NTMobileの実用化に向けた統合的枠組の検討

納堂　博史[†1,a)]　杉原　史人[†2]　鈴木　秀和[†1]　内藤　克浩[†3]　渡邊　晃[†1,b)]

概要：筆者らは，NAT の有無や IPv4/IPv6 を問わず，相互に通信接続性と移動透過性を実現する技術として NTMobile（Network Traversal with Mobility）を提案してきた．NTMobile は，Linux カーネルモジュールで動作するカーネル実装型，アプリケーションレベルで動作するフレームワーク組込型，VPN サービスとして動作する VpnService 利用型などの実装モデルがある．また，NTMobile 端末の認証においては，従来通りのパスワードを利用する方法のほか，公開鍵証明書を利用する方法が検討されている．動作仕様においても，通信を行う双方の端末が NAT 配下にいる場合と，少なくとも片方の NTMobile 端末がグローバルネットワークにいる場合とでは，トンネル構築の方法が異なっている．今後，NTMobile を実用化していく上では，これらの仕様が統合され，1 つのパッケージとして提供できることが望ましい．そこで，本稿ではこれまで様々な方式が混在していた NTMobile の仕様を改めて整理し直し，統合的な枠組みを設計した．本稿の枠組みにより，NTMobile は，実装方法や認証方法の違いに関わらず，動作が共通化され，実用化に近づいた．

キーワード：移動透過性，NAT 越え，仮想 IP アドレス，トンネル

Study of the Total Framework towards Practical Implementation in NTMobile

Hiroshi Nodo[†1,a)]　Humihito Sugihara[†2]　Hidekazu Suzuki[†1]　Katsuhiro Naito[†3]
Akira Watanabe[†1,b)]

Abstract: We have been proposing NTMobile as the technology that solves the NAT traversal problem in IPv4 netoworks and provides flexible mobility in IPv4 and IPv6 networks. NTMobile has some types of implementation models; a kernel implementation type that runs on Linux kernel, a framework built-in type that operates on the application level, and VpnService type that acts as a VPN service etc.. For the authentication of NTMobile terminals, a method utilizing the public key certicate is now under study together with the current password method. In operational specifications, tunnel construction methods are different, in the case both terminals are under NATs, and in the case one terminal is in a global network. In order to commercialize NTMobile, their specifications should be integrated and provided as one package. Therefore in this paper, we have reorganized these technologies and designed the total framework. With this framework, it is said that NTMobile has approached to a practical use.

Keywords: Migration Transparency, NAT Traversal, Virtual IP Address, Tunnel

1. はじめに

スマートフォンとタブレット端末の普及や無線技術の躍進により，モバイルデバイスのインターネット利用は拡大

[†1] 現在，名城大学大学院理工学研究科
Presently with Graduate School of Science and Technology, Meijo University
[†2] 現在，三重大学大学院工学研究科
Presently with Graduate School of Engineering, Mie University
[†3] 現在，愛知工業大学情報科学部
Presently with Faculty of Information Science, Aichi Institute of Technology
[a)] hiroshi.noudou@wata-lab.meijo-u.ac.jp
[b)] wtnbakr@meijo-u.ac.jp

の一途を辿っている．このため，IPv4 ネットワークでは，IP アドレスの数が不足し，プライベートアドレスにより，IP アドレスを節約しながら利用している．しかし，2011 年に IANA（Internet Assigned Numbers Authority）[*1] の新規に割り振り可能な IPv4 アドレスが枯渇して以降，RIR（Regional Internet Registry）においても次々と割り振り可能な IPv4 アドレスが枯渇している [1]．また，IPv4 ネットワークにおいては，NAT（Network Address Translation）の存在により，NAT の内側の端末へ NAT の外側から通信を開始できないといった NAT 越え問題もある．近年，CGN（Carrier Grade NAT）のように，通信事業者単位で NAT を導入する動きもあり [2]，NAT が多段になることを想定していない既存の NAT 越え技術では対応できない可能性がある [3]．また，NAT に改造を施す NAT 越え技術では，通信事業者側の NAT にも改造が必要となり，導入の敷居が高くなる．

IPv4 アドレス枯渇問題を根本的に解決するため，IPv6 が策定され，1999 年から IANA で IPv6 アドレスの割り振りが行われている．IPv6 は IPv4 と比較して膨大なアドレス空間を有しており，枯渇の心配はないが，IPv4 と IPv6 には互換性がなく，IPv4 利用者と IPv6 利用者は直接通信ができない．総務省の報告 [4] によると，2013 年 3 月末において，ISP（Internet Service Provider）における IPv6 インターネット接続サービス等を標準提供しているのは約 50%であるが，小規模事業者に限ると 30%ほどに留まっている．CATV 事業者の場合，IPv6 対応サービスを提供しているのは 15%に満たない．また，同報告書によると，Google の計測結果から，全世界の利用者のうち，インターネット接続で IPv6 を利用しているのは約 1.5%と報告されている．このように，未だに IP ネットワークの主流は IPv4 であると言えるだろう．以上のことから，IPv6 は徐々に普及して行くが，IPv4 が消滅することはなく，半永久的に IPv4 ネットワークと IPv6 ネットワークが共存していくことが推測される．

一方で，モバイルデバイスは持ち運んで利用されることが一般的であるため，移動通信の需要が増大している．IP ネットワークにおいては，IP アドレスが位置識別子と通信識別子を兼ねているため，通信中に IP アドレスが変わると通信が切断するという問題がある．モバイルデバイスは，4G や WiFi などの複数の無線インターフェースが搭載されており，通信の分散が行われているが，無線インターフェースを切り替えると IP アドレスが変わってしまうため，この問題が現れる．

これらの問題を解決するため，様々な技術が実現されている [5–10] が，IPv4 ネットワークと IPv6 ネットワークの一方でのみで動作するものであったり，NAT 越えと移動透過性を同時に実現したり，通信相手が一般端末の場合に移動透過性を実現したりできない，通信経路が冗長化する，既存の通信インフラをそのまま利用できないなどの問題がある．

著者らは，IPv4/IPv6 ネットワークにおいて，相互に通信接続性を確保でき，同時に NAT 越え，移動透過性を実現する NTMobile（Network Traversal with Mobility）を提案してきた [11–15]．NTMobile は，移動端末（以後 NTM 端末）に仮想 IP アドレスを割り当て，アプリケーションはこの仮想 IP アドレスを用いて通信を行う．実際の通信は実 IP アドレスでカプセル化されて転送されるため，アプリケーションは常に同じ仮想 IP アドレスで通信を継続することとなり，実 IP アドレスが変化しても通信が切断されることはない．NTMobile では，原則としてエンドエンド通信を行うよう設計されており，通信経路の冗長化が起きにくい．また，NAT などの既設の通信インフラに改造を施すことなくこれらの機能を実現しているほか，NTM 端末における実装方法には Linux カーネルモジュールで動作するカーネル実装型 [16]，アプリケーションレベルで動作するフレームワーク組込型 [17,18]，VPN サービスとして動作する VpnService 利用型 [19] などがあり，導入時の敷居を低くしている．既に Linux のほか，Android および iPhone においても実装および動作検証が行われており，その有効性が確認されている．

このように，多彩な実装方法により導入の敷居が低くなっている中，実用化にあたって新たな要求仕様が浮上してきた．例えば，グループセキュリティなどの今後の拡張性を鑑みて，NTM 端末に公開鍵を持たせたいという要求がある．NTM 端末同士で通信を行う時に暗号化に要する一時鍵を確実に交換するために動作シーケンスをよりわかりやすくシンプルにしたいという要求もある．そこで，本稿では，実用化に向けて複数の方式が混在していた NTMobile の仕様を整理し，1 つの統合した枠組みとして再設計を行った．この時，多くの OS および言語に対応するために実装についても再検討を行った．

以下，2 章で NTMobile の概要を説明し，3 章で NTMobile の実装方式および動作を説明する．4 章で NTMobile の統合的な枠組みを説明し，5 章で実装について述べ，最後に 6 章で全体をまとめる．

2. NTMobile

図 1 に NTMobile の概要を示す．NTMobile は，NTM 端末のほか，グローバルのデュアルスタックネットワーク上に設置され，NTM 端末の端末情報の管理やトンネル経路指示を出す DC（Direction Coordinator），IPv4/IPv6 間の通信，一般端末との通信，および NTM 端末が互いに異なる NAT 配下にいる場合などに通信を中継する RS（Relay Server），NTM 端末の認証を行う AS（Account Server），

[*1] http://www.iana.org/

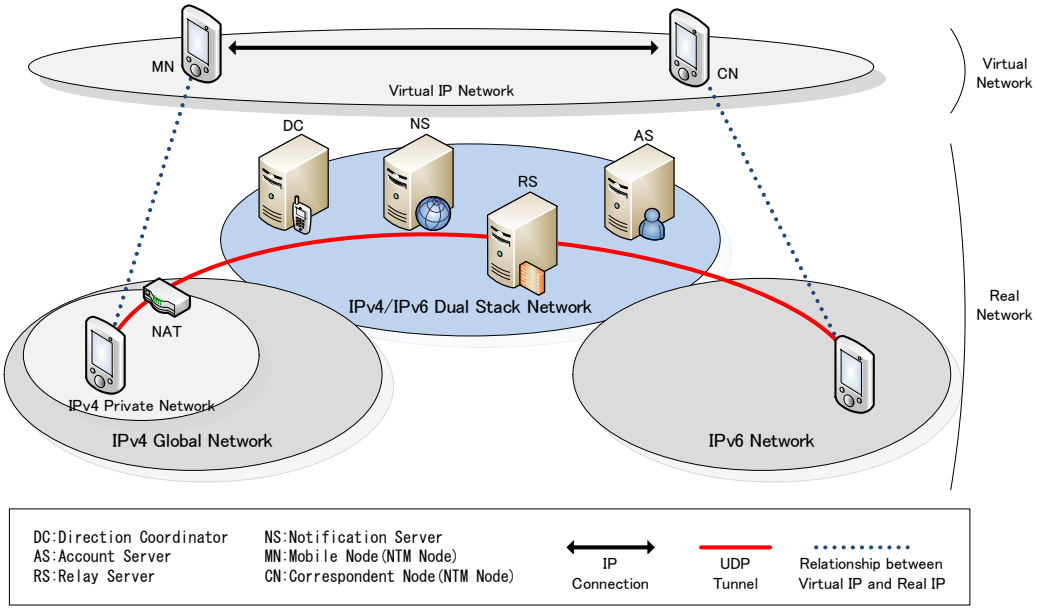


図 1　NTMobile の概要
Fig. 1　Overview of NTMobile

NAT 越えに関するオプション装置である NS（Notification Server）で構成される．また，DC や RS，NS は複数台設置可能で，負荷分散可能な仕組みとなっている．また，AS は NTMobile において 1 台だけ存在するサーバーであるが，一定時間内の再認証は DC で可能であり，大きな負荷がかからないよう設計されている．

2.1　動作概要

NTM 端末は，初回認証時に AS に問い合わせることで，自身を管理する DC などの初期情報を取得する．DC には自身の実 IP アドレスなどの情報を通知し，仮想 IP アドレスを取得する．NTM 端末のアプリケーションは，仮想 IP アドレスを用いて通信を行うことで，移動や無線インターフェースの切り替えに伴う実 IP アドレスの変化を考慮する必要がない．実際の通信は実 IP アドレスを用い，UDP でカプセル化される．原則としてエンドエンド通信を行うが，自身と通信相手がいずれも NAT 配下にいる場合や，IPv4/IPv6 間の通信，通信相手が一般端末の場合には RS を経由する．なお，自身と通信相手が両方とも NAT 配下にいる場合であっても，NAT の種類によってはエンドエンド通信に即座に切り替わる [20]．さらに，RS を経由する際も最適な RS が選択され [21]，通信品質を保つよう設計されている．

