

GSCIP の Windows への実装に関する検討

細尾 幸宏[†] 鈴木 秀和[‡] 渡邊 晃[†]

名城大学理工学部[†] 名城大学理工学研究科[‡]

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、個人単位や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 for Communication for IP) を提案している[1]。現在、GSCIP は FreeBSD に実装され動作検証を行っており、有効なアーキテクチャであることが確認されている。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。そこで本稿では GSCIP を Windows に実装する方法について検討した。

2. GSCIP

GSCIP では、共通暗号鍵と通信グループを 1 対 1 に対応付けることにより、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。

GSCIP を構成するプロトコル郡として DPRP (Dynamic Process Resolution Protocol)、Mobile PPC (Mobile Peer to Peer Communication) および NAT-f (NAT-free Protocol) がある。DPRP は通信に先立って通信経路上の GSCIP 対応装置が情報を交換し、認証や通信の可否を判断する処理を行う。Mobile PPC は通信中に一方の端末が移動した場合、エンド端末同士で IP アドレスの変

化情報を交換し、端末の上位ソフトウェアに対して IP アドレスの変化を隠蔽し、通信の継続を実現する。NAT-f はプライベートアドレス空間にいる相手装置と通信を開始するとき、NAT-f に対応した NAT ルータに NAT テーブルを強制的に生成させることにより、プライベートアドレス空間とグローバルアドレス空間の違いを意識するとこない通信を実現する。

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

Windows は OS がブラックボックスになっており、FreeBSD に実装された GSCIP のように直接 IP 層を改造して実装することができない。しかし、Windows には機能を拡張するために複数のインタフェースが外部に公開されている。GSCIP はこの中でネットワークの機能を拡張できる NDIS (Network Driver Interface Specification) を用いて実現することができる。NDIS の概要を図 1 に示す。NDIS は Windows カーネルのネットワークスタック内での処理手順などを規定したネットワークドライバの仕様とそれらドライバとのインタフェースを規定したものである。NDIS が規定する NDIS ドライバは図 1 中の中間ドライバやミニポートドライバを指し、NDIS インタ

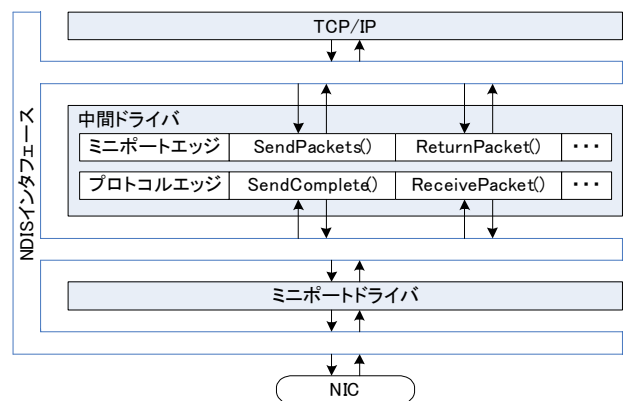


図1 NDISの概要

“A Study of Implementation of GSCIP for Windows”

[†] Yukihiro Hosoo and Akira Watanabe
Faculty of Science and Technology, Meijo University

[‡] Hidekazu Suzuki
Graduate School of Science and Technology,
Meijo University

フェースは NDIS ドライバ間の通信の中継やライブラリを提供する。NDIS はデータリンク層の機能の一部であり、NDIS ドライバはここで動作する。NDIS ドライバは通信時のパケットの送受信時に呼び出されて動作を行うだけではなく、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、登録しておくことができる。登録されたモジュールは NDIS インタフェースが決まった動作時に呼び出し、そこで動作を行う。

GSCIP は中間ドライバに実装し、IP 層の改造と同様の動作を実現する。中間ドライバは TCP/IP のようなプロトコルドライバと NIC を操作するミニポートドライバの間でデータ転送を中継するように動作する。

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成否に関する情報をすぐには受け取らず、後で実行されるミニポートドライバからの結果通知処理によって結果を取得する。受信時はミニポートドライバがメモリなどのリソースを管理し、パケットの受信を上位モジュールへ通知する。上位モジュールはそのパケットの処理終了をミニポートドライバに通知し、そこでリソースを開放する。

