GSCIP の Windows への実装に関する検討

040427177　細尾幸宏
渡邊研究室

1. はじめに

企業ネットワークのセキュリティを確保するために通信グループを定義することは有効な方法である．しかし，IPsec のような既存の技術では通信グループの構成が頻繁に変化する場合や，個人単位と部門単位のグループ定義が混在した場合，それらに柔軟に対応するためには管理負荷が高くなり，導入が難しくなる．

我々は FPN（Flexible Private Network）と呼ぶ柔軟性とセキュリティを兼ね備えたネットワークの概念を提唱し，FPN を実現するためのネットワークアーキテクチャとして GSCIP（Grouping for Secure Communication for IP）を提案している[1]．今後 GSCIP をより多くの人に利用してもらい，評価を受けるためには Windows に実装することが必須である．

本稿では GSCIP を Windows に実装する方法について検討し，GSCIP の基幹プロトコルである DPRP（Dynamic Process Resolution Protocol）の実装と評価実験を行ったので報告する．

2. GSCIP

GSCIP では，定義した通信グループに所属する端末は共通の暗号鍵としてグループ鍵を持ち，グループ鍵と通信グループを 1 対 1 に対応付ける．これにより，IP アドレスに依存しない通信グループを定義することができ，IPsec に比べて大幅に管理負荷を軽減することができる．

現在，GSCIP は FreeBSD の IP 層に実装されており，基本動作を確認済みである．GSCIP モジュールは IP 層の一部を改造し，適切な場所から呼び出すサブルーチンとして実現されている．

3. Windows への実装方法

Windows は OS がブラックボックスになっており，FreeBSD のように直接 IP 層を改造して実装することができない．その代わりに，Windows は機能を拡張できるインタフェースが公開されている．GSCIP はこの中でネットワーク機能を拡張できる NDIS（Network Driver Interface Specification）に着目し，これを用いて FreeBSD の場合と同等の機能を実現することができる．NDIS はネットワークドライバの仕様やそれらドライバとのインタフェースを規定した仕様である．NDIS はデータリンク層の機能の一部であり，中間ドライバとミニポートドライバがある．NDIS ドライバはネットワークドライバとして必要な機能を実現するモジュール群として作成して登録しておき，登録されたモジュールは NDIS のインタフェースを介して決まった動作時に呼び出される．

GSCIP は NDIS の中間ドライバとして実装する．

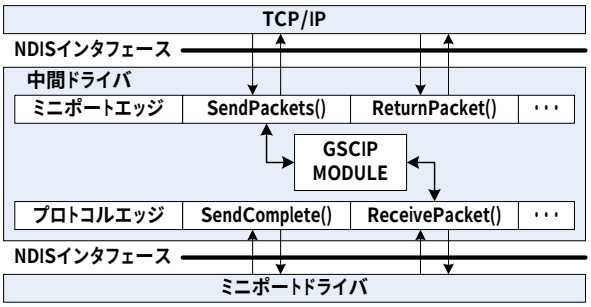


図 1．NDIS への実装

FreeBSD で開発した GSCIP のモジュールはほぼそのまま Windows へ流用可能であるが，Windows と FreeBSD で提供されている API の違いへの対応，データリンク層で動作するために MAC ヘッダに対する処理の追加，処理対象パケットのフィルタリングを行うなどの処理が必要になる．

また，NDIS ドライバにはパケット送受信時に FreeBSD の IP 層にはない特有の動作がある．プロトコルスタックの上位モジュールはパケットの送信を行う際に送信処理の結果をすぐには受け取らず，後で実行される結果通知処理によって結果を取得する．受信時は上位モジュールが受信パケットへの処理終了を通知すると，ミニポートドライバがリソースを開放する．

GSCIP では通信開始時に DPRP によって通信相手とのネゴシエーションを行い，グループの認証や通信の可否を判断する．このとき，トリガとなった通信パケットを待避し，ネゴシエーションパケットを送信するが，このパケットは GSCIP が動作するスタックより上位モジュールにその送信結果を知らせる必要はない．また，上記ネゴシエーションパケットの送信完了通知を上位モジュールが受け取ると管理していないパケットの通知を取得したことに起因してクラッシュを起こす可能性がある．そこで，ネゴシエーションパケットについては送信完了通知処理時と受信処理時にパケットの判別を行い，下位モジュールで全ての処理を完結させる必要がある．