2.2　端末の認証情報登録

NTM 端末の認証情報（アカウント）は，予め AS に登録されている必要がある．現在，メールアドレスとパスワードで AS にアカウント登録ができる．

2.3　端末の認証

NTM 端末は，端末起動時に自身を管理する DC にログイン処理を行う．DC は，データベースに当該端末情報があればそれを用い，無ければ AS に問い合わせることで認証を行う．正規の端末と認証されれば，その結果を NTM 端末に返す．これらの通信は TLS 通信を用いて安全に行われる．この認証で NTM 端末と DC の共通鍵を共有する．その後，NTM 端末は DC に対して自身の端末情報を登録し，仮想 IP アドレスを取得する．これらの通信には前述の共通鍵が用いられ，安全かつ高速に行われる．

2.4　端末の通信

NTMobile では，DNS クエリをフックして NTMobile 独自のアドレス解決処理が開始される．通信を行う NTM 端末（以下、MN（Mobile Node））はフックした DNS クエリから通信相手の NTM 端末（以下、CN（Correspondent Node））の FQDN を抽出し，MN を管理する DC（以下、DC_{MN}）へ通信要求を送信する．DC_{MN} は，受信した FQDN の NS レコードから CN を管理する DC（以下、DC_{CN}）の FQDN を取得し，CN の端末情報を DC_{CN} から取得する．DC_{CN} の発見には DNS の NS レコードを利用する．これにより，DC_{MN} は MN と CN の情報を取得できたので，最適な経路を構築するよう MN と CN に通知する．通知されたメッセージにより，MN と CN は UDP トンネルを構築し，MN は DNS クエリの回答としてアプリケーションに CN の仮想 IP アドレスを通知する．以上の処理によって，MN のアプリケーションは仮想 IP アドレスを用いた IP 通信を行い，実 IP アドレスは隠蔽される．

実 IP アドレスが変化した時は，同様の処理によって UDP トンネルを再構築することで，アプリケーションは通信を継続することができる．

3. 新たな要求仕様

NTMobile を実用化するにあたり，下記のような要求仕様が出された．これらはすべて実用化に向けて検討しなくてはならない事項である．そこで，本稿ではこれらすべてを満たす枠組みを検討した．

- サーバー間の信頼関係
  DC 間や AS-DC 間の信頼関係を構築するための動作仕様が求められる．
- OpenID への対応
  近年，OpenID Foundation[*2] のような OpenID に対応したサービスが増えており，AS への認証情報登録に活用 [22] できる拡張性が求められる．
- 公開鍵への対応
  現在の NTMobile ではパスワードを基本として認証を行い，共通鍵を利用して安全な通信を実現している．また，通信ごとに一時共通鍵を MN 側で 2 つ生成し，NTM 端末に 2 つ，RS に 1 つ配布することで，DC/RS の管理者であってもエンドエンドの通信内容が秘匿されるよう設計されている．一方，Linux Box のような装置の場合，パスワード認証より公開鍵を埋め込んだ方が安全性が高まり管理が容易であるほか，グループ認証等の今後の拡張性を考えると，公開鍵認証への対応が求められる．
- 動作シーケンスの統一
  MN や CN の属するネットワークに応じて動作シーケンスが異なる場合がある．NAT の有無や IPv4/IPv6 ネットワークの差を NTMobile が吸収し，アプリケーションには隠蔽する必要があるため，すべてを同一とすることはできないが，可能な限り共通化されることが望ましい．また，NTMobile には NAT 越えの方法が複数あるが，どのような場合にどの方法で NAT 越えを実現するのかを整理する必要がある．
- 実装方式によらない相互互換性
  NTMobile は，カーネル実装型，フレームワーク組込型，VpnService 利用型など複数の実装方式がある．カーネル実装型は，LinuxOS においてアプリケーションの改造を要さず，高速に動作する一方，root 権限を要する．フレームワーク組込型は，アプリケーションの改造が必要となるが，iOS や Android 上において一般権限で動作する．VpnService 利用型は，アプリケーションの改造を要さず，Android 上において一般権限で動作する．このように，実装方式ごとにメリットとデメリットがあり，今後も複数の実装方式をサポートすることが NTMobile の普及のためには必要である．この場合，実装方式にかかわらず相互に互換性を有する必要がある．

[*2] http://openid.net/

4. 統合的枠組みの検討

本章では，これまで独立して検討がなされてきた NTMobile の実装方法や NAT 越え手法を統合する．

4.1 動作シーケンス

4.1.1 NTM 端末の認証

図 2 に NTM 端末認証のシーケンス図を示す．NTM 端末は，Login Request によって AS へログインを行う．AS は，正しく認証された場合，NTM 端末と NTM 端末を管理する DC との間の共通鍵を生成し，Key Distribution に格納して DC に送信する．DC から確認応答として NTM ACK を受信したら，NTM 端末に共通鍵と NTM 端末の FQDN を Login Response によって通知する．NTM 端末は通知された FQDN の NS レコードより，DC の FQDN を取得する．以上の処理により，NTM 端末は自身を管理する DC の FQDN と，DC と安全に通信するための共通鍵を得る．一度ログインして共通鍵を取得していれば，共通鍵の有効期限が切れるまで再ログインの必要はない．NTM 端末と AS との間の通信は，TLS によって AS の正当性確認を行った上で暗号化される．また，AS と DC の間の通信は TLS 相互認証および暗号化がなされ，安全に通信が行われる．

NTM 端末に公開鍵を持たせる場合は，NTM 端末と AS 間で TLS 相互認証ができるため，Login Request にパスワードを格納する必要はない．

OpenID を利用する場合は，AS に OpenID 認証を要求し，正しく認証された場合に Login Request 以降の処理を行う．この場合においても，Login Request にパスワードを格納する必要はない．

このように，NTM 端末がパスワード認証をする場合，公開鍵を持った上で OpenID を利用する場合と利用しない場合で，統一的なシーケンスとなるようにした．これにより，NTMobile は多彩な認証方法をサポートすることができる．一般利用には取り扱いが簡単なパスワード方式を利用し，企業利用等で堅牢なセキュリティが求められる場合および LinuxBox には公開鍵方式を利用するなどの選択ができ，相互に通信を行うことができる．

4.1.2 ネットワーク情報の登録と NAT 越え

図 3 にネットワーク情報登録のシーケンス図を示す．NTM 端末は，端末起動時及びネットワーク切り替え時に，Registration Request を DC に送信する．DC は，このメッセージによって NTM 端末の実 IP アドレスや NAT の外側 IP アドレスを取得し，仮想 IP アドレスを割り当てる．割り

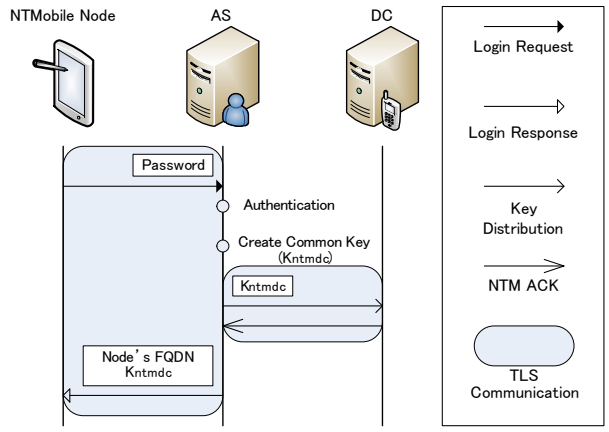


図 2　端末認証シーケンス
Fig. 2　Sequence of Node Authentication

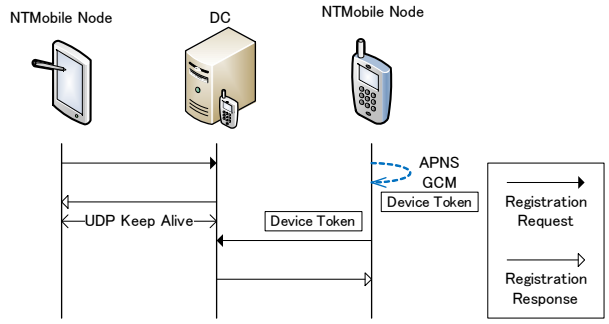


図 3　情報登録シーケンス
Fig. 3　Registration Sequence

当てた仮想 IP アドレスは，Registration Response によって NTM 端末に通知される．NTM 端末は，Registration Response を受信した後，DC との通信経路を確保するために 20 秒間隔で UDP によるキープアライブを行う．

NTMobile には，UDP のキープアライブパケットによるネットワークの輻輳を避けるため，TCP を用いたキープアライブを行う NS（Notification Server）が準備されている [23]．NAT のアドレス変換テーブルの維持時間は，UDP と比較して TCP の方が長いため，キープアライブパケットの送信間隔を長くすることができる．このときの情報登録シーケンスを**図 4** に示す．NTM 端末は，TCP によるキープアライブを希望する場合，その旨を Registration Request に記載して DC に通知する．DC は，NTM 端末と NS との間の通信に用いる共通鍵を生成し，NS および NTM 端末に通知する．NTM 端末は通知された NS の FQDN と共通鍵を利用して，NS に TCP で Notification Registration を送信する．NS からの確認応答を待ち，以降 NTM 端末と NS 間で約 1 時間間隔で TCP によるキープアライブが行われる．

NTM 端末が Android 端末や iPhone の場合は，GCM（Google Cloud Messaging），APNS（Apple Push Notification Service）といったプッシュ通知サービスを利用す

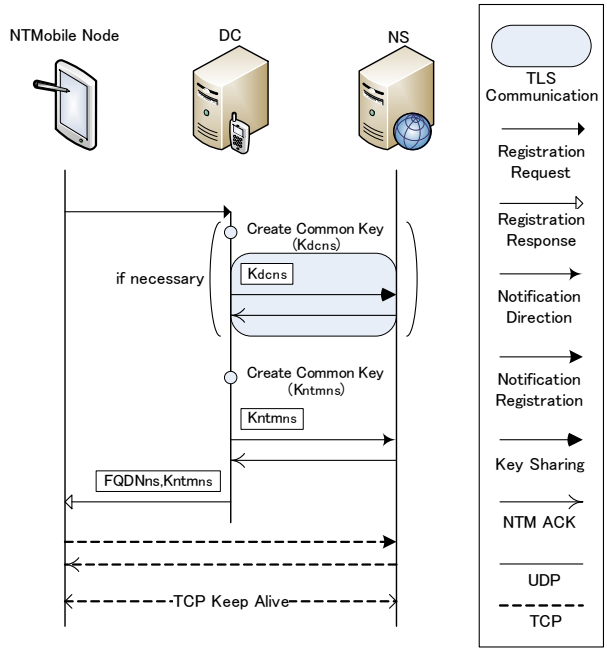


図 4　NS を利用した情報登録シーケンス
Fig. 4　Registration Sequence using NS

る [14]．この場合，事前にプッシュ通知サービスを利用するためのデバイストークンを取得しておき，Registration Request で DC に通知する．DC との通信が必要な場合は，このプッシュ通知サービスが利用されるため，NTM 端末と DC との間でキープアライブは不要である．

このように，複数の NAT 越え方式が共存できるようにシーケンスを整理した．

4.1.3　通信経路の生成

図 5 に名前解決シーケンスを示す．MN は，CN と通信するために CN の FQDN（以下，FQDN_{CN} とする．）を用いて DNS によるアドレス解決を行う．この DNS 要求をフックして，DC_{MN} に Direction Request を送信する．DC_{MN} は，FQDN_{CN} の NS レコードから DC_{CN} の FQDN を取得し，CN の端末情報を取得する．以上により，DC_{MN} は MN と CN の端末情報を取得でき，通信経路を決定することができる．

なお，NTM 端末に公開鍵を持たせる場合は，Direction Request に MN の公開鍵を格納する．これは，後述するシーケンスに則って CN へ配送するためである．

