを各ルータで保持するため，ネットワーク全体に負荷がかかってしまう問題がある．DiffServ 型はエッジ側で各フローに対して優先度を設定できるが，優先度の低いフローは優先度の高いフローに圧迫されてしまう問題がある．

アプリケーションで帯域を予約する方式には RTSP(RealTime Streaming Protocol)などを用いてアプリケーションレイヤで制御する方式が一般的である．しかしアプリケーションレイヤでの帯域予約だけでは伝送経路上のネットワークの状況を見ていないため，レート制御ができない．

ここではベストエフォート型のネットワークを対象にネットワークとコンピューティングリソースが協調するリアルタイムな制御により，QoS を維持する方式を提案する．

例えば網状態の観測結果に基づき(1)空いているパスを動的に割り当て，(2)網の混雑区間で動的に冗長伝送を実現，(3)可用帯域に合わせて圧縮レートの変更を行うシステムを想定する．

具体的な構成は，図 2.1 のようにネットワーク制御部とコンピュータ処理部などの全体リソースを管理するリソースマネジメント(RM)，パケットに対する処理を行うコンピューティングリソース(CR)，ネットワーク制御・転送を行うネットワークリソース(NR)，の 3 つの層に分散させ，各層が協調して動作し，網状態の観測結果に基づき処理を行う．

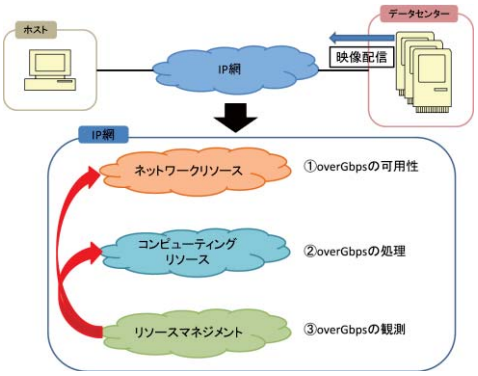

図 2.1 提案方式概要

1 神奈川工科大学　情報学部
Department of Information Engineering, Kanagawa Institute of Technology
a) iwasaki@nwlab.org

ⓒ2014 Information Processing Society of Japan

2.1 提案方式実現に向けた課題

提案方式を実現するにあたっていくつかの課題がある．
①ネットワークリソース部における overGbps でのネットワーク制御技術の可用性
②コンピューティングリソース部における overGbps でのコンピューティング処理の実現方法と配置法
③リソースマネジメント部の overGbps の観測手法

具体的な各層の処理を以下に示す．

ネットワークリソースはリソースマネジメントサーバの指示により，伝送経路のパスの切り替え，及びパケットの転送処理を行う

また，ネットワークリソースは単純な負荷分散のみであれば，従来のポリシールーティング機能を用いて構築することも可能であるが，本提案方式では網のパス設定に加えて途中のコンピューティングリソース内のノードに対して動的な接続切断が必要である．そのため網内でパケットの中身を書きかえることのできる OpenFlow を適用することにした．OpenFlow については 2．2 で述べる．

コンピューティングリソースは図 2.4 のように，リソースマネジメントサーバの指示により，入ってきたパケットに対してリアルタイム処理（圧縮・伸長・伝送方式の変換・映像フォーマット変換・暗号化）を行い網状に出力するものである．

例えば，4K 解像度(3840×2160)のストリーミング映像をハイビジョン画質に圧縮したり，その逆でハイビジョン画質を 4K 解像度に伸長したりする．

図 2.4　コンピューティングリソース

また，コンピューティングリソースにおける伝送方式の変換の一例として我々が実現した冗長伝送方式について述べる．[1]

図 2.5 のように，送信端末と受信端末の伝送パスにおいてパケット毎に，同一の内容を n 回，送出時刻をずらして転送する遅延制御を行う事で，単一パスでありながら，擬似的なマルチパス伝送を行う．しかし，この制御を End-End 間で行うと，全ての区間で帯域を n 倍消費するというデメリットがあるため，途中のバーストエラー障害がある区間だけノードで遅延制御するマルチパス伝送を実現し，有効性を評価した．このノード内で行っている遅延制御もコンピューティングリソースで実現する機能の 1 つである．

