GSCIP の Windows への実装

三宅　智朗

Implementation of GSCIP in Windows

Tomoro Miyake

第 1 章　　はじめに

インターネットが企業活動に必須の要素となっている現在，企業はネットワークセキュリティへの対応を迫られている．しかし，今までのセキュリティ対策はファイアウォール，IDS(Intrusion Detection System)など外部からの不正アクセス，ウイルス対策に目が向けられ，イントラネット内部への対策はあまり重要視されてこなかった．イントラネット内部のセキュリティ対策としては，簡単なユーザ認証，アクセス管理などしか行われていないのが現状であり，実際には，情報漏洩，ウイルス感染等の被害は内部利用者によるものが多く，イントラネット内部のセキュリティ強化が重要となっている．ネットワークセキュリティの代表的な既存技術としては IPsec(Security Architecture for Internet Protocol)が挙げられる．現在では VPN(Virtual Private Network)を構築する際に広く利用されている．IPsec の動作モードにはパケットデータ部のみを暗号化するトランスポートモードと，ヘッダを含めたパケット全体を丸ごと「データ」として暗号化し新たな IP ヘッダを付加するトンネルモードがあり，トンネルモードは主として VPN で使用される．しかし，設定が煩雑であり，通信する二点間で全ての設定が一致していなければ通信が成立しない．このため VPN のような特定 2 点間ならともかく，多台数間の IPsec 通信を手動設定で行うのは実用上ほぼ不可能である．また，IPsec はトランスポートモードとトンネルモードに互換性が無く，個人単位の通信グループとドメイン単位の通信グループが混在するような環境には向いていない．つまり，IPsec は管理負荷が大きくシステム構成が頻繁に変わる場合には適していないといえる．

そこで，我々はシステム構成の変化やユーザの移動が発生してもネットワーク管理の負荷が発生しない柔軟性と，セキュリティ兼ね備えたグルーピング通信を可能とするネットワークシステムの概念として FPN(Flexible Private Network)を提案している[1]，[2]．また，FPN を実現する手段として，GSCIP(Grouping for Secure Communication for Internet Protocol)と呼ぶ一連のセキュア通信アーキテクチャを検討している．GSCIP はシステム構成が変化しても動的に動作処理情報を生成する動的処理解決プロトコル DPRP(Dynamic Process Resolution Protocol)[3]，ノードが移動して IP アドレス変化しても P2P で通信の継続が可能な Mobile PPC(Mobile Peer-to-Peer Communication)[4]，グローバルアドレス(Global Address;以下 GA)空間からプライベートアドレス(Private Address;以下 PA)空間への通信開始を可能とする NAT-f(NAT free protocol)[5]，などのプロトコル群によって構成される．現在，GSCIP は FreeBSD に実装して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 をより多くの人に評価してもらう為には，Windows に機能を実装させることが必須である．そこで本稿では，GSCIP を Windows に実装する方式について検討を行った．

第 2 章　　FPN と GSCIP

2．1　FPN(Flexible Private Network)

FPN とは，ユビキタスネットワーク環境において，フレキシブル(柔軟)かつセキュアな通信グループを実現することができるネットワークの概念である. 図 1 に FPN の概念を示す. FPN では，個人単位とサブネット単位の要素が混在した環境でも通信グループの定義ができる．また，ホストがサブネットの内外を自由に移動してもグループの定義は維持される．ホストおよびサブネットは複数のグループに重複帰属することができ，ホストやサブネットといったグループ単位の違いを意識する必要がない．FPN ではこのようなネットワーク環境を実現するために，以下に示す位置透過性，移動透過性，アドレス空間透過性の実現を目指している．