図 6 および**図 7** に経路決定後，UDP トンネルによる通信経路生成までのシーケンスを示す．図 6 は，MN と CN のうち，CN が NAT 配下に属している場合の例である．DC_{MN} は DC_{CN} を経由して CN から MN にトンネル構築をするよう Route Direction で指示を出す．DC_{CN} は，4.1.2 項で示した NAT 越え方法に則り CN へ Route Direction を転送する．DC_{MN} は，CN へ Route Direction の転送が完了したことを確認応答によって確認すると，MN へ Tunnel Request の受信待ちをするよう Route Direction

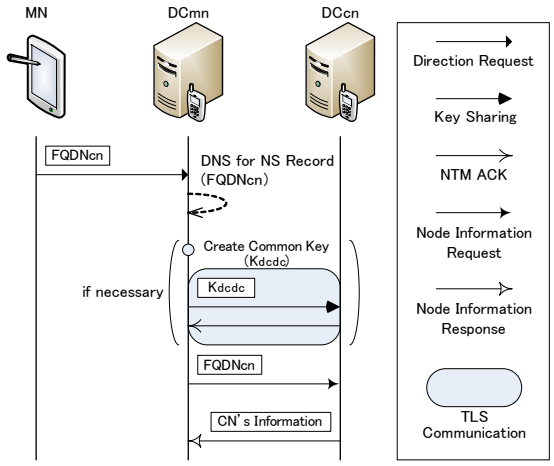


図 5　名前解決シーケンス
Fig. 5　Sequence of Name Resolution

で指示する．次に，CN はアプリケーションパケットの暗号化に用いる共通鍵 K_{mncn} を生成し，Route Direction で配布された共通鍵 K_{tmp} で暗号化した上で Tunnel Request に格納する．Tunnel Request は，Route Direction で配布された別の共通鍵 K_{tun} で暗号化されて MN へ送信される．以上の処理により，MN と CN との間で UDP トンネルの構築，および通信に用いる共通鍵 K_{mncn} の共有が完了したので，MN はフックした DNS 要求への回答として CN の仮想 IP アドレスをアプリケーションへ返す．

図 7 は，MN および CN が両方とも NAT 配下に属している場合の例である．DC_{MN} は RS へ Relay Direction を送信して中継指示を出す．DC_{MN} は RS からの確認応答を待ち，CN へ Tunnel Request の受信待ちをするよう Route Direction を送信する．確認応答の後，MN に対し，RS を経由して CN へ Tunnel Request を送信するよう Route Direction を送信する．Route Direction を受信した MN は，アプリケーションパケットの暗号化に用いる共通鍵 K_{mncn} を生成し，Route Direction で配布された共通鍵 K_{tmp} で暗号化した上で Tunnel Request に格納する．Tunnel Request は，Route Direction で配布された別の共通鍵 K_{tun} で暗号化されて RS へ送信される．RS は受信した Tunnel Request をそのまま CN へ転送する．CN からの Tunnel Response も同様に転送する．以上の処理により，RS を経由した UDP トンネルの構築，および通信に用いる共通鍵 K_{mncn} の共有が完了したので，MN はフックした DNS 要求への回答として CN の仮想 IP アドレスを返す．

NTM 端末が公開鍵を利用する場合は，MN からの Direction Request および CN 宛ての Route Direction に MN の公開鍵を格納して CN へ届ける．また，CN からの確認応答および MN 宛ての Route Direction に CN の公開鍵を格納して届ける．この結果，MN と CN は互いにそれぞれの通信相手の公開鍵を取得できるので，共通鍵 K_{mncn} を通信相手の公開鍵で暗号化し，Tunnel Request で送り届ける．

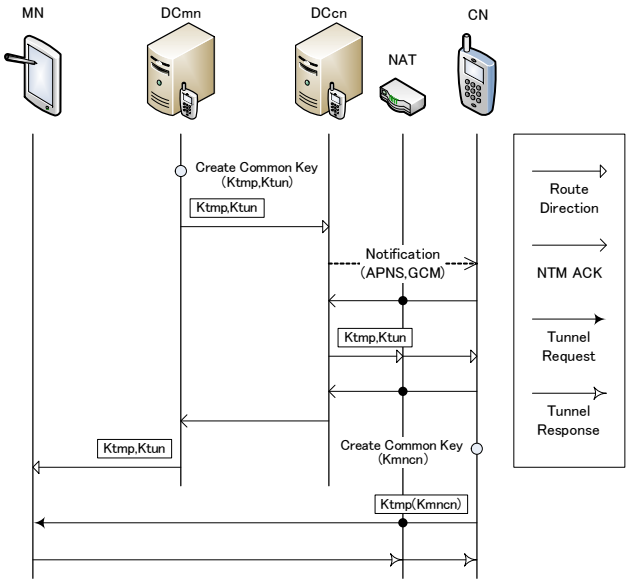


図 6　トンネル生成シーケンス（CN が NAT 配下）
Fig. 6　Sequence of Tunnel Creation(if CN is under NATs)

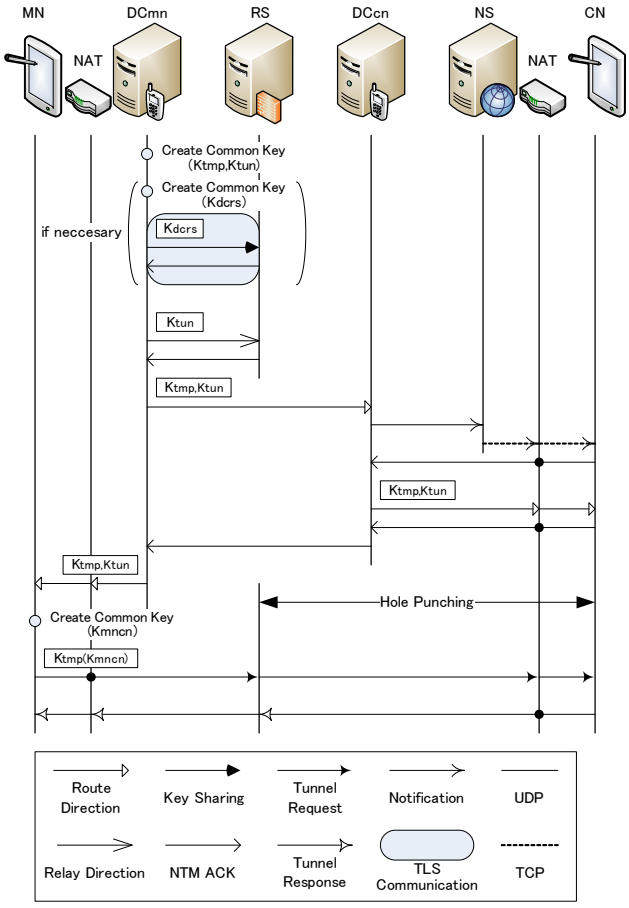


図 7　トンネル生成シーケンス（MN と CN が NAT 配下）
Fig. 7　Sequence of Tunnel Creation(if both MN and CN are under NATs)

4.2 共通鍵の整理

NTMobile で利用されるすべての装置群は，NTM 端末の除き予め公開鍵証明書を取得している．この公開鍵は，共通鍵の共有に用いられ，装置群の信頼関係および通信の安全性はこの共通鍵によって担保される．**表 1** に NTMobile で用いる共通鍵をまとめる．NTM 端末を除く装置群は，初回通信時に TLS 相互認証による Key Sharing で共通鍵を共有する．以降の通信は，この共通鍵を用いて暗号化される．NTM 端末は，DC との共通鍵 K_{ntmdc} を AS から TLS 認証で取得する．NS を利用する場合は，NS との共通鍵 K_{ntmns} を DC から取得する．また，通信を行うとき，MN または CN は，Tunnel Request の暗号化に用いる共通鍵 K_{tun} を DC から取得する．K_{tun} は MN と CN のほか，RS を利用する場合には RS とも共有される．同時に，アプリケーションパケットの送受信に用いる共通鍵 K_{mncn} を暗号化する共通鍵 K_{tmp} を DC から取得する．ここで，K_{tmp} は RS とは共有されない点が重要である．K_{mncn} は MN または CN で生成され，K_{tmp} で暗号化される．暗号化された K_{mncn} は，Tunnel Request によって通信相手端末と共有される．Tunnel Request は MN-CN 間で直接または RS 経由で送信されるが，RS は K_{mncn} を複合するための共通鍵 K_{tmp} を持たない，このため，DC や RS の管理者であっても K_{mncn} は知り得ず，MN-CN 間で安全な通信を行うことができる．

4.3 要求仕様に対する考察

本節では，3 章で記した要求仕様について考察する．

- 装置群の信頼関係
  NTM 端末を除く装置群には公開鍵証明書を持たせる．初回通信時に，相互認証により共通鍵を共有し，以後の通信はこの共通鍵を利用することで，装置間の信頼関係を実現した．
- OpenID への対応
  NTM 端末の認証を NTMobile から分離することで，OpenID 認証にも対応できるようにした．OpenID を利用すると，AS にユーザーの個人情報を登録する必要がなくなり，よりセキュリティが向上する．NTM 端末が一度認証され，DC と共通鍵が共有されれば，共通鍵の有効期限までは再認証の必要はない．
- 公開鍵への対応
  NTM 端末に公開鍵を持たせる場合と持たせない場合とでシーケンスを統合した．トンネル生成時にお互いの公開鍵を交換することで，アプリケーションパケットの暗号化に用いる共通鍵の共有を公開鍵でも実現できるようにした．
- 動作シーケンスの統一
  NAT 越え手法の違いによる部分を除き，動作シーケンスを統一した．また，NTM 端末の動作仕様について，RS 利用の有無に関係なく同じ動作に統一した．
- 実装方式によらない相互互換性
  NTMobile の実装整理し，相互互換性を実現した．これについては，5 章で述べる．

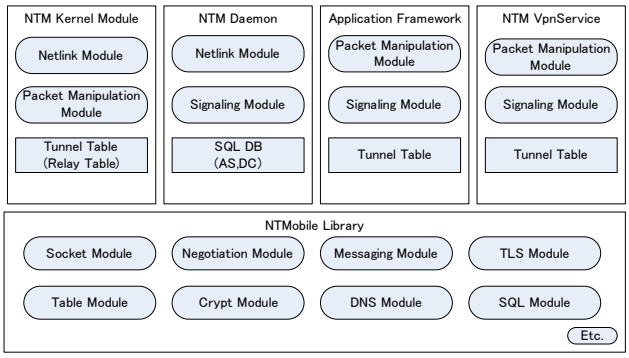


図 8 NTMobile の実装設計
Fig. 8 Architecture of NTMobile Implementation

5. 実装設計

図 8 に NTMobile の実装設計を示す．実装方式に依存しない部分を NTMobile Library として切り離して実装を行った．各種サーバーや NTM 端末のデーモンプログラムは共通の NTMobile Library を利用する．メッセージフォーマットやパケットの送受信処理などをライブラリとして実装することで，すべての実装方式において相互互換性が保たれる設計とした．

この実装方式により，NTMobile はすべての実装方式間の相互互換性が保たれている．また，32bitOS を前提にしていた実装を見直し，32bit/64bit の両方の OS に対応するよう実装を進めているほか，Windows などでも利用できるように OS のクロスプラットフォーム化も進めている．

6. まとめ

本稿では，これまで様々な実装方式や NAT 越え方法が混在していた NTMobile について，統合的な枠組みを提示した．NTMobile は，様々な実装方式があり，また IPv4 ネットワークにおける NAT 越え手法も 3 つの方式が準備されている．従来の動作シーケンスを見直し，これらの違いを極力最小となるよう設計した．また，今後の展望として OpenID への対応や公開鍵証明書を用いる方式へ拡張できる柔軟性も確保できた．これまで，NTMobile の全容を知るためには個々の論文を集め，それらを組み立てて理解する必要があったが，本稿により，NTMobile の理解と実装が容易になった．

謝辞 本研究は，日本私立学校振興・共済事業団平成 27 年度学術研究振興資金（若手研究者奨励金）の助成を受けたものである．記して謝意を表する．

表 1　NTMobile で利用する共通鍵
Table 1　Common Key in NTMobile

	Creator	Common Owner	Scene of Use	Create Timing
K_{ntmdc}	AS	MN/CN,DC	Communication between MN/CN and DC	
K_{dcns}	DC	DC,NS	Communication between DC and NS	
K_{ntmns}	DC	MN/CN,NS	Communication between MN/CN and NS	first communication
K_{dcdc}	DC_{MN}	DC_{MN},DC_{CN}	Communication between DC_{MN} and DC_{CN}	(only once until key's expiration)
K_{dcrs}	DC_{MN}	DC_{MN},RS	Communication between DC and RS	
K_{tun}	DC_{MN}	MN,CN,RS	Tunnel Request,Hole Punching for RS	
K_{tmp}	DC_{MN}	MN,CN	Encrypt for K_{mncn}	before Tunnel Request
K_{mncn}	MN or CN	MN,CN	Communication between MN and CN	(every negotiation)

参考文献

[1] Huston, G.: IPv4 Address Report. 入手先 ⟨http://www.potaroo.net/tools/ipv4/⟩（参照 2015/10/03）.

