ネットワーク仮想化を用いた ホームネットワーク管理手法の提案

長谷 錦[1] 今野 裕太[1] 吉村 悠[1] 佐藤 健哉[1]

概要：今日ではネットワークはその発展・普及とともに人々の生活において欠かせないものとなっている．最近では，ネットワークに接続し便利な機能を提供する生活家電などの普及が進んでおり，ホームネットワークというものの重要度が増している．今後もホームネットワークに接続することでの利用を前提とした機器の増加や，それぞれのサービスの高品質化などによるホームネットワーク内で扱う情報量・種類の増加は避けられないものとなる．ホームネットワーク内の機器の増加によりネットワークは複雑化し，ホームネットワークの管理は難しくなる．さらにホームネットワーク内のデータ量・種類の増加により，通信品質の低下という問題が発生する．本稿ではネットワーク仮想化技術の一つであるネットワークスライスを，SDN（Software-Defined Networking）の代表的な技術である OpenFlow 用いることで実現し，ホームネットワーク内の機器を機能ごとにスライスに振り分け，スライスごとの自動的なネットワーク制御を行うことでネットワーク知識の浅い人であってもホームネットワークの管理を容易にする手法を提案する．提案手法を用いることで，ネットワークスライスを用いた自動的なネットワーク制御が可能であることを証明し，ホームネットワーク内に発生する上述した問題が解決できることを示した．

A Proposal of the Home Network Management Method Using Virtualization

NISHIKI HASE[1] YUTA KONNO[1] HARUKA YOSHIMURA[1] KENYA SATO[1]

1. はじめに

近年，ネットワークに接続可能な家電が多く発売されている．このような家電はネットワークに接続することで，家電間で情報の共有を行い，それを用いた制御が可能となる．人々の生活を便利で豊かなものにするために，ホームネットワークに接続できる家電機器は増加し，ホームネットワークの発展は進み，機器間での情報共有や連携が多く行われる．ドアホンのカメラの高性能化，空調機器のセンサーの増加などがその例である．他にも NAS などの記憶装置から，家中どこでも情報を受け取れるような環境になっている．

また，映像コンテンツなどのサービスの高品質化により，ホームネットワーク内で扱う情報のデータ量は増加している．4K や 8K といったテレビの高画質化や，ハイレゾといった音声データの高解像度化のように，一般家庭においてもこれまでは高価であった商品も比較的安価で手に入れることができるようになった．

ホームネットワークがより便利になっていくにつれ，ホームネットワーク上で扱う情報が複雑化し，機器ごとに QoS 制御など適切な管理が必要となるが，接続する機器数の増加によりその管理コストは大きくなる．ネットワーク仮想化技術はそのように複雑化しているネットワーク管理を自動化し，より柔軟な管理を可能とする技術として注目されている．本研究ではネットワーク仮想化技術を用いたホームネットワーク管理手法を提案する．

2. 問題点

本章ではホームネットワークに発生する問題について指摘する．

2.1 ホームネットワークの複雑化

ホームネットワークが複雑化するという傾向は今後も続

[1] 同志社大学大学院理工学研究科情報工学専攻

き，ネットワーク知識の浅い人が管理することの多いホームネットワークにおいては大きな問題となる．

ホームネットワークは小規模ながらも制御系やストリームデータ伝送系など，様々なアプリケーションが混在し，異なる性質の通信が発生する傾向がある [1]．外部との通信を行う機会の多いホームネットワークでは，セキュリティの保障も問題となる．悪意ある第三者によって通信内容を盗み見されることでパスワードや個人情報が漏出，外部からホームネットワーク内の機器の遠隔操作などによる被害を受ける可能性も高い．

加えて，ホームネットワークはネットワークの構成が頻繁に変更されるという特徴がある．新たに購入した機器の設置や，すでにある機器を部屋の模様替えに合わせて移動させたりという場面も少なくない．ホームネットワークに新たに機器を追加する場合，その機能に合わせて適切にネットワーク設定を追加していく必要があるが，何度も新しい機器を追加していくうちにネットワーク構造は煩雑化し，それぞれの機器のネットワーク設定や連携する機器間の接続関係を適切に設定できなくなるといった問題が発生する．