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、送受信時の NDIS 特有の動作に対応する必要がある。

GSCIP 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う。このとき、トリガとなった通信パケットを一時的にカーネル内に待避し、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、上位モジュールは上記ネゴシエーションパケットの受信を通知されると管理していないパケットを受信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。

これらをふまえた NDIS への実装を図 2 に示す。

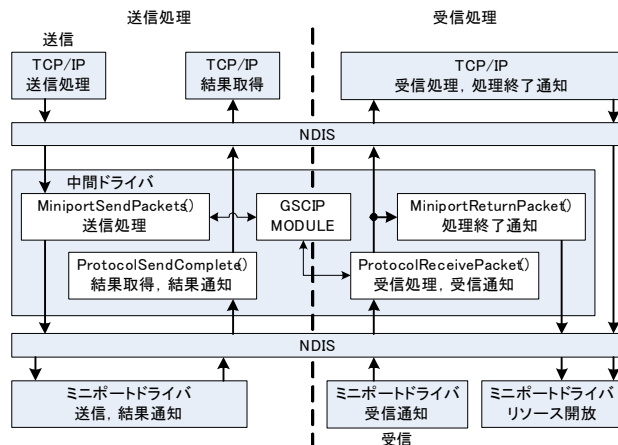


図 2 GSCIP の実装と動作

パケット送信時には NDIS から MiniportSendPackets() が呼び出される。ここで GSCIP モジュールを呼び出し、パケットの待避や暗号化、ネゴシエーションパケットの生成などの処理を行う。送信処理終了後、ミニポートドライバから通知される処理結果を ProtocolSendComplete() で取得する。ネゴシエーションパケットに関してはここで通知を破棄するが、その他の通信パケットは上位モジュールへ通知する。

パケット受信時には NDIS から ProtocolReceivePacket() が呼び出されるので、ここから GSCIP モジュールを呼び出す。GSCIP モジュールは受信パケットがネゴシエーションパケットの場合は上位モジュールへ通知をせずに MiniportReturnPacket() を経由して Miniport Driver に処理終了の通知を行い、その他の通信パケットについては上位モジュールへ通知する。これらの動作によって本来の通信に影響を与えず、DPRP ネゴシエーション処理を行うことができる。

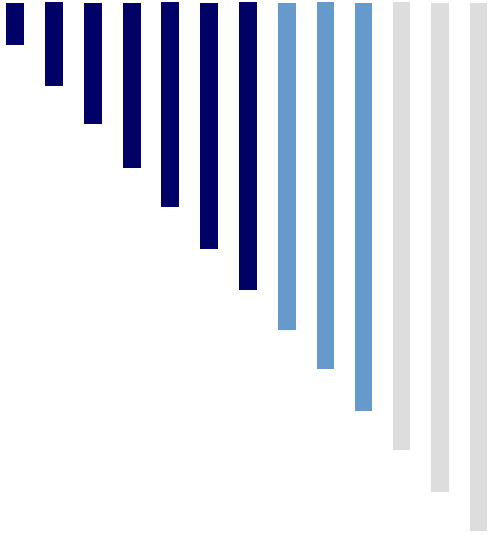
現在、GSCIP の基幹プロトコルである DPRP の実装を完了し、基本的な動作を確認済みである。

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。今後は全 GSCIP 機能の実装を完了させ、性能評価を行う。

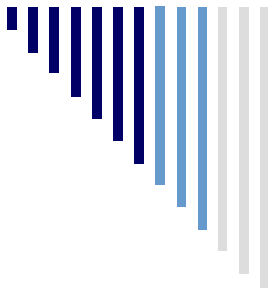
参考文献

- [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.