4. まとめ

FreeBSD に実装された GSCIP を Windows の NDIS を用いて実装する方法についての検討を行った．今後は GSCIP の全ての機能の実装を完了さる．

参考文献

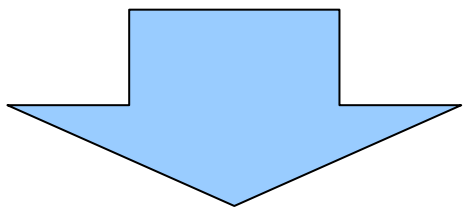
[1] 鈴木秀和，渡邊晃：フレキシブルプライベートネットワークにおける動的処理解決プロトコル DPRP の実装と評価，情報処理学会論文誌，Vol.47, No.11，pp.2976-2991，Nov.2006.

GSCIPのWindowsへの実装に関する検討

渡邊研究室
細尾 幸宏

研究背景

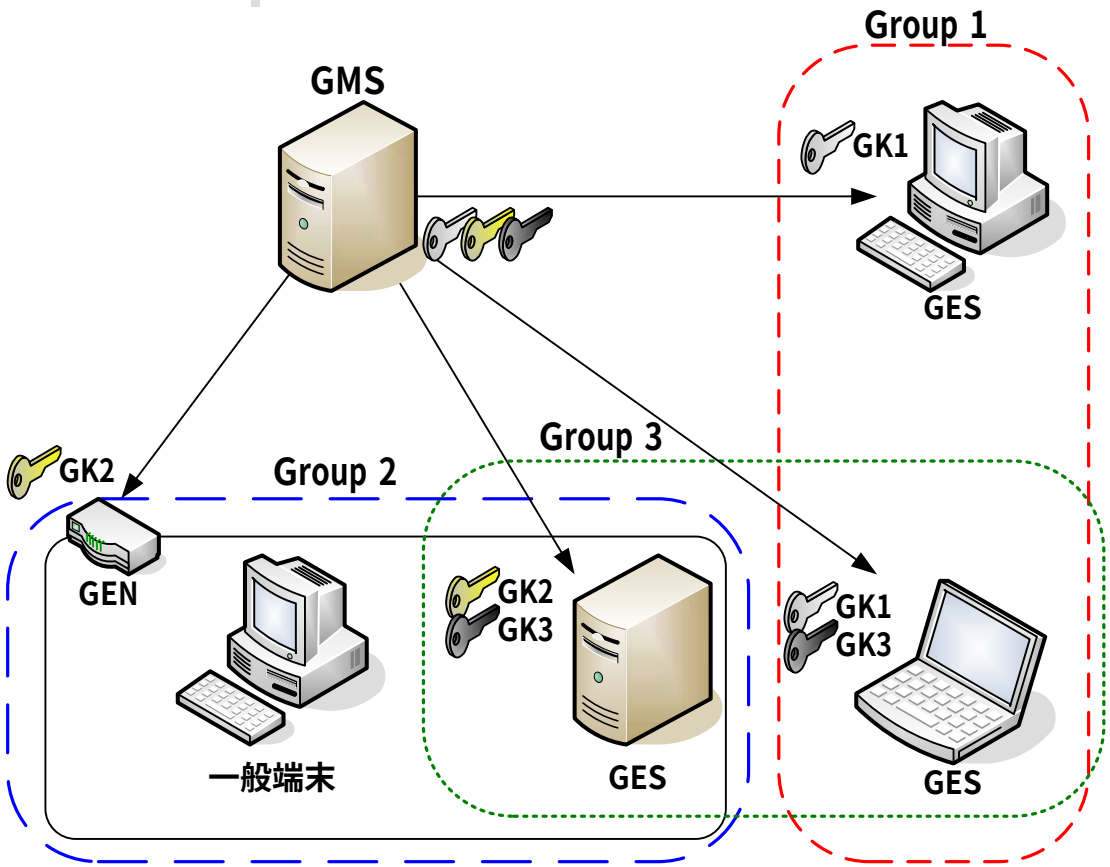
- ユビキタスネットワークの普及
  - 安全な通信
  - 移動しながらの通信
  - どこからでも自由なアクセス



柔軟性とセキュリティを兼ね備えたグループ通信を実現する

GSCIP (Grouping for Secure Communication for IP)