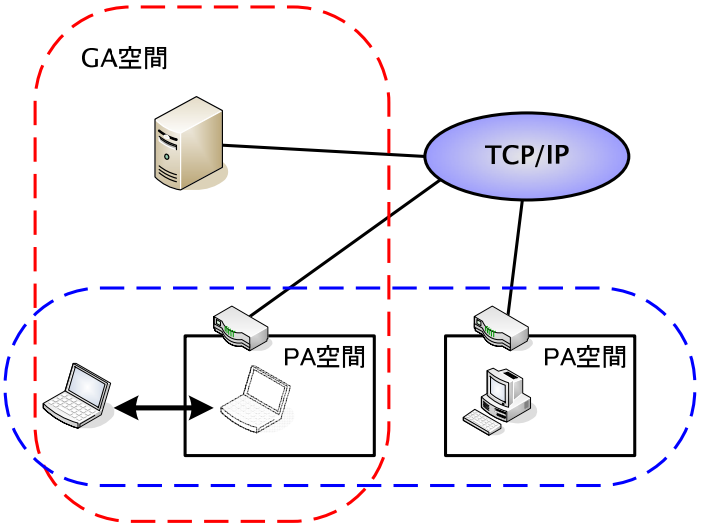

図 1　FPN の概念

(1)　位置透過性

全てのホストやサブネットは移動することを想定しており，ホストが特定のサブネットの内外を移動するなどしてネットワーク構成が変化しても，システムが動的にその変化を学習し，通信グループの関係を維持することができる．このとき設定情報をネットワーク管理者が更新する必要はない．この機能を位置透過性と呼ぶ.

(2)　移動透過性

通信端末が通信中に別のネットワークへ移動し，IP アドレスが変化すると，それまでの通信が継続できない．これは，TCP/IP 通信においては，通信端末を識別する IP アドレス自体に位置の情報を含んでいるため，ネットワークの移動前後で IP アドレスが異なり，別の通信と判断されてしまうためである．しかし，ユビキタスネットワーク環境においては，移動により IP アドレスが変化しても，上位アプリケーションには IP アドレスが変化したことを隠蔽し，それまでの通信を継続できることが望ましい．この機能を移動透過性と呼ぶ.

(3)　アドレス空間透過性

IPv4 の通信環境では，グローバルアドレス空間とプライベートアドレス空間があり，両者を接続するためにアドレス変換装置(以下 NAT)が存在し，その間の通信に制約がある．その理由は，NAT のアドレス変換テーブルが，プライベートアドレス空間からグローバルアドレス空間へのアクセスで始まる場合のみに生成されるためで，グローバルアドレス空間からプライベートアドレス空間へ通信を開始することができない．だがグローバルアドレス空間とプライベートアドレス空間を意識せず通信できる事が望ましい．この機能をアドレス空間透過性と呼ぶ．

FPN では以上三つの透過性の実現を目的とし，GSCIP を構成するプロトコル群にも統一性を持たせている．

2．2　GSCIP

FPN を実現するためには様々な方式が考えられる．厳重なセキュリティを要求するのであれば IPsec，モバイル環境を実現するには Mobile IP という既存技術があり，両者を合わせた技術も研究されている．しかし，これまで述べたような FPN で要求される柔軟性に対応できる技術はまだ無いため，我々は GSCIP という独自のセキュア通信アーキテクチャを提案している．図 2 に GSCIP の概念を示す．GSCIP では，同一のグループ鍵を持つ GSCIP 構成装置(以下 GE)同士が同一の通信グループを形成する．また，グループ鍵と通信グループは一対一に対応し，同一の通信グループの GE 同士は，グループ鍵を用いて暗号化通信を行う．GSCIP では，GE がサブネット単位であったり，個人単位であったり，両者が混在しても，ネットワーク構成の変化に対して柔軟に対応できる．セキュリティを確保するため，グループ鍵は定期的に更新される． GE は，グループ鍵により管理されるため，IP アドレスに依存しないグループ定義が可能であり，管理負荷を軽減できる効果もある．GSCIP は FPN システムを実現するためのセキュア通信アーキテクチャであり，以下に述べるようなプロトコルの集合体である．各プロトコルは個々に独立しており，単独での利用，他システムへの応用などを考えることも可能である．

