

# 透過型アドホックネットワーク構築ミドルウェア ATMOS

藤田 祥<sup>†1</sup> 江崎 浩<sup>†1</sup>

MANET は戦場や被災地にネットワークを構築する技術として注目されてきたが、日常的に MANET を利用するためには解決しなければならない課題が多い。特に、従来のネットワーク環境向けに作られたソフトウェアを MANET 環境では十分に利用することができず、また単一のネットワーク処理ロジックしかサポートしないソフトウェアでは異なる様々な MANET 環境に適応することができない。そこで我々はまずインターネットアーキテクチャのリンク層とネットワーク層にリンク補完層を導入し、上位層には従来のネットワークと同じリンク性質を見せながら MANET 環境の不完全なリンクを利用する手法を提案した。次にリンク補完層の処理をミドルウェアとしてユーザ空間に実装し、アプリケーションの通信パターンなどに応じてネットワーク処理のロジックを変更、拡張を可能にした。ミドルウェア上に IETF で仕様策定中のプロトコルである DYMO と SMF を実装し、従来のネットワーク向けに設計されたアプリケーションの mDNS が実際に MANET 環境上でそのまま利用できることを確認した。

## ATMOS: A middleware for Transparent MOBILE ad-hoc networking Systems

SHO FUJITA<sup>†1</sup> and HIROSHI ESAKI<sup>†1</sup>

Although MANET has been promising for constructing networks for battlefields and places struck by disaster, it is not regarded as being for daily use at all. We address two barriers that prevent MANET from being used for daily use. The first barrier is that MANET is not designed for being compatible with softwares that are developed for traditional networks. And the second barrier is that softwares for MANET environments must be extensible to support various situations. We introduce *link complement layer* between link layer and network layer in order to hide complexities of links in MANET environments. Then, we design and implement a middleware controlling the link complement layer in user space in order to allow users to dynamically change the network configuration. We implement DYMO and SMF on the middleware, and then demonstrate that mDNS can work in a MANET environment without any modification.

### 1. はじめに

これまで Mobile Ad-hoc Network (以下, MANET) は戦場や被災地など従来のネットワークインフラが存在しない状況でもネットワークを構築できる技術として注目を集めてきた。しかし、一度ネットワークインフラを失ってしまうとソフトウェアを新たに流通させるのが難しいので、MANET を非常時に機能させるためには普段から利用環境を整えておく必要がある。また、MANET では利用できるリンク技術の制約が少なく、移動端末をネットワークに含めることもマルチホップ通信によって広い地域をカバーできるので、MANET は日常生活にも有用な費用効果の高いネットワークになりうる<sup>3)</sup>。以上の理由から本研究では MANET を日常的な用途へ普及させることを目的とする。

我々は、MANET の普及を妨げているの要因として、まず従来のネットワーク環境向けに作られたソフトウェアは MANET 環境のメリットを十分に利用できないことを取り上げる。特に、自動設定やサービス発見などの根本となるアプリケーションが十分に利用できないことが問題となる。次に、MANET 環境ではインターネットアーキテクチャの各層の依存性が高いため、有効なネットワーク処理ロジックがアプリケーションの通信パターンやノードの移動パターン依存していることを取り上げる。

本研究の貢献は、リンク補完層を導入し MANET 環境で従来のネットワーク環境向けに作られた膨大なソフトウェア資産を利用出来るようにしたことと、リンク補完層を制御するミドルウェア ATMOS を様々なプラットフォームに実装しことを同時に達成したことである。

本論文の構成は以下の通りである。2 節では

<sup>†1</sup> 東京大学 情報理工学系研究科

MANET の性質とその普及を妨げている要因について述べる。3 節で MANET 環境のリンクを使いやすく抽象化するリンク補完アーキテクチャについて述べる。4 節では拡張性の高いアドホックネットワーク向けミドルウェアである ATMOS について述べる。5 節で評価を行い、6 節で関連研究について述べる。最後に 7 節で本論文をまとめ、今後の課題について述べる。

## 2. MANET 普及への課題

MANET は戦場や被災地でのネットワーク構築など特殊な利用は見込まれてはいるが、これまで日常的な利用については全く普及が進んでいない。本節では、まず MANET の性質について述べ、次に MANET の日常的な利用を妨げている要因について考察する。