2.2 通信品質

前述した問題によりホームネットワークの設定が適切に行われていないと，動画・音声の遅れや乱れといったような通信品質の低下という問題が発生する．同じネットワーク内で同時に複数の作業が行われる状況，例えば片方は音声通話を行い，もう片方は映像コンテンツを視聴しているような場面を想定する．同じ帯域を利用しているのでお互いの通信で帯域が圧迫され，音声データが途切れて会話に支障をきたすような事態が発生する．

ネットワークに接続できる家庭内の機器の数が増加し，この先より多くの情報を扱うことが予想されるホームネットワークにおいては，通信品質の低下というのは重大な問題となる．

2.3 関連研究

ホームネットワークを最適に利用することを主眼に置いた研究として，Software-Defined Networking（以下，SDN）技術を用いてホームネットワークを管理するものがある．OpenSDN の一般的なプロトコルである OpenFlow を用いて動的な経路選択を行うことでホームネットワークの最適化を行っている [2]．他にも，ビデオストリーミング機器などのリッチな機器がホームネットワークに接続されることを考え，仮想化を用いてホームネットワーク上の通信を制御する研究もなされている [3]．

本研究では単にホームネットワーク上での情報のやり取りを最適化するだけでなく，機器の増加や機能の高性能化に伴って複雑になるホームネットワークの構築・管理を，ネットワーク仮想化技術の一つであるスライスを用いることで自動で行い，ネットワーク知識の浅い人であっても容易に管理を行うことができる手法を提案する．

3. 提案手法

本章ではネットワーク仮想化を利用してホームネットワークを管理する提案手法を解説する．

3.1 概要

提案手法の概要を図 1 に示す．図中における楕円はホームネットワーク内の機器を示している．提案手法では，ホームネットワークに接続している各機器をその機器の持つ機能ごとに仮想化された各ネットワークに割り当てることによって，機器の追加などのホームネットワークの管理を容易にし，新たに機器が追加された際にもその機器の持つ役割を把握し自動で適するスライスに割り当てる．さらにネットワークごとに帯域などを保障することにより通信品質を向上させる．

第 2 章で述べた問題を解決するために提案手法では OpenFlow を利用し，ネットワークの仮想化・スライス化を行う．ホームネットワークを仮想化しスライス化することでホームネットワークの管理を容易にし，ネットワークが複雑になるという問題を解決する [4]．その上でスライスごとに帯域等を保障することで，通信量の増加により発生する通信品質の低下という問題を解決する．

3.2 仮想化

本研究では，ホームネットワークに対して仮想化技術を用いてネットワークスライスを作成する．仮想化とは主にコンピュータのリソースを抽象化することを指し，代表的なものに一台の物理サーバを複数の仮想的なサーバに分割して利用するサーバの仮想化などがある．仮想化を用いることで，一般的にコストの削減や，拡張性，耐障害性等の