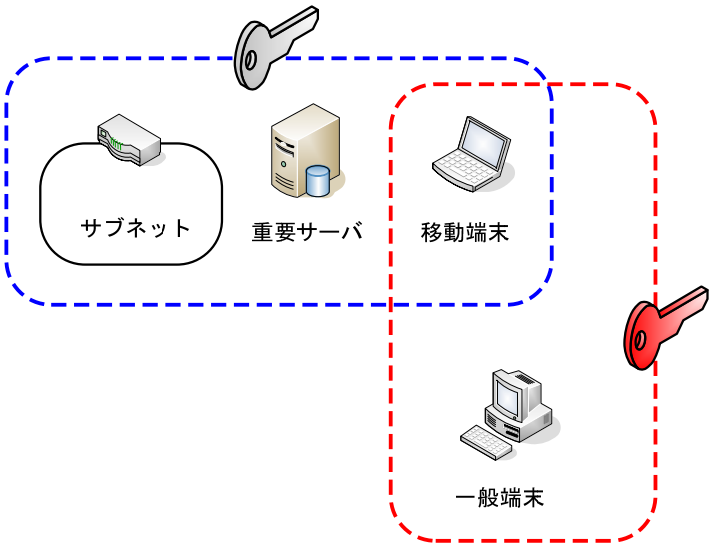

図 2　GSCIP の概念

2．2．1　DPRP

DPRP は端末間の通信開始に先立ち, 通信経路上のすべての GE 間で設定されている情報を相互に交換して，通信パケットの処理内容を決定し，動作処理情報テーブルを生成する．テーブルは送信元／宛先 IP アドレスとポート番号，プロトコル番号，処理内容，グループ鍵情報などの情報から構成されている．このうち動作処理情報は処理内容およびグループ鍵情報のことを示す．DPRP によって，位置透過性が実現される．

2．2．2　Mobile PPC

移動ノード(Mobile Node;以下 MN)と，通信相手ノード(Corresponded Node;以下 CN)は通信開始時に，移動前後の IP アドレスの対応関係を記したアドレス変換テーブル CIT(Connection ID Table)を IP 層において生成する．また，CIT はコネクション単位で生成される．通信中に MN が別のネットワークに移動した場合， MN は, 新しい IP アドレスとコネクション識別子の情報を含む CIT UPDATE (以下，CU) パケットを生成し，CN に宛てて送信する．CU は CN に対して移動を通知するとともに CIT の更新を要求する．図 3 に MN の IP アドレスが MN1 から MN2 へと変化した場合の IP アドレス変換処理を示す．CN から送信されるパケットの宛先 IP アドレスは，IP 層で CIT の情報を参照し移動後の IP アドレス MN2 へ変換される．このパケットを受信した MN は，同様に CIT を参照しパケットの宛先 IP アドレスを移動前の IP アドレス MN1 へ変換を行い上位層へ渡す．逆方向のパケットについても上記と同様なアドレス変換を行う．また，CIT は IP 層において，IP ルーティングテーブル，及び ARP キャッシュテーブルよりも階層的に上部に位置するため，自動的に移動後のアドレスについて両テーブルを参照することができる．このように IP 層内においてアドレス変換が行われるため，正しくルーティングが行われ，また，上位層へはアドレスの変化が隠蔽される．Mobile PPC によって， 移動透過性が実現される．

