NTMobileにおけるDNS実装の変更が不要なデータベース型端末情報管理手法の検討

細尾 幸宏[1,a)]　鈴木 秀和[1]　内藤 克浩[2]　旭 健作[1]　渡邊 晃[1,b)]

概要：端末が接続するネットワーク環境にかかわらず通信開始を保証する通信接続性や，移動してネットワークが切り替わっても通信が継続できる移動透過性への需要が高まっている．我々は，IPv4/IPv6 ネットワークが混在する環境においてアドレス空間やアドレス体系に依存しない通信接続性と移動透過性を実現する NTMobile(Network Traversal with Mobility) を提案している．現状の NTMobile では端末情報の収集と管理を既存の DNS に依存する方法をとっており，通信開始時の情報収集における確実性や情報管理の柔軟性に欠けるという課題があった．そこで本稿では，情報管理を DNS の仕組みに依存する方法からデータベースを利用する方法へ移行し，情報収集方法も変更することにより課題を解決する手法を提案する．また，提案手法の実装を行い，動作検証を行ったので報告する．

キーワード：移動透過性，NAT 越え，DNS，データベース

Node Information Management Method by Database without Modifying DNS in NTMobile

Yukihiro Hosoo[1,a)]　Hidekazu Suzuki[1]　Katsuhiro Naito[2]　Kensaku Asahi[1]
Akira Watanabe[1,b)]

Abstract: There is a growing demand for connectivity that ensures the start of communication, regardless of the network environment, and mobility that can continue the communication even if the node switchs its network. We have been proposing Network Traversal with Mobility (NTMobile) that can provide connectivity and mobility in IPv4 and IPv6 networks. In current NTMobile, the collection and management of node information depends on the existing DNS mechanism, that leads to poor flexibility and reliability. In this paper, database mechanism is introduced on behalf of the DNS mechanism to improve the system. we have implemented the proposed method and verified the operation.

Keywords: Mobility, NAT Traversal, DNS, Database

1. はじめに

スマートフォンなどの高性能な携帯端末の普及に伴い，ユーザがインターネットを利用する形態が大きく変化している．TCP/IP の当初の想定を越えており，様々な課題が明確になっている．IPv4 グローバルアドレスの枯渇問題に対応するためプライベートアドレスを導入して IPv4 の延命をはかってきた．しかし，グローバルアドレス側からプライベートアドレス側に対して通信を開始できない NAT 越え問題が生じ，IPv4 における大きな制約となっている．出先からホームネットワークへアクセスできるネットワーク環境への要求が高まっている上に，今後はサービスプロバイダのネットワーク自体をプライベートアドレス空間で

[1] 名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University
[2] 三重大学大学院工学研究科
Graduate School of Engineering, Mie University
[a)] m0830033@ccalumni.meijo-u.ac.jp
[b)] wtnbakr@meijo-u.ac.jp

実現することが検討されており（Large Scale NAT），NAT 越え問題の解決は重要度を増している．長期的には IPv6 への移行は避けられないが，IPv4 と IPv6 は互換性が無く直接通信ができないため IPv6 の導入は進んでいない．そのため，IPv4 と IPv6 の混在環境における通信接続性を保証することは極めて重要である．

IP ネットワークでは通信端末のインタフェースに割り当てられた IP アドレスを端末の識別と位置情報の両方に使用している．端末が接続するネットワークの切り替えやインタフェースの切り替えを行うと IP アドレスが変化し，通信を継続できない．このような問題を解決する技術は移動透過性技術と呼ばれ，現在までに様々な移動透過性技術が提案されてきた [1]．しかし，これらの多くは IPv6 が今後の主流になることを想定した IPv6 のみに対応した方式であり，NAT 越え問題が存在する IPv4 における移動透過性技術は少ない．

IPv4 と IPv6 の混在環境において移動透過性を実現する技術として DSMIP(Mobile IPv6 Support for Dual Stack Hosts and Routers:RFC5555) が標準化されている [2]．DSMIP は Mobile IPv6[3] をベースにしており，Mobile IPv4[4] の課題をそのまま引き継いでいる．Mobile IPv6 では経路最適化など様々な検討が加えられているが，IPv4 においては NAT のような IPv4 特有の制約があるため経路最適化などの機能が未定義である．また，NAT が混在する環境でのネットワーク切り替えを行うためには端末のホームアドレスをグローバル IP アドレスとするか，NAT を改造する必要があり IPv4 が主体の現在のネットワークにおいては課題が多い．

著者らは，IPv4/IPv6 ネットワークが混在する環境において通信接続性の確保と移動透過性を同時に実現する NTMobile(Network Traversal with Mobility) を提案している．NTMobile は NAT 越え技術を兼ね備えており，NAT 配下の NTMobile 対応端末（以後 NTM 端末）に対する接続性を確保できる．また，IPv4 と IPv6 ネットワーク間の接続性を確保することができる．さらに，通信中にどのようなネットワークへ接続を切り替えても通信を継続できる．NTMobile では，NTM 端末上のアプリケーションが仮想 IP アドレスに基づいた通信を行う．実際の通信は通信端末が接続しているネットワークに応じて構築した最適なトンネル経路を用いて転送する．トンネル通信は可能な限りエンドツーエンドで構築するため，冗長な経路になりにくい．この方法により実ネットワークの違いや，実 IP アドレスの変化を隠蔽する．

NTMobile では，トンネル構築に必要な NTM 端末の端末情報を既存の DNS の仕組みに従って独自に定義した NTMobile 専用レコード（以後 NTM レコード）として登録している．DNS レコード形式で扱うことにより，FQDN(Fully Qualified Domain Name) から NTM 端末が

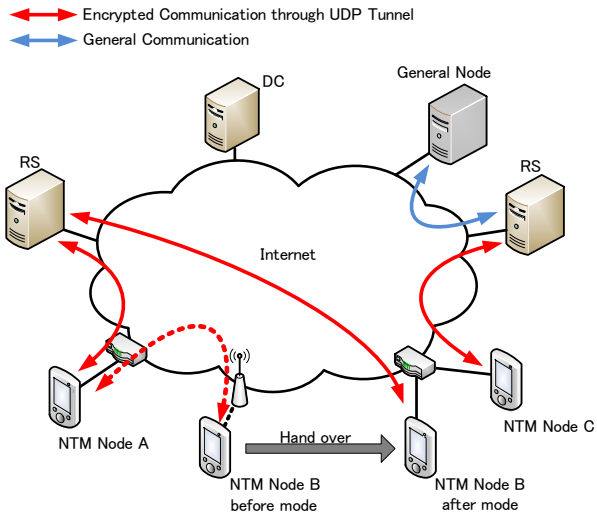


図 1 NTMobile の概要
Fig. 1 Overview of NTMobile.

接続しているネットワークのプライマリ DNS サーバを経由して NTM 端末情報を取得することができる．通信を開始する NTM 端末は通信相手の FQDN から通信相手の端末情報を収集し，トンネル構築を行う．ここで，接続しているネットワークのアドレス体系が IPv4 と IPv6 のように異なる場合でも通信接続性を確保するには NTM 端末が DNS 問い合わせにより A，AAAA，および NTM レコードの 3 種全てを問い合わせる必要があり，通信開始時のオーバヘッドが大きくなるという課題があった．また，DNS のキャッシュによって最新の情報を取得できない場合があるという課題があった．

そこで，本論文ではトンネル構築に必要な端末情報の収集を DNS の仕組みに依存しない方法に変更する．具体的には DNS レコード形式の端末情報管理をデータベースに移行し，DNS のキャッシュの影響を受けない独自の端末情報収集方法とすることを提案する．また，提案方式を実装し，動作検証を行ったので報告する．

以下，2 章で NTMobile について概説し，3 章で提案方式について述べ，4 章で動作検証結果を述べ，5 章でまとめる．

2. NTMobile

2.1 概要

NTMobile の概要を図 1 に示す．NTMobile は NTM 端末，DC(Direction Coordinator)，RS(Relay Server) によって構成される．DC や RS はグローバルネットワークに設置し，ネットワークの規模に応じて複数台設置による負荷分散を行うことができる．DC は DDNS(Dynamic DNS) の機能を包含しており，NTM 端末の A レコードと AAAA レコードに加え NTM レコードを登録している．これらのレコード情報は通信接続性の確保のために最新の状態を

保つよう動的に更新を行う．DC は DNS サーバとして管理するドメインを元に NTM 端末に一意な FQDN を割り当てる．この FQDN と各レコード情報を関連付けて管理することにより，FQDN から DNS 問い合わせを利用して NTM 端末情報への到達性を確保している．これにより，分散管理された異なる DC に所属する NTM 端末同士は FQDN を元に互いの NTM 端末情報を共有でき，通信接続性の確保に利用している．NTM レコードには NTM 端末を一意に識別する Node ID，実 IP アドレス，仮想 IP アドレス，NAT の外側の実 IP アドレス，アドレス情報を管理している DC の実 IP アドレスが記載されている．DC は NTM 端末のアドレス管理のほかに暗号鍵の生成，配布を行う．また，NTM 端末の通信開始要求を受けて通信トンネル構築時に適切なトンネル経路を判断し，指示を行う役割を担っている．DC が各 NTM 端末に割り当てる仮想 IP アドレスは一意なアドレスであり，各 DC は自身に割り当てられたアドレス空間から重複が起きないように割り当てを行う [5].

NTM 端末は実ネットワークから割り当てられる実 IP アドレスと，DC から割り当てられる仮想 IP アドレスの 2 種類の IP アドレスを保持する．NTM 端末上で動作するアプリケーションは仮想 IP アドレスに基づいた通信を行う．仮想 IP アドレスに基づくパケットは，NTM 端末間に構築される UDP トンネルによって転送される．トンネルの経路は通信を行う NTM 端末のどちらか一方がグローバルネットワークに接続している場合には必ずエンドツーエンドのトンネル通信を行うことができる．

RS は通信を行う NTM 端末が互いに異なる NAT 配下に接続している場合や NTMobile 未実装の一般端末と通信を行う場合，さらには一方が IPv4 もう一方が IPv6 ネットワークに接続している場合，通信の中継を行う装置である．ただし，NTM 端末同士が互いに異なる NAT 配下に接続していても NAT の種類によってはエンドツーエンドのトンネル通信に切り替えることが可能である [6].

RS を IPv4/IPv6 デュアルスタックネットワークに設置することにより，IPv4 アドレスを使用する NTM 端末と IPv6 アドレスを使用する NTM 端末間の通信を実現することができる [7]．アプリケーション間では，IPv4 または IPv6 の仮想 IP アドレスに基づいた通信が行われるため，端末が接続しているネットワークや使用している IP アドレスの体系の違いに影響されることはない．このため，通信中に接続しているネットワークが IPv4 と IPv6 の間で切り替わることがあっても通信を継続することができる．

NTMobile では，NTM 端末と DC，DC 同士，DC と RS 間にはそれぞれ事前に信頼関係があることを前提とする．信頼関係を確認後, DC は NTM 端末に対して FQDN, Node ID，仮想 IP アドレスを配布する．NTMobile で使用される制御メッセージは各端末間で共有している暗号鍵を用いて暗号化される．

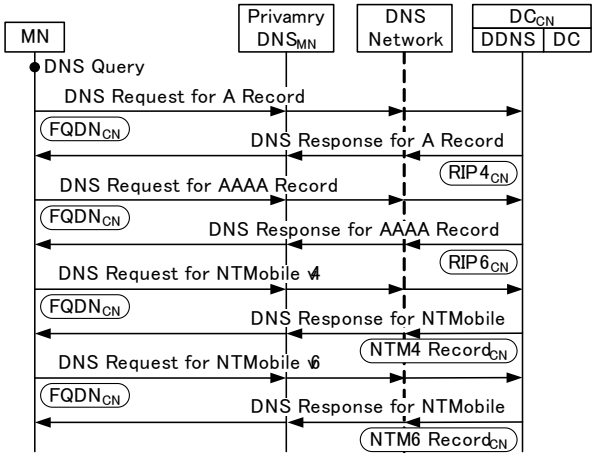


図 2 名前解決処理
Fig. 2 Name resolution process.

2.2 通信シーケンス

以後の説明では，通信開始側の NTM 端末を MN(Mobile Node)，通信相手側の NTM 端末を CN(Correspondent Node) として説明する．また，端末 N の FQDN を $FQDN_N$, Node ID を NID_N, 実IPv4アドレスを $RIP4_N$，仮想 IPv4 アドレスを $VIP4_N$，端末 N が NAT 配下に接続している場合の NAT の実 IPv4 アドレスを $RIP4_{NAT_N}$ とし，アドレス情報を管理している DC を DC_N，その実 IPv4 アドレスを $RIP4_{DC_N}$ とする．NTM レコードは IPv4 用の NTM4 レコードと IPv6 用の NTM6 レコードの 2 つが定義されている．端末 N の NTM4 レコードには，NID_N，$RIP4_N$，$VIP4_N$，$RIP4_{NAT_N}$，$RIP4_{DC_N}$ が含まれ，NTM6 レコードにはそれぞれの IPv6 アドレスが含まれる．N1 と N2 がトンネル通信時に用いる Path ID を PID_{N1-N2} と表す．Path ID は NTM 端末間の通信を一意に識別するための識別子である．

2.2.1 登録処理

NTM 端末は通信接続性の確保のために，端末起動時に DC に対して実 IP アドレスなどの端末情報を登録する．MN は $FQDN_{MN}$，NID_{MN}，$RIP4_{MN}$，$RIP6_{MN}$ などを記載した NTM Registration Request を DC_{MN} に送信する．DC_{MN} は NTM Registration Request を受信すると，DNS サーバに登録されている NTM レコードを更新する．このとき，NTM Registration Request の IP ヘッダに格納されている送信元 IP アドレスが，メッセージ内の RIP と異なる場合，MN がプライベートアドレス空間にいると判断し，IP ヘッダの送信元 IP アドレスを NAT の外側の IP アドレスとして NTM レコードを更新する．その後，MN へ応答として NTM Registration Response を返す．

登録完了後は MN と DC_{MN} は定期的にメッセージの交換をすることで制御メッセージ用の通信経路を確保する

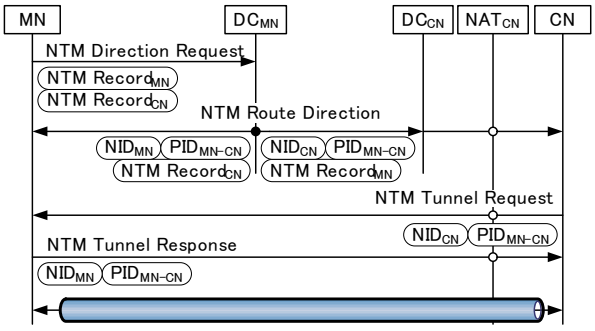


図 3 NTM 端末間のトンネル構築手順
Fig. 3 Tunnel establishment procedure between NTM nodes.

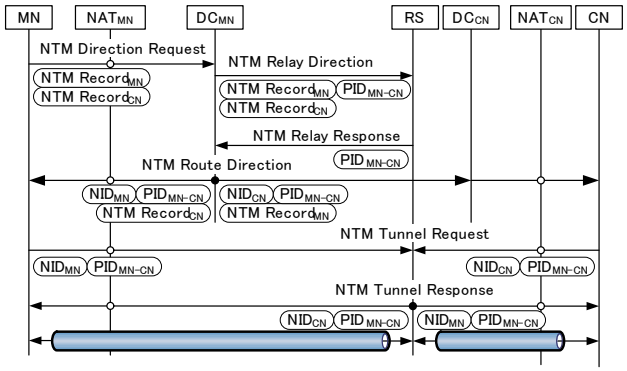


図 4 RS 経由のトンネル構築手順
Fig. 4 Tunnel establishment procedure via RS.