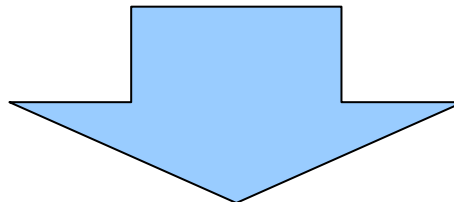
GSCIPのWindowsへ の実装に関する検討

名城大学理工学部情報工学科
細尾幸宏 鈴木秀和 渡邊晃

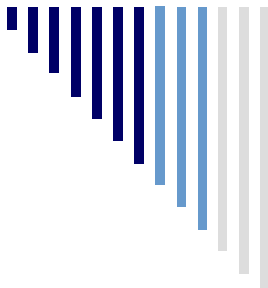


研究背景

- 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

□ 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

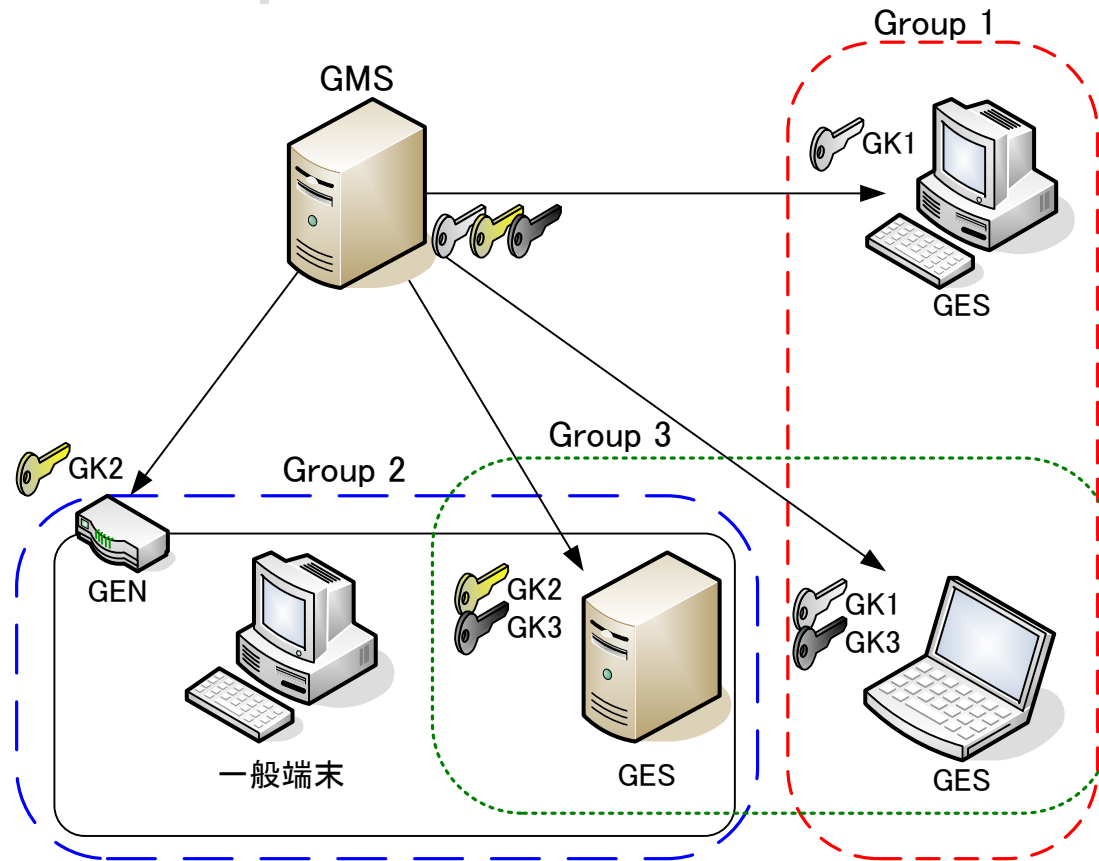
■ Mobile PPC (Mobile Peer to Peer Communication)