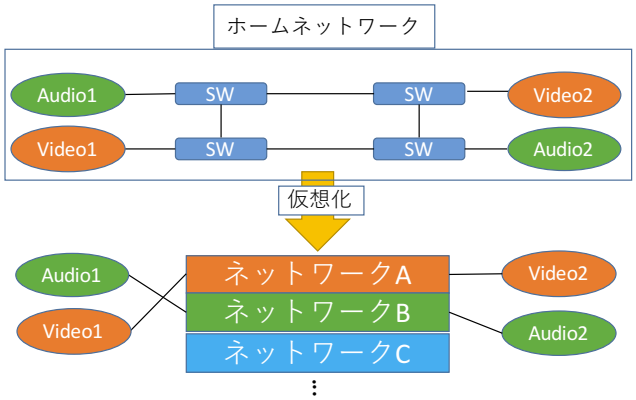

図 1 提案手法概要図

向上が図られる．本研究では，ネットワークの仮想化を用いてネットワークスライスを実現する．

3.3 ネットワークスライス

ネットワークスライスの概要を図 2 に示す．ネットワークスライスとは，仮想化技術を用いて一つの物理ネットワークを複数の独立した仮想ネットワークに分割する技術のことで，それぞれのネットワークをスライスと呼ぶ．それぞれのネットワーク上に自由にネットワーク設定・構成をとることが可能である．ネットワークスライス技術を利用すれば，それぞれ独立したネットワークを構築することができるため，安全性の高いネットワークが実現できる．さらに，現在インターネットで使われている TCP/IP 以外のプロトコルを使用することもできるので，権利のない第三者が不法にアクセスすることはほぼ不可能になる．本研究においてスライスを用いる利点として，スライス単位でのネットワーク設定が可能であるということがあげられる．スライス単位でのネットワーク設定を行うことにより，各機器ごとに個別の設定を行う必要が無くなる．本論文内ではスライスの実装は二つのみで行っているが，実際のホームネットワークに即してスライスは容易に追加可能であり，より細かく機能ごとにスライスを追加することで，機器や機能が増加してもホームネットワークの管理は簡単に行うことができる．新たな機器の増加，サービスの高品質化などに伴ってホームネットワークに要求される機能というのは増加する．そういった部分を考慮すると，スライスを用いた管理手法はホームネットワーク内の機能の増加，統合，分離などに対応しうるものである．

スライスを実現する代表的な既存技術に VLAN がある．VLAN はネットワークをスイッチのポート単位や MAC アドレス単位で分割することができる．また，VLAN タグと呼ばれる ID をパケットに付加することでスイッチをまたがったスライスを作ることも可能である．

しかし，VLAN にはスライス数の上限がプロトコル上の制約により有限であるといった点や，ネットワーク知識の浅い一般の人が設定するには難しいという問題点がある．そこで本研究では，ネットワークスライスの実現には OpenFlow を用いる．

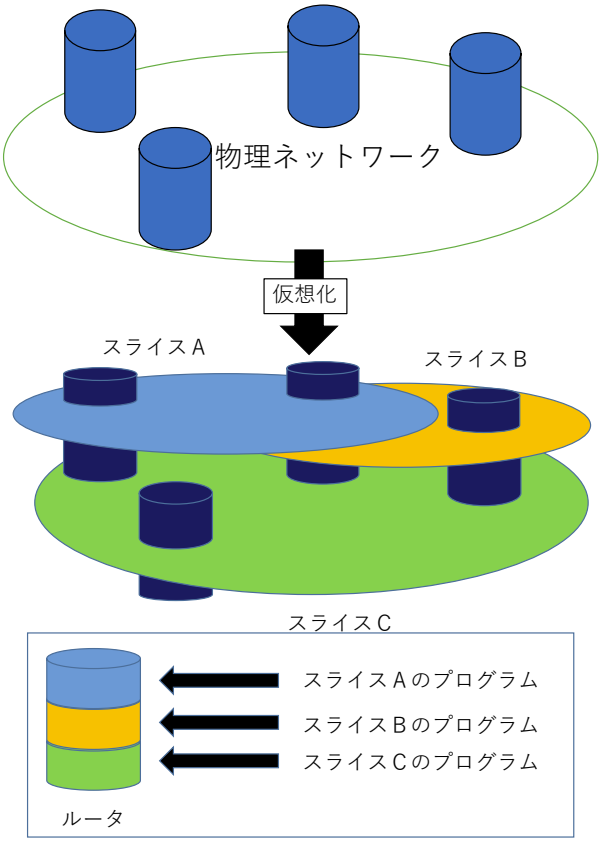

図 2　ネットワークスライス

3.4 OpenFlow

本研究ではネットワークスライスを実現するために，ルータやスイッチなどのネットワーク機器をソフトウェアによって集中的に制御することで，ネットワークの構造や構成，背低などを柔軟に変更することを可能とする技術である SDN の代表的なプロトコルである OpenFlow を用いる．OpenFlow とはネットワークスイッチの動作制御を可能とする標準プロトコルの一つであり，OpenFlow を使うことで各スイッチの動作をソフトウェアから自由に制御することが可能となる．OpenFlow の概要を図 3 に示す．OpenFlow は以下のような特徴を持つ [5]．OpenFlow の概要を図 3 に示す．

伝送部と制御部の分離

ルーター等の従来のネットワーク機器は，パケットの操作内容を制御をするネットワーク経路制御機能と実際にパケットを転送する物理的なデータ伝送機能が同一機器に組み込まれている．OpenFlow では，図 3.3 のように OpenFlow コントローラと呼ばれる制御部と