[2] Jiang, S., Guo, D. and Carpenter, B.: An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition, RFC 6264 (Informational) (2011).

[3] 屏雄一郎，大岸智彦，勝野　聡：ISP への NAT 導入によるユーザ影響評価，電子情報通信学会誌，Vol. 93, No. 6, pp. 473–478 (2010).

[4] 総務省（オンライン）：IPv6 によるインターネットの利用高度化に関する研究会　第二次プログレスレポート．入手先 ⟨http://www.soumu.go.jp/main_content/000239088.pdf⟩（参照 2015/10/03）.

[5] Leung, K., Dommety, G., Narayanan, V. and Petrescu, A.: Network Mobility (NEMO) Extensions for Mobile IPv4, RFC 5177 (Proposed Standard) (2008).

[6] Moskowitz, R., Nikander, P., Jokela, P. and Henderson, T.: Host Identity Protocol, RFC 5201 (Experimental) (2008). Obsoleted by RFC 7401, updated by RFC 6253.

[7] Rosenberg, J.: Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols, RFC 5245 (Proposed Standard) (2010).

[8] Soliman, H.: Mobile IPv6 Support for Dual Stack Hosts and Routers, RFC 5555 (Proposed Standard) (2009).

[9] 関　顕生，岩田裕貴，森廣勇人，前田香織，近堂　徹，岸場清悟，西村浩二，相原玲二：IPv4 拡張した移動透過通信アーキテクチャ MAT の設計と性能評価，情報処理学会論文誌，Vol. 52, No. 3, pp. 1323–1333 (2011).

[10] カフレベド，福島裕介，原井洋明：Design and Implementation of ID-Layer Multipath Forwarding in HIMALIS Architecture (情報ネットワーク)，電子情報通信学会技術研究報告 = IEICE technical report : 信学技報，Vol. 114, No. 478, pp. 301–306 (2015).

[11] 内藤克浩，上醉尾一真，西尾拓也，水谷智大，鈴木秀和，渡邊　晃，森香津夫，小林英雄：NTMobile における移動透過性の実現と実装，情報処理学会論文誌，Vol. 54, No. 1, pp. 380–393 (2013).

[12] 鈴木秀和，上醉尾一真，水谷智大，西尾拓也，内藤克浩，渡邊　晃：NTMobile における通信接続性の確立手法と実装，情報処理学会論文誌，Vol. 54, No. 1, pp. 367–379 (2013).

[13] 鈴木秀和，内藤克浩，渡邊　晃：ユーザ空間における移動透過通信技術の設計と実装，マルチメディア、分散協調とモバイルシンポジウム 2014 論文集，Vol. 2014, pp. 1319–1325 (2014).

[14] Naito, K., Mori, K., Kobayashi, H., Kamienoo, K., Suzuki, H. and Watanabe, A.: End-to-end IP mobility platform in application layer for iOS and Android OS, *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, pp. 92–97 (online), DOI: 10.1109/CCNC.2014.6866554 (2014).

[15] Miyazaki, Y., Sugihara, F., Naito, K., Suzuki, H. and Watanabe, A.: Certificate based key exchange scheme for encrypted communication in NTMobile networks, *Proceedings of the 12th IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, No. RS8-5, pp. 1–5 (2015).

[16] 上醉尾一真，鈴木秀和，内藤克浩，渡邊　晃：IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価，情報処理学会論文誌，Vol. 54, No. 10, pp. 2288–2299 (2013).

[17] Naito, K., Mori, K., Kobayashi, H., Kamienoo, K., Suzuki, H. and Watanabe, A.: End-to-end IP mobility platform in application layer for iOS and Android OS, *Proceedings of the 11th IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 92–97 (2014).

[18] Kamienoo, K., Suzuki, H., Naito, K. and Watanabe, A.: Development of Mobile Communication Framework based on NTMobile, *Proceedings of the 7th International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, pp. 27–32 (2014).

[19] 山田貴之，鈴木秀和，内藤克浩，渡邊　晃：IPv4/IPv6 混在環境に対応した VpnService 型 NTMobile の性能評価，マルチメディア，分散，協調とモバイル (DICOMO2015) シンポジウム論文集，Vol. 2015, pp. 1784–1791 (2015).

[20] 納堂博史，鈴木秀和，内藤克浩，渡邊　晃：NTMobile における自律的経路最適化の提案，情報処理学会論文誌，Vol. 54, No. 1, pp. 394–403 (2013).

[21] 三宅佑佳，鈴木秀和，内藤克浩，渡邊　晃：NTMobile における端末移動後を考慮したリレーサーバ選択手法の提案，マルチメディア，分散，協調とモバイル (DICOMO2015) シンポジウム論文集，Vol. 2015, pp. 1792–1799 (2015).

[22] 上野泰輔，大久保陽平，鈴木秀和，内藤克浩，渡邊　晃：OpenID Connect を用いた NTMobile ユーザ認証スキームの検討，平成 27 年度電気・電子・情報関係学会東海支部連合大会講演論文集，No. B3-8 (2015).

[23] 杉原史人，内藤克浩，鈴木秀和，渡邊　晃，森香津夫，小林英雄：NTMobile における組み込み機器向けトラフィック削減手法の提案，マルチメディア、分散協調とモバイルシンポジウム 2014 論文集，Vol. 2014, pp. 1313–1318 (2014).

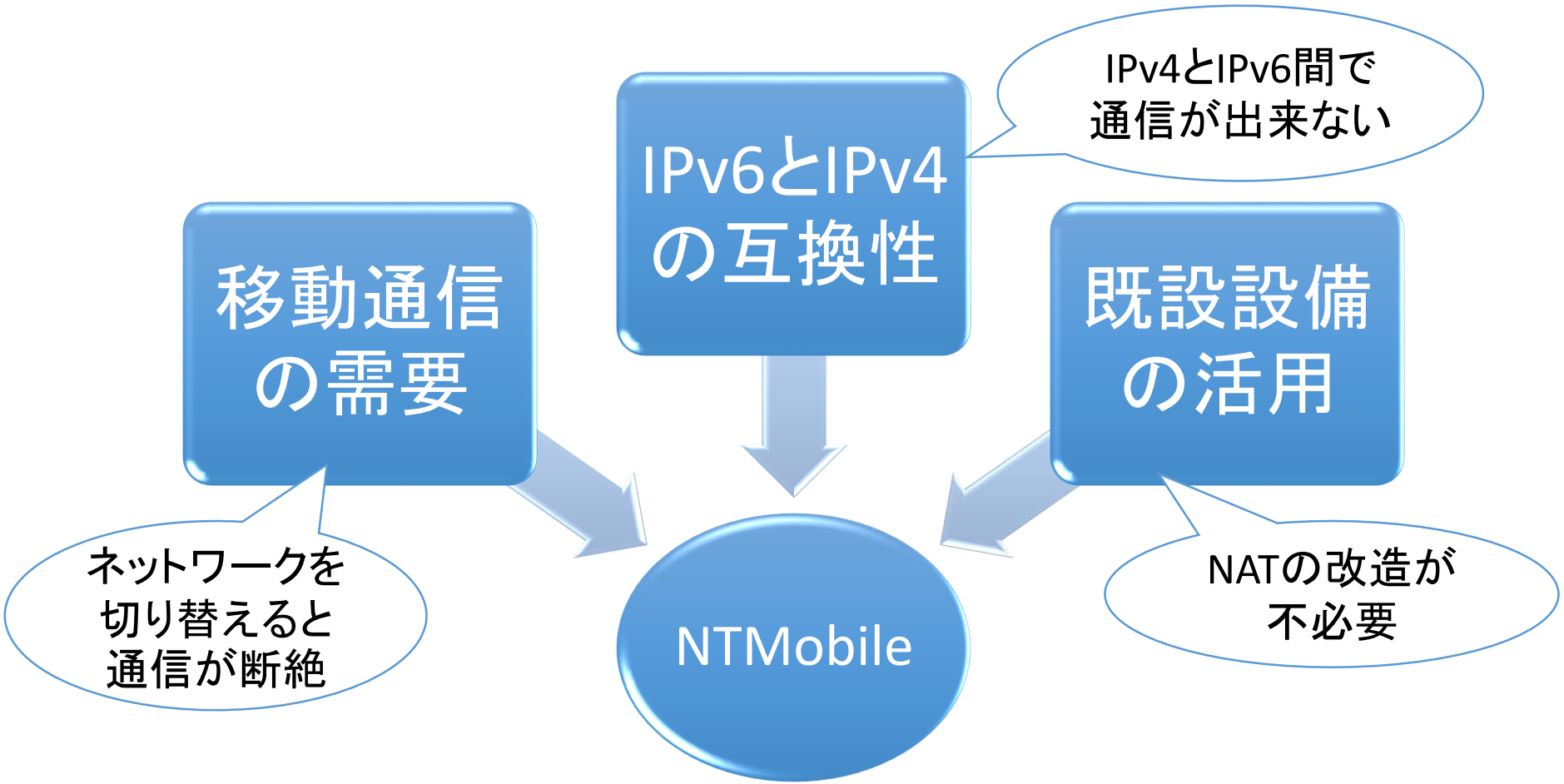
NTMobileの実用化に向けた統合的枠組の検討

名城大学　納堂 博史　　三重大学　杉原 史人　名城大学　鈴木 秀和
愛知工業大学　内藤 克浩　名城大学　渡邊 晃

目次

- はじめに
- NTMobileについて
- 統合的枠組について
- 実装について
- 考察
- まとめ

研究背景(1)



IPv4/IPv6問わず既存インフラで
通信接続性と移動透過性を実現

研究背景(2)

