

NTMobile を用いたエンドツーエンド通信によるチャットアプリケーションの提案

中村 隼大^{†*}, 鈴木 秀和[†], 内藤 克浩[‡], 渡邊 晃[†] ([†] 名城大学, [‡] 愛知工業大学)

Proposal of Chat Application of End To End Communication using NTMobile

Hayata Nakamura[†], Hidekazu Suzuki[†], Katsuhiko Naito[‡], Akira Watanabe[†] ([†] Meijo University, [‡] Aichi Institute of Technology)

1 はじめに

モバイルネットワークの普及に伴い、チャットが重要なコミュニケーションツールとなっている。チャットはクライアントサーバシステムで実現するのが一般的である。しかし、サーバから情報漏洩する懸念や、サーバの障害・二重化等に対する管理負荷が大きいという課題がある。

筆者らは、端末がどのようなネットワーク環境にいてもエンドツーエンドで通信を行うことができる NTMobile(Network Traversal with Mobility) を提案している [1]。そこで、本稿では NTMobile を用いたエンドツーエンド通信によるスマートフォン向けチャット通信方式を提案し、その利点を考察する。

2 従来のチャットアプリケーションの実現方法

従来のチャットアプリケーションは、インターネット上に存在する IRC(Internet Relay Chat) サーバとクライアントから成るクライアントサーバシステムで実現されている。このシステムでは、クライアントがサーバに対してメッセージを送付し、サーバが各クライアントに同一メッセージを配信する。サーバはクライアントに対して Notification^{*1} と呼ぶメッセージを送信し、アプリケーションを起動させる。これでクライアントはメッセージの受信を知り、サーバにチャットデータを取りに行くことができる。

現在のネットワークには通信経路上に NAT(Network Address Translation) が存在することが多く、エンドツーエンドの通信が阻害されている。そのため、インターネット上のサーバを介して通信を行う必要があった。クライアントサーバシステムでは、サーバの管理者が情報を取得できるため、セキュリティ上問題があるという指摘がある。また、サーバの二重化等の管理が必須である。

3 NTMobile によるエンドツーエンド通信

NTMobile は、主に NTMobile を実装した端末 (NTM 端末) と NTM 端末に対してアドレス情報の管理やトンネル構築指示を行う DC(Direction Coordinator) により構成される。NTM 端末はスマートフォンなどの移動端末を想定しており、インターネット上に配置された DC から仮想 IP アドレスが割り当てられる。アプリケーションは仮想 IP アドレスによりセッションを確立する。通信開始側の NTM 端末 (イニシエータ) は通信開始時に DC からの指示に従って通信相手の NTM 端末 (レスポンド) との間に実 IP アドレスによる UDP トンネルを構築する。実際の通信は仮想 IP アドレスによるパケットを実 IP アドレスでカプセル化することにより実現する。通信経路上に NAT が存在している場合であっても、NAT によるアドレス変換はカプセル化パケットの外側 IP ヘッダに対して行われるため、ア

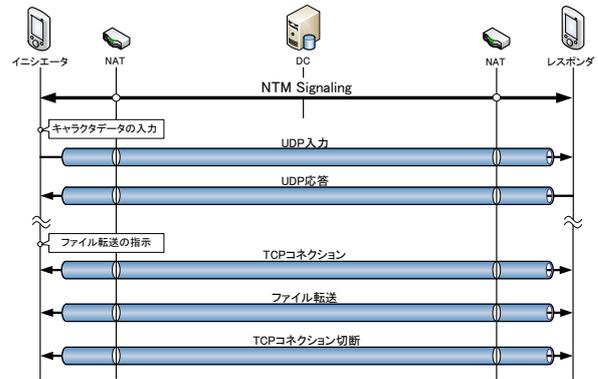


Fig. 1 End To End Chat Sequence

アプリケーションは NAT に影響されることなくエンドツーエンドの通信を行うことができる。NTMobile を適用することにより、ユーザはサーバを経由することなく直接通信を行うことができる。