GSCIPの概要

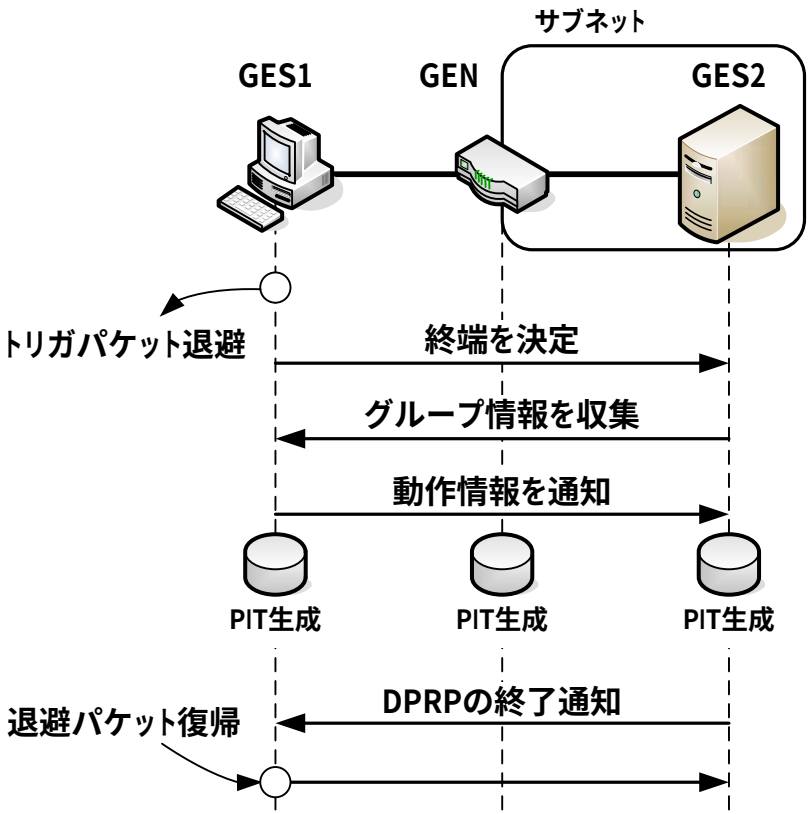


GE　：GSCIP対応装置
GES：ソフトウェア型
GEN：ルータ型
GMS：管理装置

- GMSがグループ鍵GKを各GEへ配送
- GKによって通信グループを構築
- 同一グループ間の通信はGKにより暗号化
- GKと通信グループを1：1に対応付け
- IPアドレスに依存しないグループ定義

DPRP (Dynamic Process Resolution Protocol)

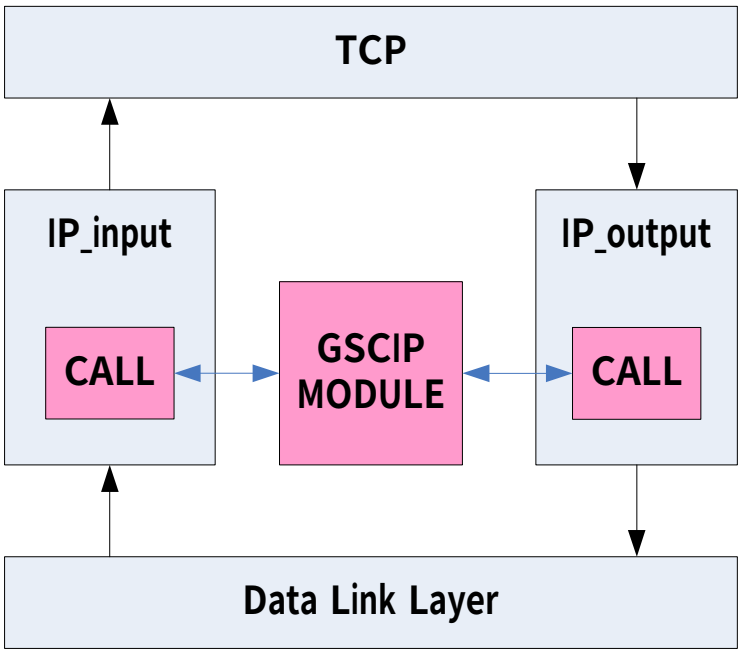
- **通信開始の際に各GEの情報を知るためにDPRPを行う**



PIT (Process Information Table)
通信パケットに対する処理を定義する動作処理情報 (暗号化/復号, 透過中継, 破棄) 等を格納

- **終端GEを決定**
- **経路上の各GEのグループ情報を収集し, 動作処理情報を決定**
- **グループ情報によって通信相手が同一グループであるか確認, 認証**
- **動作処理情報テーブルPIT を生成**
- **以降の通信はPITに定義された動作処理情報に従って動作**