### 2.1 MANET

従来のネットワークは、双方向かつ推移的かつ安定なリンク層を仮定して構築されてきた<sup>4)</sup>。つまり、ノード 1 が 2 のフレームを受信出来るならばノード 2 も 1 のフレームを受信出来（双方向）、ノード 1 と 2 が通信可能でノード 2 と 3 が通信可能ならばノード 1 と 3 は通信可能であり（推移的）、ノード間の接続性はほとんど変わらない（安定）。これら 3 つの仮定は、有線リンクでは自然であり容易に成り立つが、無線リンクでは一般に難しく成り立つとしても高コストである。図 1 は IEEE802.11 BSS モードを使ったネットワークのリンク利用状況を表している。IEEE802.11 BSS モードでは、アクセスポイント（以下、AP）と呼ばれる特別なノードとアソシエーション処理を完了したノードだけがそのネットワークに参加し、参加したノード間は全て AP を経由して通信する。参加ノードは AP を中心としたスター型のリンクを構成するので上記の 3 つの仮定が満たされる。一方で、利用可能なリンクは限定されて AP に全ての負荷が集中するので AP が通信スループットのボトルネックになったり、通信範囲が狭くなったりしてしまう。実際に、図 1 のノード 6 は AP の通信可能範囲外にいるためにネットワークに参加出来ず通信可能範囲にいるノード 2 と 3 とも通信出来ない。確かに、無線メディアを使ったり

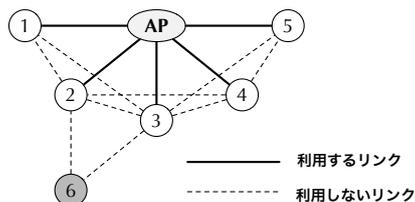


図 1 IEEE802.11 BSS モードのリンク利用状況

リンク層技術が 3 つの仮定を満たしている限り、ネットワーク層のソフトウェアはそれを従来の有線リンクを等価に扱うことが出来た。IEEE802.11 の BSS モードが爆発的に普及した背景にこの後方互換性があったことは間違いない。しかし、一方でリンク層に 3 つの仮定を押し付ける限り無線メディアの持つ可能性を最大限に引き出す事は出来ない。

MANET は 3 つの性質を仮定していないため、リンク層は制約が減り設計の自由度は増える。図 2 はノード配置は図 1 の AP を他ノードと同じ機能を持つノード 0 に置き換え、リンク層に IEEE802.11 IBSS モードを使ったネットワークのリンク使用状況である。各ノードは自由に他ノードとリンクを形成することができるので、図 1 のように全てのトラフィックが単一ノードに集中することがなく通信可能範囲も広がっている。しかし一方で、片方向リンクの検出やマルチホップ通信、そしてリンク状態のモニタなど IEEE802.11 BSS モードでは必要なかった複雑な処理を各ノードがこなさなければならない。

### 2.2 普及への課題

被災地や戦場などのシナリオではそもそも従来のネットワークが利用できないので MANET を利用していたが、日常的には従来のネットワークが同時に存在していることが多い。よって日常的に MANET を利用する機会を増やすためには、従来のネットワークによるリンクは失わずにそのまま活用しながらこの新しく利用可能になったリンクを有効活用しなければならない。また、アプリケーションや移動パターンをある程度仮定できる特殊な用途では適切なネットワーク処理ロジックをあらかじめ決定しておくことができるが、仮定できない状況では単一のネットワーク処理ロジックを提供するのではなく状況に合わせて複数から選択したり拡張したりできることが望ましい。しかし、現在の MANET にはこの両面で課題を抱えている。具体的な課題はそれぞれ以下の通りである。