4 NTMobile を用いたチャット通信

<4・1>概要 イニシエータがレスポンドの FQDN を指定すると NTMobile のシグナリング機能により、両者の間にエンドツーエンドのトンネル経路が構築される。このトンネルを経由して、キャラクターデータやファイル転送を直接実行する。

<4・2>通信方式 Fig. 1 にエンドツーエンド通信によるチャットのシーケンスを示す。NTM Signaling 処理の詳細説明については本稿の本質ではないため省略する。NTM Signaling 処理後、送信データがチャットのキャラクターデータであれば、単一パケットの通信で済むため UDP で通信を行う。この場合、正常に受信できたことを確認するためレスポンドはアプリケーションレベルで応答を返す。送信データがファイル (長データ) であれば TCP で通信を行う。TCP 通信の場合、送達確認は TCP の機能に任せる。

本提案方式ではチャットサーバが不要であるため、サーバの管理が不要で、サーバからの情報漏洩等の心配がない。また、サーバを介する場合に比べ、シーケンスを大幅に簡略化できる。

5 まとめ

本稿では、NTMobile を用いてエンドツーエンドでチャット通信を実現する方式を提案した。今後は、相手端末が起動していない場合や複数人でのチャットを行う場合等について検討する。また、提案手法の実装・性能評価を行っていく。

文 献

[1] 鈴木 秀和, 他: 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.

^{*1} Google の提供する GCM, Apple の提供する APNS などのサービスが存在する。

NTMobileを用いたエンドツーエンド通信 によるチャットアプリケーションの提案

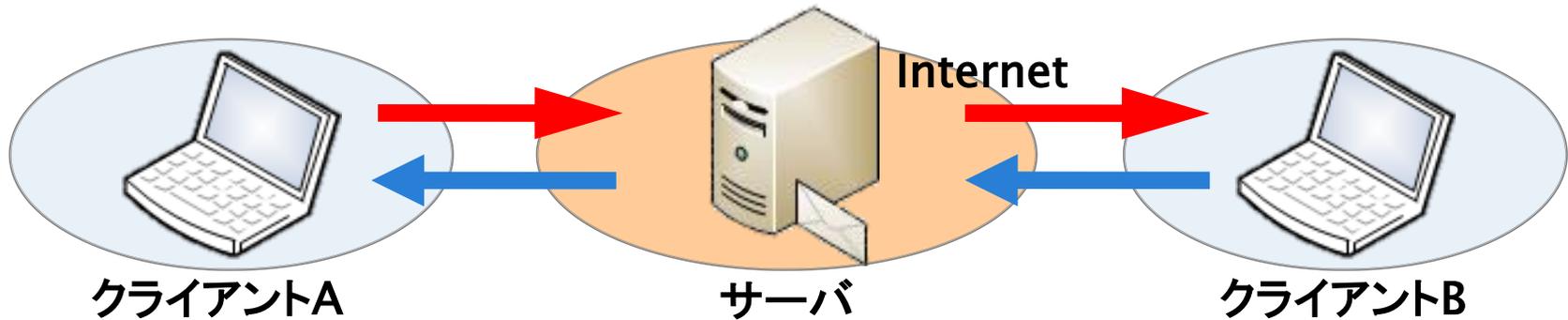
中村 隼大[†] 鈴木 秀和[†] 内藤 克浩^{††} 渡邊 晃[†]