- 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

GSCIPの概要

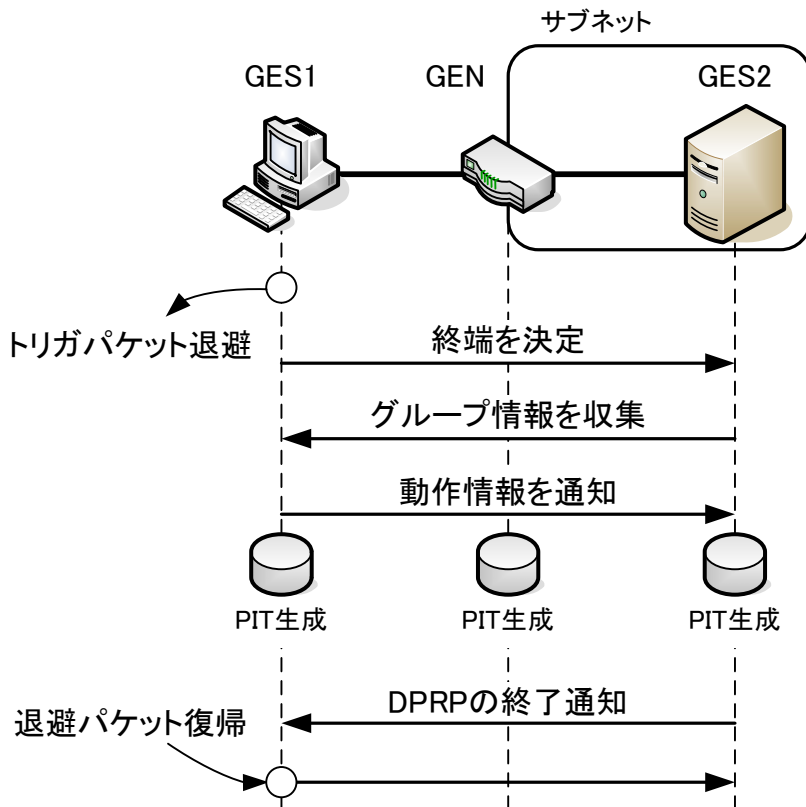


GE : GSCIP対応装置
GES:ソフトウェア型
GEN:ルータ型
GMS:管理装置

- 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

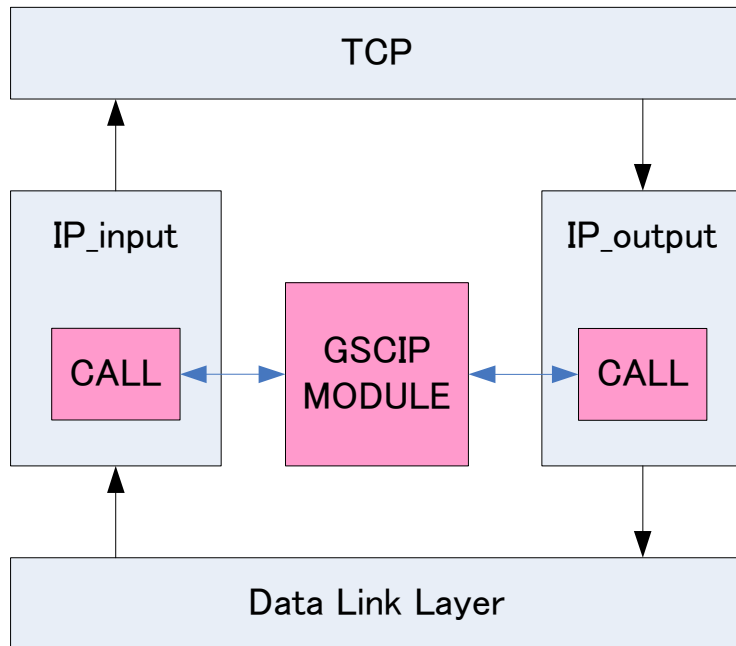


PIT (Process Information Table)