GSCIPの現状

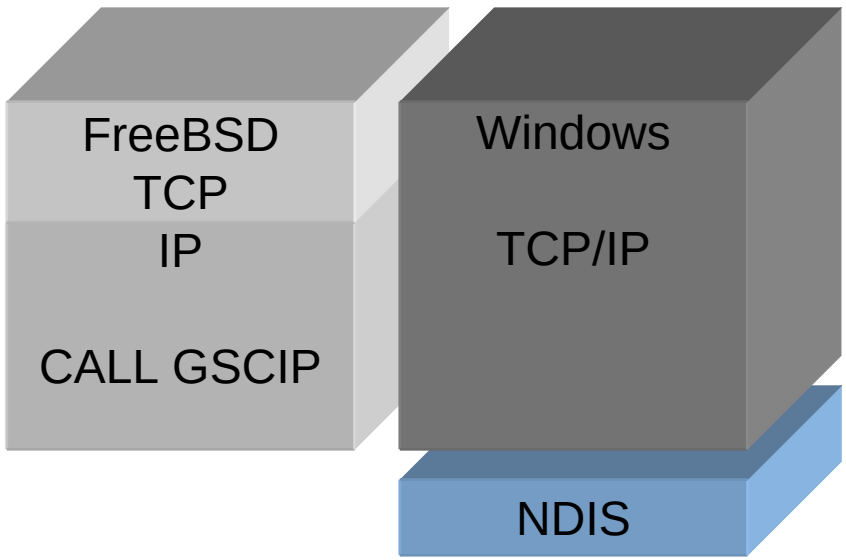


- FreeBSDではIP層にモジュール呼び出しを追加
- 動作と有効性を確認済み

GSCIPの評価や普及にはWindowsへの実装が不可欠

Windows

- WindowsはTCP/IPなどのOSがブラックボックス
  - FreeBSDのようにIP層を直接改造できない



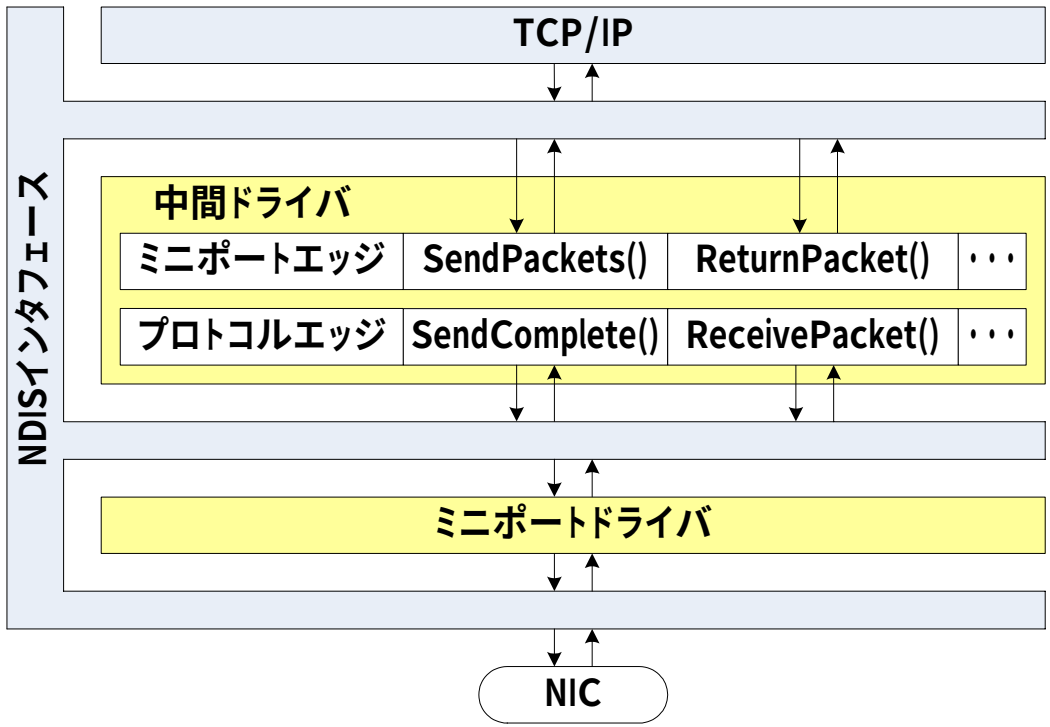
外部に仕様が公開されたインタフェース

ネットワークの機能拡張ができるインタフェース

NDIS (Network Driver Interface Specification)