NTMobileの機能が個別に検討・実装

- 様々な実装方式の混在
- IPv4において複数のNAT越え手法が混在
- 通信をする端末ペアの位置関係で異なるUDPトンネルの構築手順

実用化に向けた新たな要求仕様

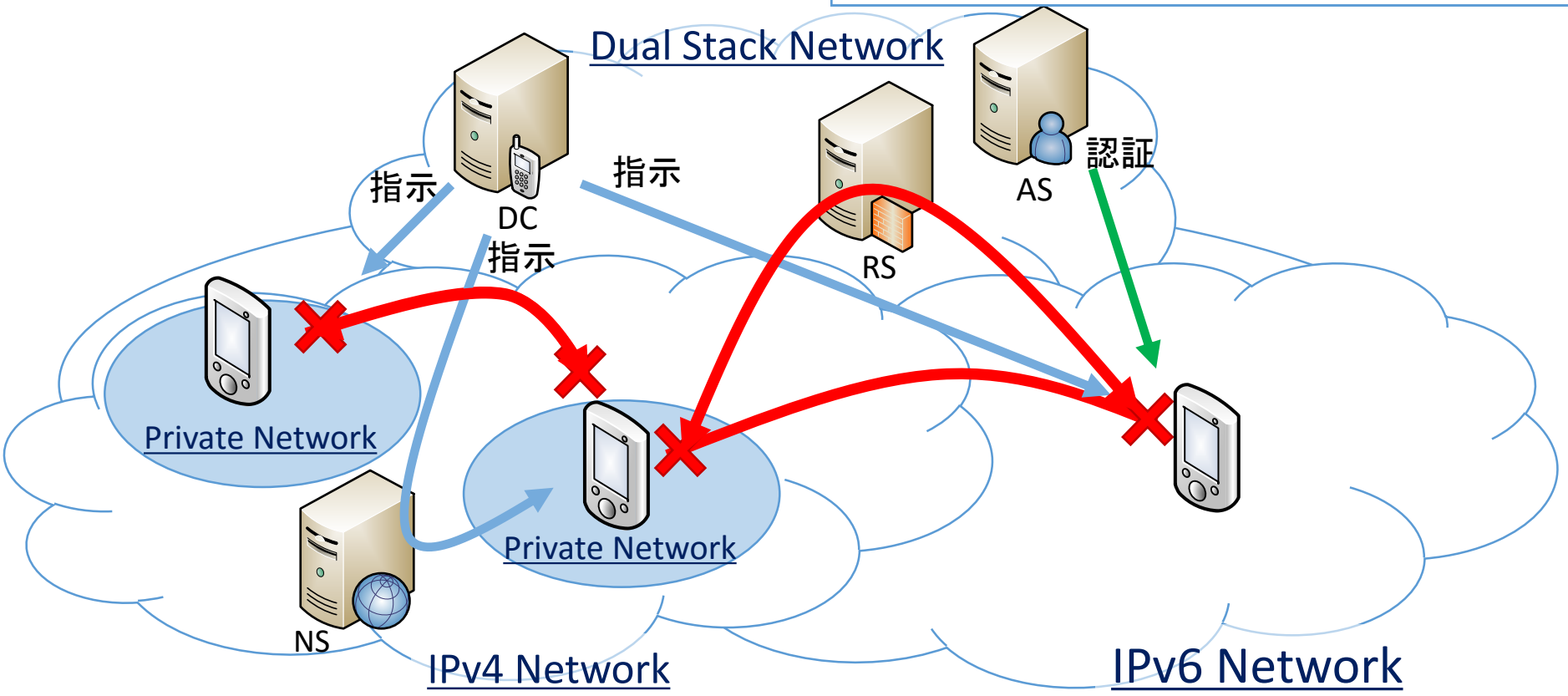
- 公開鍵の利用、OpenIDの利用

1つの統合的な枠組みとして再設計

NTMobileの概要

アプリケーションは実IPアドレスで通信

実際の通信は実IPを用いUDPでカプセル化



DC:Direction Coordinator　RS:Relay Server　NS:Notification Server　AS:Account Server

NTMobileの動作方式・実装

実装方式

- Linuxカーネル実装型
  - LinuxOS対象
  - 高速
  - アプリケーションの改造不要
- フレームワーク実装型
  - Unix対象
  - アプリケーションの改造要
- VPNサービス利用型
  - Android対象
  - アプリケーションの改造不要

NAT越え方式

- UDPキープアライブ
  - 20秒毎
  - PC対象
- TCPキープアライブ
  - NSが必要
  - 1時間毎
  - LinuxBox等対象
- プッシュ通知サービス
  - GCM/APNS利用
  - キープアライブ不要
  - スマートフォン対象

トンネル構築

- 一方の端末がNAT下
  - NAT配下端末からトンネル構築要求パケット送信
- 両方の端末がNAT下またはIPv4-IPv6間
  - 両方の端末からRSにトンネル構築要求パケット送信
  - 両端末宛てにトンネル構築応答を返送

APNS:Apple Push Notification Service　GCM:Google Cloud Messaging

実用化に向けた要求仕様

公開鍵認証

- グループ認証
- パスワード管理の代替(LinuxBox等)
- アプリケーションパケットのより強固なセキュリティ

OpenIDの利用

- ユーザー認証をNTMobileから切り離す
  - ユーザー管理のコスト削減
  - 多彩な認証方法の提供(導入の敷居を下げる)

実装方式・動作仕様の統一

- 実装方式に係らず相互に互換性がある
- 動作仕様を可能な限り統一する

統合的枠組の設計

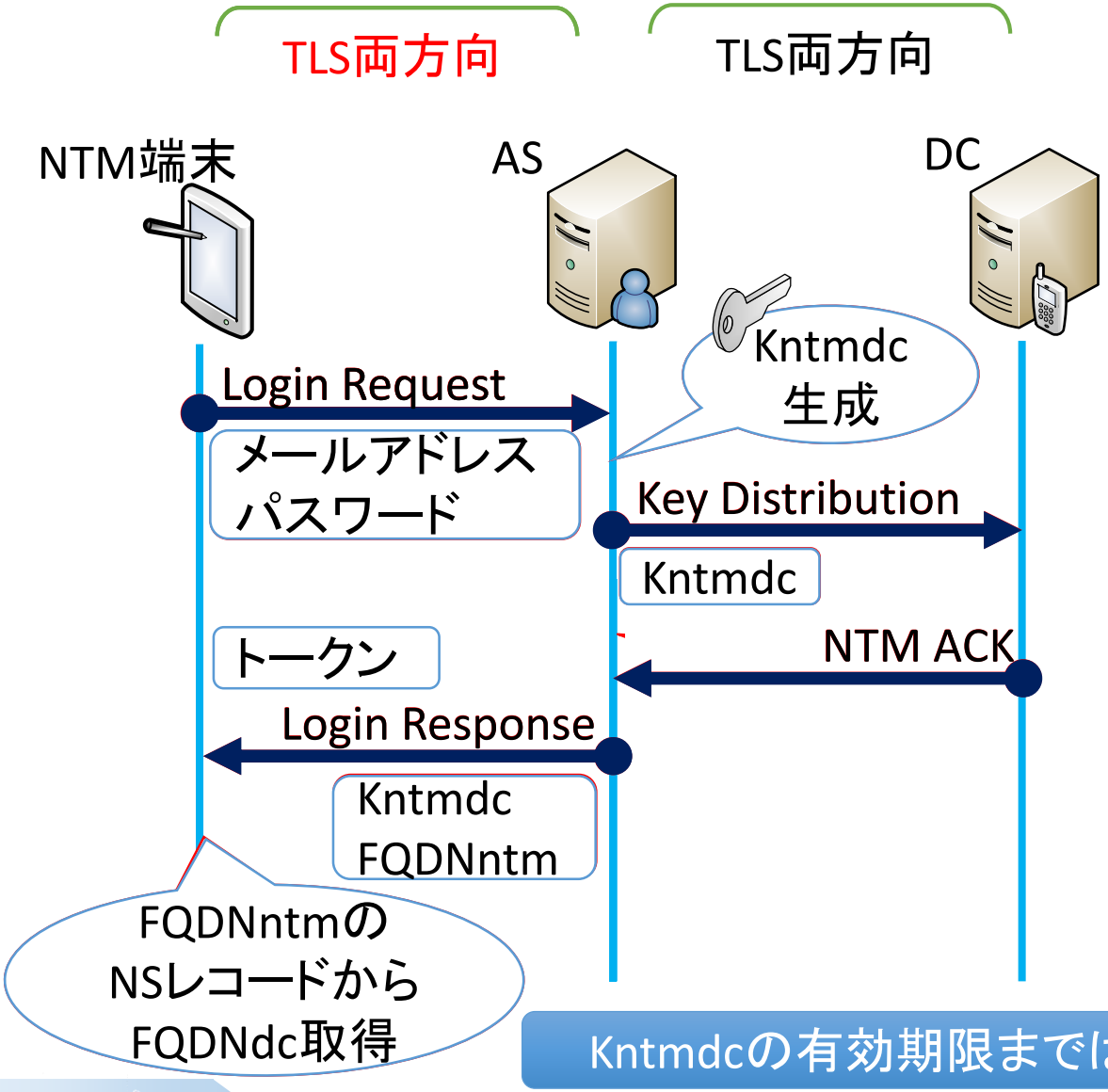
動作シーケンスの見直し

- OpenIDに対応できること
- 公開鍵認証に対応できること(従来方式と混在可)
- 複数のNAT越え方式が混在できること
- 可能な限りシーケンスを統一すること

実装の見直し

- 複数の実装方式で統一できる部分はコードを共有すること

動作シーケンス(端末認証)



ASへログイン要求

TLS両方向通信

NTM端末-DC間の鍵生成

メールアドレス・パスワード認証

DCへ鍵配送

NTM端末-DC間の通信で利用

確認応答

NTM端末へログイン応答

FQDNntmとKntmdcを通知

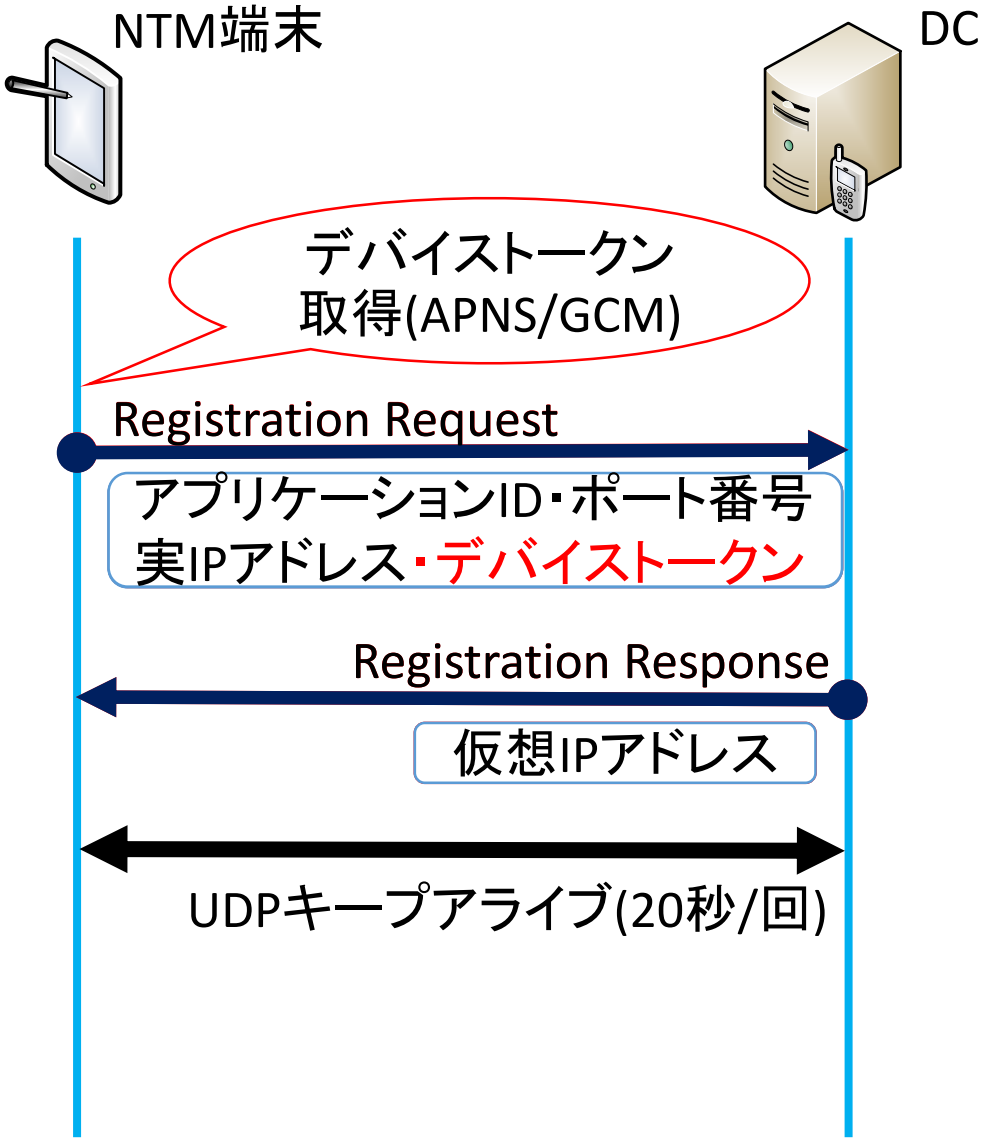
DCのFQDN取得

FQDNntmのNSレコードから取得

Kntmdcの有効期限までは再認証不要

OP:OpenID Provider

動作シーケンス(情報登録)