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

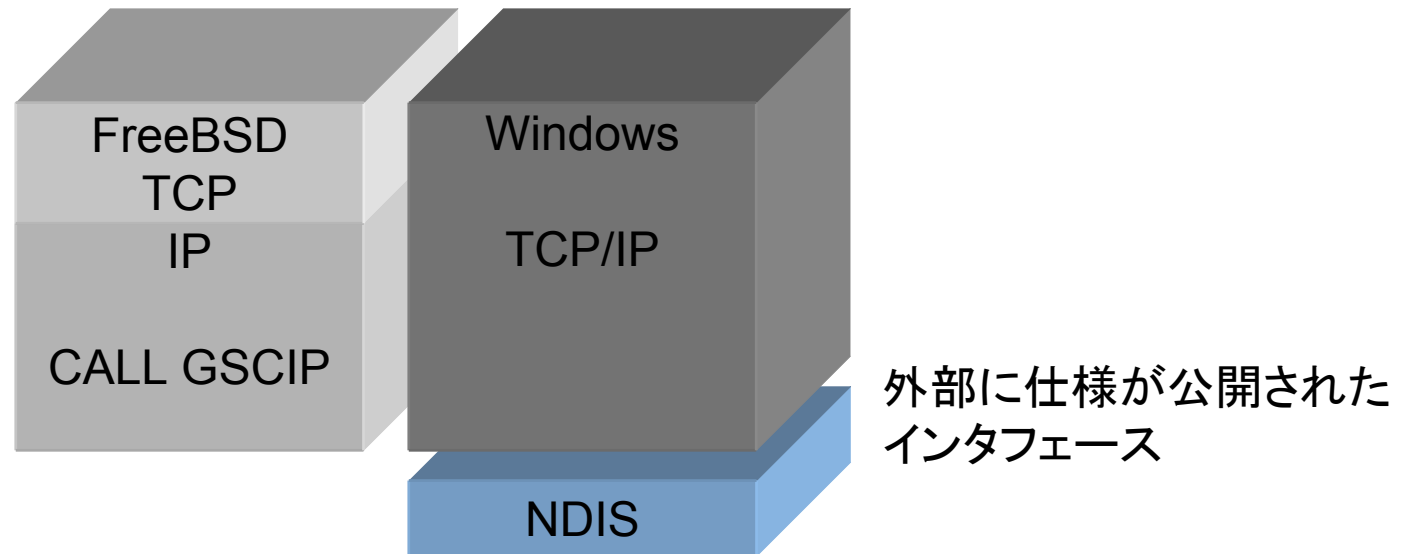


- FreeBSDではIP層にモジュール呼び出しを追加
- 動作と有効性を確認済み

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