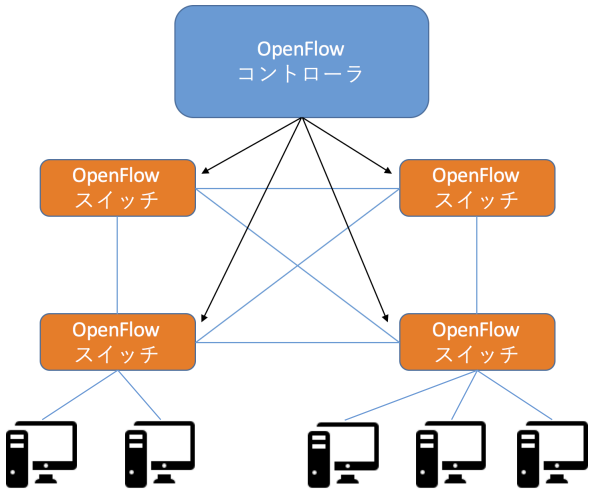

図 3　OpenFlow

OpenFlow スイッチと呼ばれる転送部が別機器として分離している構造となっており，経路の集中管理により物理配置に縛られることなく切り替えが可能である．

プログラマブルな伝送制御

OpenFlow コントローラはオープンなソフトウェアによって構成されプログラミングが可能である．パケット伝送用メソッドなど様々な API が用意され開発者は容易に OpenFlow コントローラに意図するネットワーク制御機能を実装することができる．開発者またはネットワーク管理者は OpenFlow コントローラを通してソフトウェアによってネットワークを柔軟に制御できる．

フローテーブルの利用

OpenFlow では各 OpenFlow スイッチに備えられたフローテーブルを用いてフロー単位の伝送制御を行う．フローテーブルの構造を図 4 に示す．フローテーブルにはフローエントリが設定される．フローエントリはマッチフィールド，アクション，カウンタで構成される．マッチフィールドには宛先 IP アドレスや送信元 MAC アドレスなどの条件ルールが，アクションにはパケットに対する制御内容が記載されている．パケットを受け取ったスイッチは，フローテーブルからパケットと合致するマッチングルールを持つフローエントリを検索し該当するフローエントリのアクションをそのパケットに対して実行する．OpenFlow コントローラはフローエントリを作成し各スイッチのフローテーブルに書き込むことで特定の条件を持った一連のパケット群単位の伝送制御，つまりフロー単位での伝送制御を行うことができる．またフローエントリのカウンタにはそのフローの送信パケット数など各種統計情報が記載されていてこの情報を収集，統合することで OpenFlow コントローラは様々なネットワーク情報を把握することができる．

OpenFlow のパケット伝送

ここで OpenFlow ネットワークにおけるパケット伝送の基本動作手順を以下に示す．

(1) 送信元ホストからパケットが隣接スイッチに届く
(2) ホストからパケットを受け取ったスイッチは，フローテーブルからパケットに合うマッチングルールを持つフローエントリを探索する．該当するフローエントリがあれば，そのフローエントリのアクションに従いパケットを伝送する．該当するフローエントリがなければ，コントローラへ Packet-In メッセージを送信する．
(3) Packet-In メッセージにはスイッチが受け取ったパケットの情報（表 1）が記載されており OpenFlow コントローラは，その情報を基にパケットに対するマッチングルール，アクションを作成し，各 OpenFlow スイッチに Flow-Mod メッセージによってフローエントリを追加する．以後同様のパケットは追加したフローエントリによって処理されるが Packet-in メッセージを送った OpenFlow スイッチはフローエントリ検索を終えてしまっているため，最初のパケットはコントローラが Packet-Out メッセージによってアクションを明示的に実行させる．
(4) 以降パケットは各スイッチのフローエントリに従い伝送されていく．

表 1　Packet-In メッセージ内容

項目	内容
datapath_id	メッセージ送信元スイッチの ID
transaction_id	トランザクションの ID
in_port	パケットを受信したポート番号
total_len	受信したパケットのサイズ
data	受信データの文字列
ipv4_protocol	IP プロトコル番号
ipv4_saddr	送信元 IP アドレス
ipv4_daddr	送信先 IP アドレス
tcp_src_port	送信元の TCP ポート番号
tcp_dst_port	送信先の TCP ポート番号
udp_src_port	送信元の UDP ポート番号
udp_dst_port	送信先の UDP ポート番号
macsa	送信元の MAC アドレス
macda	送信先の MAC アドレス

3.5　スライスによるホームネットワークの管理

スライスの作成ホームネットワーク内に存在する機能・サービスごとに適したネットワーク設定のスライスが作成される．例えば音声データを主に扱う Audio スライスであれば，スループットは考慮せずパケット到達率を優先するといったような設定を行う．