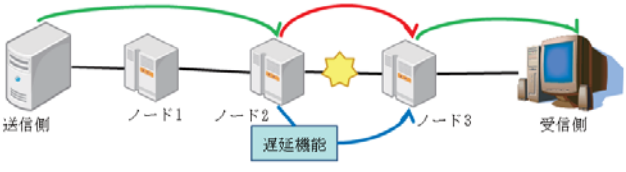

図 2.5 遅延制御によるマルチパス伝送

リソースマネジメントでは IP 網の状況を観測し，観測結果に基づきネットワークリソースとコンピューティングリソースに指示を行う．

実現に向けた課題として overGbps の観測をどうするかが重要であるため，２つの手法を検討した．

1 つは図 2.2 のように，伝送経路上にリソースマネジメントサーバ(RMS)を設置し，伝送している物理ポートの使用状況を管理プロトコルによって収集する Inband による観測手法である．管理プロトコルとして代表的なものは SNMP (Simple Network Management Protocol)がある．

2 つ目は図 2.3 のように，伝送経路上にスプリッタを設置し，観測専用の RMS を設置し，専用の観測プロトコルを用いて収集する Out of band による観測手法である．

前者の Inband による観測手法は，SNMP という既存のプロトコルを使用するためシンプルに構築でき，コスト面においても優れているが，SNMP の仕様上，インタフェースのカウンタ情報が反映されるまでに時間がかかるため，高精度な観測が行えない．また，スイッチの実装方法によってカウンタの更新時間に差が出てしまう．

後者の Out of band による観測手法は専用の装置とプロトコルを使うため，高精度な観測が行えるが高コストである．

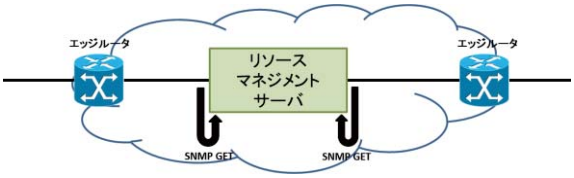

図 2.2　Inband による観測

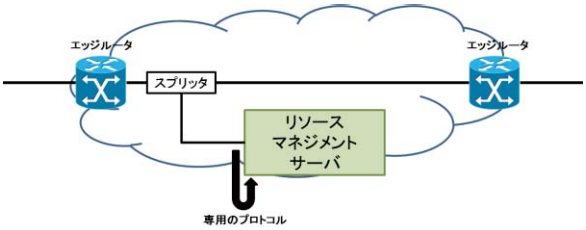

図 2.3　Out of band による観測

2.2 OpenFlow の適用

OpenFlow とは近年，次世代のネットワーク制御技術として注目されている SDN (Software Defined Network)の制御技術の 1 つである．OpenFlow は，従来のように MAC (Media Access Control)アドレスによるスイッチングや IP アドレスによるルーティングによりパケットを転送するのではなく，L1 ポート，VLAN (Virtual Local Area Network)タグ，MAC アドレス，IP アドレスなどの情報をフローとして扱い，それを基にパケットを転送する．

OpenFlow の代表的な特徴として，従来のネットワークではデータ転送と制御を行うプレーンを 1 つのプレーン（ネットワークリソース）として扱っていたが OpenFlow ではパケットの転送を行うデータプレーンとパケットの制御を行うコントロールプレーンを分離し，専用の装置が連携して処理することで，柔軟なネットワーク制御を可能としている．具体的にはデータプレーンの処理を担当する OFS (Open Flow Switch)とコントロールプレーンの処理を担当

ⓒ2014 Information Processing Society of Japan

する OFC (Open Flow Controller)から構成されている．

OpenFlow の基本的な動作は match と action である．match とは OFS がもつフローテーブル内で受信したフローに一致することを指す．action は match したフローに対する処理を行うことである．matching ルールには表 2.1 の種類があり，action には表 2.2 の種類がある．

OpenFlow による処理の流れは図 2.6 のように，パケットを受信した OFS は自身の持つフローテーブルを参照し，パケットに合致するルールが無いかを探す．

match したルールがある場合，それに従いパケットに対する action を行う．

match するものが無い場合，OFC にパケットに対する処理を問い合わせる Packet_in イベントを発生させる．Packet_in イベントを受信した OFC はパケットに対する振る舞いをソフトウェアのプログラムに従って決定し，結果を OFS に応答する．その後，OFS は OFC から送られてきたフローを新しいフローエントリーとしてフローテーブルに記憶し，パケットに対して action を行う．

表 2.1　matching ルール

要素	レイヤ	内容
Ingress Port	L1	L1物理ポート
Ether src	L2	送信元MACアドレス
Ether dst		宛先MACアドレス
Ether type		イーサネットの種別
VLAN id		VLAN ID
VLAN priority		VLAN 優先度
IP src	L3	送信元IPアドレス
IP dst		宛先IPアドレス
IP proto		IPプロトコル種別
IP ToS bits		ToS値
TCP/UDP src port	L4	送信元L4ポート番号
TCP/UDP dst port		宛先L4ポート番号