情報登録要求

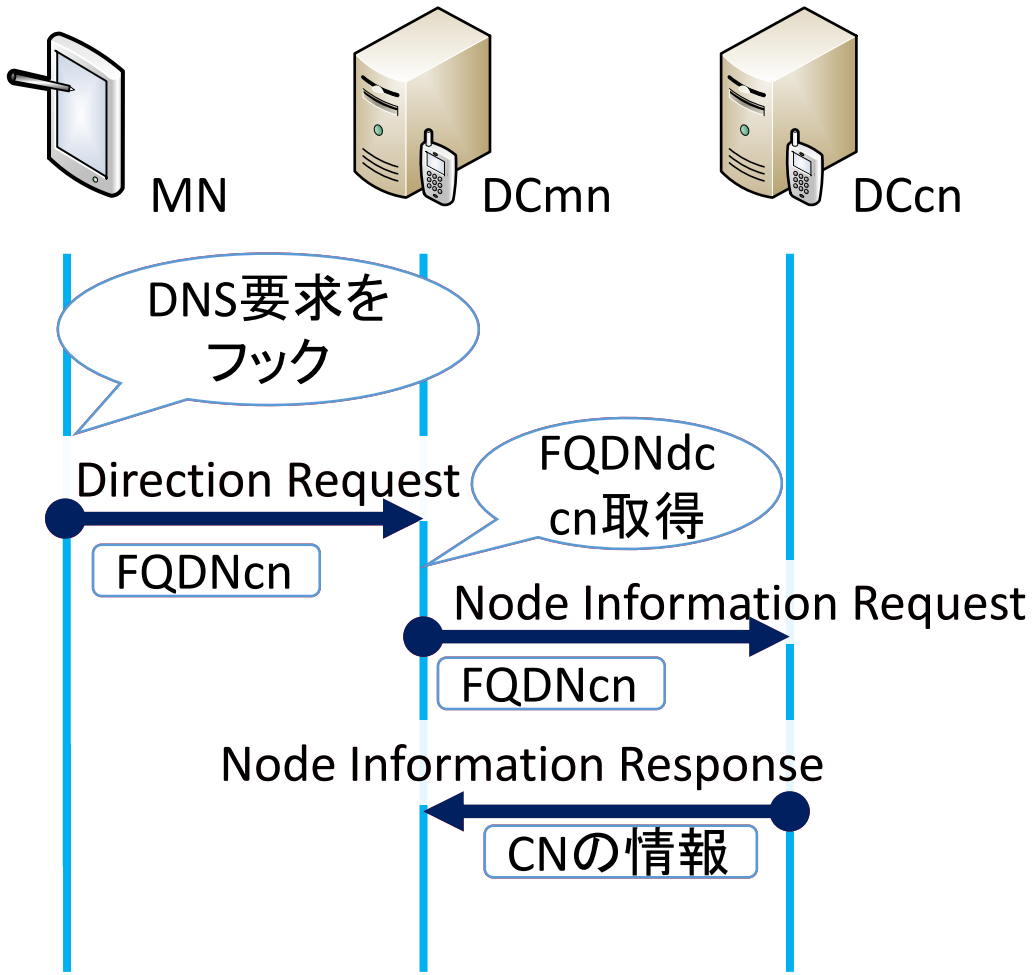
実IPアドレス・NATのIPアドレス
アプリケーションID・ポート番号
を取得

情報登録応答

仮想IPアドレスを通知

端末起動時と移動時に実行

動作シーケンス(名前解決)



FQDNcnの名前解決をフック

DNS要求をフック

DCmnへ経路指示要求

トンネル構築の指示要求

FQDNcnからFQDNdccn取得

FQDNcnのNSレコードを探索

DCcnへCNの端末情報要求

CNの情報を要求

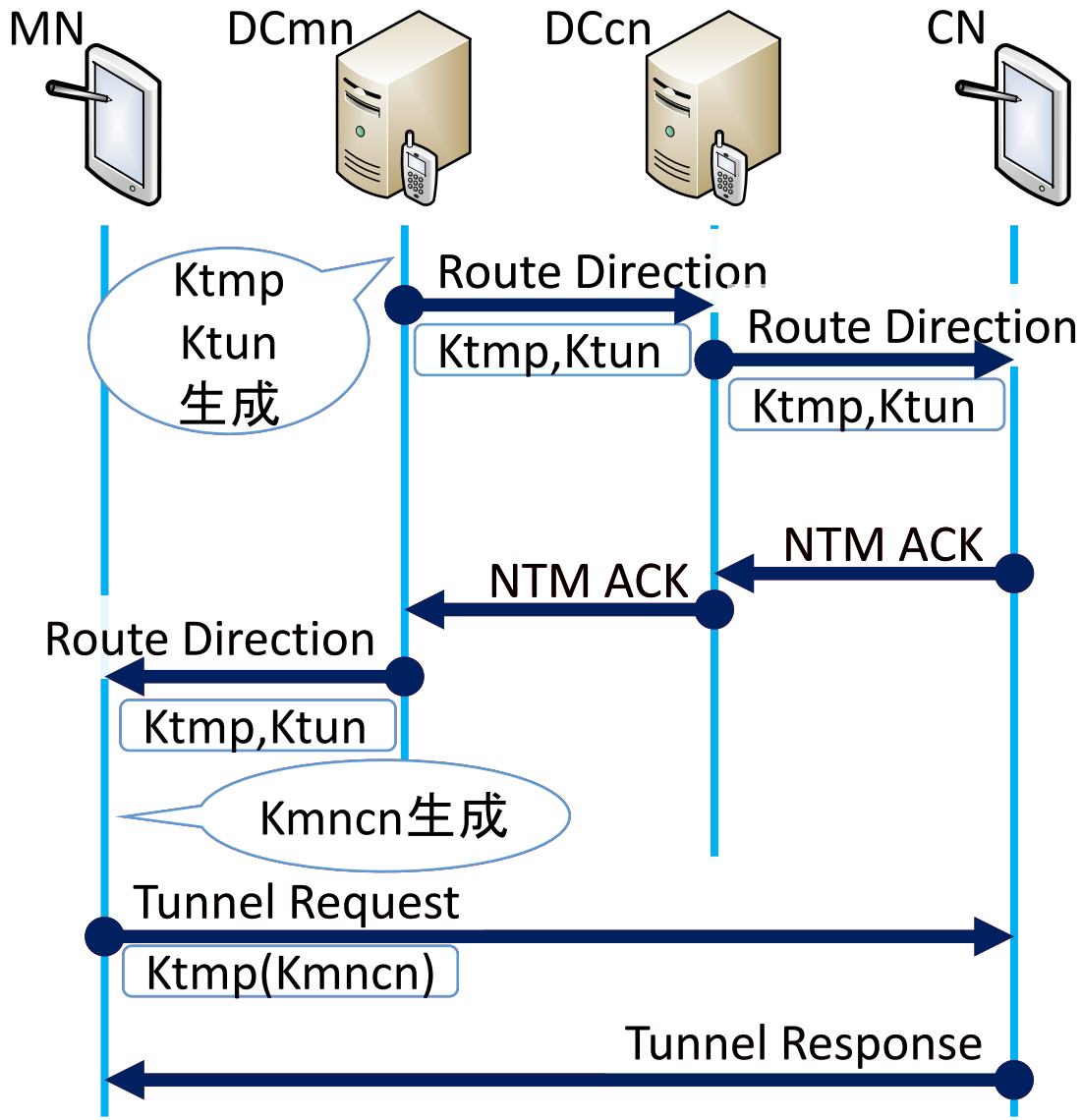
DCmnへCNの端末情報応答

CNの仮想IPアドレス等を応答

**CNの情報が集まったので
トンネル構築指示へ**

MN:Mobile Node　CN:Correspondent Node

動作シーケンス(経路生成)(直接通信)



経路生成用の共通鍵生成

トンネル要求
Kmncnの各暗号化用

CNへ経路指示

DCcnを経由
(DCcnはCNと通信可能)

確認応答

DCcnを経由
(CNの公開鍵を格納)

MNへ経路指示

通信用の共通鍵生成

KmncnはKtmpで暗号化

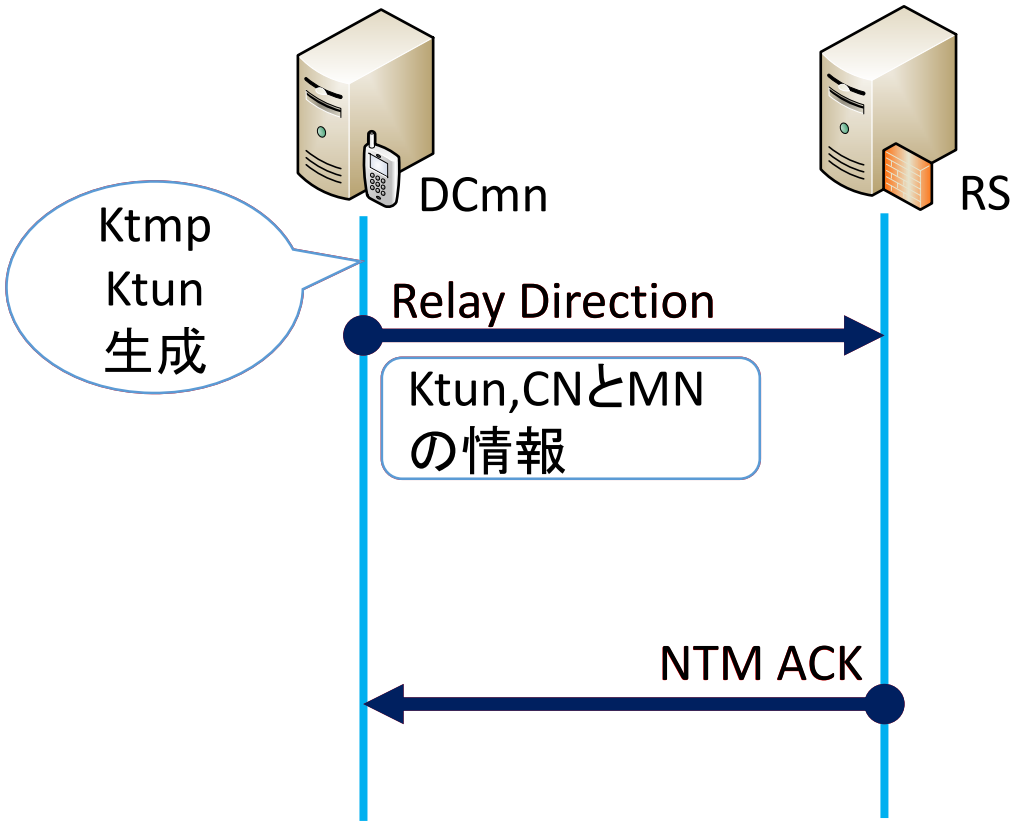
トンネル要求

メッセージはKtunで暗号化

トンネル応答

Kmncnで暗号化
(鍵の共有確認)