機器の割り振り通信に用いられる IP パケットの構造は，「実データ」と「IP ヘッダ」の二つに大きく分類される．IP ヘッダは各 IP パケットの先頭に付加されるもので，パケットの宛先アドレスなどを保持し，それにより経路選択などの IP の機能が実現される．本研究では，接続された機器がどのスライスに属するかという割り振りは，接続された機器から送られてくるパ

図 4　フローテーブル構造

ケットの ToS（Type of Service）フィールドを参照する．機器の割り振りが行われるのは，接続されて最初の通信が行われた時のみで，それ以降の通信時は ToS フィールドの参照はされない．

通信機器のスライスへの割り振りが完了すると，スライスごとに帯域制御等が行われる．コントローラがスイッチに対し，例えば Audio スライスであれば安定性を保証するといったようなルールが適用され．通信制御が行われる．

4. 設計と実装

4.1 構成と条件

本提案手法は OpenFlow コントローラ，OpenFlow スイッチ，ホームネットワーク機器によって構成される．本稿ではホームネットワーク内に存在する機能を大きく二つに分け，音声を主に扱う Audio スライスと，映像を主に扱う Video スライスの 2 つを実装する．

- **OpenFlow コントローラ（以下，コントローラ）**
  スライスごとにフローエントリを設定し伝送制御を行う．
- **OpenFlow スイッチ（以下，スイッチ）**
  フローテーブルに従ってパケットの伝送を行う．
- **ホームネットワーク機器**
  ホームネットワークを構成している機器．スイッチに接続されており，Audio スライスか Video スライスのいずれかに分類される．
- **Audio スライス**
  Audio の機能をもつホームネットワーク機器が属するネットワーク．
- **Video スライス**
  Video の機能をもつホームネットワーク機器が属するネットワーク．

4.2 動作の流れ

以下に動作手順を示す．

(1) ホームネットワーク機器をスイッチに接続する．
(2) 接続された機器はスイッチに対して空のパケットを送信する．
(3) 機器から送信されたパケットの ToS フィールドを参照し，その値によって Audio スライスか Video スライスのどちらに属するのかを判断する．
(4) 判断の結果から，接続された機器の MAC アドレスや接続されたスイッチのポート番号等の情報を適したスライスに追加する．
(5) スライスごとにネットワーク設定に応じて通信を行う．

動作手順を図 5 に示す．

4.3 実装環境

実装環境を図 6 に示す．本研究ではコントローラのフレームワークである Trema を使用して実装を行う．開発言語は Ruby を使用した．Trema には，開発用ツールが付属しておりその中に Linux の仮想ネットワーク機能 (Linux network namespace) を利用し，1 台の PC 上で仮想のスイッチやホストを配置した仮想ネットワークを作り，動作をテストする仮想 OpenFlow ネットワーク機能がある．Trema で提案手法の機能をコントローラに実装し，trema の仮想 OpenFlow ネットワーク内で動作させる．なお仮想のスイッチにはソフトウェア OpenFlow スイッチである Open vSwitch を使用する．

4.4 ネットワークスライス実装

ネットワークスライスはスイッチ番号とポート番号，それに MAC アドレスのセットで管理される（図 7）．例えば図中の 0x2:4 の欄は，スイッチ 2 の 4 番ポートに MAC アドレス 11：11：11：11：11：11 の機器が接続されており，それが AudioSlice に属しているという見方になる．あるスイッチのあるポートがどのスライスに属するかはあらかじ