(Keep Alive)．これにより，DC_{MN} は NAT 配下にいる MN との制御メッセージ用経路を維持する．

2.2.2 名前解決処理

図 2 に名前解決処理手順を示す．NTMobile は，DNS による名前解決をトリガとして NTMobile 特有のネゴシエーションを実行する [8]．MN は DNS による名前解決処理を検出すると，そこに含まれる $FQDN_{CN}$ を元に A レコード, AAAA レコード, NTM4 レコードと NTM6 レコードの問い合わせを行い情報を収集する．NTM 端末の各 DNS レコードを登録している DC は DDNS の機能を包含し DNS サーバとして運用するため，DNS の探索により $FQDN_{CN}$ のドメインを管理する DC に到達し，各レコードを取得することができる．A レコードおよび AAAA レコードにより CN の実 IP アドレスを取得する．NTM4 レコードにより IPv4 上のトンネル構築を行うための情報を取得し，NTM6 レコードにより IPv6 上のトンネル構築を行うための情報を取得する．DC_{MN} は，MN が収集した情報を元に最適なトンネル経路判断を行うため，CN に関する全てのレコードを問い合わせる．このとき，NTM レコードを取得できなかった場合には通信相手が NTMobile 未実装端末であると判断する．DNS サーバからの応答は一時的に待避し，取得した情報を元に 2.2.3 項で説明するトンネル構築処理を行う．

トンネル構築後，MN は待避していた DNS サーバからの応答に含まれる RIP を VIP に書き換えて DNS リゾルバへ渡す．これにより，MN のアプリケーションは通信相手の IP アドレスとして VIP を認識することになる．

2.2.3 トンネル構築

図 3 にトンネル構築シーケンスを示す．NTM 端末は名前解決による情報収集完了後，DC にトンネル構築の指示を要求する．MN は 2.2.2 項で収集した CN の端末情報と自身の端末情報を DC_{MN} へ NTM Direction Request として送信する．DC_{MN} はメッセージ内の両端末の情報から最適なトンネル経路を判断する．DC_{MN} は経路判断を元にトンネル構築に必要な情報を載せた NTM Route Direction を MN と CN へ送信する．図 3 では CN が NAT 配下であるため，CN 宛の NTM Route Direction は DC_{CN} を中継させる．

NTM Route Direction を受け取った NTM 端末はメッセージに含まれる通信相手の情報をカーネル空間のトンネルテーブルに保持する．また，NTM Route Direction の指示に従い，NTM Tunnel Request/NTM Tunnel Reqsponse を交換することによりトンネルを構築する．

両端末が直接通信できない場合，図 4 に示すように DC_{MN} は RS に対して MN と CN の通信を中継するよう指示し，MN と CN には NTM Route Direction により RS との間にトンネルを構築するように指示する．通信を行う両端末が IPv4 と IPv6 のようにアドレス体系が異なるネットワークに接続している場合でも RS がトンネル通信を中継することにより接続性を確保することができる．

2.2.4 トンネル通信

アプリケーションは通信相手として仮想 IP アドレスを認識しているため，アプリケーションが生成したパケットの宛先 IP アドレスには仮想 IP アドレスが記載される．MN は端末内の IP 層において宛先のアドレスになっている $VIP4_{CN}$ をキーにトンネルテーブルを検索し，該当エントリに従ってカプセル化および暗号化を行う．カプセル化の際には IP ヘッダ，UDP ヘッダと NTM ヘッダが付加される．CN はカプセル化されたパケットを受信すると，IP 層において NTM ヘッダに記載されている PID_{MN-CN} をキーにトンネルテーブルを検索し，該当エントリに従ってデカプセル化および復号処理を行う．その後，抽出したアプリケーションパケットを上位アプリケーションへ渡す．

2.2.5 ハンドオーバ時の動作

NTMobile では，通信中に NTM 端末がネットワークを切り替えた場合，通信開始時と同じトンネル構築処理を行うことによりトンネルの再構築を行う [9]．このとき，MN は通信開始時に CN のレコード情報は取得済みであるため，2.2.2 項の名前解決処理は省略される．MN はこれと並行して 2.2.1 項で説明した登録処理を行い，DC に登録されている NTM レコードを最新の情報に更新する．

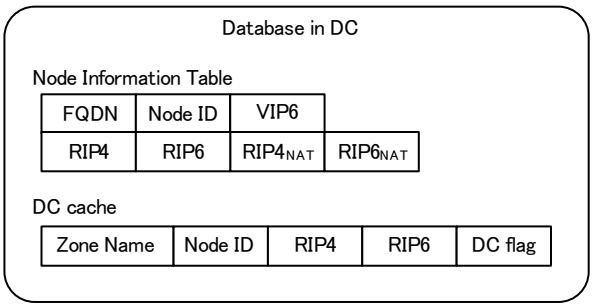


図 5 データベースの構成
Fig. 5 Database composition.

2.3 NTMobile の課題

現状の NTMobile では名前解決処理において以下のような課題がある．すなわち，通信開始時に NTM 端末が通信相手の FQDN から通信相手の A レコード，AAAA レコード，NTM4，NTM6 レコードと 4 回の DNS レコード問い合わせを行う．ここで，NTM 端末が携帯端末等であることを想定すると，通信遅延の大きい 3G 回線を使用している可能性が高く，接続時間が大きくなる可能性がある．また，NTM 端末が DNS による問い合わせを受けると，DNS サーバにキャッシュが残る場合がある．この場合，キャッシュが残っている期間は NTM 端末が移動していても移動する前の情報しか得ることができず，通信の接続性を確保することができない．このため，DNS レコードの TTL を短く設定する必要があるためキャッシュを利用できず，毎回 DNS 問い合わせによる探索を必要とする．さらに，DNS レコードのキャッシュについては実装によって TTL の扱いが異なる場合があり，想定を超えて長時間キャッシュが残る場合もあり，信頼性に欠ける．

この他にも，NTM レコードとして追加で取り扱っている端末情報が DNS サーバによって公開されているため，誰でも FQDN から NTM 端末情報を取得可能であり，セキュリティ上の問題がある．

3. 提案方式

3.1 提案方式の方針

2.3 項で示した課題を回避するため，NTM 端末から行っていた複数の DNS 問い合わせを DC_{MN} に依頼する方法に変更する．DC_{MN} は NS レコードを用いて DC_{CN} を発見し，DC_{CN} から直接情報を収集する．このために DC_{MN} と DC_{CN} 間で独自のメッセージを定義する．DC はグローバルネットワーク上に設置するため，NS レコードの問い合わせは比較的高速に行うことができる．この方法により，NTM 端末情報を DNS 問い合わせによって取得する必要がなくなるため，NTM 端末情報を DNS レコードではなくデータベースとして DC に保持させる．

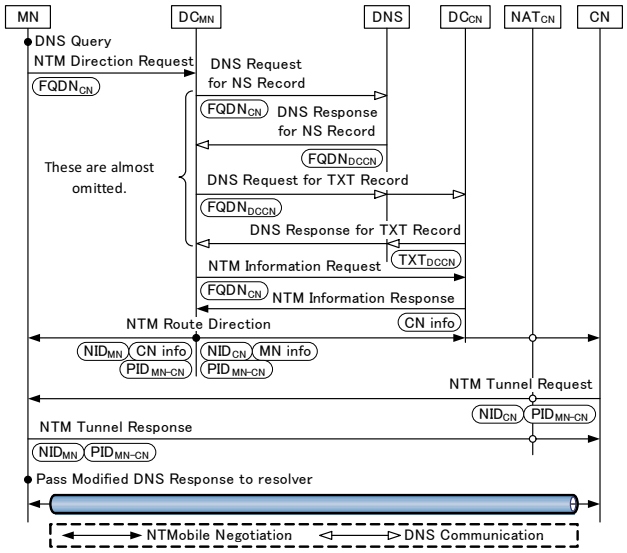


図 6 提案シーケンス
Fig. 6 Proposal sequence.

3.2 データベースによる情報管理

図 5 にデータベースの構成を示す．DC のデータベースでは 2 種類のテーブルで情報を管理する．Node information テーブルには DC に登録を行っている NTM 端末の端末情報を記録し，NTMobile における経路判断およびトンネル構築に利用する．DC キャッシュは DC および DNS サーバの情報を一時的に保持するテーブルであり，DC が新たに管理する情報である．DC キャッシュは NS レコードと TXT レコードの問い合わせ結果から判断して登録し，今後同一ドメインの FQDN に対して通信を開始する際に参照する．Zone Name は DC および DNS サーバのドメイン名であり，管理しているゾーン名を指す．この情報は各ゾーンを管理する DNS の IP アドレス等の情報と NTMobile への対応状況を含むキャッシュのように扱う．この情報を管理することにより，NTM 端末が通信を開始する際，通信相手の FQDN のドメインが DC キャッシュに存在するかを確認し，情報があれば NS レコードや TXT レコードを問い合わせる必要がなくなる．

3.3 提案シーケンス

提案シーケンスを図 6 に示す．MN, CN の DC に対する登録処理および Keep Alive は従来と同様である．登録処理を受け取った DC はその情報を Node Information テーブルに登録しておく．

3.3.1 名前解決処理

MN はアプリケーションからの DNS 名前解決処理を検出すると，$FQDN_{CN}$ を抽出して独自のネゴシエーションを開始する．MN は NTM Direction Request に $FQDN_{MN}$ と $FQDN_{CN}$ を記載して DC_{MN} へ送り，名前解決およびトンネル構築指示を依頼する．DC_{MN} は NTM Direction

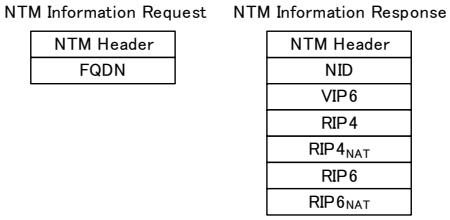


図 7 メッセージフォーマット
Fig. 7 Message format.

Request に記載されている $FQDN_{MN}$ で Node Information テーブルを検索することにより MN の端末情報を取得し，$FQDN_{CN}$ のドメインで DC キャッシュを検索する．DC キャッシュに情報が無い場合，$FQDN_{CN}$ の NS レコードを DNS クエリにより問い合わせる．DNS からの NS レコードの応答には DNS サーバの名前だけが含まれ，IP アドレスが含まれていない場合がある．その場合には DNS サーバの名前から DNS クエリにより再度 IP アドレスを問い合わせる．

ここで，CN が一般端末であった場合，DNS サーバが一般のものである場合があるため，NS レコードによる DNS サーバの名前解決後，その DNS サーバが DC であるかどうかの判断を行う必要がある．よって，再度 DNS クエリによって DC_{CN} の TXT レコードの問い合わせを行う．DC には DNS の資源レコードの 1 つである TXT レコードとして NTMobile の DC であることを示す文字列を含んで登録する．

これにより，DC_{MN} は NS レコードによって特定した DNS サーバが DC であるか一般の DNS であるかを判断できる．DC_{MN} が収集した DC_{CN} の情報は，今後の通信確立時およびハンドオーバによるトンネル再構築時に利用できるため，DC_{MN} 内に DC キャッシュとして保持しておく．

特定した DNS サーバが DC であった場合，DC_{MN} は NTM Information Request に $FQDN_{CN}$ を載せ，DC_{CN} に CN の端末情報を要求する．DC_{CN} は，$FQDN_{CN}$ が示す CN の端末情報を Node Information テーブルから検索し，NTM Information Response に載せて DC_{MN} へ送り返す．これにより DC_{MN} は CN の端末情報の取得を完了する．NTM Information Request/Response のメッセージフォーマットは図 7 に示す通りである．NTM Information Request には通信相手の FQDN を含み，NTM Information Response にはその FQDN に対応した NTM 端末情報を含む．これにより，NTM 端末の端末情報収集が DNS のキャッシュの影響をうけることを防ぐことができる．

特定した DNS サーバが一般の DNS サーバであった場合，DC_{MN} は DNS サーバに直接 $FQDN_{CN}$ の A レコードと AAAA レコードのみを問い合わせる．

3.3.2 トンネル構築

DC は従来の NTMobile と同様に収集した MN と CN の端末情報を元に経路を判断し，NTM Route Direction によって MN と CN が通信可能なトンネル構築を指示する．

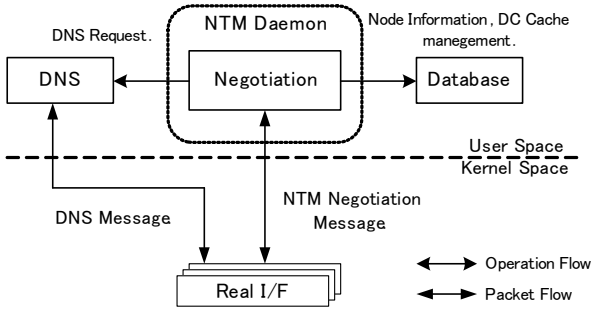


図 8 モジュール構成図
Fig. 8 Module Configuration.

3.3.3 ハンドオーバ時の動作

MN がネットワークを切り替えた場合，変化したアドレス情報などを載せた NTM Registration Request を DC_{MN} に送り，Node Information テーブルを最新の情報に更新する．次に，MN は 3.3.1 項の名前解決処理と同様に $FQDN_{CN}$ を NTM Direction Request に載せて DC_{MN} に送信する．DC_{MN} は $FQDN_{CN}$ が示す CN の端末情報を再度収集する必要があるが，DC キャッシュに DC の管理ゾーンに関する情報を保持しているため，ただちに NTM Information Request/Response により CN の端末情報を受け取ることができる．更新された MN の端末情報と最新の CN の端末情報からトンネル経路を判断し，NTM Route Direction によって新たなトンネル構築を指示する．NTM 端末は，指示に従ってトンネルを再構築する．

3.4 内部ネットワークに対するの名前解決

提案の通信シーケンスでは，A レコードなどの名前解決を NTM Direction Request によってグローバルネットワークに配置されている DC に任せる形になっている．しかし，ネットワークによっては DNS サーバがネットワーク内部に設置され，プライマリ DNS サーバとして登録されている場合がある．このような場合に対応するため，NTMobile の名前解決処理と並行して，プライマリ DNS への問い合わせもそのまま実行させる．プライマリ DNS 問い合わせの応答がある場合は，NTMobile の名前解決より先に終了する場合が多いため，一時的に待避しておく．

NTMobile の処理完了後，MN は DC_{MN} の名前解決の結果によって内部ネットワーク側との直接通信を行うか NTMobile のトンネル通信を行うかを選択する．DC_{MN} が CN の名前解決に成功すればトンネル構築の指示を行う．DC_{MN} から名前解決ができない場合，待避していた DNS 応答の結果をアプリケーションに通知し，実アドレスによる通信を行う．

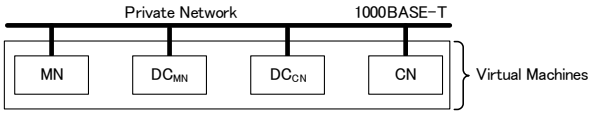


図 9 試験ネットワーク構成
Fig. 9 Test Network configuration.

表 1 測定結果
Table 1 Result of overhed time by negotiation.

	計測結果 [ms]
ネゴシエーション時間合計	50.6
DC 処理時間	32.0
DNS 解決時間	7.5

4. 評価

4.1 実装と動作検証

提案方式を実装し，動作検証を行った．図 8 に DC のモジュール構成図を示す．DC にはトンネル構築などのネゴシエーションを行う NTM デーモンをデーモンプログラムとして実装している．また，DC は DNS 問い合わせを行うため，DNS サーバ機能を搭載している．提案方式のデータベースによる端末情報管理のため，新たにデータベースソフトの MySQL 5.1 を DC にインストールした．NTM デーモンには新たに扱う NS レコード，TXT レコードおよびデータベースを扱うモジュール群を追加し，通信シーケンスの変更を行った．また，NTM 端末とのネゴシエーションを行う NTM デーモンに対しても通信シーケンスの変更に対応するよう実装を行った．

動作検証として通信開始における動作を確認し，性能評価を行った．図 9 に試験ネットワーク構成を示す．1 台の実機 PC 上にインストールした VMware 5.0 を利用して NTM 端末 2 台および DC2 台を仮想マシンとして構築し，同一のプライベートネットワークへブリッジ接続した．また，ネゴシエーション時に使用する暗号化アルゴリズムとして AES-CFB，認証アルゴリズムは HMAC-MD5，鍵長 128bit とし，事前に設定した．

4.2 性能評価

表 1 に名前解決処理に要した時間の測定結果を示す．測定回数は 10 回の平均値を示している．ネゴシエーション時間合計とは NTM 端末が DNS 問い合わせを検出した時から DNS 応答に仮想 IP アドレスを書き込んでリゾルバに渡すまでの時間である．DC 処理時間とはネゴシエーション時間の一部で，DC が NTM Direction Request を受け取ってから NTM Route Direction の送信を完了するまでの処理時間である．DNS 解決時間は DC 処理時間の一部で，DC キャッシュが無い場合，DNS サーバの探索から NTM Information Request を送信するまでの処理時間を示す．

表 2 従来方式と提案方式の比較
Table 2 Comparison.