表 2.2　action

action	内容
Forward	パケットを指定した宛先ポートに出力
Drop	パケットを破棄する
Modify-Field	パケットの中身を書き換える

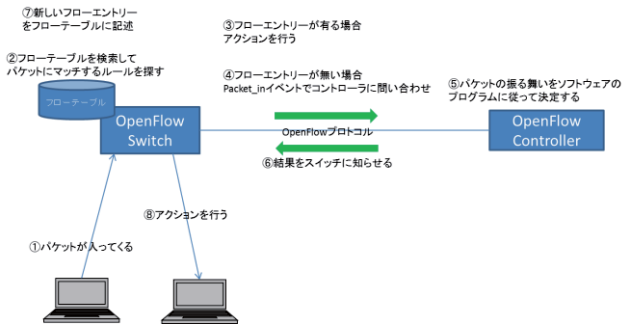

図 2.6　OpenFlow の処理の流れ

3.　提案方式の実現に向けたこれまでの取り組み

広帯域ストリーム伝送に向けたアプリケーションネットワーク制御方式を実現するために以下のように取り組んだ内容を概説する．

ネットワークリソース部の overGbps の可用性については 3.1 で述べる「OpenFlow ベースの overGbps の可用性」で検証した．

リソースマネジメント部の overGbps の観測をどうするかについては in band による観測との比較として 3.2 で述べる「overGbps の Out of band での高精細観測」で検証した．

コンピューティングリソース部の overGbps でのコンピューティング処理の実現方法と配置法については 4. において図 3.1 に示すトランスコーダを実装した映像伝送システムを題材に構築し，アプリケーションネットワーク制御方式について検証する．

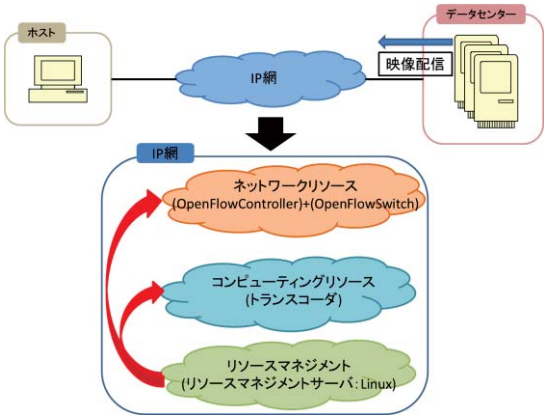

図 3.1　アプリケーションネットワーク制御方式実装

3.1　OpenFlow ベースの overGbps の可用性

ネットワークリソース部の overGbps の可用性について検証するため，ストリーミングゲームクラウドシステムを構築した[2]．コンセプトは，自宅のゲーム機器を広帯域ネットワーク経由でゲームクラウドに接続すると自宅のゲーム機器とゲームクラウドが連携して多画面ゲームができるシステムである．

これを実現するにあたっては，自宅側とゲームクラウドから送出される映像の同期の点から低遅延性の確保が必要となるため，ゲームクラウドからの映像を非圧縮で送る事とした．ゲームクラウドから送る映像は最大ハイビジョン品質 4 本分の映像となるため，ネットワークの帯域を 1.6Gbps×4 だけ確保する事となるため，常時確保するのは効率的ではない．このため，OpenFlow 技術を使ってゲームをしている最中だけネットワーク帯域を確保するシステムとして作成した．

提案システムは，図 3.2 のようにゲーム機 1 台をマスタとして自宅のクライアント部に置き，残り 4 台をスレーブとしてゲームクラウド部に置き，ネットワークセンタ(NC)を介して映像を伝送する．クライアント部のマスタのゲーム機はゲームクラウド部のスレーブのゲーム機器群と制御情報のやり取りをする．また，ゲームクラウド部から送られてくる IP パケット化されたゲームの映像を 4K Gateway で非圧縮 HD(1920×1080) 4 本分の映像信号に変換し・分配する．NC 部は OpenFlow コントローラの Trema によって制

ⓒ2014 Information Processing Society of Japan

御される複数台の 10GbE と 1Gbps イーサネット(GbE)のインタフェースを持つ OpenFlow スイッチから構成される．

OpenFlow スイッチがゲーム機器間のゲーム制御情報を検出すると，Trema からの指示によって 10GbE のネットワークを用いて広帯域の映像パスを設定する．同時にゲーム終了の検出のためゲーム制御情報が途切れて一定時間経つとパスを解除するためのタイマーを起動する．ゲーム中はクラウド部のゲーム機は同期を取りながら 4K Gateway で各ゲーム機から送信された映像信号を非圧縮のまま 1 つにまとめ IP パケットに変換する．映像伝送レートは最大で 6.4Gbps である．