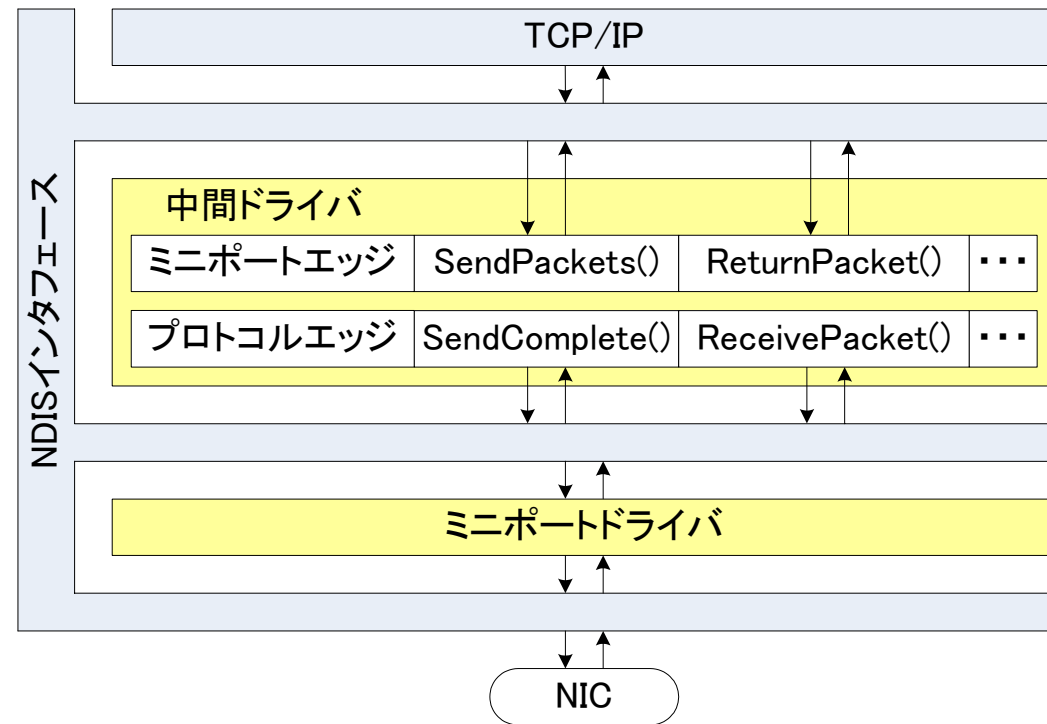


ネットワークの機能拡張ができるインタフェース

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

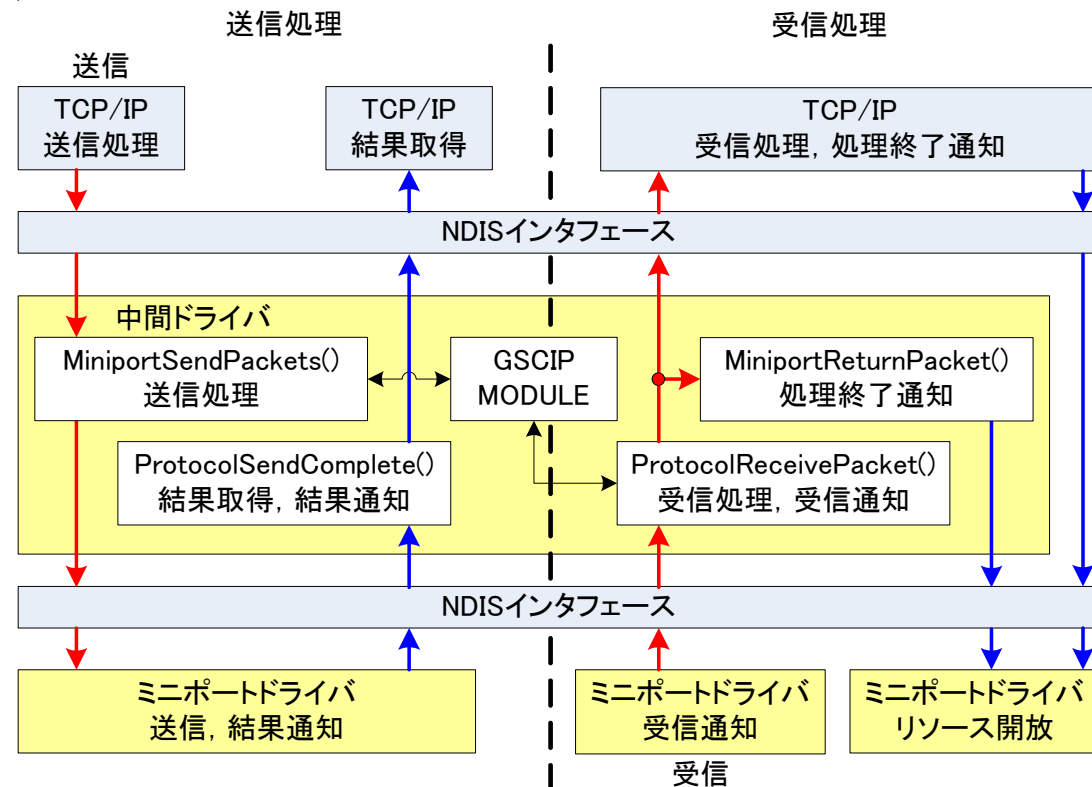
□送信動作

□MiniportSendPackets()