- MANET によって増えたリンク活用することが難しい
- 通信ロジックを変更させることが困難

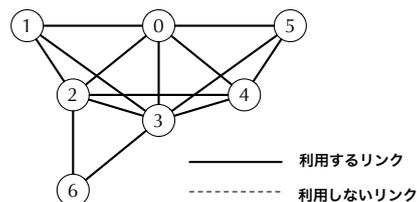


図 2 IEEE802.11 IBSS モードのリンク利用状況

### 2.2.1 従来のアプリケーションへの互換性

MANET によって新しく利用可能になったリンクを既存のアプリケーションが活用することは難しい。従来のネットワークではリンク層によって到達性が提供される範囲（以下、リンクローカルスコープ）がサブネットワーク全体をカバーしていた。しかしリンクローカルスコープは、一般にマルチホップ通信で通信範囲を広げる MANET 全体の中ではごく一部に過ぎず、MANET 全体の対象とした通信手段としては不十分である。MANET のマルチキャスト制御手法が議論されている<sup>11)</sup>がその対象となるマルチキャスト IP アドレスはサイトローカルスコープであり、やはりリンクローカルスコープと MANET 全体は明確に区別されている。よって、従来のネットワークで使われてきた DHCP などの自動設定と mDNS などのサービス発見はリンクローカルスコープを利用するため、特にマルチホップを許可することで新しく利用可能になったリンクには全く機能しない。自動設定やサービス発見は通信を始める前に必要な根本的な機能であり、これらがなく MANET を使い難いものになっている。

確かに MANET 独自の自動設定機構とサービス発見機構については多くの研究が存在する<sup>15)7)</sup>が、MANET のために全く別のソフトウェアを準備しなければいけないことは日常的な利用への普及を妨げる大きな要因の1つになっていると考えている。

### 2.2.2 ネットワーク処理ロジックの拡張性

単一のネットワーク処理ロジックでは想定される様々な MANET 環境をカバーすることが難しい。実際に、アプリケーションやノードの移動モデルが適切なルーティング手法は大きく異なることが分かっている<sup>2)</sup>。

## 3. リンク補完層

我々は、従来ネットワークのために作られたソフトウェア資産を生かしつつ無線リンクの持つ可能性を最大限に活用するために、ネットワーク層とリンク層の間に新たにリンク補完層を導入する。リンク補完層は、双方向かつ推移的かつ安定的であるという3つの仮定を必ずしも満たさない不完全なリンク層を補完しネットワーク層の間に完全なリンク層を提供する。リンク補完層は実リンクの上に仮想リンクを構築し、仮想リンクに接続するネットワークの仮想インターフェースを提供する。リンク補完層のアーキテクチャは図4のようになっている。各ノードはネットワーク層の経路表に加えて、リンク補完層には仮想リンク1つあたりに1つ経路表を持つ。このアーキテクチャは同じ 2.5

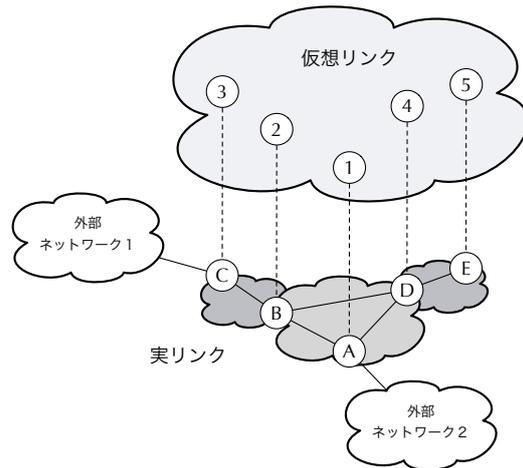


図3 実リンクと仮想リンク

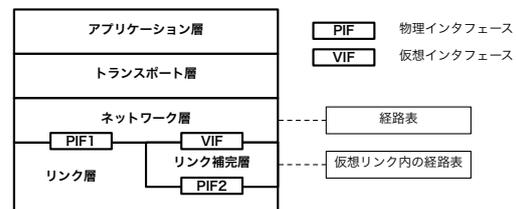


図4 リンク補完層アーキテクチャ

層の技術である MPLS<sup>13)</sup> のものと似ている。仮想リンクへ入るパケットにはリンク補完ヘッダが挿入され、仮想リンク内のルーティングはネットワーク層の経路表ではなくリンク補完層の経路表が利用されるからである。