	DNS レコードを用いた方式	提案方式
運用管理の容易さ	△	○
DC 管理の柔軟性	×	○
NTM 端末情報の秘匿性	×	○
通信開始の確実性	△	○
通信開始のオーバヘッド	△	○

測定は仮想マシンによって行ったため通信遅延はほとんどないが，実ネットワークでは通信遅延が大きく影響を及ぼす．従来の DNS レコードを用いる方式においては，3G ネットワークに接続した携帯端末から A, AAAA, NTM4, NTM6 の各レコードを取得するまでに 4 往復の問い合わせを必要とし，実測値によるとそれにかかる時間は合計 450~490ms 程度であった．また，3G ネットワークと無線 LAN にそれぞれ接続した携帯端末同士の NTM Tunnel Request/Response の 1 往復に要する時間はおよそ 300ms であった [10]．一方，提案方式では最初から DC にトンネル構築の指示を依頼するため，3G ネットワークでの 4 往復分の通信時間が発生しない．また，提案方式では CN の端末情報を取得するため DC 同士で 1 往復の通信を行う．グローバルネットワーク上における国内の一般的な RTT は 20ms 程度であるため，この時間が加算される．DNS レコード問い合わせ時間を 480ms と仮定し，この値から MN と DC_{MN} 間で行われる NTM　Direction Request および NTM Route Direction による 1 往復の通信時間を 120ms と仮定する．これらの値から実ネットワークにおけるネゴシエーション時間合計を推測すると，従来方式が約 900ms に対し，提案方式では約 440ms と推測できる．

提案方式においては DC_{MN} は 2 回目の問い合わせ以降は DC キャッシュに保持する情報を用いて CN の端末情報を取得することができる．しかし，DC キャッシュに該当する情報が無い場合は DNS の仕組みを用いて DC を発見する必要があり，NS，A，AAAA，TXT レコード問い合わせによる 4 往復の通信が追加される．この時間を推測すると，通信時間の増加は 80ms 程度である．この場合，表 1 の DNS 解決時間の部分が 80ms となり，その分ネゴシエーション時間合計も増加する．提案方式では DC キャッシュの時間を長めに設定できるため，通信開始時のオーバヘッドを大きく削減することができる．

4.3 提案方式の利点

従来の DNS レコードを用いた方式と提案方式の比較を表 2 に示す．

- 運用管理の容易さ
  通信シーケンスの変更に伴い，通信トンネル構築に必要な NTM 端末情報は DDNS による管理からデータ

ベースによる管理に変更となり，Dynamic DNS に動的な情報の登録および更新を行うことがなくなった．そのため，これまではDC に DDNS の機能が必要だったが通常の DNS サーバでの運用が可能になった．

- DC 管理の柔軟性
  DNS レコードとして登録，管理されていた NTM 端末の端末情報をデータベースによる管理に移行することにより自由に端末情報を管理できるようになり，柔軟性が確保された．
- NTM 端末情報の秘匿性
  従来の DNS レコードを用いた管理方法では DNS 問い合わせによって NTM 端末の端末情報が公開されている状況にあった．提案方式では NTM 端末情報をデータベースに格納することで NTMobile の問い合わせに対してのみ NTM 端末情報を提供することができるようになった．
- 通信開始時の確実性
  DNS サーバに残るキャッシュの影響を受けることなく通信相手の端末情報を取得することができるようになったため，より確実な接続性を確保することができるようになった．
- 通信開始のオーバヘッド
  携帯端末のネットワークへの接続状況によっては通信遅延が大きい環境が想定されるため，通信開始に時間を要する可能性があった．提案方式で NTM 端末が複数の DNS 問い合わせによって NTM 端末情報を収集する必要がなくなった．DC はいずれもグローバルネットワークに設置するため，遅延は安定して小さくなる．

5. まとめ

本稿では，アドレス空間や体系に依存しない通信接続性の確立と移動透過性を同時に実現する NTMobile における通信端末情報の管理および収集方法が抱えていた課題の解決手法について提案を行った．提案方式では通信開始におけるオーバヘッドを大きく削減するとともに，DNS 問い合わせを使用することによるキャッシュの影響により通信接続性の確保ができない場合がある課題を解決する通信シーケンスの提案を行った．さらに，DNS レコードとして公開されていた端末情報をデータベースに格納し，秘匿性を確保するとともに運用管理の容易さと柔軟性を向上させた．また，提案方式を実装し，動作検証を行った．今後は実ネットワーク環境での詳細な性能評価を行う．

参考文献

[1] Le, D.,Fu, X. and Hogrefe, D.:A Review of Mobility Support Paradigms for the Internet, *IEEE Communications Surveys*, Vol. 8, No. 1, pp. 38-51(2006).

[2] Soliman, H.: Mobile IPv6 Support for Dual Stack Hosts and Routers, *RFC 5555,IETF*(2009).

[3] Johnson, D., Perkins, C. and Arkko, J.: Mobility Support in IPv6, *RFC 3775, IETF* (2004).

[4] Perkins, C:IP Mobility Support for IPv4, Revised, *RFC 5944, IETF*(2010).

[5] 西尾拓也，内藤克浩，水谷智大，鈴木秀和，渡邊　晃，森香津夫，小林英雄 : NTMobile における端末アドレスの移動管理と実装，マルチメディア，分散，協調とモバイル (DICOMO2011) シンポジウム論文集，Vol.2011，No.1，pp. 1139-1145(2011).

[6] 納堂　博史，鈴木秀和，内藤克浩，渡邊　晃 : NTMobile の経路最適化の検討，情報処理学会研究報告，Vol. 2011-MBL-61,No. 33, pp. 1—8 (2011).

[7] 上醉尾一真，鈴木秀和，内藤克浩，渡邊　晃 : IPv6 ネットワークにおける NTMobile の検討，情報処理学会研究報告，Vol.2011-MBL-61, No.33, pp.1-8(2011).

[8] 鈴木秀和，水谷智大，西尾拓也，内藤克浩，渡邊　晃:NTMobile における相互接続性の確立手法と実装，マルチメディア，分散，協調とモバイル (DICOMO2011) シンポジウム論文集，Vol.2011，No. 1，pp. 1339-1348(2011).

[9] 内藤克浩, 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, DICOMO2011 論文集, Vol.2001, pp.1349-1359(2011).

[10] 上醉尾一真，鈴木秀和，内藤克浩，渡邊　晃 : IPv4/IPv6 混在環境における NTMobile の検討，情報処理学会第 74 回全国大会論文集，pp.3-221-3-222(2011).

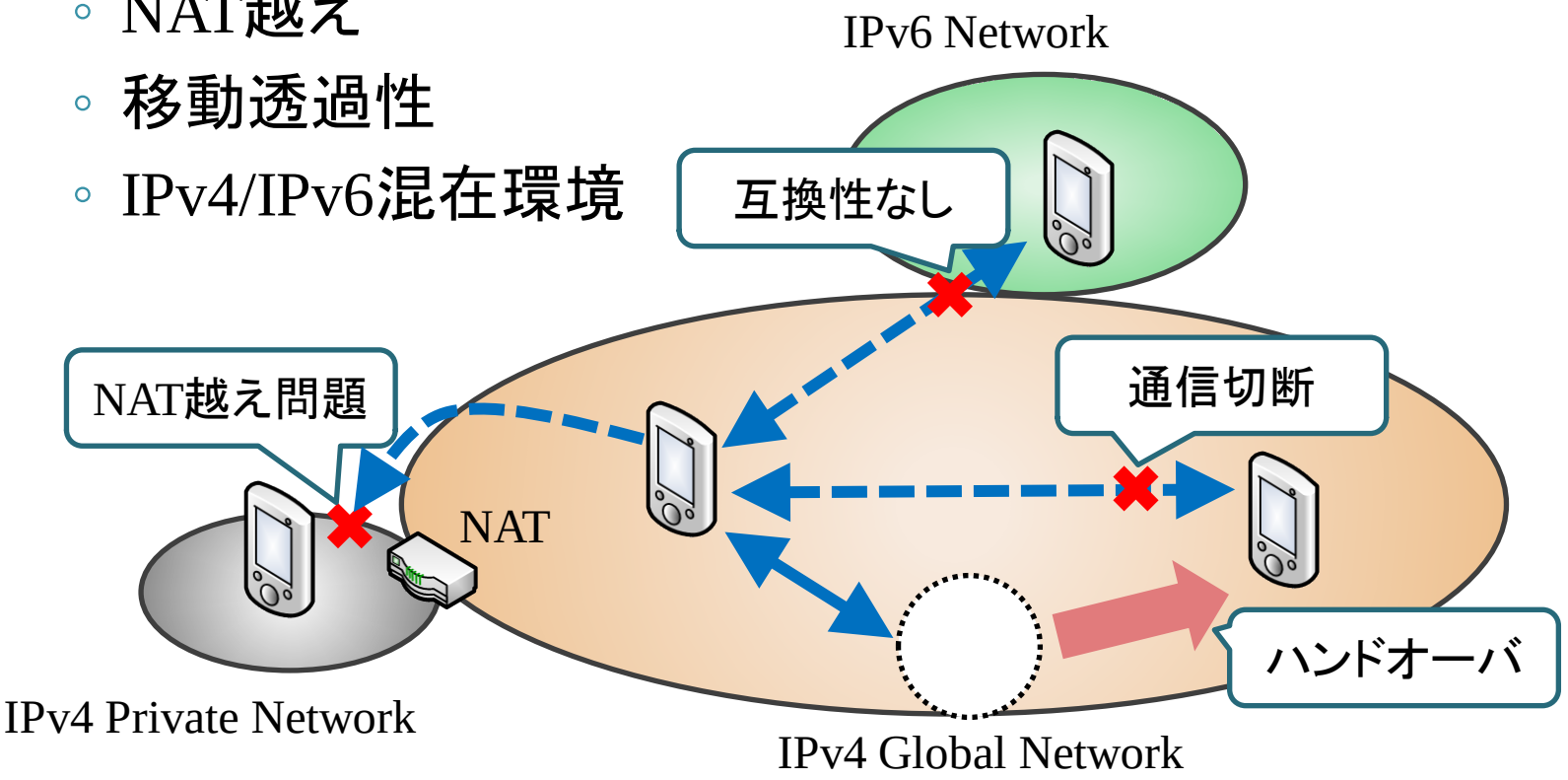
NTMobileにおけるDNS実装変更が不要なデータベース型端末情報管理手法の検討

細尾幸宏† 鈴木秀和† 内藤克浩‡
旭建作† 渡邊晃†

† 名城大学大学院 理工学研究科
‡ 三重大学大学院 工学研究科

はじめに

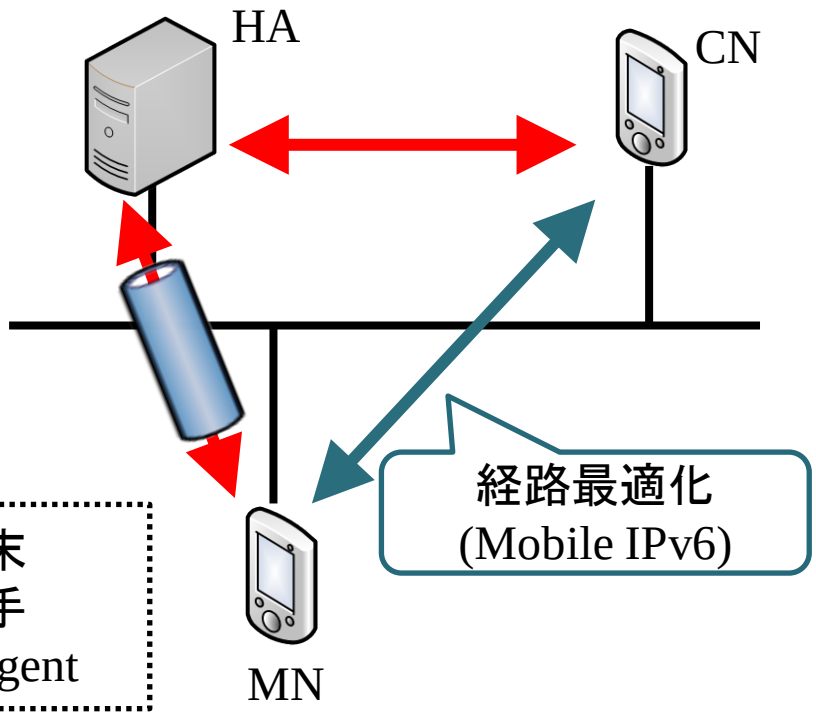
- IPネットワークの課題
  - NAT越え
  - 移動透過性
  - IPv4/IPv6混在環境



IPv4/IPv6混在環境における接続性と移動透過性
NTMobile (Network Traversal with Mobility)

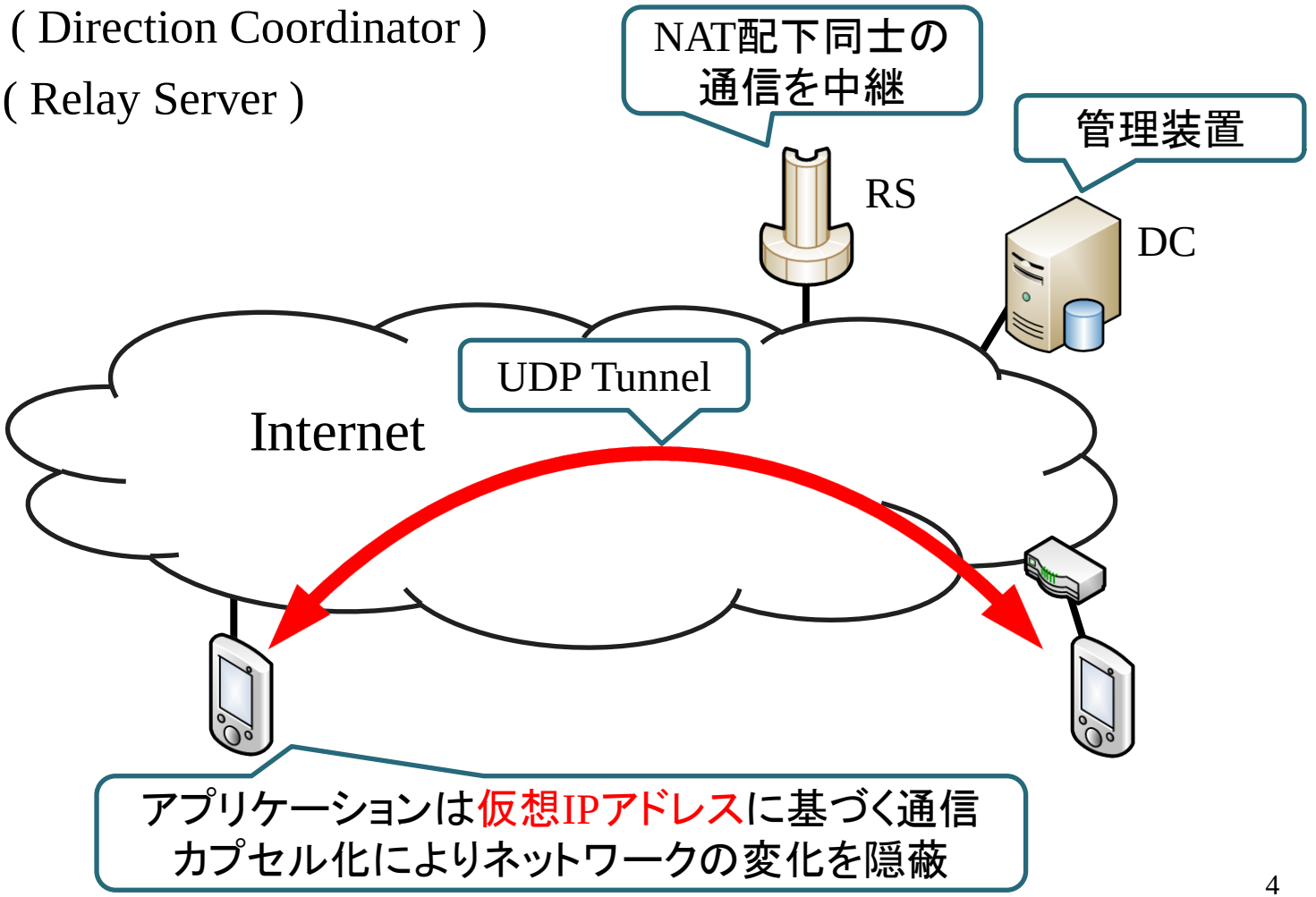
類似研究

- DSMIP(Dual Stack Mobile IPv6)
  - IPv4/IPv6混在環境において移動透過性を実現
  - HA経由によりCNに対してMNの移動を隠蔽
  - IPv4における課題
    - 経路最適化
    - HAの一点障害