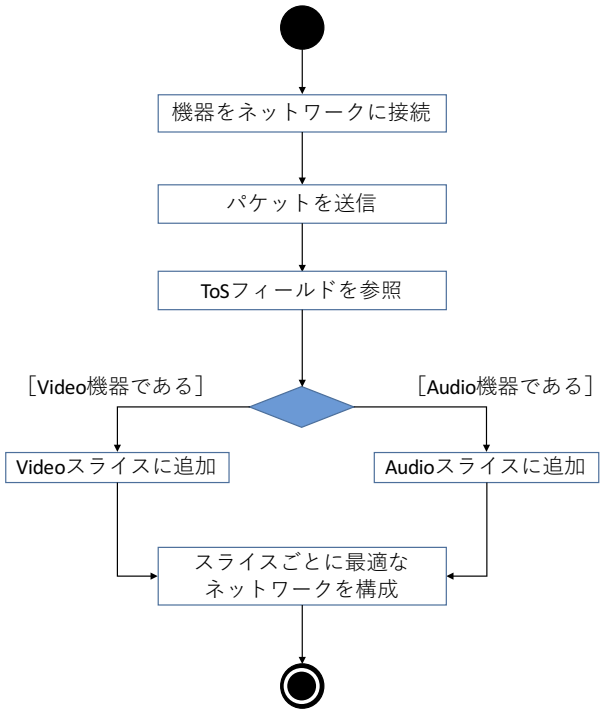

図 5 動作手順

OS	Ubuntu14.04LTS
CPU	Intel Core i7-6500U@ 2.5GHz × 2
メモリ	8.00GB
開発言語	Ruby
OpenFlowコントローラ	Trema Version 0.9.0
OpenFlowスイッチ	Open vSwitch Version 2.0.2

図 6 実装環境

Audio Slice		Video Slice	
portNum	macAddress	portNum	macAddress
0x2:4	11:11:11:11:11:11	0x4:3	33:33:33:33:33:33
0x5:3	22:22:22:22:22:22	0x3:4	44:44:44:44:44:44

図 7　スライステーブル

め設定しておくのではなく，スイッチに機器が接続されてからコントローラによって判別されるものなので，今回の実装では図のようなスライスが構成されるが，ネットワーク構成を変更すればスイッチ・ポート番号の部分は適宜変更される．

5.　実験と評価

本章では提案手法の優位性を評価する．

5.1　評価環境

評価環境を図 8 に示す OpenFlow スイッチを五台と，ホームネットワーク機器の役割を持つ仮想ホストを四台図のように配置する．機器はそれぞれ Audio スライスに属する Audio1，Audio2 と Video スライスに属する Video1，Video 2 を用意し，それぞれが別々のスイッチに接続されている．そこに前章で述べたネットワークスライスを実装している．

5.2　評価方法

図 6.1 で示した評価環境下において Audio1 から Audio2 へ，Video1 から Video2 へ 1000 パケットを送信する．今回の評価環境では，ホームネットワーク内でビデオ視聴と音声通話が同時に行われた場合を想定する．音声通話の安定性を重視するために，Video スライスと Audio スライスの 2 つにおいて，Audio スライスに帯域制御を行った．提案手法を用い帯域制御を行っていないものと，帯域制御を行ったもの 2 つの結果を示す．評価項目としてパケット到達率を用いる．

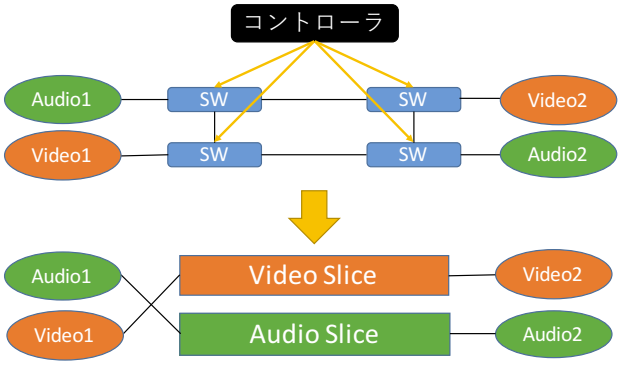

図 8　評価環境

5.3　評価結果

評価結果を図 11 に示す．提案手法を用いた通信では，Audio スライスのパケット到達率が 100 %であることが示された．Video スライスにおいてパケット到達率は低下しているが，映像データではパケット落ちの影響は音声データに比べて小さく問題にはならない．

6.　考察

評価結果をもとに提案手法の有効性を確認する．

6.1　ホームネットワークの管理

評価結果より正しく伝送制御が行われていることが示せた．提案手法では，ネットワーク内のスイッチであればどのスイッチに接続しても自動で適したスライスに追加されるので，配線をどのようにするかといった手間を省くことが可能となった．自動でスライスに追加されることで，各機器に対してネットワークの設定や連携する機器間の接続関係の設定を個別に行う必要がなくなり，ホームネットワーク内の機器の管理は容易になった．また，本研究の提案手法を用いることで，ホームネットワーク内の機器の増加により管理が複雑になり，ネットワーク知識の浅い人物が管理する際にネットワーク設定や連携する機器間の接続が適切に行われていないといった問題を解決している．