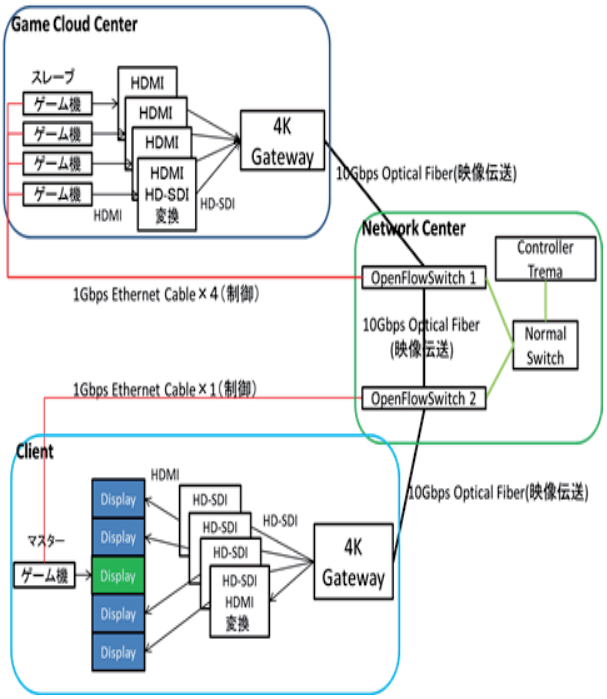

図 3.2　ハイビジョン多画面映像伝送によるゲームクラウドシステム

図 3.3 にゲームを体験している模様を示す．中央がクライアントのゲーム機器の映像出力であり，両端の 4 台がゲームクラウドから伝送されてきた映像で，同期遅延は 1 フレーム（33ms）以内に十分収まった．

この結果より，OpenFlow スイッチを用いて overGbps の広帯域ストリームデータをパケットロス無しでスイッチング制御が行えることを確認した．

図 3.3　ゲームクラウドシステムの体験模様

3.2　overGbps の Out of band での高精度観測

リソースマネジメント部の overGbps の観測を検証するため，2013 年度に神奈川工科大学内に以下の特徴を有するストリーミングクラウド実験設備を配備し，機能要素が連携して集中制御する事で広帯域なストリームデータを安定的かつ即時に配信可能な実験設備として構築した．

・10Gbps 対応のオープンフロースイッチをベースにしたストリーム指向仮想ネットワーク制御機能

・高精度ネットワークモニタを外部テストベッド内のモニタ設備と連携させた多面的高精度ネットワーク品質監視機能

・4K リアルタイム映像伝送機器　PFU 社 QG70[3]

・4K 映像蓄積配信装置　NTT-IT 社 SHS-XMS[4]

ストリーミングクラウド設備は，情報通信研究機構（NICT）の JGN-X テストベッドへ 10Gbps の帯域で接続を行った．また大学内に 10ns の粒度でキャプチャが可能な PRESTA 10G[(3)] のネットワークモニタ機器を導入した．これにより，高精度なネットワーク観測が可能となり，特に広帯域ストリーム伝送での伝送特性を細かく観測する事が可能である．また，本装置は，JGN-X 内の大手町，堂島，天神の 3 箇所にも設置されており，これを多地点で連携して観測する事ができる．

上記で説明した設備を使って，奈良先端科学技術大学院大学（NAIST）との間で，4K 映像を題材にリアルタイムおよび蓄積配信実験を行った[5]．

3.2.1　リアルタイム 4K 映像伝送実験

NAIST との間で，図 3.4 に示す太平洋周りおよび北陸周りのパスを設定し，QG70 を用いて 4K テレビ会議のような掛け合い環境を作成した．それぞれのパスのラウンドトリップ時間は，12ms と 24ms である．