動作シーケンス(経路生成)(viaRS)(1)



経路生成用の共通鍵生成

Ktun: TunnelRequest暗号化用
Ktmp:通信に用いる共通鍵(Kmncn)
の暗号化用

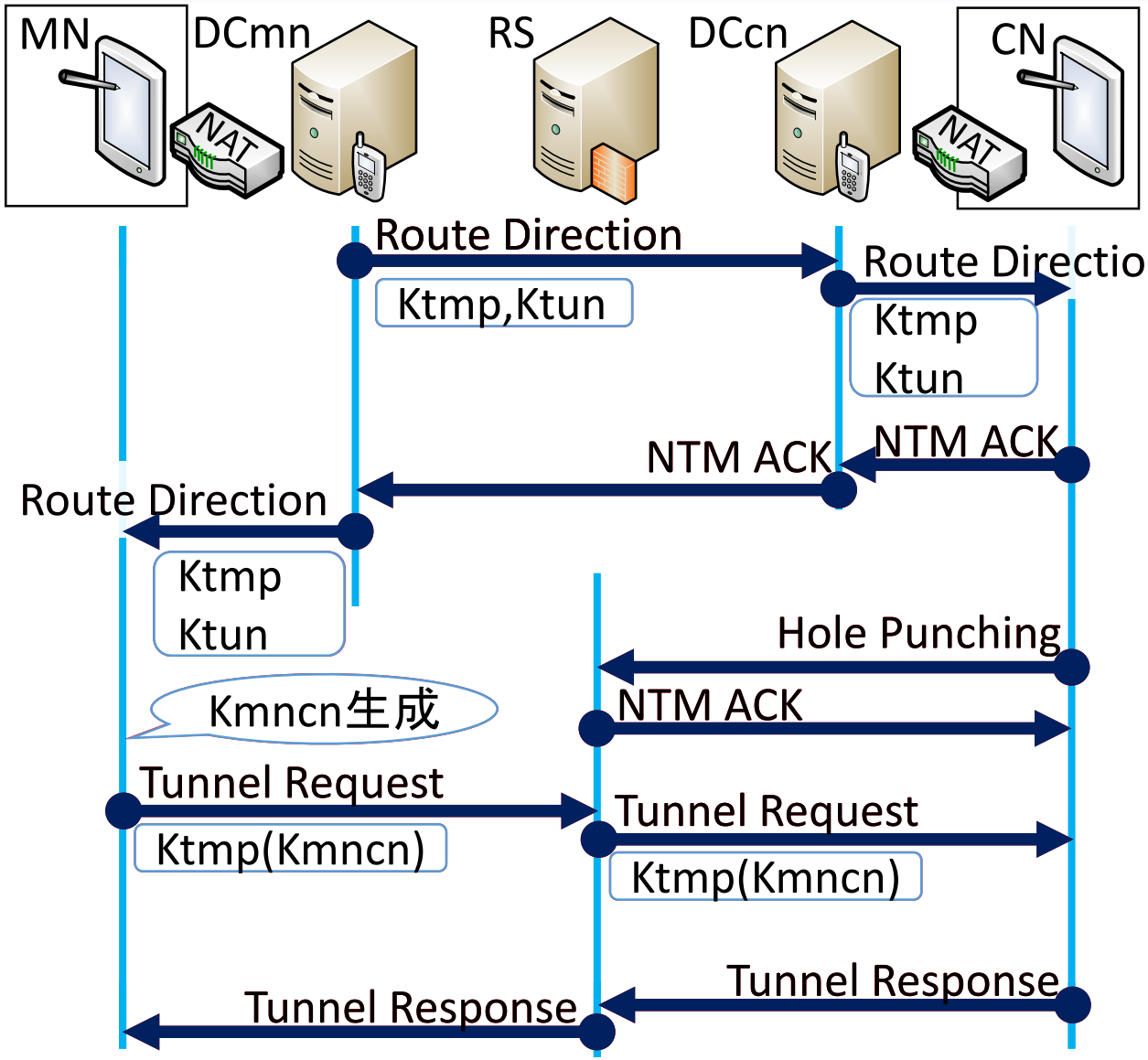
RSへ中継指示

・Ktun（TunnelRequest暗号化用)
のみ配布
・Ktmpは配布しない
→TunnelRequest格納の
Kmncnを復号できない

確認応答

中継指示が完了したので
経路生成指示へ

動作シーケンス(経路生成)(viaRS)(2)



CNへ経路指示

Ktmp,Ktunを配送

確認応答

CNの公開鍵を配送

MNへ経路指示

Ktmp,Ktunを配送

UDPホールパンチング

RSからの通信路を確保

通信用の共通鍵生成

KmncnはKtmpで暗号化
暗号化

トンネル要求

メッセージはKtunで暗号化

トンネル応答

Kmncnで暗号化

装置間の信頼関係について(1)

NTM端末間

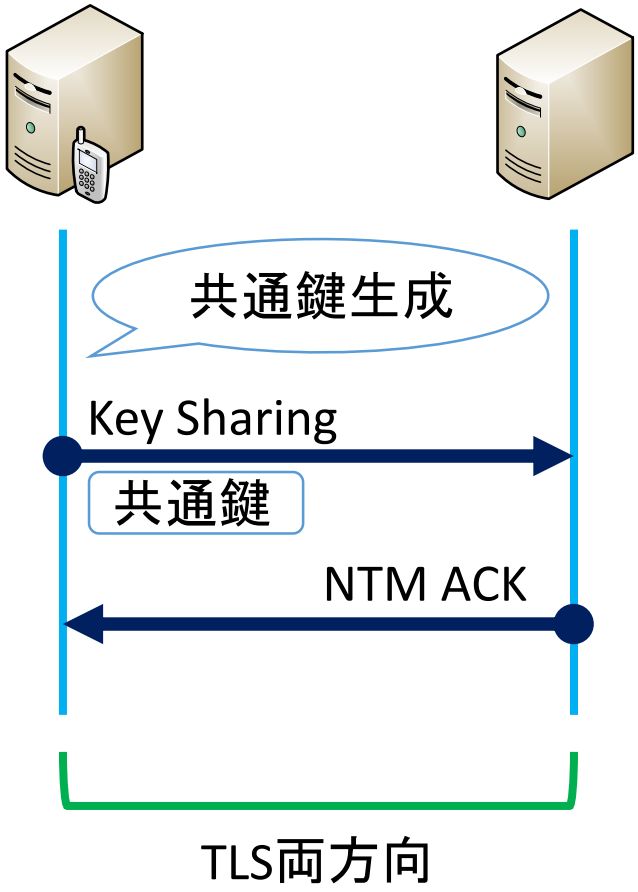
- 共通鍵(NTM端末間で交換)
- 通信ごとに共通鍵生成

NTM端末-サーバー(DC含む)間

- AS:TLS片方向認証
- その他:共通鍵(ASまたはDCから配布)

サーバー(DC含む)間

- AS-DC間:TLS両方向
- その他:初回通信時に共通鍵交換



装置間の信頼関係について(2)

通信時のセキュリティ

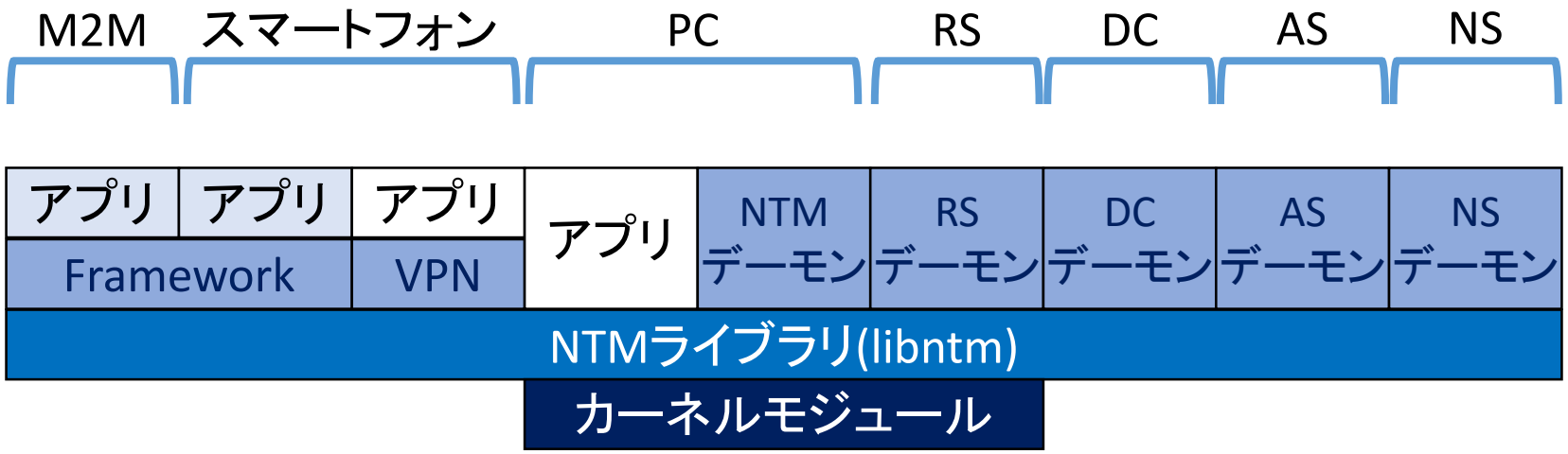
- 通信ごとに共通鍵を3つ生成
  - 通信暗号化鍵(NTM端末で生成)
  - 通信暗号化鍵を暗号化する鍵
  - 暗号化された通信暗号化鍵を送信するメッセージの暗号化鍵
- 例え管理者であっても通信に用いる共通鍵を知り得ない
  通信暗号化鍵はサーバー管理者の管理外
- NSなどのサーバー間との通信を要す場合は専用の共通鍵を使用

DCを含むサーバー間のセキュリティ

- 予めTLS相互認証で共通鍵を共有
  →安全な共通鍵交換
- 通信ごとにTLSシーケンスを行う必要無し
  →高速化

NTMobileの実装

- 2つ以上のデーモンプログラム等に共通する部分はライブラリに実装
  関数の命名規則の確認
  Doxygenによるドキュメント化対応の確認
- デーモンプログラムは基本的にライブラリを使う
  可能な限りライブラリ側でプラットフォーム間の互換性を取る
  デーモンプログラム等は、実装特有のものを除きシステムAPIを利用しない



M2M:Machine to Machine

要求仕様に対する考察

公開鍵認証

- メッセージに公開鍵を格納することで実現可能

OpenIDの利用

- OpenID認証を終えてからNTMobile処理を開始することで実現可能

実装方式・動作仕様の統一

- 実装方式を問わず同一のシーケンスを設計
  実装はNTMobileライブラリによって確実に統一される
- NS等のオプション装置が追加されても基本的なシーケンスは同じ
  追加処理が入るのみ

まとめ

- NTMobileの実装・動作方式に依存しない枠組みを設計
  動作シーケンスの統一
  共通鍵の整理(鍵の配布方法含む)
  共通する部分をライブラリとして実装
  OpenIDに対応できる拡張性の確保
  公開鍵に対応できる拡張性の確保