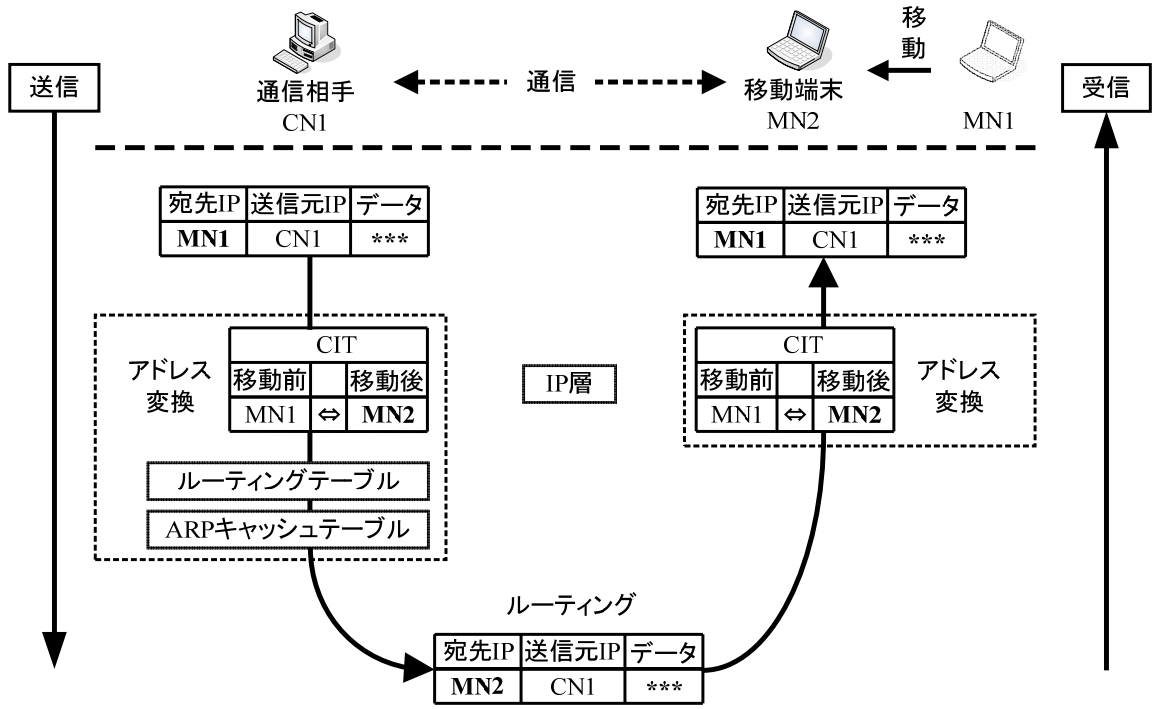

図 3　Mobile PPC における IP アドレス変換

第 3 章　　実装

3．1　Routing Table

TCP/IP では，IP アドレスに基づいてネットワーク・パケットのルーティングを行っている．このルーティング処理において重要な役割を持つのが「ルーティングテーブル」である．ルーティングテーブルには，宛先となるネットワーク・アドレスと使用するネットワーク・インターフェイスなどを記録した情報（エントリ）が多数格納されている．IP パケットのルーティング処理では，パケットの宛先 IP アドレスがどのルーティングテーブルのエントリにマッチするかを調べ，合致するものがあれば，指定されたネットワーク・インターフェイスへと送出する．また，どのエントリにもマッチしなければ，「デフォルト・ゲートウェイ」として指定されているルータへとパケットが送られる．

3．2　ARP Cache Table

IP パケットを送受信するためには，下位のデータリンク層の機能を使って IP パケットをカプセル化して送受信する必要がある．だが,そのときに使用される宛先のアドレスは，IP アドレスではなく，データリンク層のアドレスである MAC アドレスである．つまり IP パケットを送信するためには，その宛先の MAC アドレスが必要になる．このために利用されるのが ARP(Address Resolution Protocol)である．ARP により取得した情報はしばらくの間はキャッシュされる．

3．3　実装にあたっての課題

GSCIP は現在 FreeBSD の IP 層に実装し，基本動作を確認済みである．GSCIP を実現するモジュール群のことを GPACK と呼ぶ．図 4 に Free BSD における GPACK の実装概要を示す，GPACK は IP 層の適切な場所から呼び出されるサブルーチンとなっており，IP 層の一部を改造することにより実現している．一方，Windows は TCP/IP を含む OS がブラックボックスであり，改造ができない．そこで我々は，外部に公開されたインターフェースである NDIS (Network Driver Interface Specification)と呼ぶドライバに着目し，NDIS 内に GSCIP を実装することにした．ここで,実装に関して特に問題となるのが，移動端末が，通信相手と同じネットワークから別のネットワークに移動した際の Mobile PPC の処理である．NDIS は図 5 のように，TCP/IP より階層的に下部に位置する．このため,送信パケットは IP 層において移動前のアドレスについて経路決定された後で NDIS 内において移動後のアドレスにアドレス変換が行われる．このため，パケットが移動前のアドレスに向けて送信されてしまうという問題が発生する．正しくルーティングが行われるためには，アドレス変換後の，移動後のアドレスについて IP ルーティングテーブルと ARP キャッシュテーブルを参照し，MAC アドレスをデフォルト・ゲートウェイ宛てに書き換える必要がある．FreeBSD では GSCIP の呼び出し口を変えることにより，両テーブルを自動的に参照し，経路決定することができた．しかし Windows ではその方法が取れないため，以下のような実装方法を考案した．