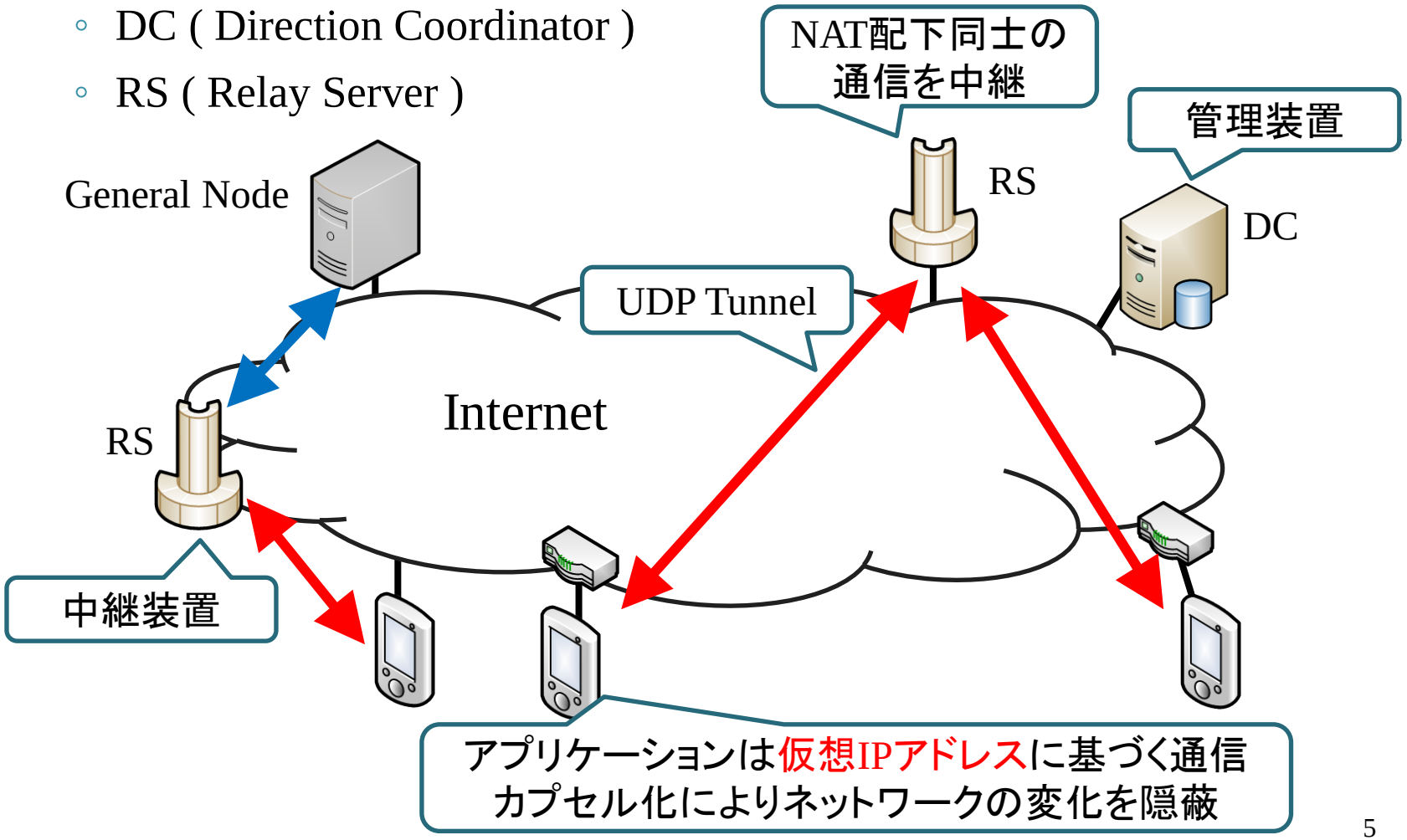
NTMobileの概要

- NTMobileの構成
  - NTMobile Node
  - DC (Direction Coordinator)
  - RS (Relay Server)



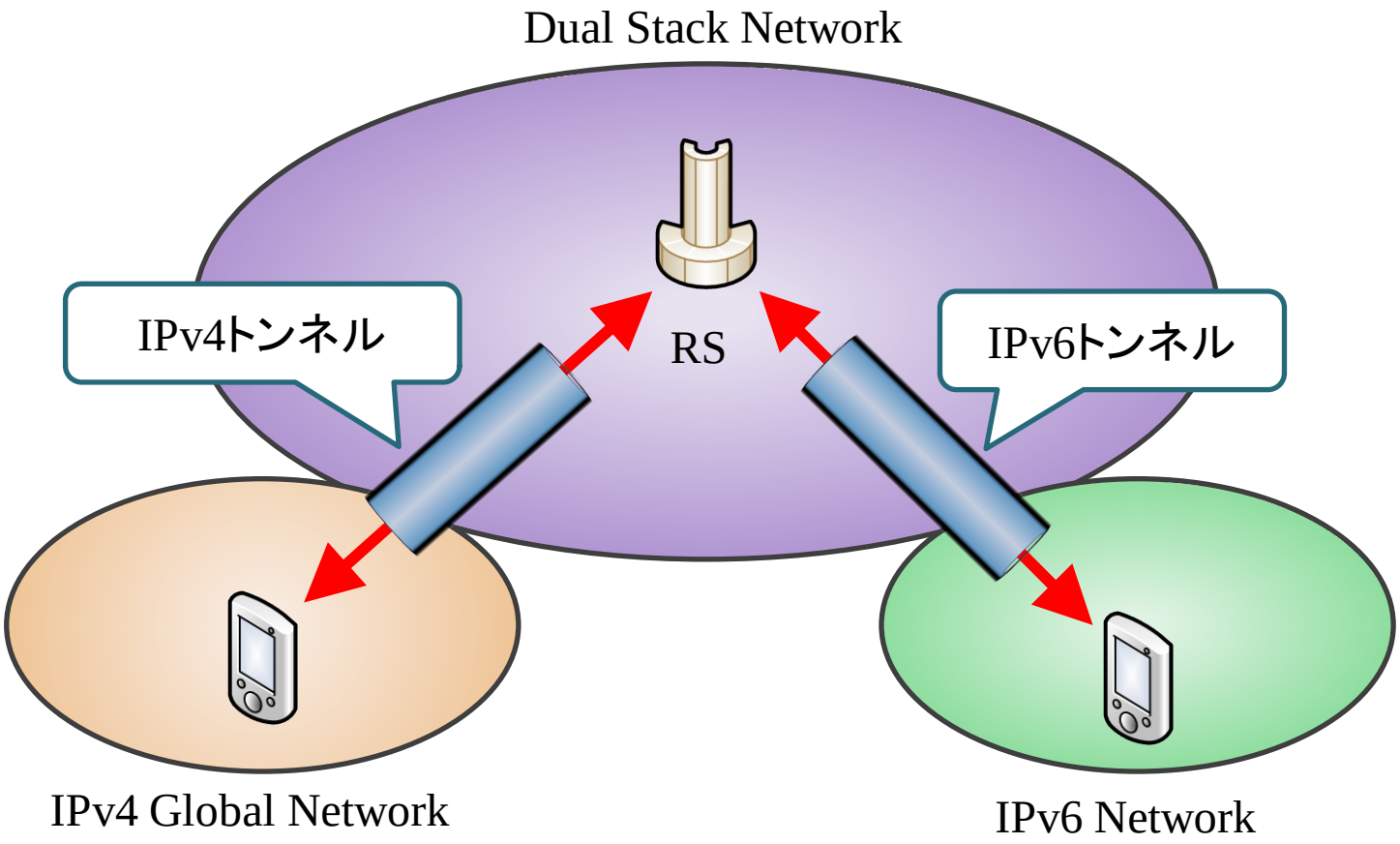
NTMobileの概要

- NTMobileの構成
  - NTMobile Node
  - DC (Direction Coordinator)
  - RS (Relay Server)