図3は仮想リンクの具体例であり、IP アドレス A～E は物理インターフェースに割り当てられたものとし IP アドレス 1～5 は仮想インターフェースに割り当てられたものとする。便宜上物理インターフェースの IP アドレスを使ってノードを識別するものとする。また、物理インターフェースを通してノード C は外部ネットワーク1へ接続され、ノード A は外部ネットワーク2へ接続されている。リンク補完処理によって実リンクの複雑な構成が隠されるので、実リンク上ではノード C と E は最低でも3ホップは離れているが仮想リンク上では隣接しているように見える。仮想リンクは3つの仮定を満たす完全なリンクであり、ノード A と C は仮想リンクを経由してユニキャストはもちろんリンクローカルマルチキャストで通信できる。以下ではリンク補完層を内で行われる具体的な処理を送信処理と受信処理に分けて述べる。ただし各ノードのリンク補完層の経路表を、宛先 IP アドレスが仮想リンク外

にある場合と自分自身のアドレスの場合は、パケットを上位のネットワーク層に渡すように設定している。

### 3.1 送信処理

ネットワーク以上の層から送信されるパケットはまずネットワーク層の経路表に基づいて処理される。もし、次ホップが仮想リンク上のノードになった場合はパケットはリンク補完層に渡され、MANET 環境に必要な処理が施される。最後にパケットはリンク補完層を挿入された後リンク層に渡される。

例としてノード B と C の間の通信を考える。ノード B は物理インターフェースに割り当てられた B と仮想インターフェースに割り当てられた 2 の 2 つの IP アドレスを持つ。同じくノード C も C と 3 の 2 つの IP アドレスを持ち、実リンクでも仮想リンクでもノード B と隣接している。ノード B から IP アドレス C に向かってパケットを送信する場合は当然次ホップの IP アドレスは C で物理インターフェースを通る。一方 IP アドレス 3 に向かって送信する場合には、次ホップの IP アドレスは 3 になり仮想インターフェースを通過してリンク補完層に渡されることになる。

### 3.2 受信処理

リンク層に取ってが受信したパケットを直接ネットワーク層に渡すべきか、リンク補完層に渡すべきかは自明である。パケットにリンク補完層が含まれている場合のみリンク補完層に渡せばよい。リンク補完層に渡されたパケットはリンク補完層の経路表に基づいて処理される。パケットの宛先が自分自身の IP アドレスだった場合や次ホップが実リンク側に存在した場合は、パケットはリンク補完ヘッダを取り除かれた後ネットワーク層に渡される。次ホップが仮想リンク内だった場合は、MANET 環境に必要な処理をしてから再び物理インターフェースから送信される。

例としてノード C が仮想リンク側から宛先 IP アドレス C のパケットを受信した場合を考える。パケットはリンク補完ヘッダを含むのでリンク補完層に渡される。リンク補完層はリンク補完ヘッダを剥がした後パケットをネットワーク層に渡し、最終的にアプリケーション層に届けられる。アプリケーションからみると、パケットが仮想リンク側から届いた場合も実リンク側から届いた場合も、受けとるデータは全く同じように扱うことができる。

また別の例としてノード B がノード A から仮想リンクを通過して宛先 IP アドレス 1 のパケットを受信すると、やはりパケットはリンク補完層に渡される。リンク補完層の経路表によって次ホップの IP アドレスが 1 に決まるので、パケットはリンク補完ヘッダを含

んだまま再び物理インターフェースから送信される。

## 4. ATMOS ミドルウェア

MANET を様々な用途に利用することを考えると単一のネットワーク処理ロジックを提供するのではなく、複数のネットワーク処理ロジックから選択したり既存のネットワークロジックを拡張したりできることが望ましい。そこで、我々はリンク補完層をカーネル空間ではなくユーザ空間にミドルウェアとして実装し A Middleware for Transparent MOBILE networking(以下、ATMOS) と名付けた。ATMOS はアプリケーションに透過的にネットワーク処理のロジックを変更することができる。

ATMOS の機能はリンク補完層のどの部分を扱うかで 3 つに分けられる。

**ネットワーク層インターフェース** リンク補完層とネットワーク層の間を繋ぐ部分(図 4 の VIF)

**リンク層インターフェース** リンク補完層とリンク層を繋ぐ部分(図 4 の PIF2)

**内部インターフェース** リンク補完層の内部を拡張する部分

以下でそれぞれについて詳しく述べる。

### 4.1 ネットワーク層インターフェース

リンク補完はユーザ空間のデーモンプロセス(以下、ATMOS デーモン)によって処理されるため、仮想リンクに入ってリンク補完の対象となるパケットをカーネルのネットワーク層からユーザ空間へ渡す機能と仮想リンクから出て通常のネットワーク処理に戻るパケットをユーザ空間からカーネルのネットワーク層へ渡す機能が必要となる。これらをネットワーク層インターフェースと呼ぶ。