- 送信 packets 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

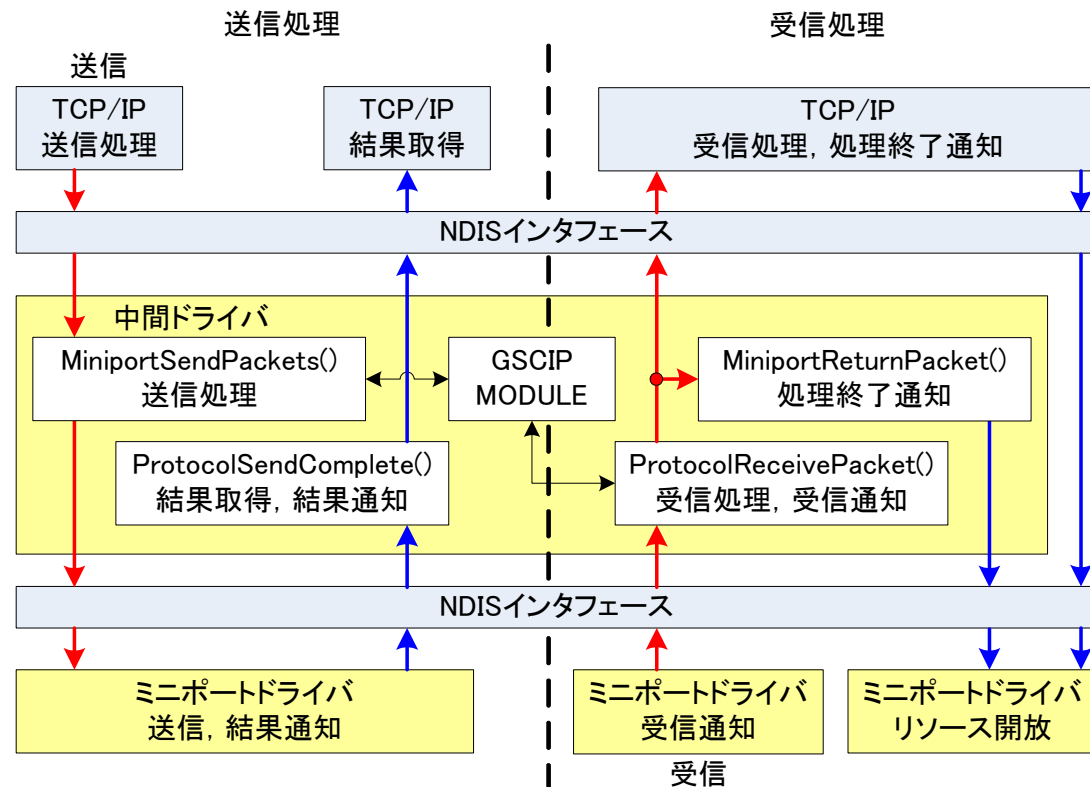
➤ 受信動作

➤ 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

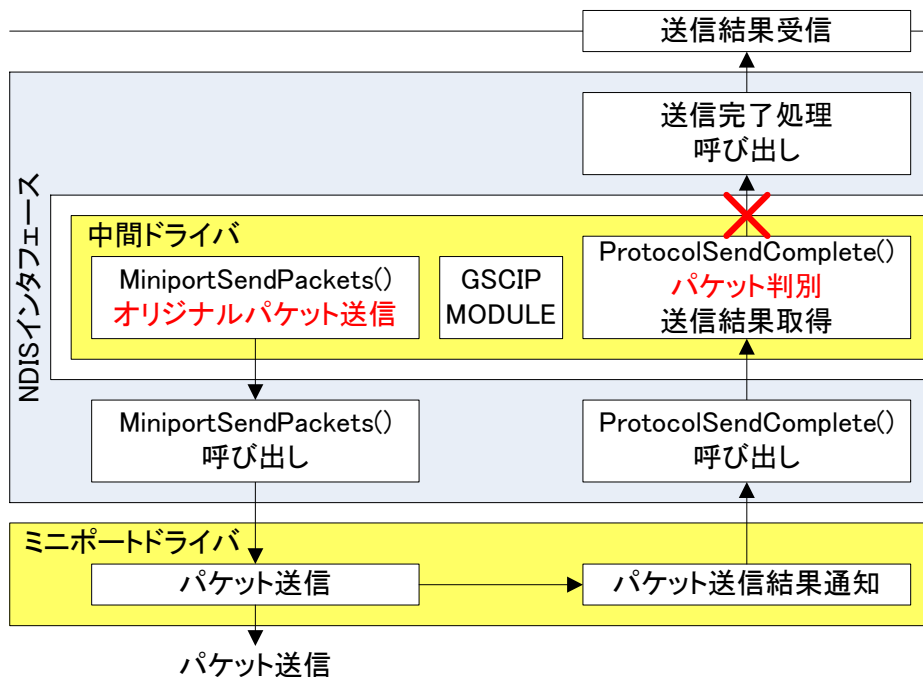
➤ MiniportReturnPacket()