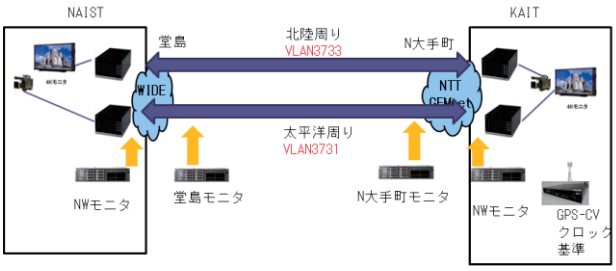

図 3.4　NAIST-KAIT 間リアルタイム映像実験

これを従来のネットワーク観測手段の 1 つである Cacti で観測すると，図 3.5 のように 6.4Gbps の綺麗な一定レートが観測できる．

ⓒ2014 Information Processing Society of Japan

図 3.5　Cacti による NAIST の観測データ

これを高精度ネットワーク測定装置で測定すると，図 3.6 のように各地点でのマイクロバースト性を同時に観測する事ができ，障害箇所の特定に役立つ事が分かった．

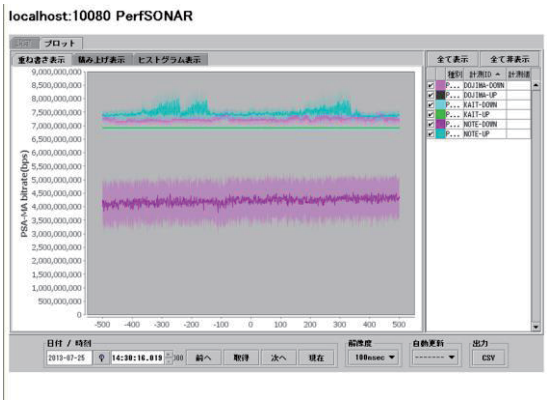

図 3.6　高精度ネットワーク測定装置の観測結果

4.　アプリケーションネットワーク制御方式の実験概要

アプリケーションネットワーク制御方式を検証するために，図 4.1 のような映像伝送システムを構築した．

映像伝送システムでは，データセンターからの映像配信サービスをイメージした 4K カメラの映像をネットワークリソースがホスト側の 4K ディスプレイに伝送するというものである．このときリソースマネジメントにより IP 網を監視し，網が混雑していないときには約 6.9Gbps の非圧縮の 4K 映像がホスト側に届けられ(Normal Path)，混雑時にはコンピューティングリソースのトランスコーダが起動し 4K 映像を変換し約 1.7Gbps の HD 画質が伝送され(Transcode Path)，ホストに届けられる直前で HD 画質を 4K 映像に再変換している．これによりホストには IP 網が混雑しているかどうかにかかわらず常に 4K 品質の映像が届けられる．

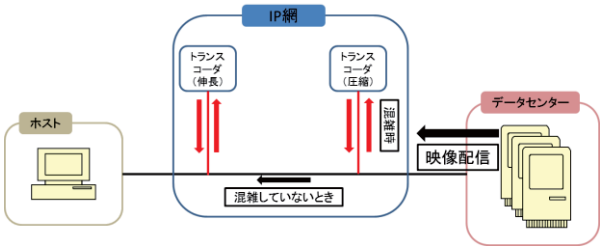

図 4.1　映像伝送システム概要図

4.1　実験の構成図と使用した機器

映像伝送システムの構成を図 4.2 に示す．

リソースマネジメントには PICA8 社の P-3290 に SNMP エージェントを設定し網の観測装置として利用した．また，リソースマネジメントサーバとして Linux がインストールされたノート PC を利用した．

ネットワークリソースには OFS として PICA8 社の P-3290 を使用し，OFC にはデスクトップパソコンを用いて Trema-edge をインストールして構築した．

コンピューティングリソースには PFU 社の QG70 と各 Converter を IP 網内のクラウド上で動作する擬似的トランスコーダとして構築した．DownConverter には AJA 社の 4K2HD を使用し，UpConverter には計測技術研究所製の FE-B1 を使用した．

その他の機材として，カメラは JVC 社の MVH502A を使用し，ディスプレイは TOSHIBA 製の 55XS5 を使用した．

4.2　本実験における各プレーンの動作

リソースマネジメントでは一定時間ごとにリソースマネジメントサーバから P-3290 に SNMP GET を送信し port のカウンタの値から網の負荷状況を計測する．トラヒック量の計算には 1 回目に取得した 64bit カウンタの値を α，2 回目に取得した 64bit カウンタの値を β，カウンタの再取得までに要した時間を t として以下の式を用いて算出した．

$$\text{トラヒック量} = (\beta - \alpha) \times 8 \div t$$

計測に使用する t の値を小さくするほどリアルタイムに計測できるが，SNMP の仕様上カウンタの値が反映されるまでに時間がかかるため，トラヒック量の精度が低下する．逆に t の時間を大きくするほど，精度が高くなるが，リアルタイム性が損なわれる．今回の実験ではパスの切り替えを確認するため，スイッチの P-3290 の SNMP エージェントの仕様上，t の値を 180 秒として計測した．