ATMOS デーモンは TUN/TAP デバイス<sup>1)</sup> を使ってネットワーク層インターフェースを実現している。TUN/TAP デバイスは様々なプラットフォームで利用可能なデバイスドライバであり、ネットワークの仮想インターフェースと対応するファイル記述子の組を提供する。仮想インターフェースから送信されたパケットはファイル記述子からバイト列として読み込むことができ、逆にファイル記述子に書き込まれたバイト列はパケットとして仮想インターフェースに到着させることができる。

### 4.2 リンク層インターフェース

ユーザ空間でパケットにリンク補完ヘッダを挿入した後は実際にそれらを物理インターフェースから送信する機能が必要である。これらをリンク層インターフェースと呼ぶ。

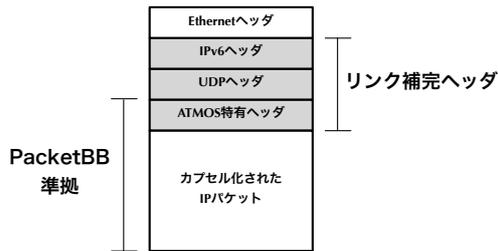


図5 リンク補完ヘッダの実装

ATMOSのリンク補完ヘッダは図5のようにネットワーク層にIPv6、トランスポート層にUDP、データ部分にTLV形式を使っていて拡張性に優れたPacketBBフォーマット<sup>5)</sup>を用いている。IPv6についてはリンクローカルに関する機能のみを使う。つまり、ユニキャストアドレスとマルチキャストアドレスはそれぞれfe80::/64とff02::/112に含まれておりホップリミットも1であるので、このパケットはオペレーティングシステムの機能では中継されない。

#### 4.3 内部インターフェース

MANET環境でルーティングやフラディングを効率的に行うためのロジックをATMOS内に実装する必要がある。ATMOSを拡張しロジックを実装するための機能を内部インターフェースと呼ぶ。

ATMOSではMANET環境の変化に応じてインタラクティブにパラメータやアルゴリズムを書き換えるためにスクリプト言語のPythonで実装した。またATMOSではClick<sup>10)</sup>のように細かな機能単位にモジュール化されており、それらのモジュールの接続関係を設定ファイルに記述することにより単純なパラメータ変更だけでなくネットワーク処理のロジックそのものを変更することができる。今回用いた設定は図6である。

特に近隣探索や経路探索を行うモジュールにはメッセージの送受信を行う機能が必要である。ATMOSメッセージの送受信に関しては以下の基本関数を提供している。

- *handleMessage(message, neighbor)* はリンク補完ヘッダを含むメッセージを受信したときにATMOSによって呼び出されるコールバック関数である。messageは受信したパケット全体のバイト列であり、neighborはリンク補完ヘッダを含むパケットを送信したノードのソケットアドレスである。
- *multicastMessage(message)* はmessageを全ての近隣ノードに送信する関数である。全ての近隣ノードに送信するために、仮想インターフェース

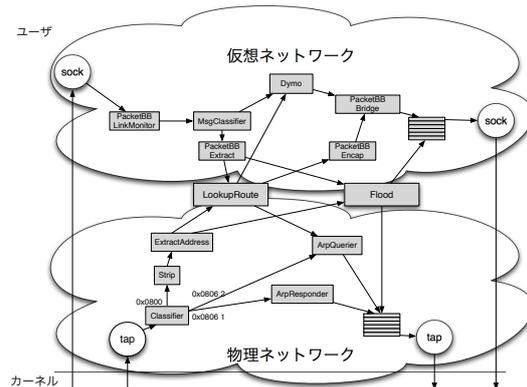


図6 設定サンプル

に対応する全ての物理インターフェースからIPv6リンクローカルマルチキャストする。

- *unicastMessage(message, neighbor)* はmessageをソケットアドレスneighborに向けてユニキャストする関数である。