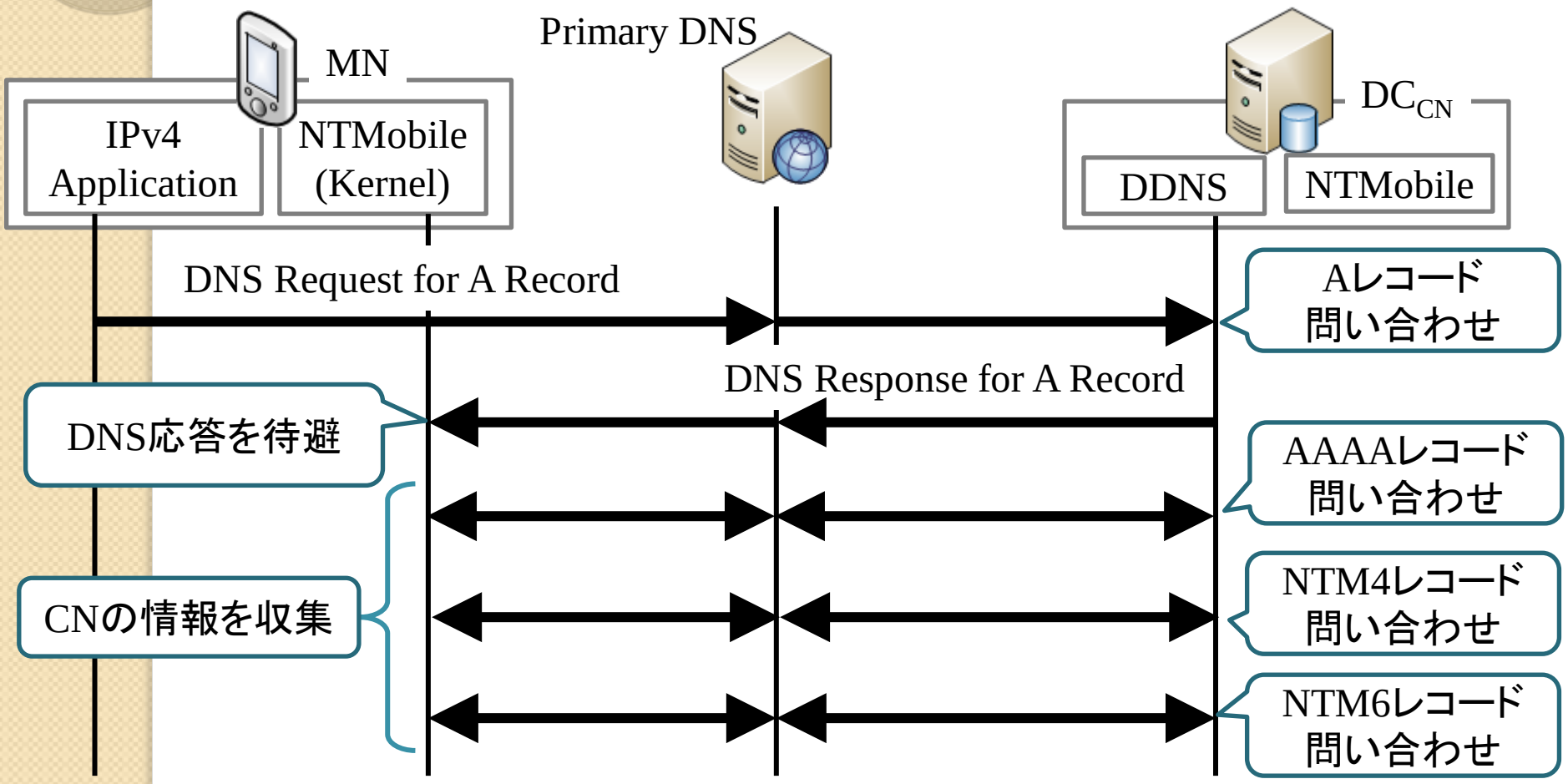
IPv4/IPv6ネットワーク間の通信

- Dual StackネットワークにRSを設置
- IPv4/IPv6それぞれのトンネルでRSを経由



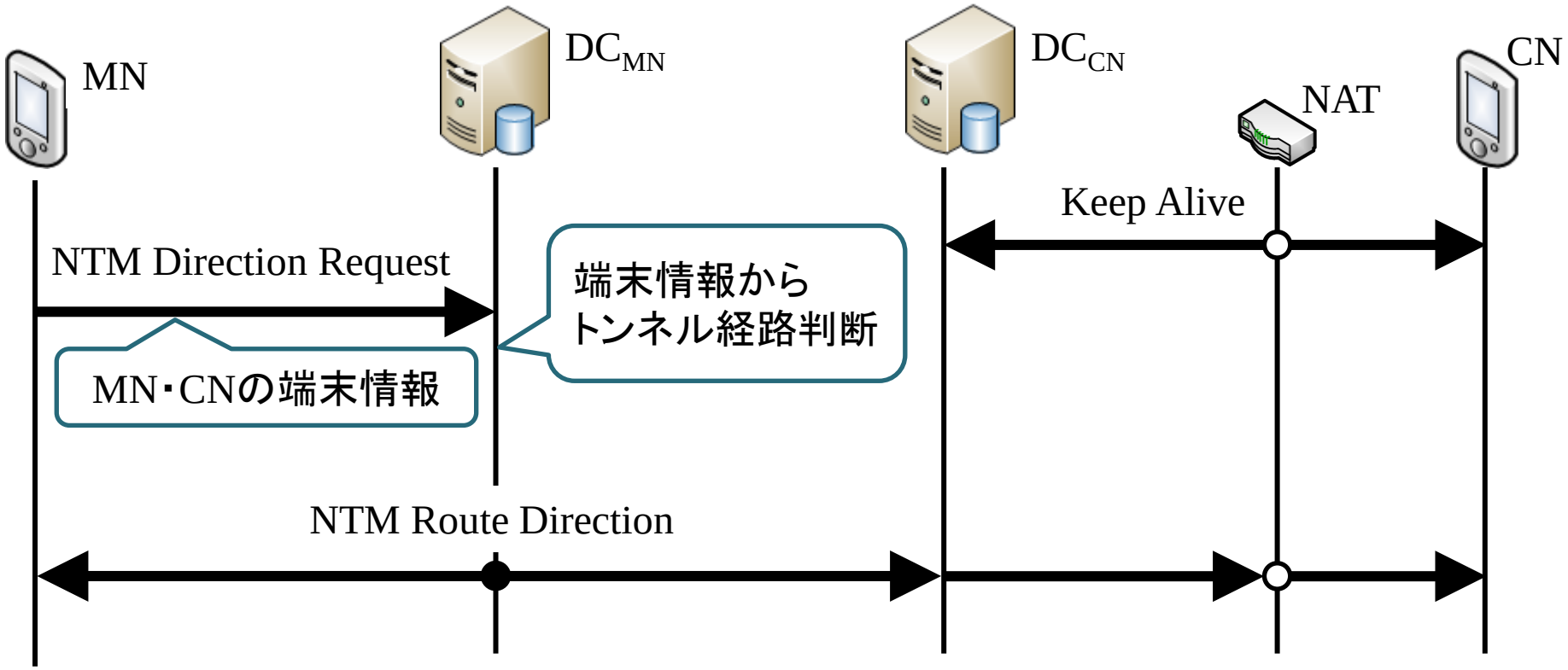
名前解決

- DNS問い合わせをトリガに動作開始
- CNのレコード情報をDNS経由で全て問い合わせる



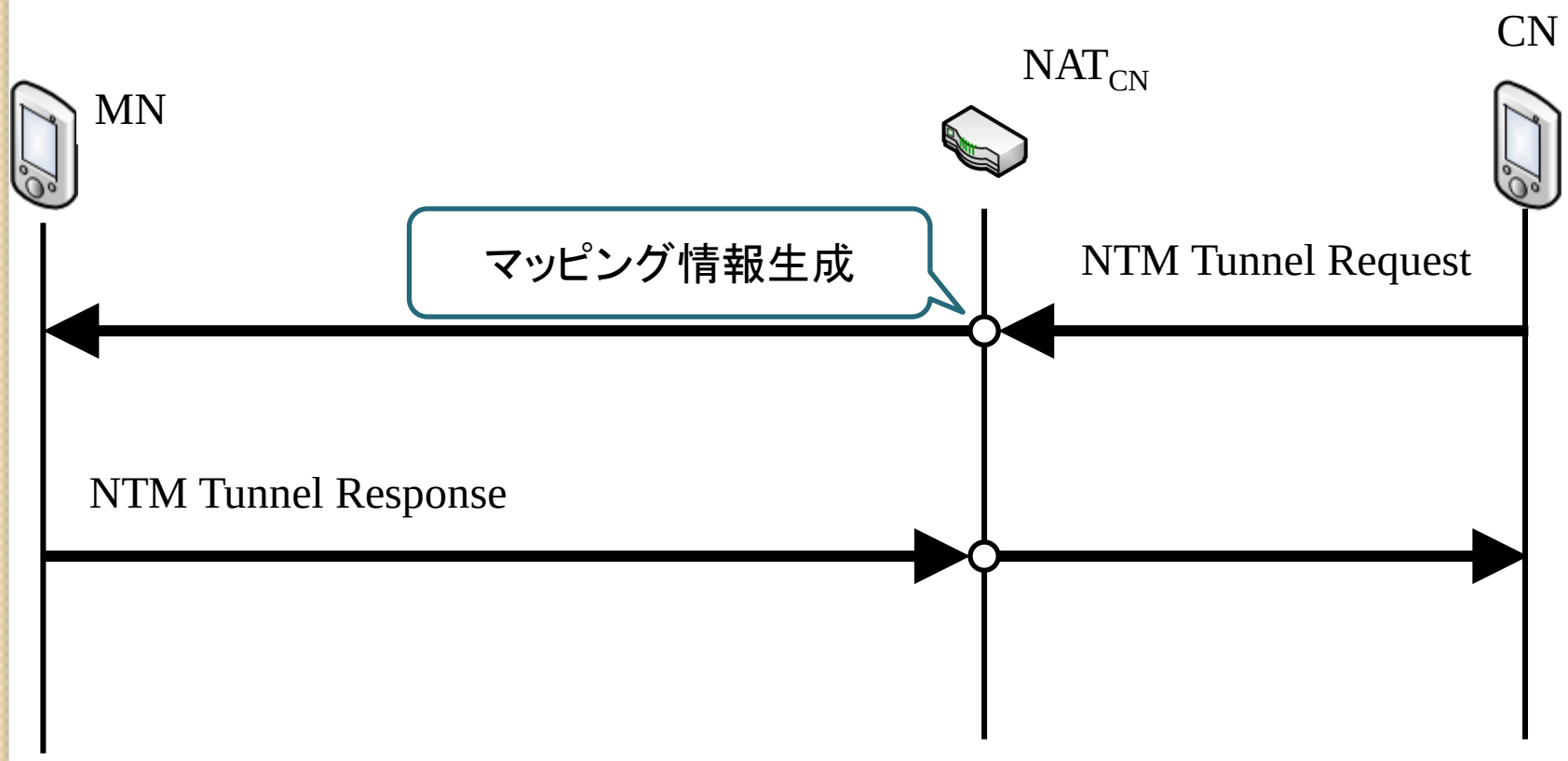
トンネル構築

- IPv4同士でNAT配下のCNに対してトンネル構築開始
- DC_{MN}がNTM Direction Requestを元に経路判断
- 両端末に指示を送る



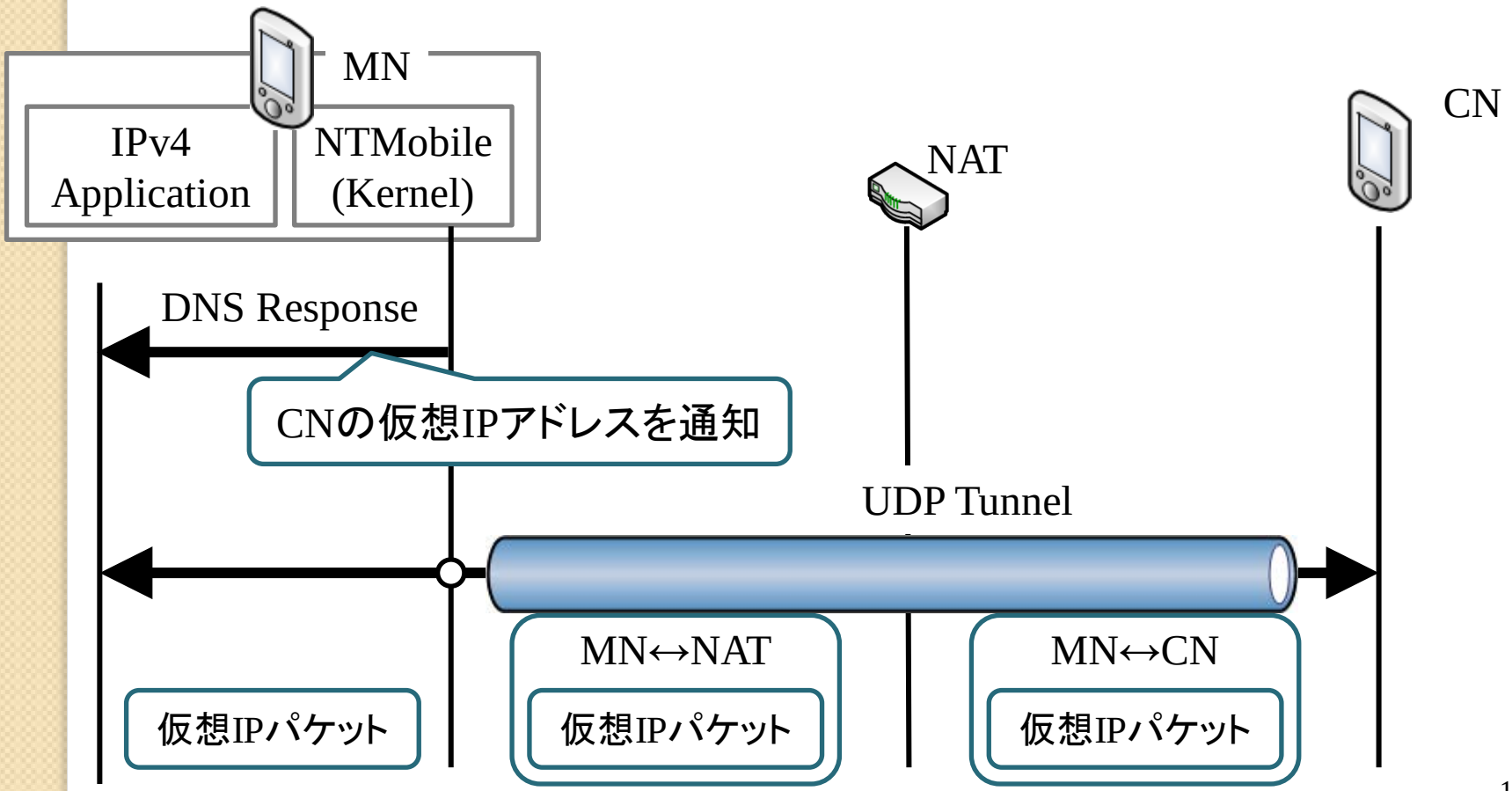
トンネル構築

- NAT配下の端末からUDPパケットの
  NTM Tunnel Requestを送信



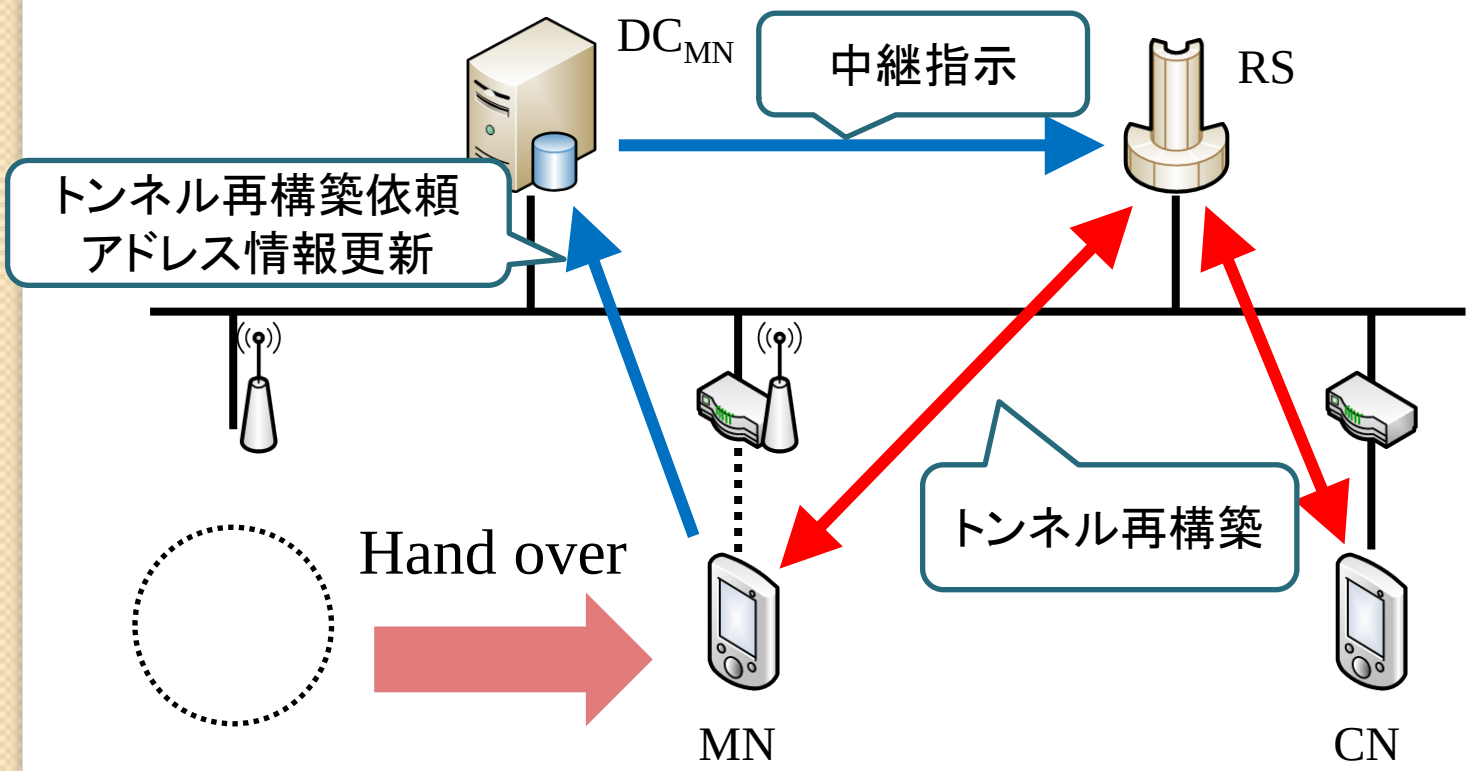
トンネル通信

- 仮想IPアドレスに書き換えたDNS応答をアプリケーションへ渡す
- カーネル空間でカプセル化



ハンドオーバ時

- MN-CN通信中にMNがNAT配下へ移動
- RS経由のトンネルを再構築



NTMobileの名前解決における課題

- DCにDDNS機能が必要
- NTMレコードをDNSレコード形式で管理
  - 管理方法に制限
- FQDNから誰でもNTMレコードを取得できる
- DNSキャッシュが有効だと最新アドレスが取得できない
  - DNSキャッシュを無効にする必要あり
  - 毎回DNSによる探索が必要
- 遅延の大きいネットワークにおける通信開始時のオーバヘッド

課題解決の方針

Watanabe Lab.

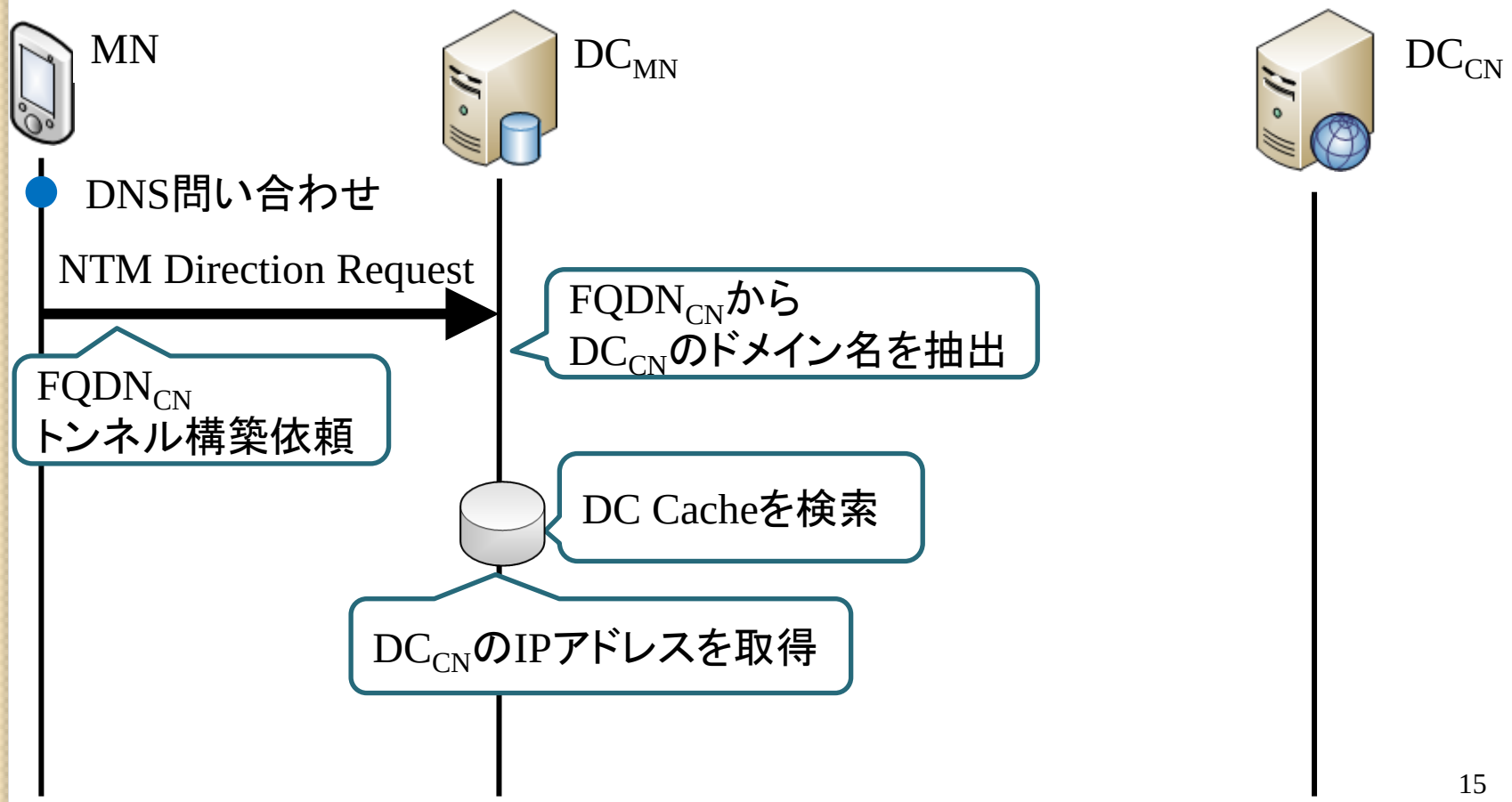
- NTM端末は通信相手の情報を収集しない
  - 通信開始時の遅延減少
- DC同士が直接通信により端末情報を取得
  - DNSキャッシュの影響をなくす
  - NSレコードによるDC探索
- データベースによる端末情報管理
  - 端末情報の秘匿性を確保
  - 柔軟な運用管理

データベース

- DNSレコード情報をデータベースに格納
- Node Information Table
  - NTMobile端末情報を管理
- DC Cache Table
  - ドメインごとのDNSに関する情報を管理
  - CNのFQDNから管理しているDNSを特定
  - DNSのNTMobile対応状況を管理
    - 直ちに端末情報を取得可能