この計測値と閾値を基にパスを切り替える必要があるかどうかを判断し，OFC に指示を出す．今回の場合，閾値はパスの切り替えを確認するための値であるため 7Gbps とした．

非圧縮の映像を伝送する Normal Path と圧縮して伝送する Transcode Path の経路を図 4.3 に示す．

パスの切り替え判定は図 4.4 のフローチャートを使用した．変数 send_path は最終的に OFC に指示を出すパスの値で，0 はパスに変更が無いことを示し，1 はトランスコーダを経由しない最短のパスを示し，2 はトランスコーダ経由のパスを示す．変数 old_send_path は前回 OFC に指示を出した send_path の値を指す．

ⓒ2014 Information Processing Society of Japan

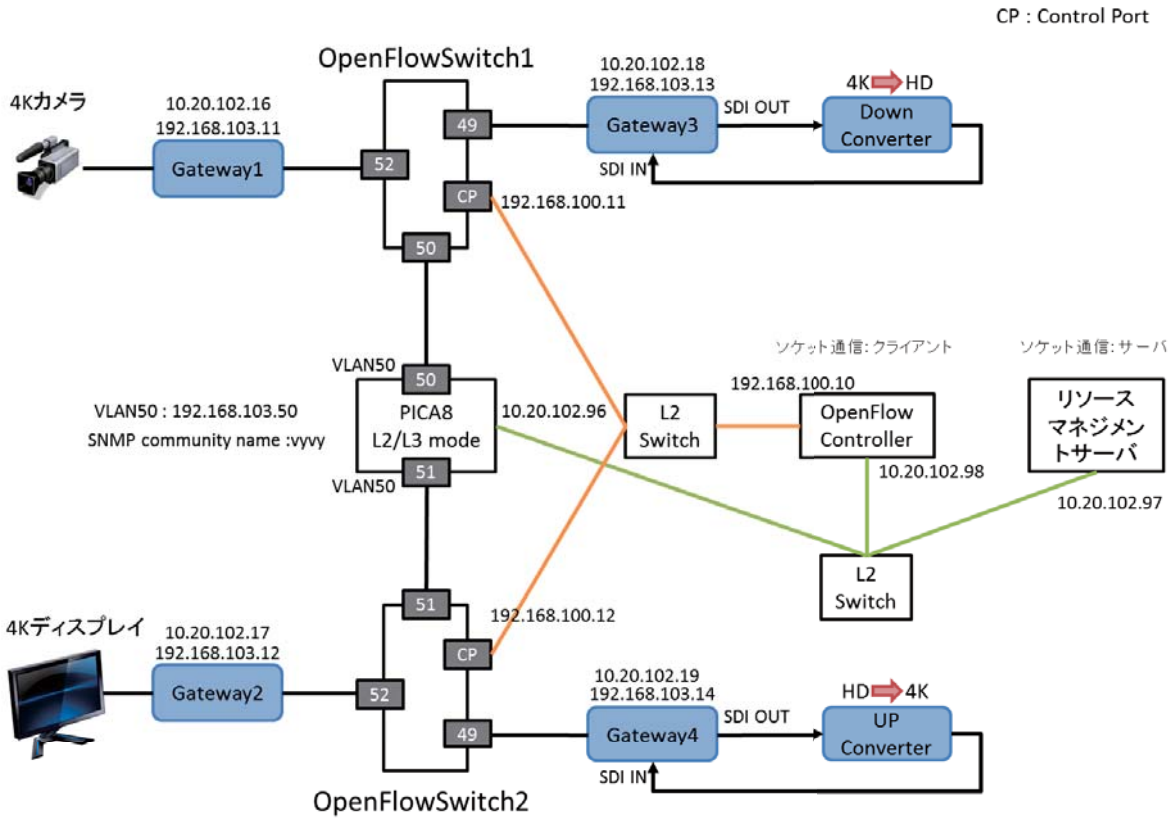

図 4.2　映像伝送システム構成図

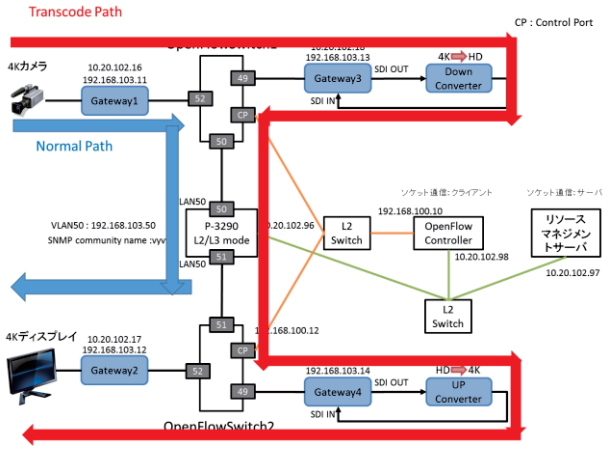

図 4.3　Normal Path と Transcode Path

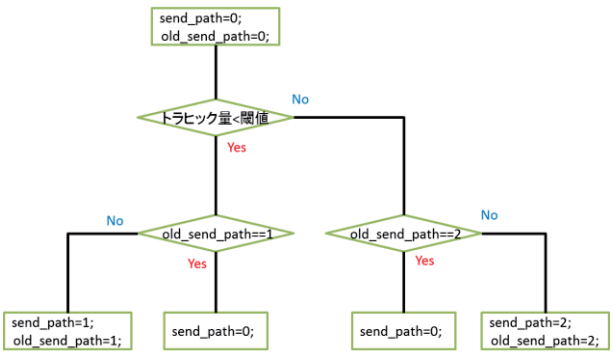

図 4.4　パス切り替えフローチャート

ネットワークリソースでは OFC が一定時間ごとにリソースマネジメントサーバ(RMS)にソケット通信を行い，パスに変更が無いかを問い合わせる．この問い合わせ結果をもとに OFS に send_openflow_message を用いてフローを登録する．

登録するフローは IP 網に負荷が無い場合の伝送経路(Normal Path)と IP 網に負荷がある場合の(Transcode Path)の 2 種類である．詳細は以下の通りである．

Normal Path の場合に登録するフロー

OFS1：Flow1　match import が 52 action outport が 50

OFS2：Flow1　match import が 51 action outport が 52

Transcode Path の場合に登録するフロー