実際にこれらの基本関数を使ってユニキャストのためのDYMOとマルチキャストのためのSMFを実装した。

##### 4.3.1 ユニキャストサポート

ユニキャストルーティングプロトコルとしてDynamic MANET On-demand routing<sup>8)</sup> (以下、DYMO)をATMOS上に実装した。DYMOはIETFのMANETワーキンググループで仕様策定されているMANET用のユニキャストルーティングプロトコルであり、Dynamic Source Routing<sup>9)</sup>やAd Hoc On Demand Distance Vector<sup>12)</sup>と同様、必要になるまで経路を確立しないリアクティブ型に分類される。一般にリアクティブ型のプロトコルは、経路が確立されるまでパケットをプロトコルスタック内にバッファする必要があるため、プラットフォーム依存の方法で実装されることが多いが、ATMOSミドルウェアを使って実装したDYMOはATMOSが動作する全てのプラットフォームでそのまま動作する。

##### 4.3.2 マルチキャストサポート

Simplified Multicast Flooding<sup>11)</sup> (以下、SMF)は同じくIETFで仕様策定されているマルチキャストプロトコルである。SMFは厳密にマルチキャストツリーを管理するプロトコルではなく、フラディングと冗長なパケット抑制を組み合わせた仕組みである。

CPU	Pentium M 1700MHz
メモリ	2G
NIC	EEPROM/1000
OS	linux 2.6.22

表 1 実験マシンの構成

## 5. 評価

ATMOS を使って、パケット処理のオーバーヘッドについての前提実験と後方互換性についての確認実験を行った。2つの実験のどちらについても参加ノードは全て ATMOS を図 6 の設定で利用している。

### 5.1 前提実験

まず、ATMOS のパケット処理オーバーヘッドを測定するための前提実験を行った。実験環境には表 1 の構成のコンピュータを 2 台を UTP で接続したものをを用いた。仮想リンク越しに netperf で TCP のスループットを計測した。10 回計測して平均が 8.90Mbps、標準偏差が 0.381Mbps でほぼ安定していた。全ての試行で ATMOS は CPU をほとんど食いつぶしており、Python インタプリタの実行速度がボトルネックになっていると考えられる。計測された ATMOS のスループットは IEEE802.11a や 11g の無線リンクの実効帯域をサポートしきれていない。しかし、無線リンクをマルチホップ使った通信のスループットでは 1 ホップで達成されるスループットの 1/3 から 1/4 程度になるのが一般的であり、MANET 環境のマルチホップパスを従来ネットワーク向けのアプリケーションで利用する目的には十分であった。今後、より貧弱な CPU を持つアクセスポイントへの対応や高速な IEEE802.11n リンクへの対応を考えた場合、パケット転送部分のコードのみ C で書き直すことを検討している。

### 5.2 mDNS 実験

マルチキャストについて後方互換性を実証するために図 7 の構成で mDNS を使った。物理インターフェースと仮想インターフェースに割り当てたアドレスの関係は図 4 と同様である。ノード A~C は ATMOS をサポートし仮想リンクに接続しているが、ノード D~F は接続していない。

ノード B が物理インターフェースから mDNS クエリを送信するとノード A, D, E, F からレスポンスを得た。一方、ノード B が仮想インターフェースから mDNS クエリを送信すると、ATMOS の SMF 機構により仮想リンク全体にクエリが配布され、ノード A, C からレスポンスを得た。このように ATMOS は従来のネットワーク接続性を壊さずに MANET 環境