NDISの概要

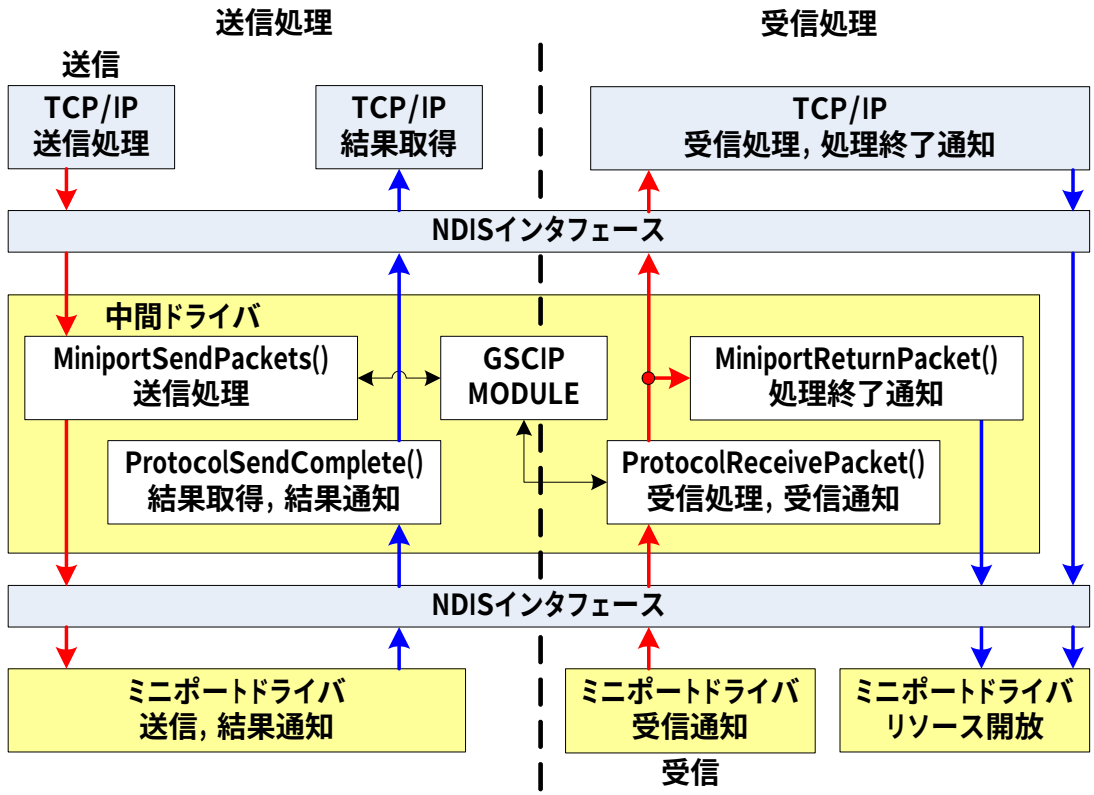
- NDISはネットワークに機能を追加できるインタフェースとそこで動作するドライバの動作手順を定める
  - データリンク層の機能の一部
  - NDISドライバは仕様として公開された機能を実行するモジュール群として作成し，NDISインタフェースに登録
  - NDISインタフェースは各モジュールを必要に応じて呼び出す