6.2　ネットワーク最適化

管理が容易となることで機能に合った適切な制御が可能になり，ネットワークに接続できる機器の増加や，機器の高性能化に伴いホームネットワーク内のトラフィックの量や種類が増加することによる通信品質の低下という問題を解決した．

6.3　今後の課題

ネットワーク仮想化を用いてホームネットワークの管理を行うという本研究の有効性を示すことができたが，本研究内で実装したネットワークスライスは，スライス内での通信を行うことはできても異なるスライス間での通信を行

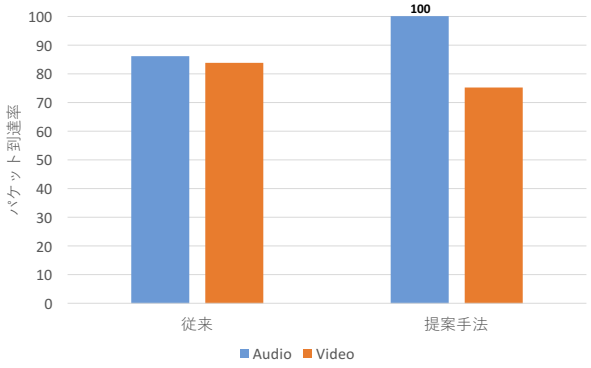

図 9　パケット到達率

うことができない．そのため，セキュリティ面を考慮すれば現在の仕様の効果は大きいが，自身のホームネットワークに対して外部から何らかの操作，メッセージの送信を行うことができない．したがって今後は，外部との通信を中継するアプリケーションをゲートウェイに用意するなどの方法を検討する．

本研究で得られた評価は，接続された機器が自動でスライスに追加され，帯域制御を行うことが可能であることを示したものである．しかし，本提案手法を使用することによるスループットの低下が発生する可能性がある．その値が各スライスの機能を満たすものであるかという評価を取得し，それに応じた制御方法を検討する．

ネットワークスライスの追加は容易であると述べたが，スライスの追加を誰が行うのかという課題がある．一般の人が行うのであれば容易にそれができる枠組みを作る必要がある．システム自体にアップデート機能を付加し，定期的にスライスリストのようなものを取得する方法などが考えられる．

7. おわりに

本研究で提案したホームネットワーク管理手法では，情報家電の増加やサービスの高品質化に伴ってホームネットワーク内に発生する諸問題を，ネットワーク仮想化技術を用いてホームネットワークを管理することで解決する手法を提案した．スライス化により機能ごとのネットワークを構築しネットワークを最適化した．さらに，新たに機器をネットワークに追加する際にも物理的配置にとらわれず容易に追加可能となった．

本研究のホームネットワーク管理手法を利用することで，従来のホームネットワーク環境のまま機器の増加や機能の高性能化が進んだ場合と比較して，配線やネットワーク設定といった機器の管理は容易になり，それに伴ってネットワークが最適に利用されていないといった問題も解決することができた．

拡張性も高く，将来のホームネットワークがより複雑なものになっても柔軟に対応することが可能であることが示された．

謝辞

本研究の一部は JSPS 科研費（JP16H02814）の助成を受けたものである．

参考文献

[1] 迫田紘志，OpenFlow 技術のホームネットワークへの適用に関する研究，JAIST 学術研究結果リポジトリ，2015-09.

[2] Niels Soetens，Jeroen Famaey，Matthias Verstappen，Steven Latre ´，SDN-based management of heterogeneous home networks，Network and Service Management (CNSM), 2015 11th International Conference on，9-13 Nov. 2015

[3] Yiakoumis, Y., Yap, K. K., Katti, S., Parulkar, G.,& McKeown,N. (2011, August). Slicing home networks. In Proceedings of the 2nd ACM SIGCOMM workshop on Homenetworks (pp. 1-6). ACM.

[4] 中尾彰宏，新世代ネットワーク構想におけるネットワーク仮想化，電子情報通信学会誌，Vol.94.No.5.2011.

[5] Nunes,INRIA,Mendonca，M.;Xuan-Nam Nguyen;Obraczka，K.;Turletti,T.，A Survey of Software-Defined Networking: Past,Present,and Future of Programmable Networks,Communications Surveys and Tutorials， IEEE，Volume:16，Issue:3，pp.1617-1634，2014.