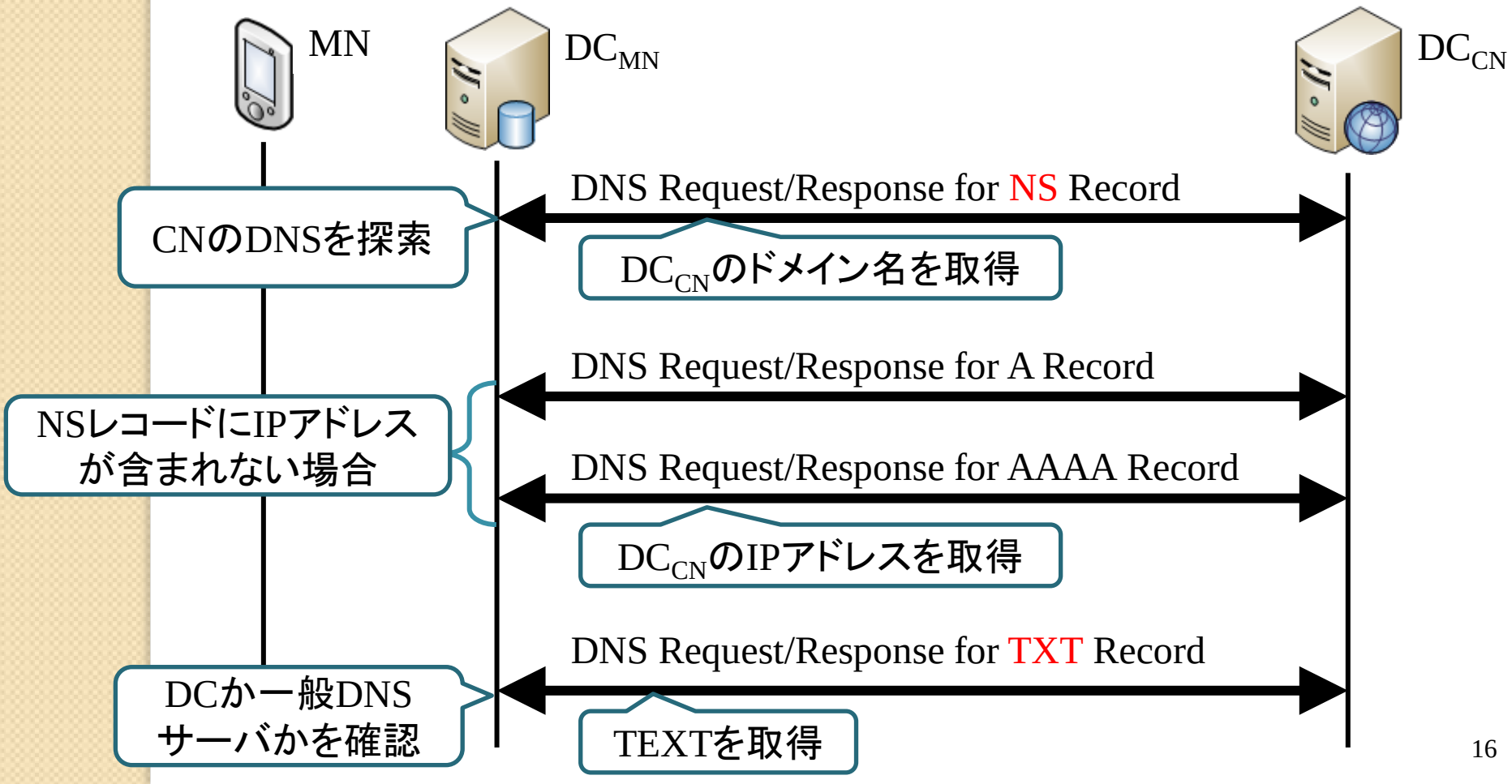
通信シーケンス

- DNS問い合わせをトリガに動作開始
- FQDNを載せてトンネル構築を依頼
- DC CacheからDCのIPアドレスを検索



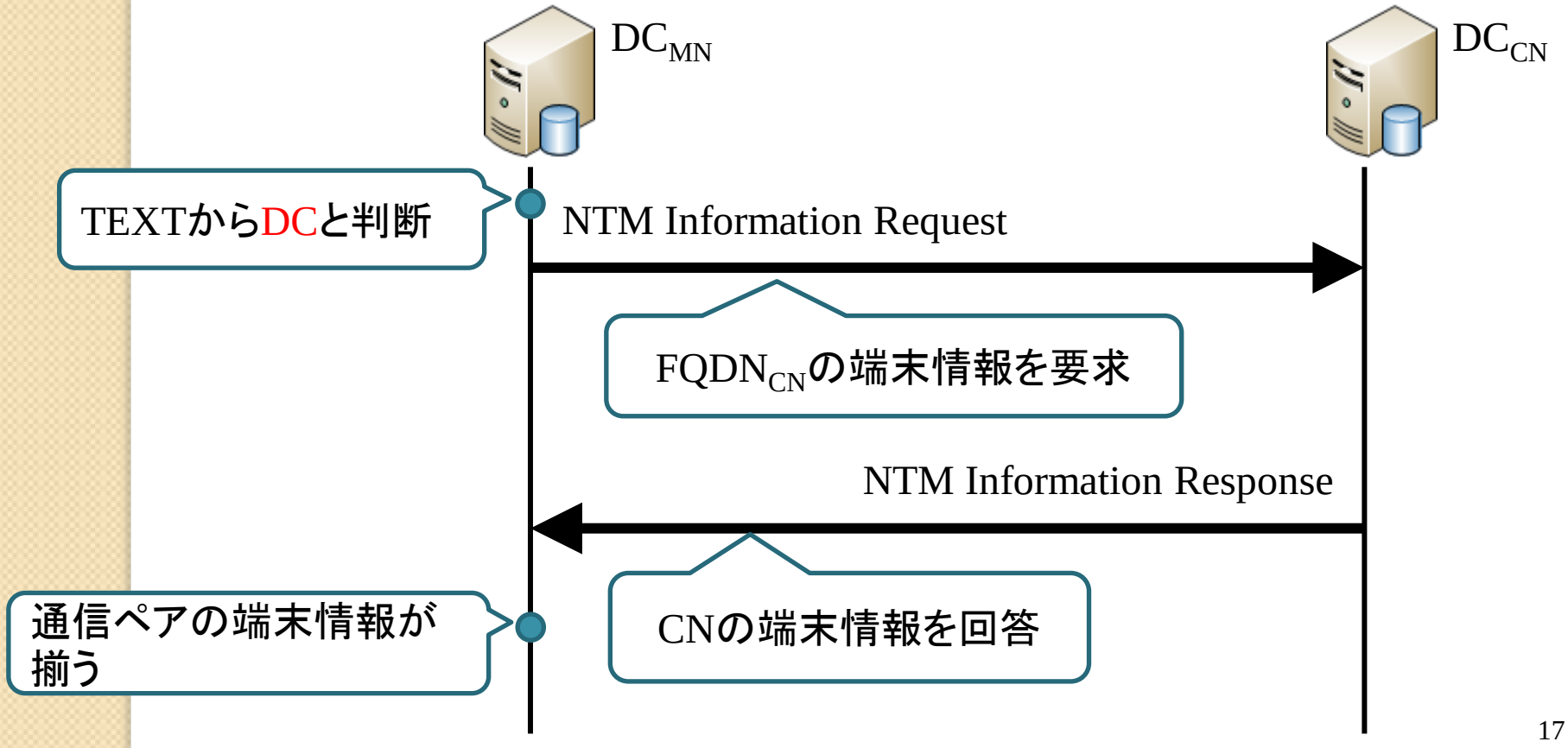
通信シーケンス

- DC CacheからIPアドレスが取得できない場合
- DC_{MN}がCNのDCを探索
  - DNSのNSレコードを利用



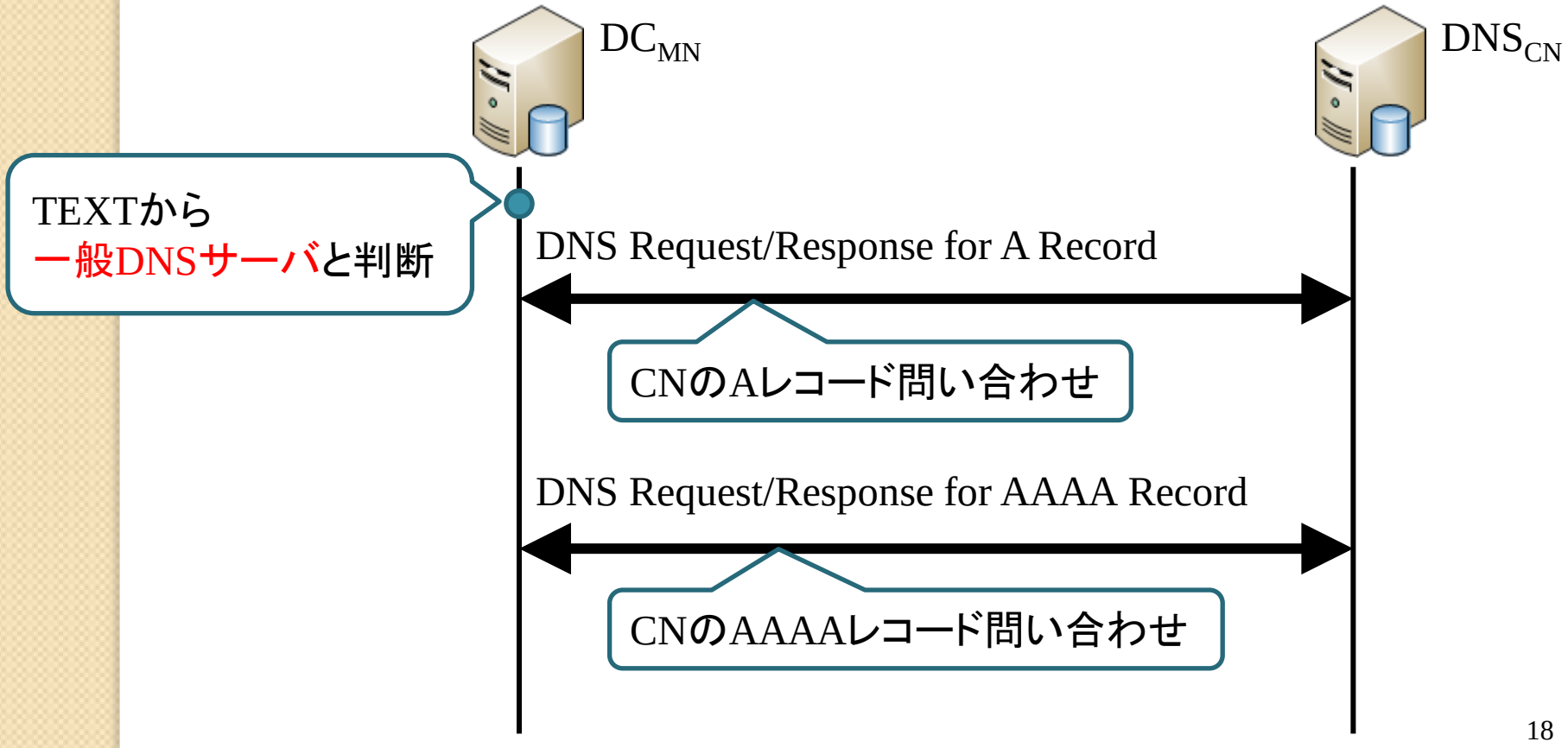
通信シーケンス

- DNSがDCの場合
- DC同士で直接通信し端末情報取得



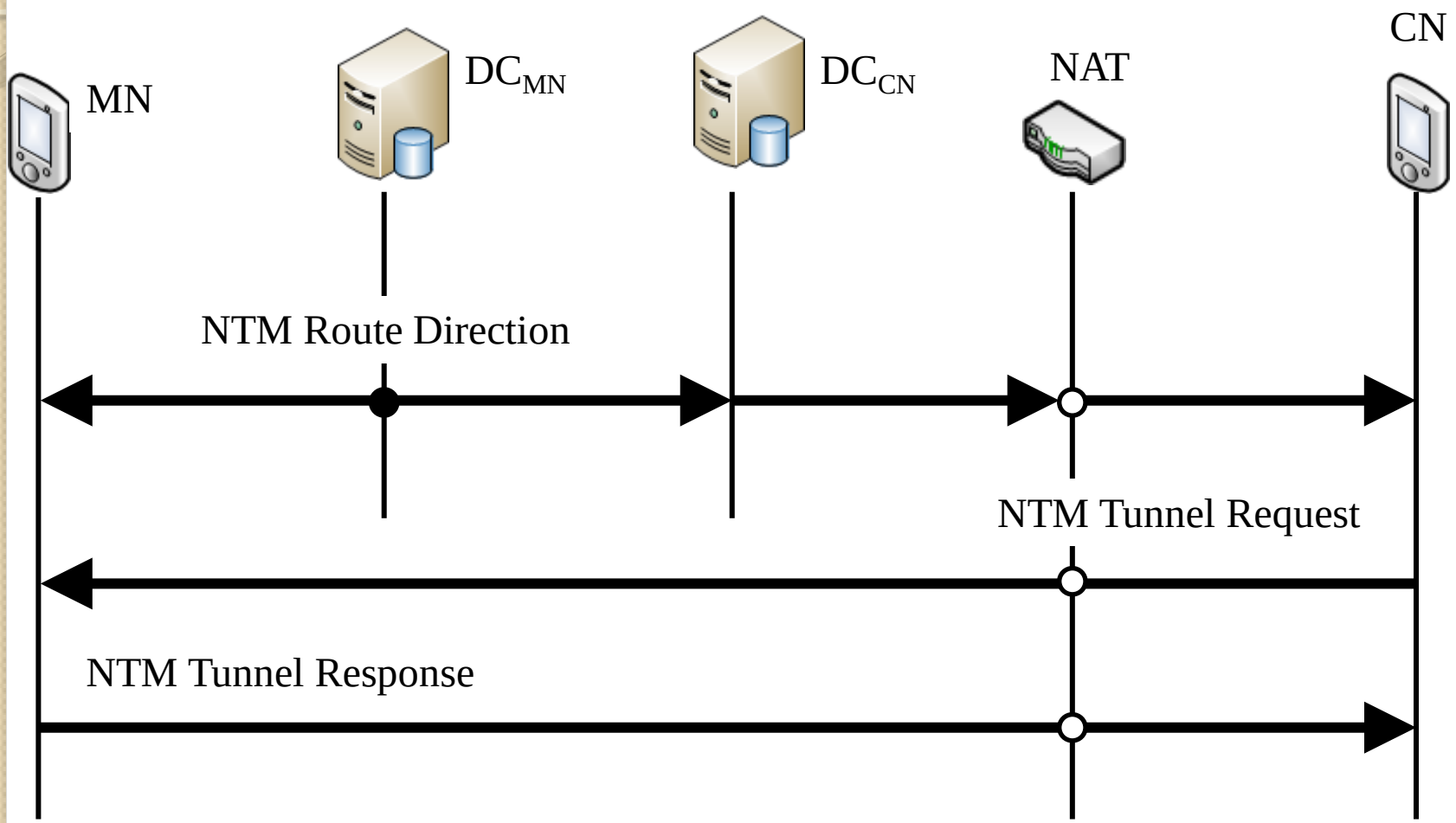
通信シーケンス

- DNSが一般DNSサーバの場合
- DC同士で直接通信し端末情報取得



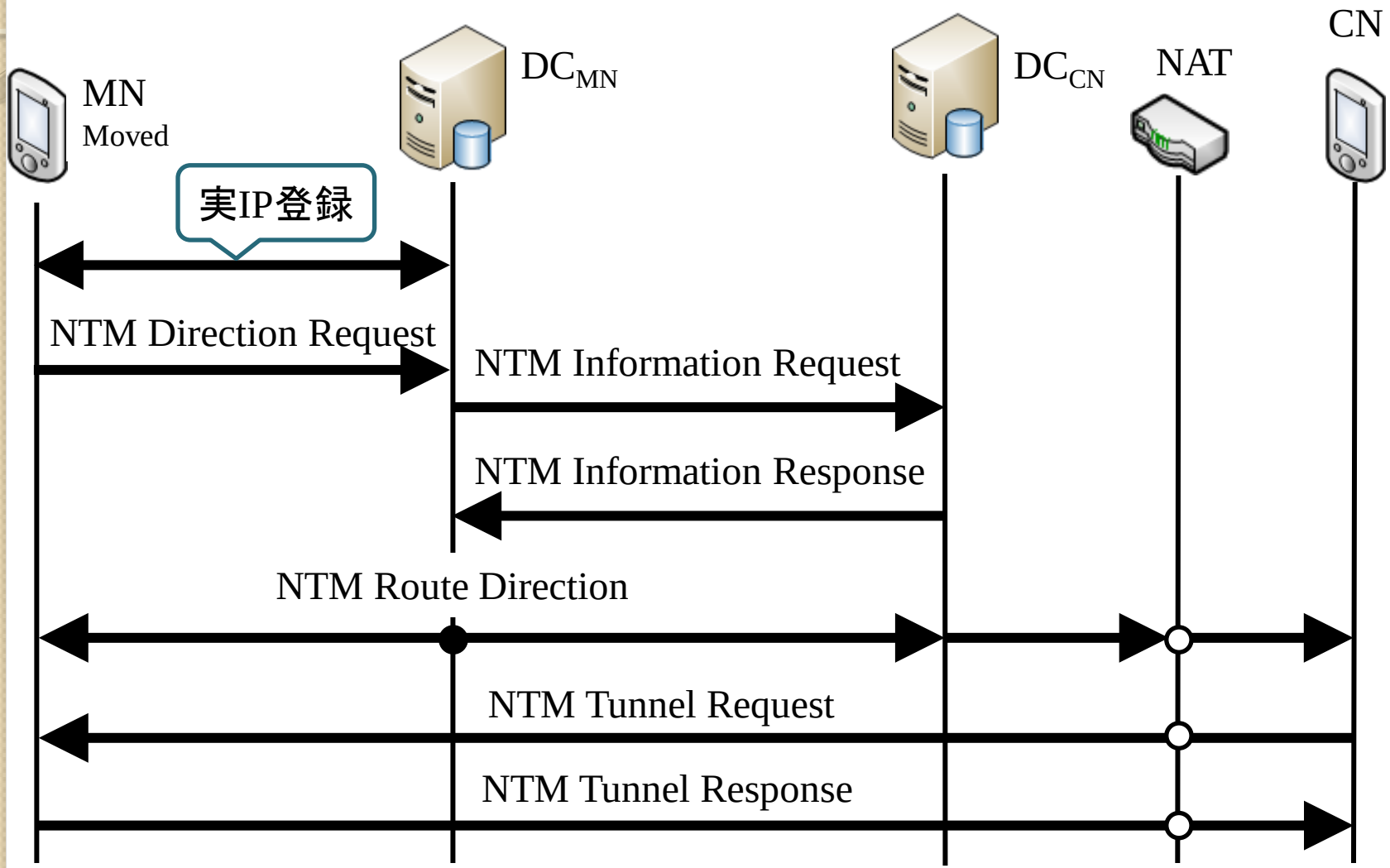
通信シーケンス

- トンネルの指示・構築は従来と同様



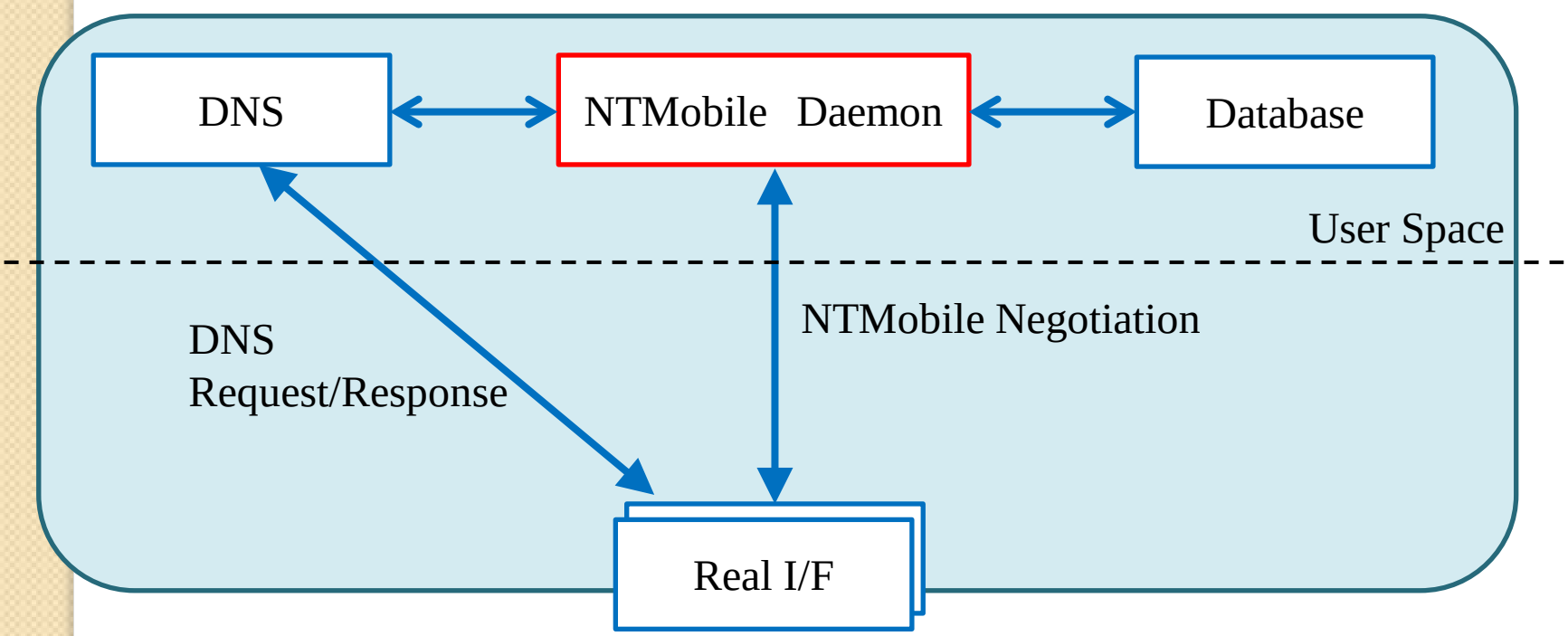
ハンドオーバ時

- 移動後の実IPアドレスを更新してトンネル再構築



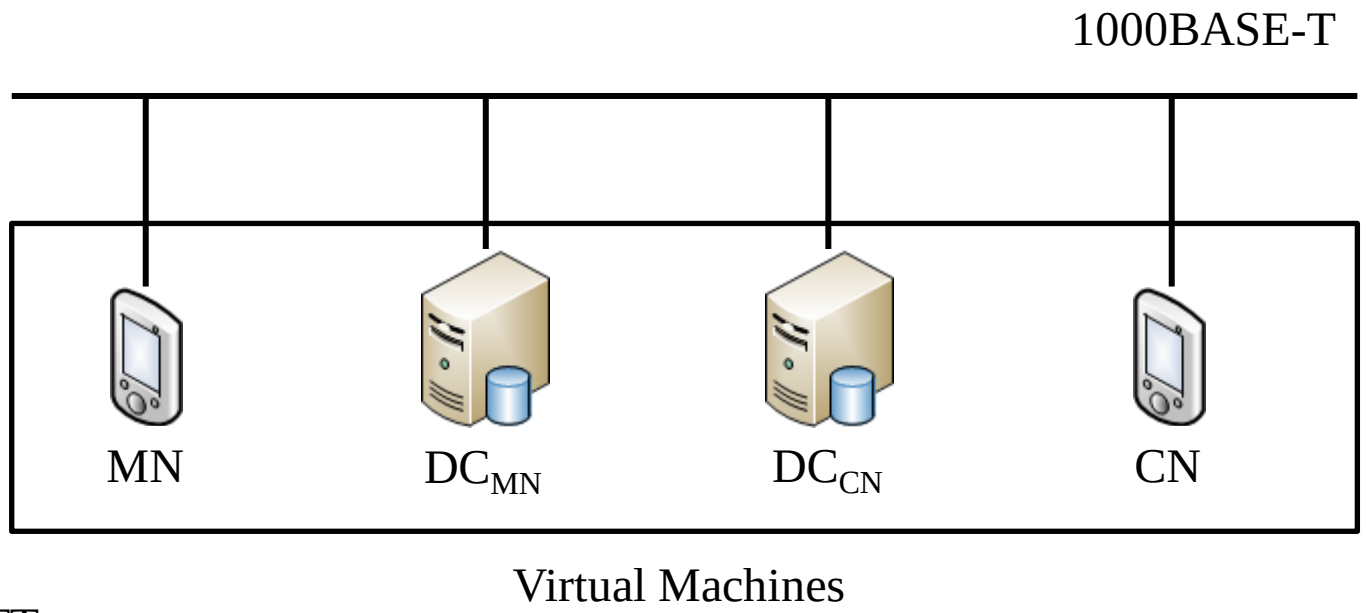
動作検証

- DCのモジュール構成
  - デーモンプログラム
    - NTMobileのネゴシエーション処理を行う
  - データベース（Mysql 5.1）