NDISへの実装と動作

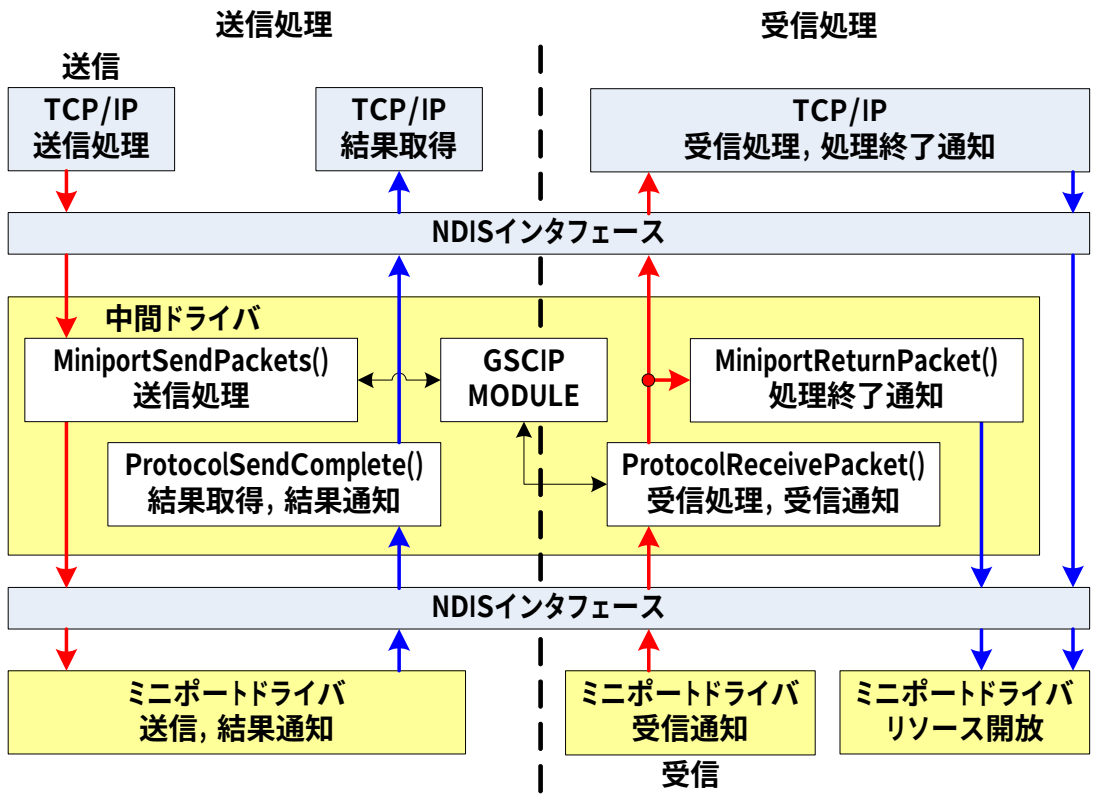
GSCIPは中間ドライバとして実装

- **送信動作**
- MiniportSendPackets()
  - **送信パケットを中継**
  - GSCIPを呼び出し，送信時の処理を行う
- ProtocolSendComplete()
  - **パケット送信処理の結果を通知**



NDISへの実装と動作

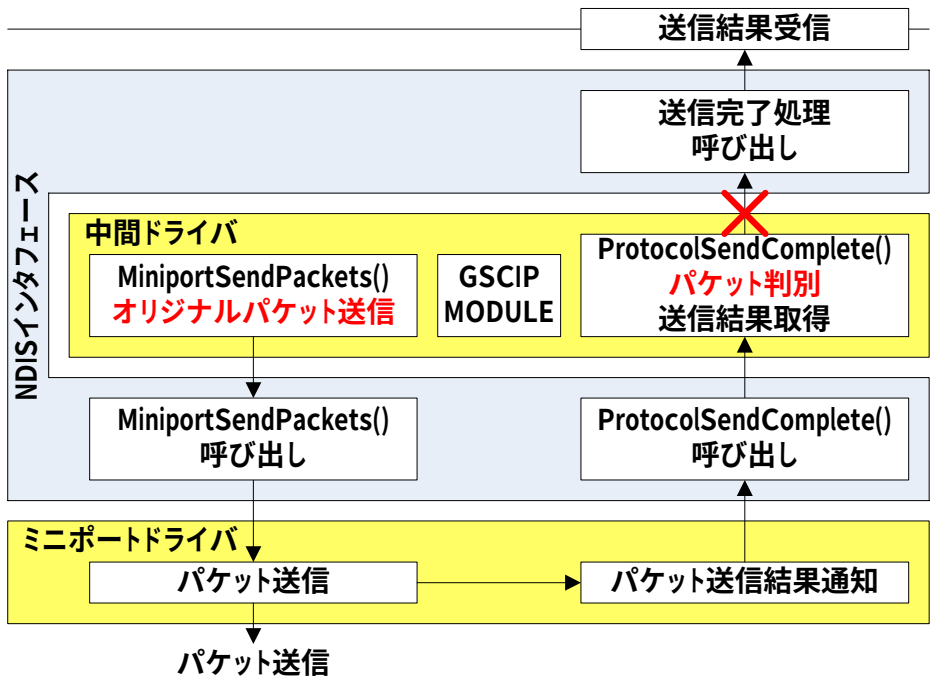
- **受信動作**
- ProtocolReceivePacket()
  - **パケットの受信を通知**
  - GSCIPを呼び出し，**受信時の処理を行う**
- MiniportReturnPacket()
  - **パケットへの処理終了を通知**