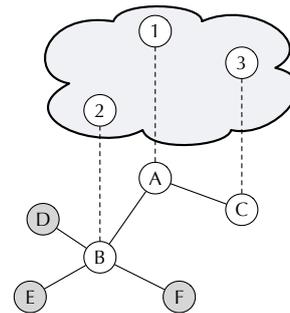


図 7 mDNS 実験構成

向けの技術を適用し、ネットワークの通信範囲を広げることが確認できた。

## 6. 関連研究

IEEE802.11s<sup>6)</sup> は IEEE802.11 系プロトコルのメッシュネットワーク拡張である。上位層から見るとマルチホップ通信は意識することではなく、その点で ATMOS と類似している。しかし、IEEE802.11s プロトコルがそのベースとなるリンク層技術を IEEE802.11 系プロトコルに限っているのに対し、ATMOS は様々なリンク層技術をサポートできる点で異なっている。また、IEEE802.11 プロトコルは仕様策定プロセスが比較的閉鎖的であり IEEE802.11 はデバイスドライバの一部として実装されバイナリで配布されることも多い。ATMOS のように処理を自由に再設定出来ることは出来ない。

protolib<sup>14)</sup> はネットワークプロトコルやアプリケーションをネットワークシミュレータを含む様々なプラットフォーム上に実装するための C++ クラスライブラリである。OLSR や SMF など様々な MANET 環境向けのプロトコルが protolib を利用して実装されている。protolib は現時点で WIN32 プラットフォームと NS2 や OPNET などのシミュレーションプラットフォームをサポートしており移植性の面では非常に優れている。しかし、既存のソフトウェアとの下方互換性やネットワーク処理アルゴリズムの動的な変更などについては考えられていない。

## 7. おわりに

本論文で我々はリンク補完層を導入し従来のネットワーク向けに開発されたソフトウェアを使って、MANET 環境を有効活用出来るようにした。また、リンク補完層上の処理ロジックを自由に組み立てるためのミドルウェアとして ATMOS を実装した。さらに実際に ATMOS 上に DYMO と SMF を実装し、mDNS

などの従来ネットワーク向けに作られたアプリケーションをそのまま利用出来る事を確認した。

今後の課題として、ほげー。

## 謝 辞

本論文を執筆するにあたり多くの方のお世話になった。特に、度重なる議論に付き合ってくれた研究室同僚の吉田 薫氏と IJ 技術研究所の宇夫 陽次朗氏、鳥慶一氏に感謝したい。

## 参 考 文 献

- 1) VTun Project. <http://vtun.sourceforge.net/>.
- 2) Josh Broth, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *International Conference on Mobile Computing and Networking*, 1998.
- 3) Raffaele Bruno, Marco Conti, Enrico Gregori, and National Research Council. Mesh Networks: Commodity Multihop Ad Hoc Network. *IEEE Communications Magazine*, 2005.
- 4) I. Chakeres, J. Macker, and T. Clausen. Mobile Ad hoc Network Architecture. *Work in Progress*, November 2007. draft-ietf-autoconf-manetarch-07.
- 5) T. Clausen, C. Dearlove, J. Dean, and C. Adjih. Generalized MANET Packet/Message Format, draft-ietf-manet-packetbb-04. IETF Internet Draft (ID). "Work in Progress", March 2008.
- 6) W. Steven Conner, Jan Kruys, Joseph Kim, and Juan Carlos Zuniga. 802.11s: WLAN Mesh Networking. Technical report, The IEEE 802 LAN/MAN Standards Committee, 2006.
- 7) S. Helal, N. Desai, V. Verma, and Choonhwa Lee. Konark - a service discovery and delivery protocol for ad-hoc networks. In *Wireless Communications and Networking*, March 2003.
- 8) I. Chakeres and C. Perkins. Dynamic MANET On-demand (DYMO) Routing. *Work in Progress*, February 2008. draft-ietf-manet-dymo-12.
- 9) D. Johnson, Y. Hu, and D. Maltz. . *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*, February 2007.
- 10) Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The Click modular router. *ACM Transactions on Computer Systems*, 18(3), August 2000.
- 11) J. Macker and SMF Design Team. Simplified Multicast Forwarding for MANET. *Work in Progress*, February 2008. draft-ietf-manet-smf-05.
- 12) C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. *Network Working Group RFC 3561*, July 2003.
- 13) E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. *Network Working Group RFC 3031*, January 2001.
- 14) Ian Taylor, Brian Adamson, Ian Downard, and Joe Macker. AgentJ: Enabling Java NS-2 Simulations for Large Scale Distributed Multimedia Applications. In *Distributed Frameworks for Multimedia Applications*, 2006.
- 15) Ulas C. Kozat and Leandros Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *INFOCOM*, volume 3, pages 1965 – 1975, March 2003.