  - DNS(Bind 9)



動作検証

- 仮想マシンによる測定環境
  - ネゴシエーションによるオーバヘッドの測定



平均RTT:

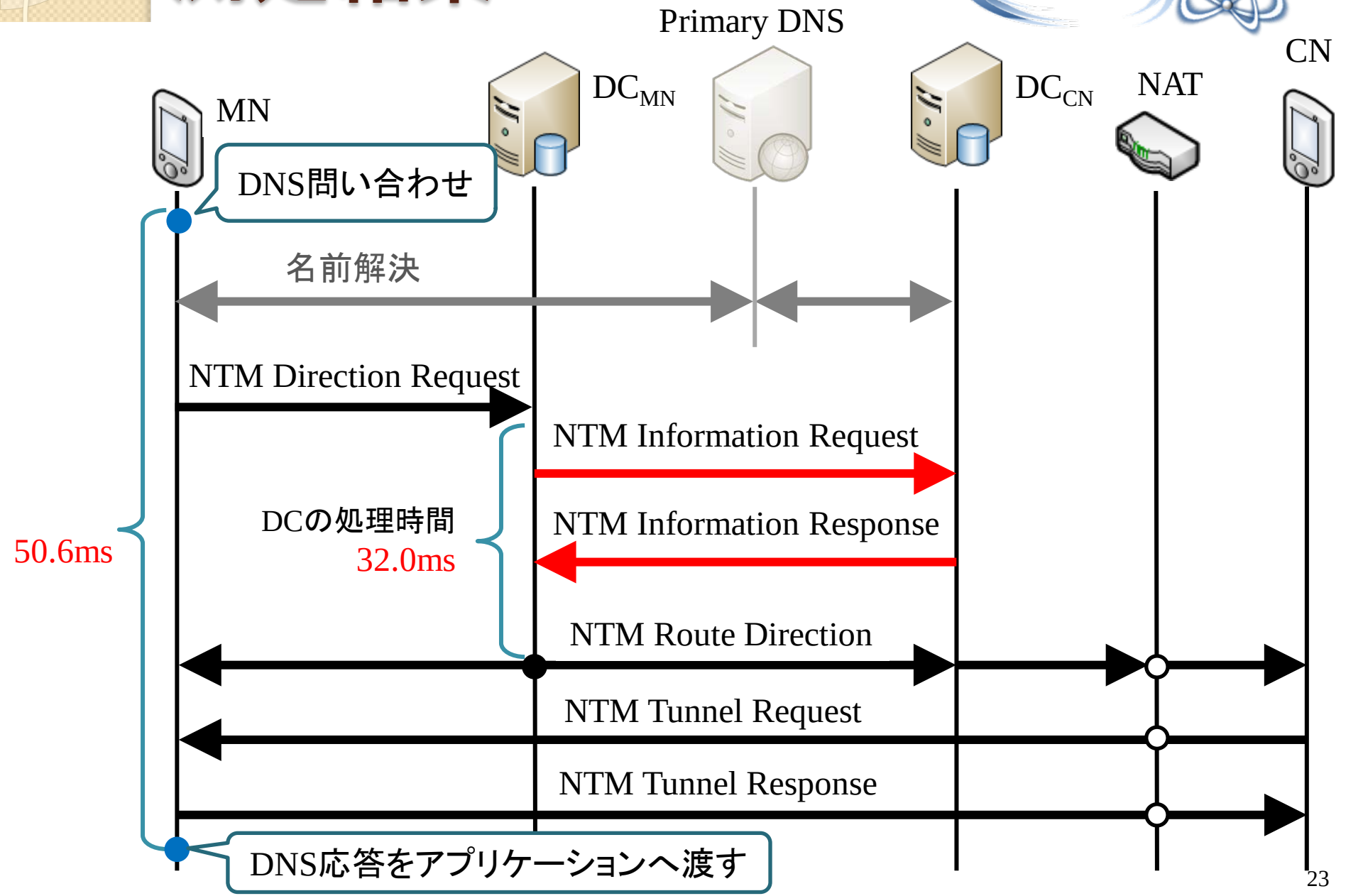
MN-MN : 0.36ms

DC_{MN}-DC_{CN} : 0.36ms

MN-DC_{MN} : 0.35ms

CN-DC_{CN} : 0.39ms

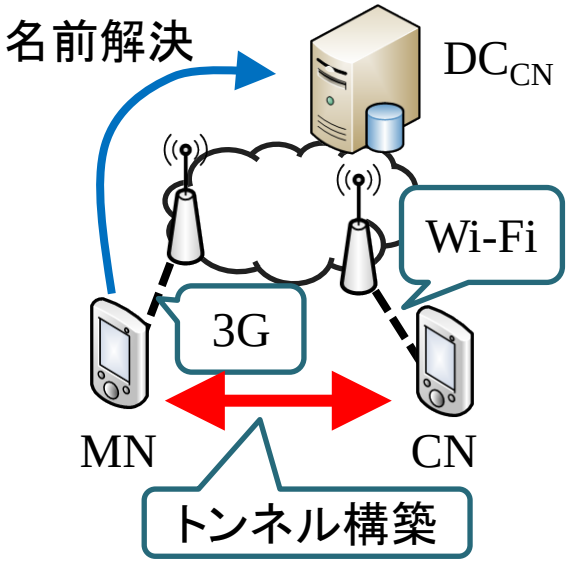
測定結果


Primary DNS
MN
DC$_{MN}$
DC$_{CN}$
NAT
CN
DNS問い合わせ
名前解決
NTM Direction Request
NTM Information Request
DCの処理時間
32.0ms
NTM Information Response
50.6ms
NTM Route Direction
NTM Tunnel Request
NTM Tunnel Response
DNS応答をアプリケーションへ渡す

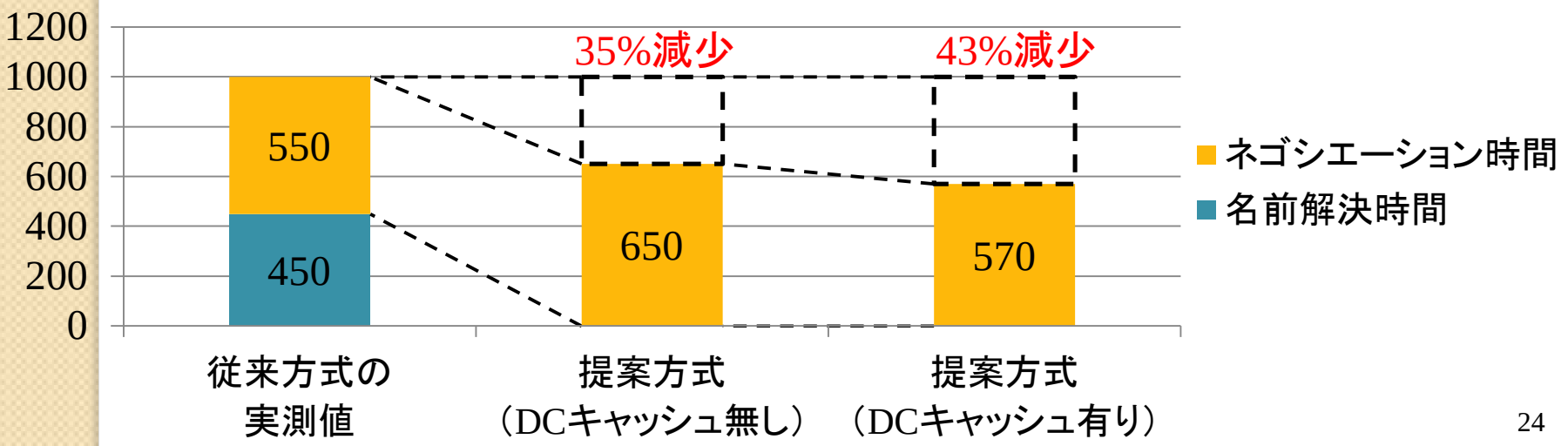
実環境における予測

- 従来方式の実測値は合計1秒
  - 名前解決時間：450ms (3G環境)
- 提案方式における予測値
  - 名前解決時間は0に減少
  - 端末情報取得に20ms増加
  - DCキャッシュ無し時に最大80ms増加

※国内の一般的なRTT：20ms



従来方式の測定環境



比較

Watanabe Lab.