OFS1：Flow1　match import が 52 action outport が 49

OFS1：Flow2　match import が 49 action outport が 50

OFS2：Flow1　match import が 51 action outport が 49

OFS2：Flow1　match import が 49 action outport が 52

OFS は OFC から受け取ったフローをフローテーブルに登録する．パケットを受け取るとフローテーブルに書かれたフローを基にパケットの転送を開始する．

コンピューティングリソースでは Down Converter と Up Converter から構成されている．Down Converter は映像データを受け取ると 4K 映像を HD 映像に変換する．Up Converter は HD 映像を受け取ると 4K 映像に変換する．

ⓒ2014 Information Processing Society of Japan

5. 実験結果

始めに，Normal Path を経由してカメラからの映像が 4K ディスプレイに表示されることを確認した．尚，カメラからの映像がディスプレイに表示されるまでの遅延に関しては図 5.1 に示すように，100ms~200ms であった．

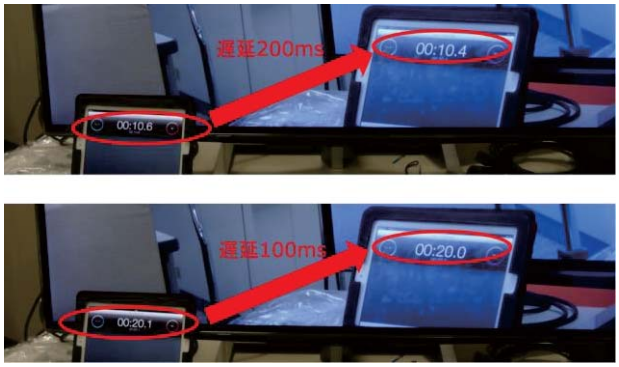

図 5.1　Normal Path の映像

次に，伝送経路の切り替えを行うため 2 台の PC を Open Flow Switch1 に接続し，jperf を用いて負荷を発生させたこれによりリソースマネジメントが IP 網の負荷を検知し，伝送経路の切り替え指示をネットワークリソースに出したことを確認した．

この指示を受けたネットワークリソースでは Normal Path を Transcode Path に切り替えるように，各 OFS に指示を出したことを確認した．

Transcode Path になったことでコンピューティングリソースが動作しパケットが圧縮されて伝送され IP 網を出る直前で 4K 映像化され，4K ディスプレイに表示されることを確認した．