- パケットへの処理終了を通知



送信処理完了通知 SendComplete()

- GSCIPの protokol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

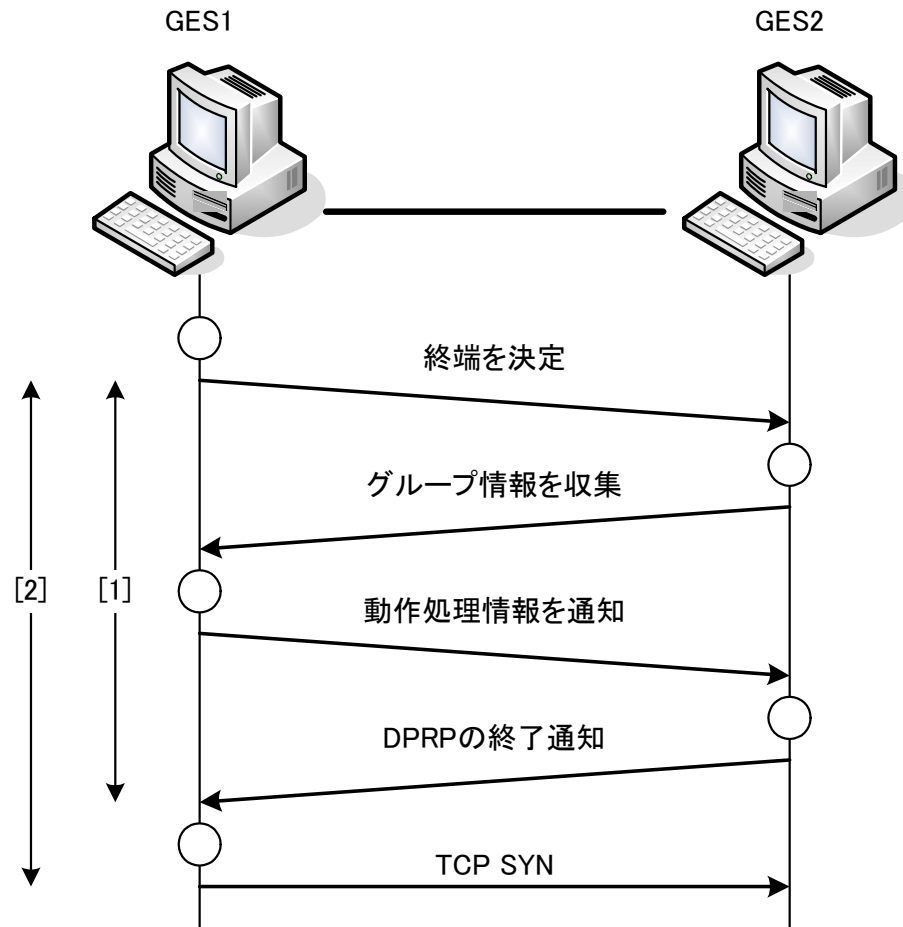
□ オーバヘッド時間

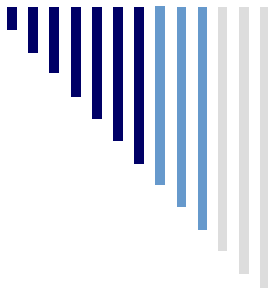
[1] 通信に先立って行われる
ネゴシエーション時間

[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ー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

オーバーヘッド時間

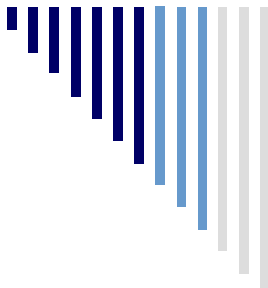
単位:ミリ秒

[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