送信処理完了通知
SendComplete()

- GSCIPのプロトコルには独自のパケットを作成し，通信を行う
  - TCP/IPが関与しないパケットに関するSend Completeが行われるとクラッシュを引き起こす原因になる

- ProtocolSendComplete()にGSCIP独自パケットの判断処理を追加
- TCP/IPが関与しないパケットを通知しない



評価

100BASE-TXのEthernet

2台のPCを直接接続

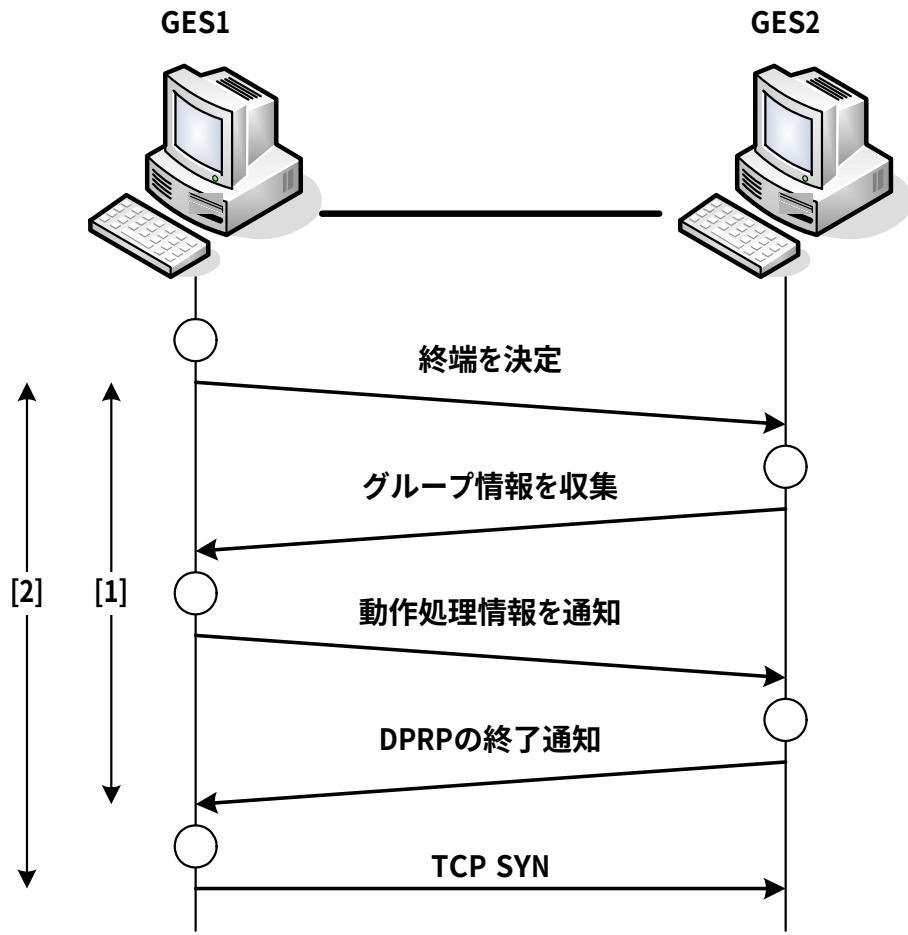
OS　WindowsXP

CPU　Pentium4　2.4GHz

メモリ 1280MB

評価

- オーバヘッド時間
  - [1] 通信に先立って行われるネゴシエーション時間
  - [2]トリガパケット送信までのオーバヘッド
- スループット
  - FTP接続で500MBのファイルをダウンロード