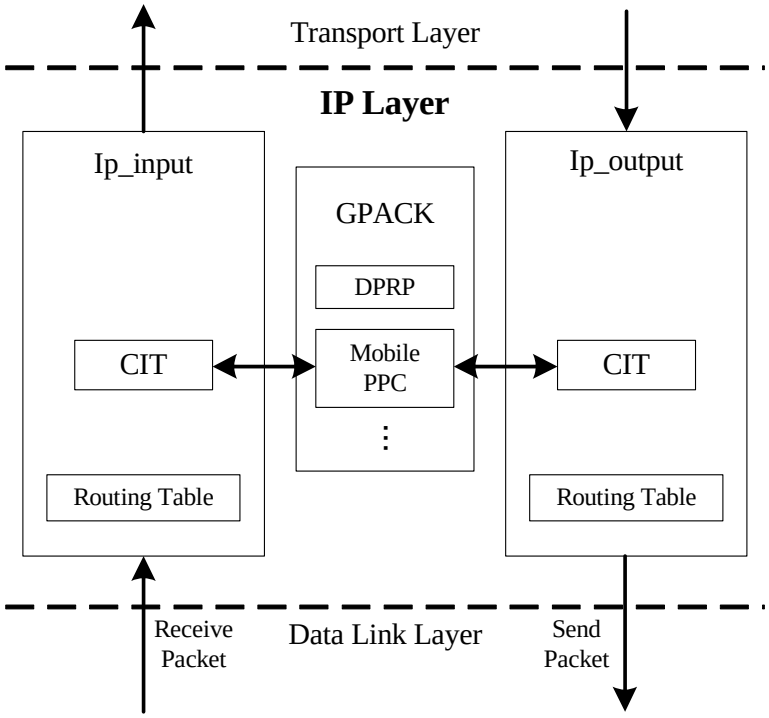

図 4　Free BSD における GPACK の実装

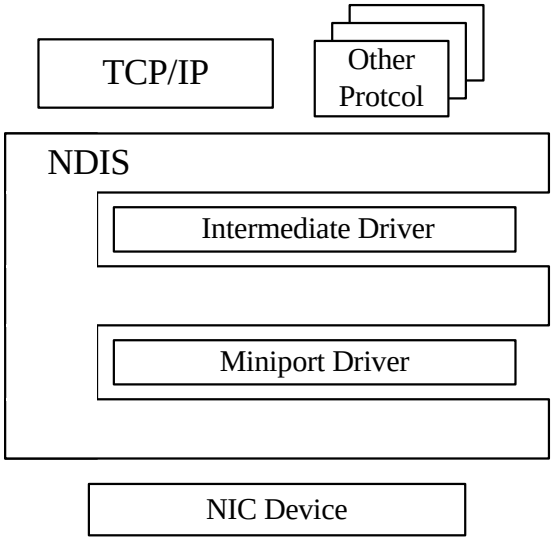

図 5　NDIS の構成

第 4 章　　実装方式

4．1　NDIS(Network Driver Interface Specification)

GSCIP の実装は，NDIS と呼ぶ外部に公開されたインターフェースを利用して実現する．NDIS とは Microsoft 社が定めたネットワークドライバの仕様であり，ネットワークドライバに機能を追加できるように定められている．また OS やアプリケーションソフトとドライバが通信するための手順，ドライバとネットワークカードが通信するための手順などを規定している．NDIS は以下の 3 つのドライバで構成されている．

(1)　**プロトコルドライバ**

アプリケーションから NIC を操作.

(2)　**ミニポートドライバ**

アダプタ経由でのデータの送信と受信など，ネットワーク アダプタの管理，中間ドライバや転送プロトコルドライバなど，上位レベルのドライバとのインターフェース通信.

(3)　**中間ドライバ**

NDIS 中間ドライバはオプションのドライバであり，プロトコルドライバに対してはミニポートドライバのように振る舞い，ミニポートドライバに対してはプロトコルドライバのように振舞うように実装される．NDIS 中間ドライバ内部での処理は自由.

4．2　実装方式

GSCIP の Windows への実装概要を図 6 に示す．GSCIP は NDIS の中間ドライバ内に実装する．Mobile PPC の処理中に IP ルーティングテーブルと ARP キャッシュテーブルの参照を実現するために，中間ドライバ内においてパケットにアドレス変換を施した後，GSCIP から一度アプリケーションに制御を渡し，そこから両テーブルの値を参照する．そして，参照した値を基に送信パケットの MAC アドレスをデフォルト・ゲートウェイ宛てに書き換えて送信する．この処理により課題となっていた Windows における移動透過性の実現が可能となる.