[†]名城大学 理工学部

^{††}愛知工業大学 情報科学部

研究背景

- ▶ ネットワーク技術が急速に発展
- ▶ チャットが重要なコミュニケーションツール
 - クライアントサーバシステムによる実現



- チャットを業務で使用するのが有用
- サーバから情報漏洩する懸念



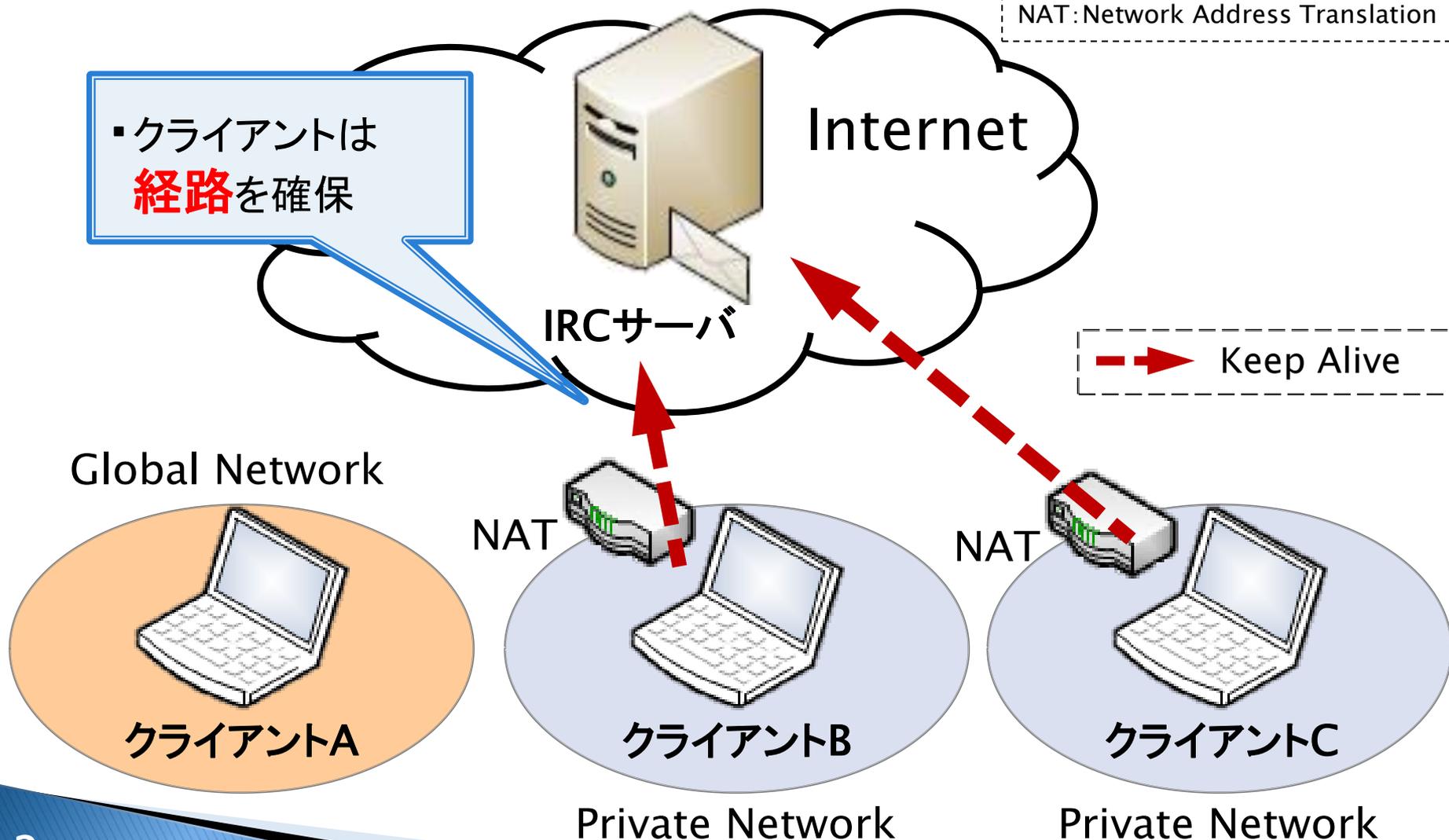
エンドツーエンド通信によるチャットの実現

従来のチャットアプリケーション

▶ クライアントサーバシステム

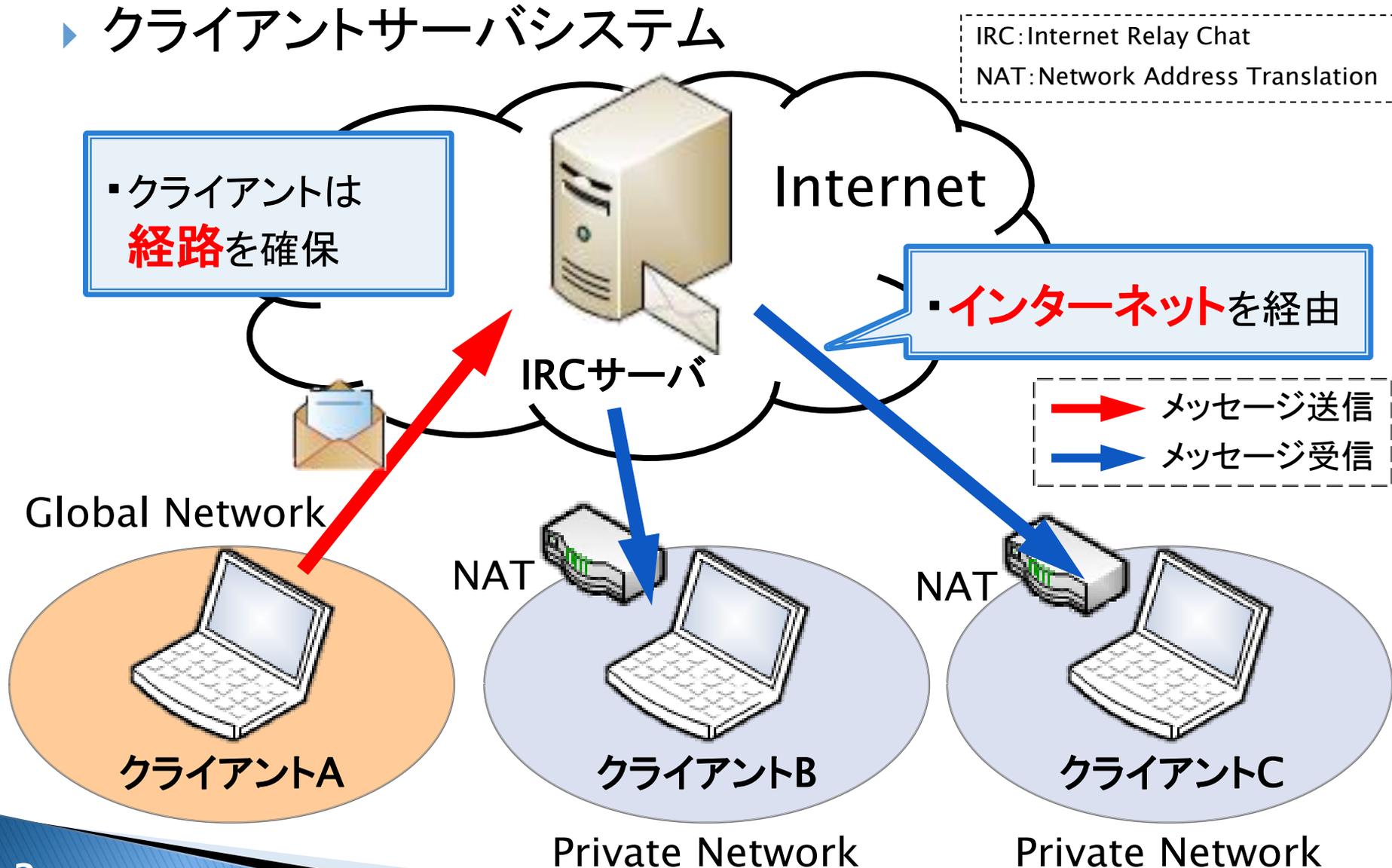