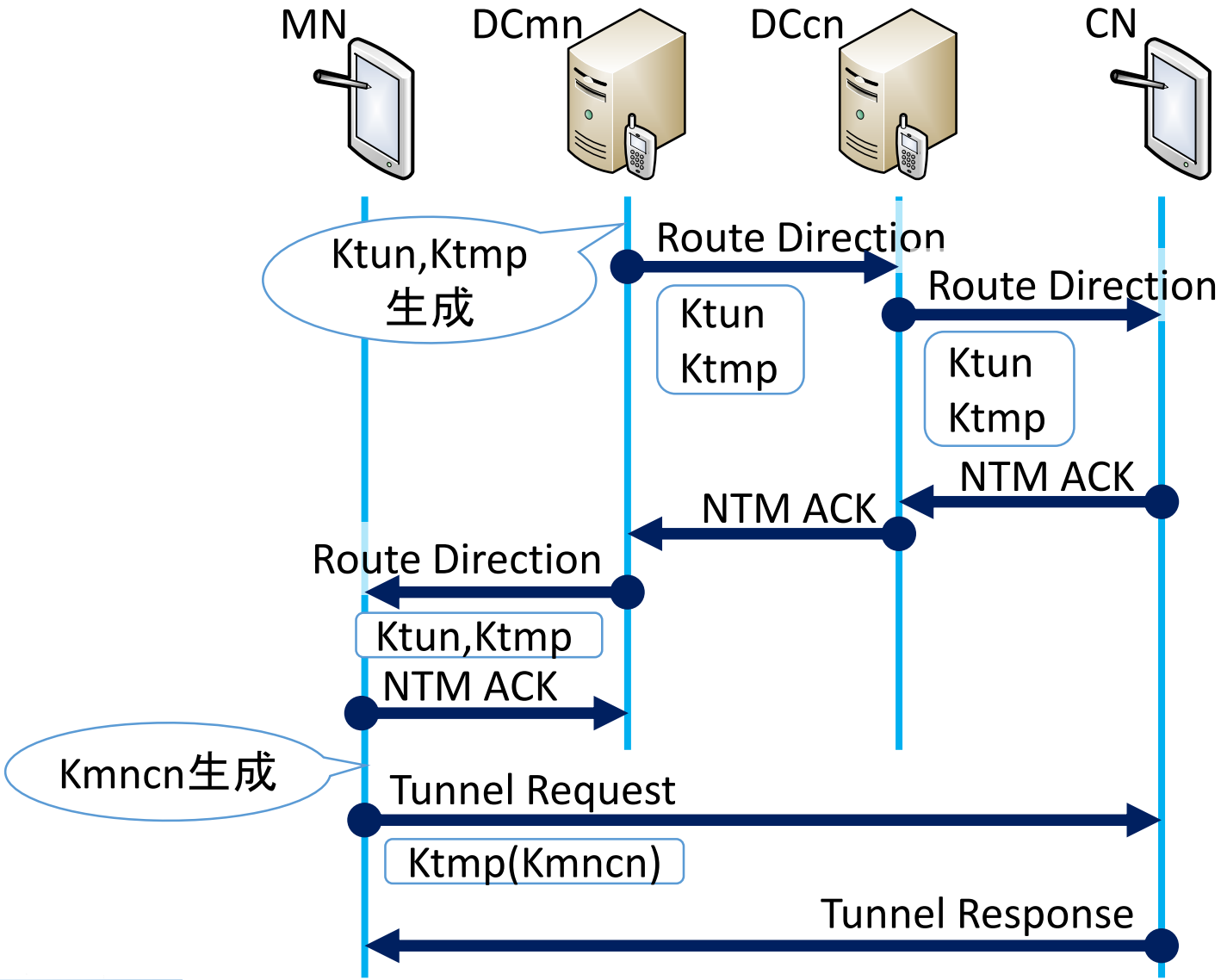
- 今後の予定
  実装(64bit可含む)および動作確認・評価

NTMobileの共通鍵

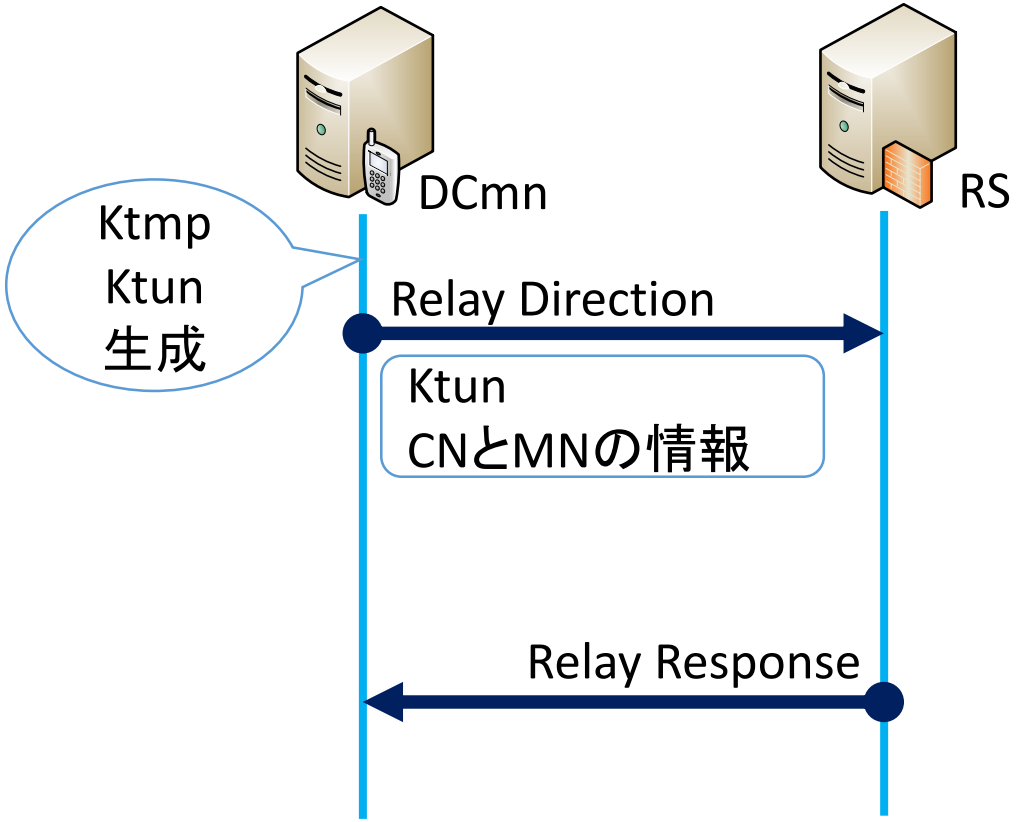
共通鍵名	生成装置	配布先	用途
Kntmdc	AS	NTM端末,DC	NTM端末とDCとの間の通信暗号化
Kdcns	DC	DC,NS	DCとNSの間の通信暗号化
Kntmns	DC	NTM端末,NS	NTM端末とNSの間の通信暗号化
Kdcdc	DCmn	DCmn,DCcn	DC間の通信暗号化
Kdcrs	DCmn	DCmn,RS	DCとRSの間の通信暗号化
Ktun	DCmn	MN,CN,RS	TunnelRequestの暗号化 RS宛てのHolePunching/ACKの暗号化
Ktmp	DCmn	MN,CN	Kmncnの暗号化(公開鍵の時は未使用)
Kmncn	MN/CN	MN,CN	MNとCNの間の通信暗号化

上5つ(網掛け部分)は、1度共通鍵共有をした場合、共通鍵の有効期限までは再共有不要
下3つは、通信ごとに生成

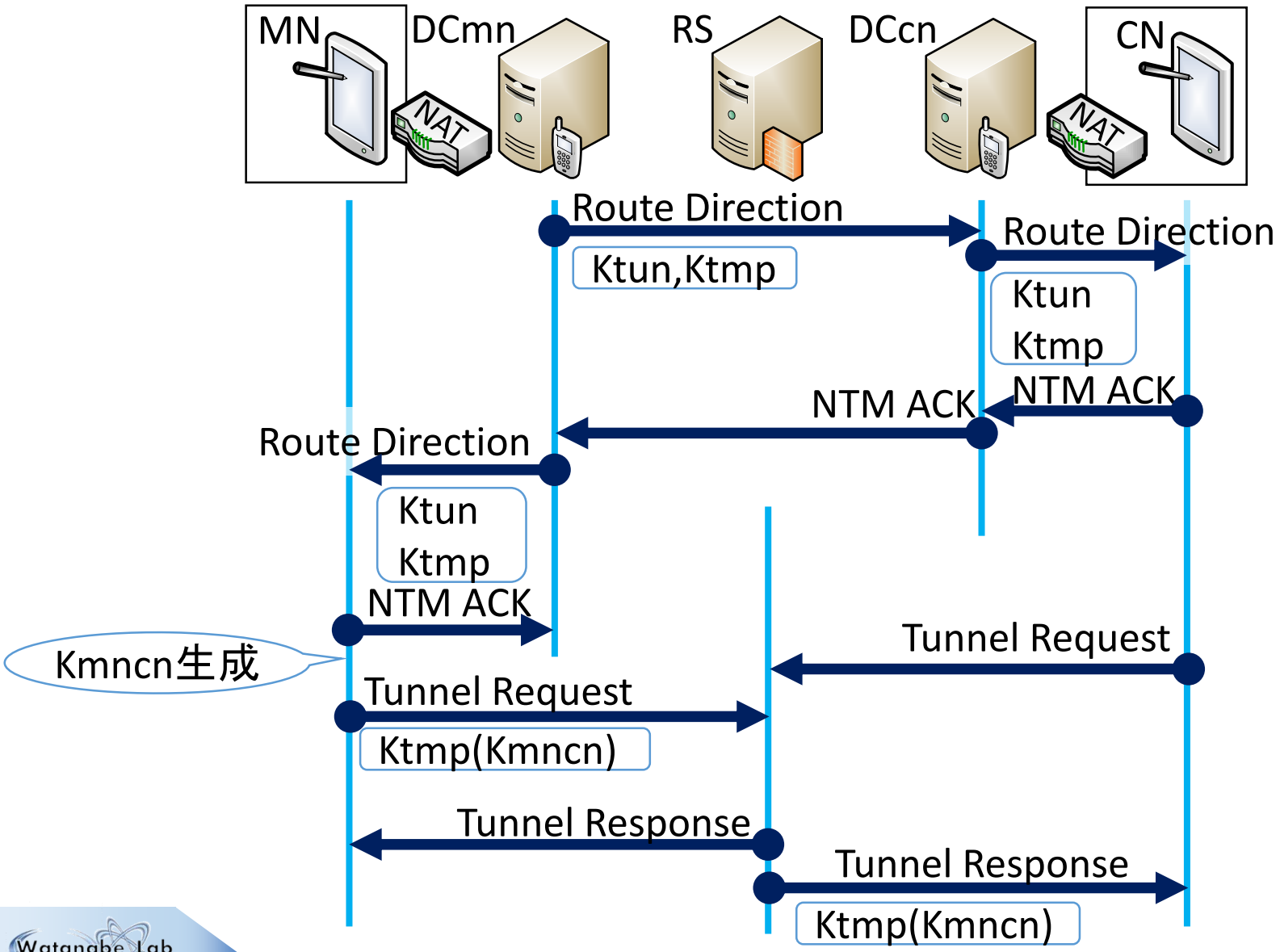
従来のシーケンス(直接通信)



従来のシーケンス(RS経由)(1)



従来のシーケンス(RS経由)(2)



プッシュ通知サービス(1)

- スマートフォンのアプリケーションを呼び出すための仕組み
  Google(Android向け)、Apple(iOS向け)が提供

スマートフォンアプリケーションは、APNS/GCMサーバーにデバイス登録(デバイスIDを取得)

以後、サーバーとスマートフォンはセッションを維持

↓

アプリケーションサーバ(NTMobileではDC)は、APNS/GCMサーバーにデバイスIDを用いてメッセージを送信

↓

APNS/GCMサーバーは、維持しているセッションにメッセージを流す。

スマートフォンアプリケーションが立ち上がってメッセージを受信する。

プッシュ通知サービス(2)