結果

- 通信に先立って発生するオーバヘッドは十分に小さい
- スループット低下率は約0.06%
- 通信に対する影響はほとんどみられない

オーバヘッド時間

単位：ミリ秒

[1]ネゴシエーション時間	[2]通信開始までの時間
0.22	0.24

スループット

単位：Mbps

GSCIP	実装時	未実装時
スループット	92.33	92.39

まとめ

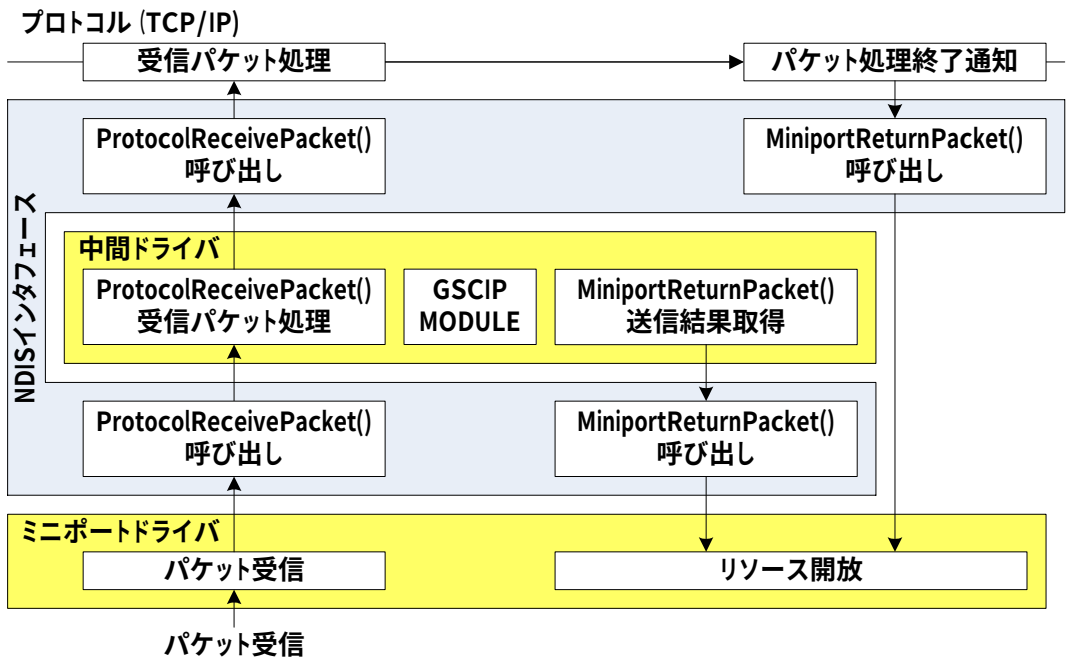
- GSCIPをWindowsのインタフェースNDISを用いて実装する方法についての検討を行った
- GSCIPの基幹プロトコルDPRPを実装し，評価を行った

- 今後はGSCIPの全機能を実装し，性能評価を行う

GSCIP

- GSCIP**構成プロトコル**
  - DPRP (Dynamic Process Resolution Protocol)
    - **ネットワークの構成変化に動的に対応**
    - **通信相手と経路上にある**GE**に対してネゴシエーションや認証を行う**
  - Mobile PPC (Mobile Peer to Peer Communication)
    - IP**アドレスの変化を隠蔽し，移動通信をエンドエンドで実現**
  - NAT-f (NAT – free Protocol)
    - **対応**NAT**ルータに外部から強制的に**NAT**テーブルを生成し，アドレス空間の違いを意識しない通信をエンドエンドで実現**

受信処理


プロトコル (TCP/IP)
受信パケット処理
パケット処理終了通知
ProtocolReceivePacket()
呼び出し
MiniportReturnPacket()
呼び出し
NDISインタフェース
中間ドライバ
ProtocolReceivePacket()
受信パケット処理
GSCIP
MODULE
MiniportReturnPacket()
送信結果取得
ProtocolReceivePacket()
呼び出し
MiniportReturnPacket()
呼び出し
ミニポートドライバ
パケット受信
リソース開放
パケット受信

Windowsへの移植

- API (Application Programming Interface)の違い
  - OSが違うため，APIを同等の処理になるように置き換え
- MACヘッダの処理を追加
- パケットの形式
  - FreeBSDでは構造体でパケットを表現
  - NDISでは構造体とメモリ領域の接続で表現