4．3　実装状況

NDIS 内を流れるパケットの解析・書き換え・追加を自由に行えるようにした．また，FreeBSD 独自の関数は，Windows では利用できないため，NDIS 内で利用できる関数に修正した．そして，GSCIP には DPRP，Mobile PPC，NAT-f の他にも， パケット長を変化させないままパケット完全性保証，本人性確認を可能とする実用的な暗号通信プロトコル PCCOM（Practical Cipher COMmunication protocol）や，非接触型 IC カードを用いて安全にグループ鍵を配送するプロトコル SPAIC（Secure Protocol for Authentication with IC Card）などがあり，現在は DPRP について実装を進めている段階である．GSCIP の Mobile PPC 以外のプロトコルについては，FreeBSD で実現済みのソースをほぼそのまま流用できる見込みである.

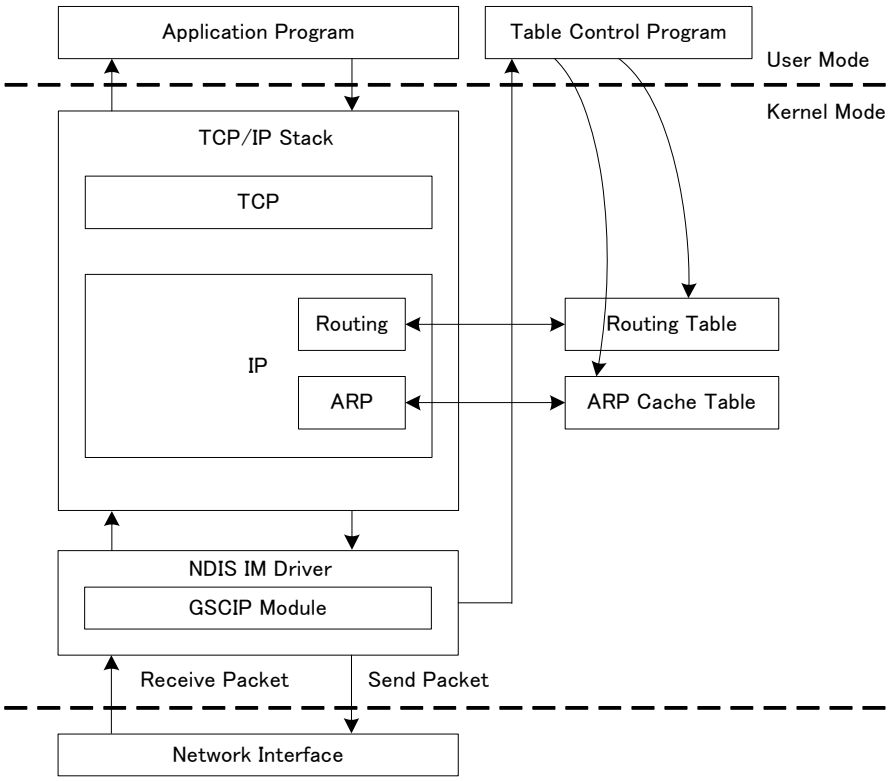

図 6　GSCIP の実装

第 5 章　　むすび

本稿では，FPN を構成するアーキテクチャである GSCIP を，Windows に実装する手法について検討した．今後は引き続き実装を完了させ，動作検証及び性能評価を行う．

参　考　文　献

[1] 鈴木秀和，竹内元規，加藤尚樹，増田真也，渡邊晃“フレキシブルプライベートネットワークを実現するセキュア通信アーキテクチャ GSCIP の提案”，2005-DICOMO2005 シンポジウム
[2] 名城大学理工学部，渡辺研究室，http://www．wata-lab．meijo-u．ac．jp/research/fpn1．html
[3] 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.
[4] 竹内元規，渡邊晃“エンドエンドで移動透過性を実現する Mobile PPC の提案と実装”，情報処理学会論文誌，Vol．47，No．12，pp．30，Dec．2006.
[5] 鈴木 秀和，渡邊 晃“アドレス空間透過性を実現する NAT-f の実装と評価”，マルチメディア，分散，協調とモバイル（DICOMO2006）シンポジウム論文集(I)，Vol．2006，No．6，pp．453-456，Jul．2006.