	DNSを用いる方式	提案方式
運用管理の容易さ	△ DDNS機能が必要	○ DNSでよい
DC管理の柔軟性	× DNSレコードとして管理	○ データベースによる自由な管理
NTM端末情報の秘匿性	× 誰でもNTMレコードが取得可能	○ NTMobileネゴシエーションに対して提供
通信開始の確実性	△ DNSキャッシュの影響を受ける	○ DNSキャッシュの影響を受けない
通信開始のオーバヘッド	△ 接続ネットワークの通信遅延の影響が大きい	○ 接続ネットワークの通信遅延の影響が小さい

まとめ

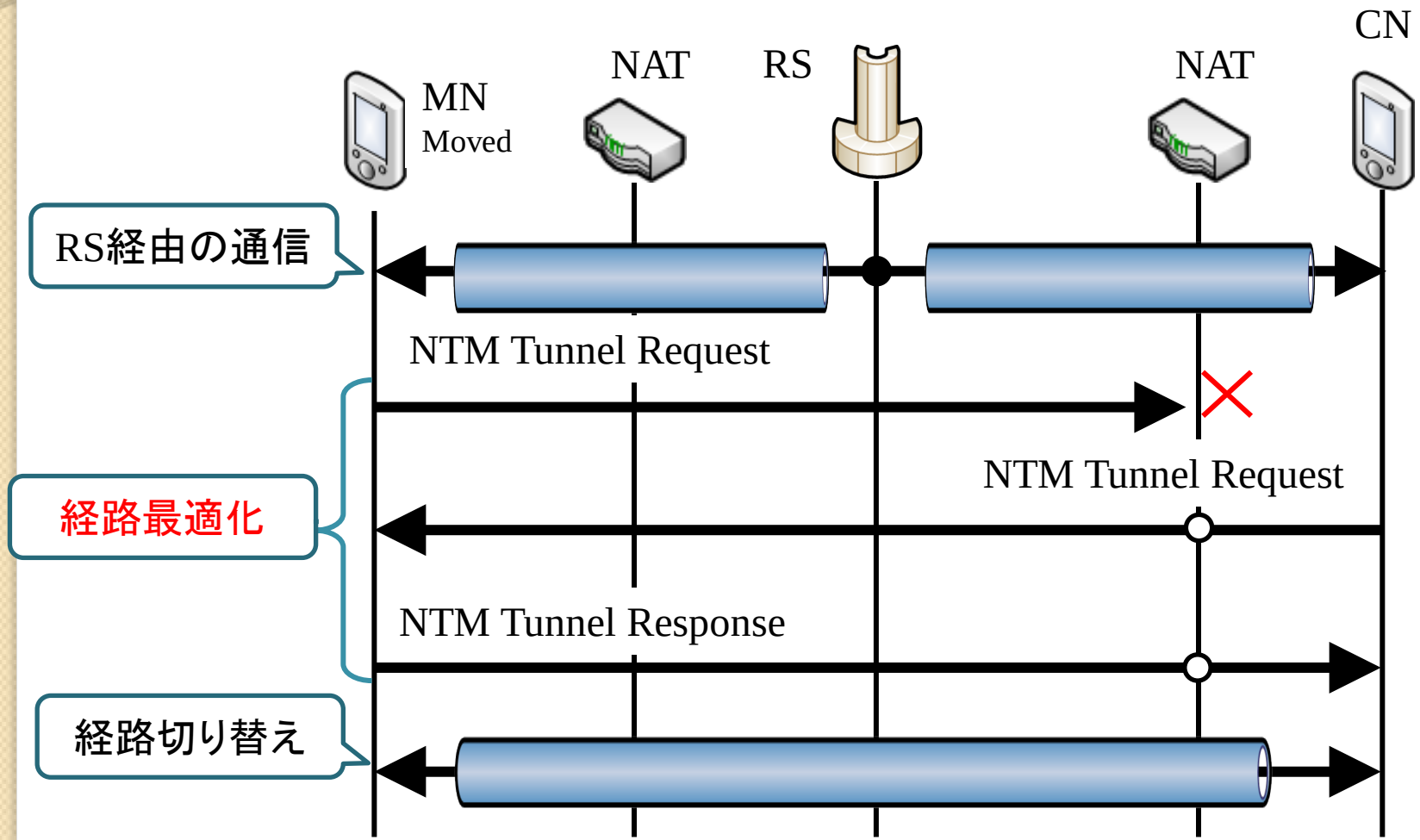
- NTMobile
  - IPv4/IPv6混在環境で通信接続性と移動透過性を実現
- 提案方式
  - DNS依存による制約を排除
  - 情報管理をデータベースへ移行
    - 自由な管理・運用
    - 端末情報の秘匿性確保
  - 通信開始時のオーバヘッドを削減
- 今後の予定
  - 提案方式の実環境における有用性の検証

ご清聴ありがとうございました

補足資料

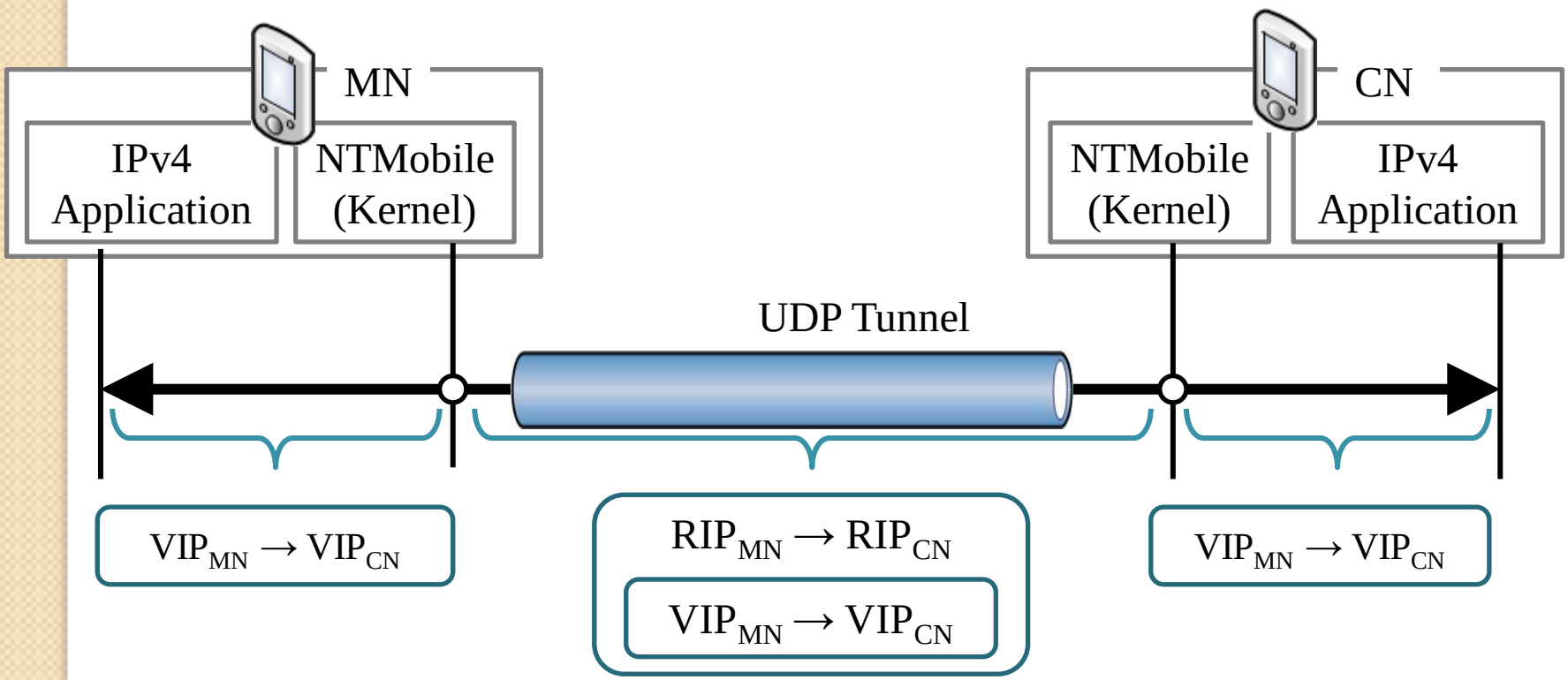
経路最適化

- RS経由の通信と並行して実行
- NATの種類に依存
- 異なるNAT配下端末同士でE2E通信



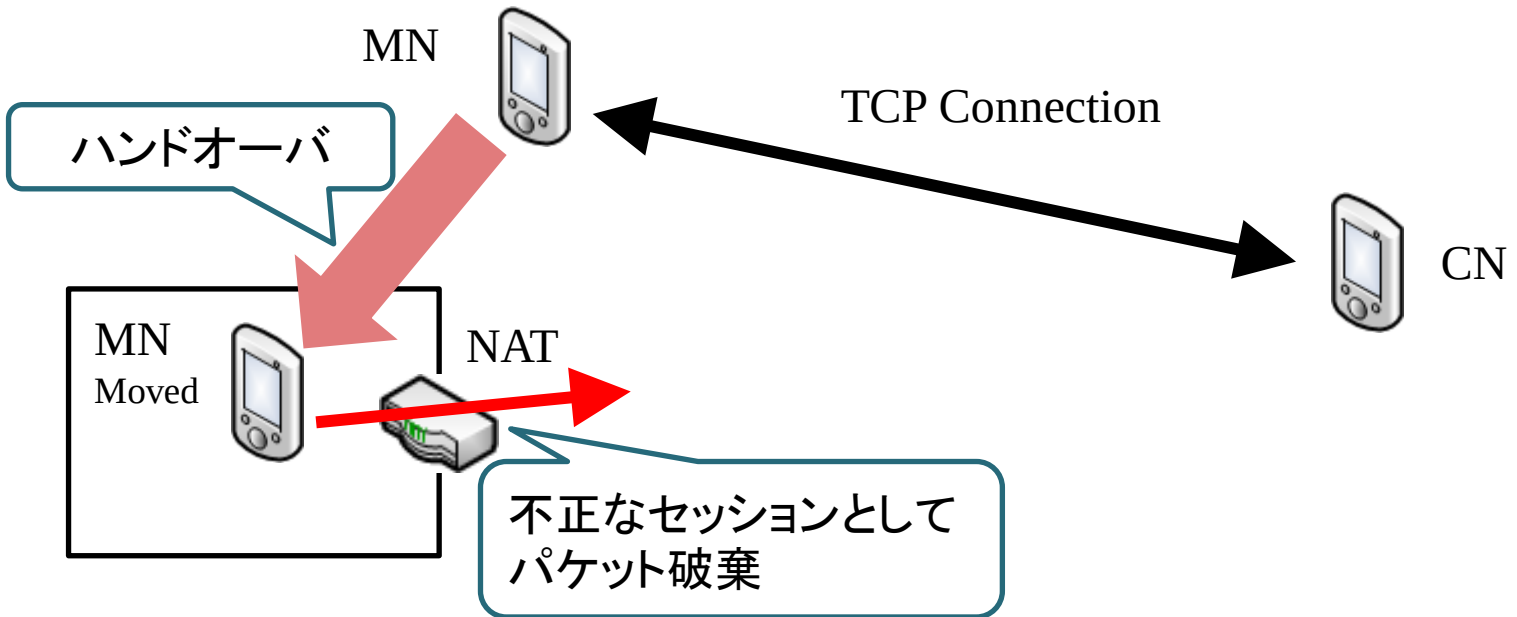
トンネル通信詳細

- アプリケーションは仮想IPアドレスを使用
- アプリケーションパケットの暗号化



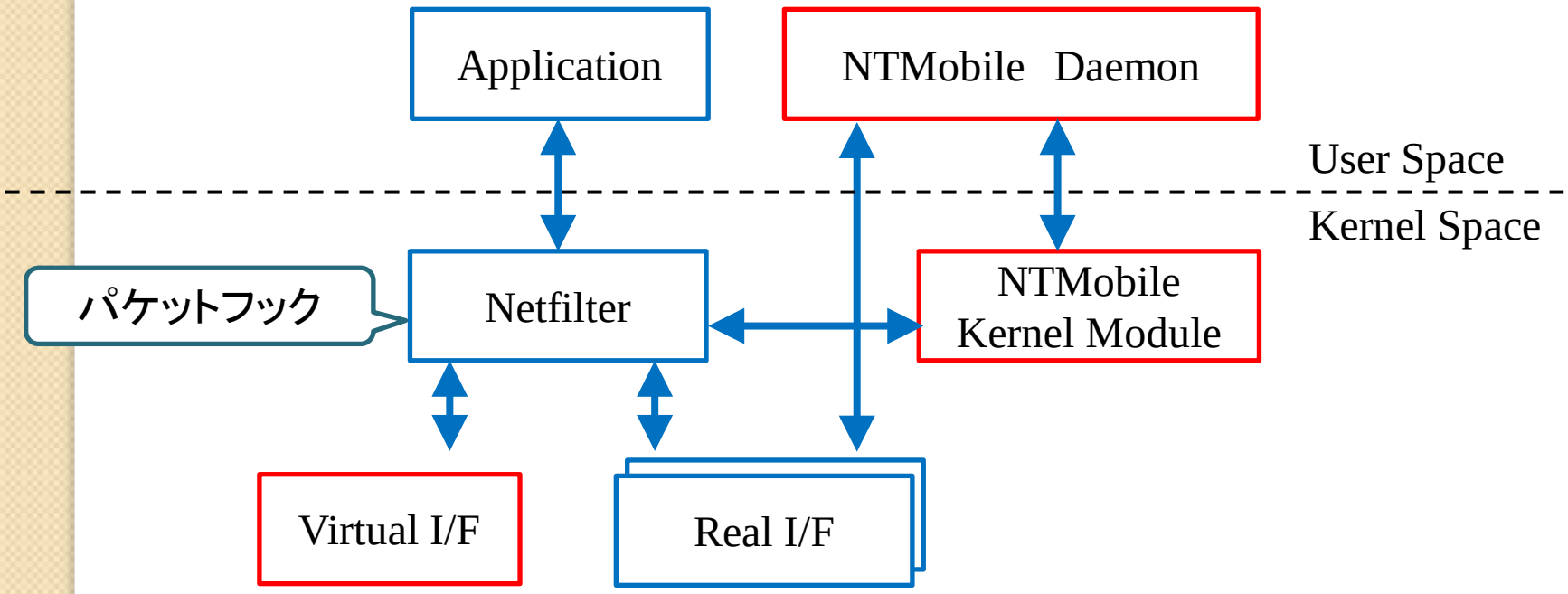
SPI (Stateful Packet Inspection)

- TCPのセッションを監視するフィルタ
- 多くのNATに搭載
- NATを跨った移動時にパケット破棄の問題
  - NTMobileではUDPトンネルを用いて解決



NTM端末のモジュール構成

- デーモンプログラム
  - トンネル構築などのネゴシエーション
  - NetlinkによりL2ハンドオーバ，IPアドレス変化を検知
- カーネルモジュール
  - カプセル化・暗号化



NTMレコード

- IPv4用のNTM4レコードとIPv6用のNTM6レコード
- Node ID
  - NTM端末を一意に識別するID
- 仮想IPアドレス (IPv4/IPv6)
  - DCが重複なくNTM端末に割り当てる
- 実IPアドレス (IPv4/IPv6)
  - NTM端末が接続ネットワークから取得
  - NTM端末のIPアドレス変化に伴い動的に更新
- NATの外側の実IPアドレス (IPv4/IPv6)
  - プライベートアドレス空間への接続を識別
- DCのIPアドレス (IPv4/IPv6)
  - NTM Route Directionの転送に使用

NTMobileとDSMIP

Watanabe Lab.

- IPv6ネットワークにおいて同様の機能
  - 移動透過性
  - 経路最適化
- IPv4ネットワークにおけるNTMobileの優位性
  - 経路最適化に対応
  - 管理，中継装置の多重化に対応
- NATが混在する環境におけるNTMobile
  - グローバルIPv4アドレスは必ずしも必要ではない
  - NATの改造は不要
    - DSMIPではどちらかに対応が必要

想定アプリケーション

- NTMobileはエンドツーエンド通信を実現
  - サーバ転送を必要としないリアルタイム性
    - 通話，動画，チャット，ゲーム
  - コンテンツ提供者のサーバ管理不要なサービス
    - ネットワークサービスへの参入が容易に

ハンドオーバ処理の最適化

- 最新実IPアドレスの更新処理を省略