尚，カメラからの映像がディスプレイに表示されるまでの遅延に関しては図 5.2 に示すように，300ms~400ms であった．

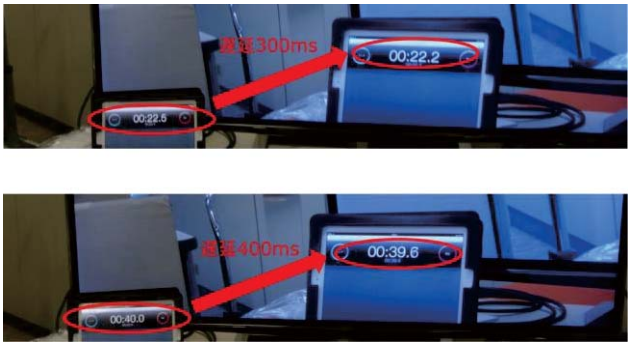

図 5.2　Transcode Path の映像

パスの切り替えにはリソースマネジメントの RMS から SNMP の観測に要する時間が 180s であった．

また，RMS から OFC へ伝送経路の切り替え指示，OFC から OFS へのフローテーブルの更新処理，Up Converter と Down Converter によるトランスコーディング時間で計 2.8s~3.5s の時間を要した．この時間は別途，Normal Path と Transcode Path を手動で切り替えるプログラムを作成し，図 5.1 や図 5.2 のようなタイムレコードを入れた動画を撮影し計測した．

実験の結果，網内のコンピューティングリソースを動的に割り当て，リアルタイム入出力を実現するパイプライン処理方式が実現できることを実証した．

6　まとめと今後の課題

提案方式による実証実験ではパスの切り替えには全体で 183s 程度かかった，この内，リソースマネジメントの IP 網の監視に 180 秒掛かるのに対して，それ以外の処理時間が 2.8s~3.5s であった．

この結果より，伝送経路の切り替えにはリソースマネジメントサーバによる IP 網の観測時間がボトルネックとなり，リアルタイム性が損なわれていることが分かった．

原因は，SNMP のカウンタの更新のために 180s 待つことが理由であるが，SNMP の仕様上，カウンタの更新の待ち時間を短くした場合，カウンタの値が正常に取得できなくなってしまう．

こうした問題を解決するため，今後は 3.2 でも挙げたように，Inband による監視ではなく図 6.1 のような Out of band による監視を行うことで，より高速に，精度の高いリソースマネジメントを構築する．最終的には映像の無瞬断伝送ができるシステムを目指す．

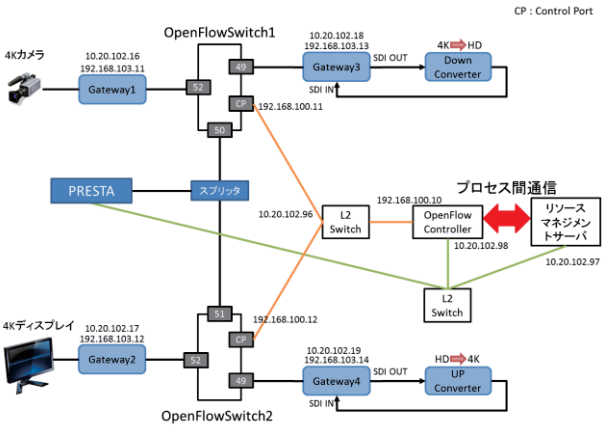

図 6.1　Out of band による監視

また，本システムを用いて JGN-X での実網での実験評価を進めていきたい．

3.2 の実験にあたっては，NICT，NAIST，NTT アイティ，(株)PFU，NTT 未来ねっと研究所の皆様の協力を頂いたことに深く感謝する．本研究の一部は，JSPS 科研費 24800069,26330121 の助成を受けたものである．

参考文献

[1] 丸山 充, 岩崎 祐也, 菅谷 祐介, 白須 雄太 “OpenFlow 技術を用いたストリーミング伝送の制御” 神奈川工科大学情報教育センター研究報告 2013,Vol.8.ISSN1882-0646.

[2] 岩崎 祐也，丸山 充，朴 美娘 “OpenFlow を用いたゲームクラウドの実現” 2013 年電子情報通信学会 総合大会 B-7-80,2013.

[3] 鍋谷栄展，小林正之，田中篤史，山崎恭啓，“Qool Tornado QG70,” PFU Technical review 2013 年 5 月 Vol.24, No.1, 45 号.

[4] 君山博之，小倉 毅，丸山 充“オーバー50Gbit/s PC クラスタ型ストリームサーバの構成法，” 情報処理学会論文誌,54(12),2413-2426 (2013-12-15) , 1882-776.

[5] 丸山 充，油谷曉，垣内正年，大槻英樹，小林和真，酒井昌男,小林正之，“リアルタイム指向ネットワークコンピューティング技術を用いたストリーミングクラウド機能の検証,” 信学技報, vol. 113, no. 256, IA2013-41, pp. 1-6, 2013.

ⓒ2014 Information Processing Society of Japan