IRC: Internet Relay Chat

NAT: Network Address Translation



従来のチャットアプリケーション

▶ クライアントサーバシステム



CS経由チャットシーケンス



イニシエータ



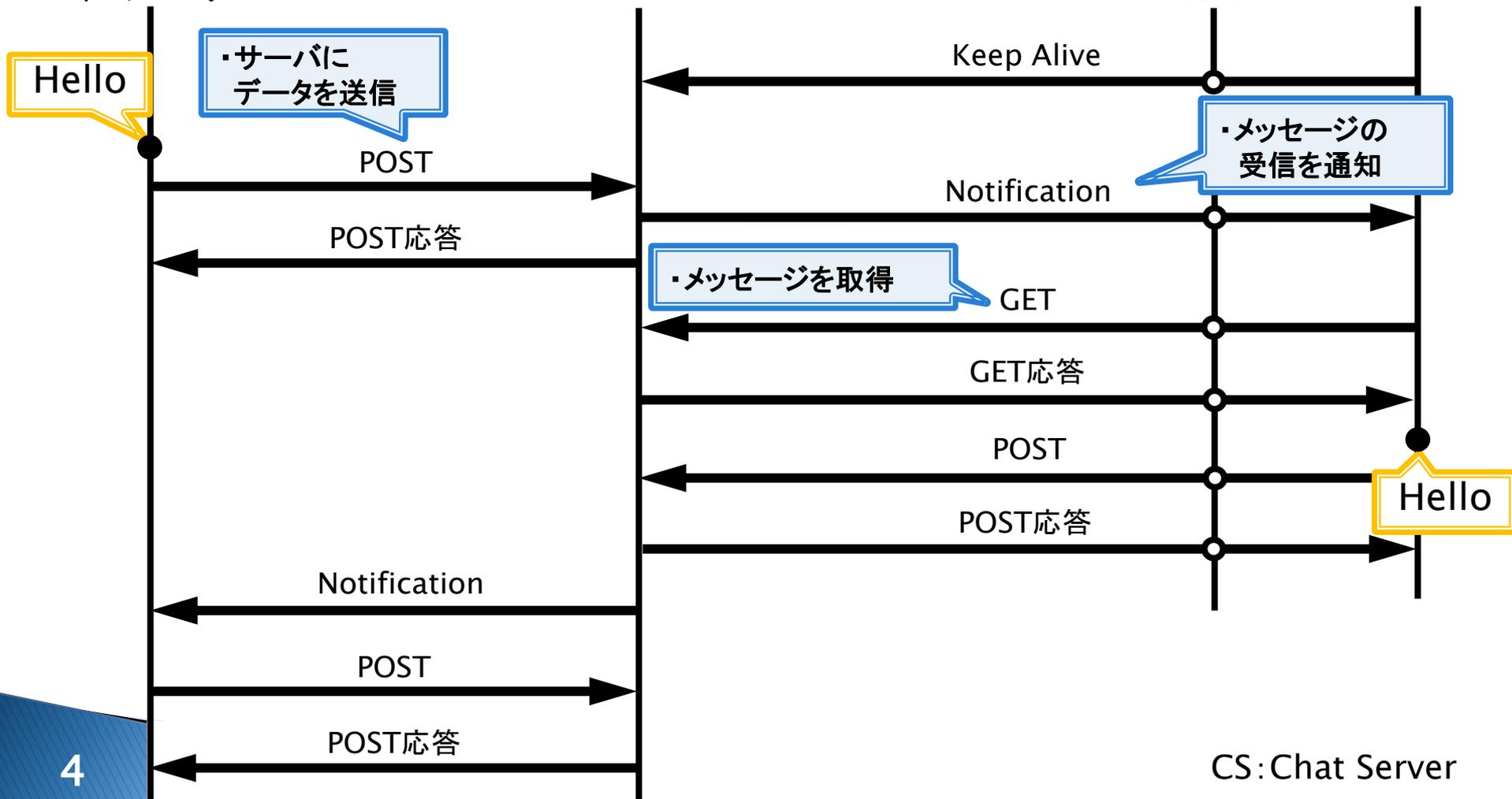
CS



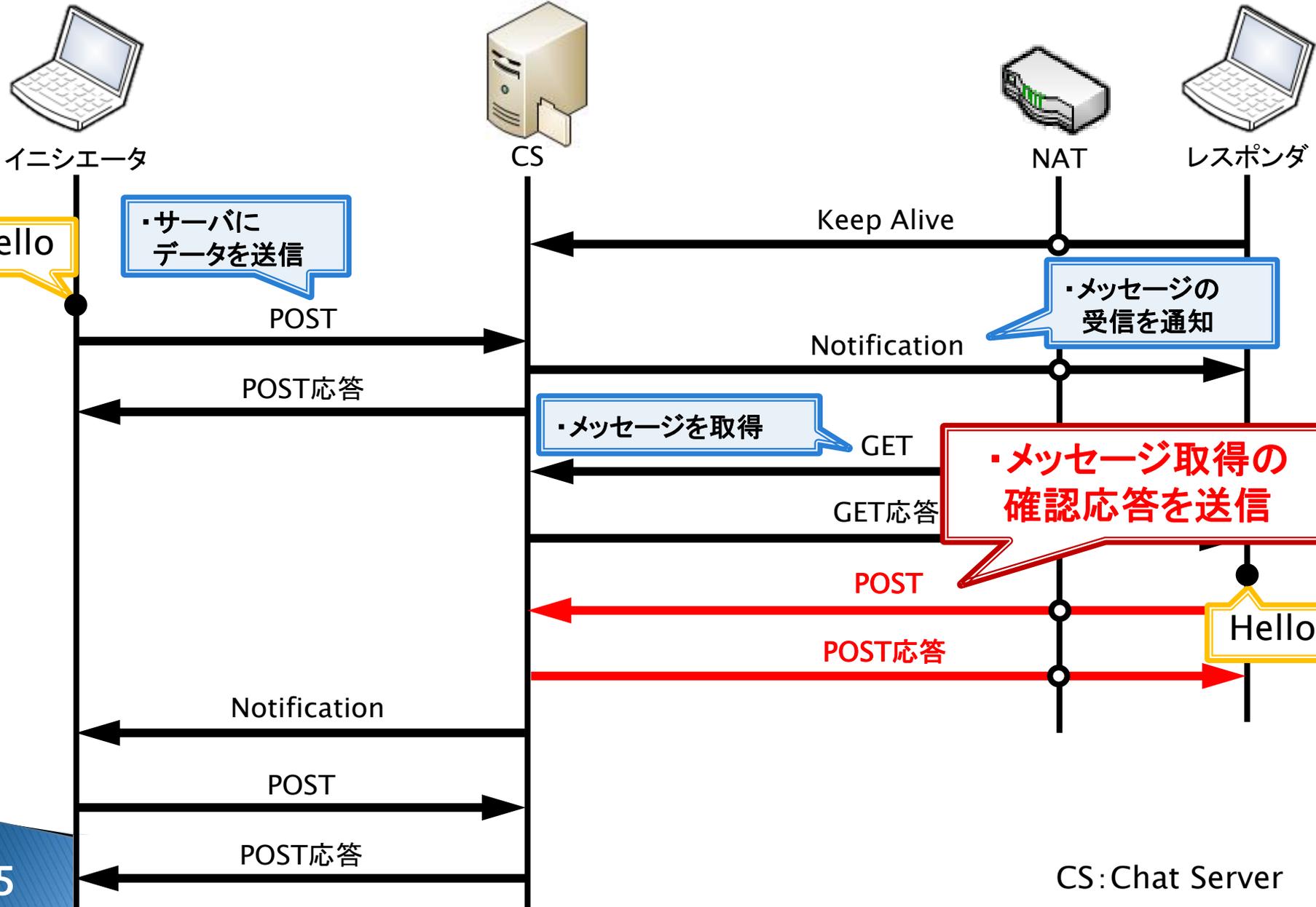
NAT



レスポнда



CS経由チャットシーケンス



CS経由チャットシーケンス



イニシエータ



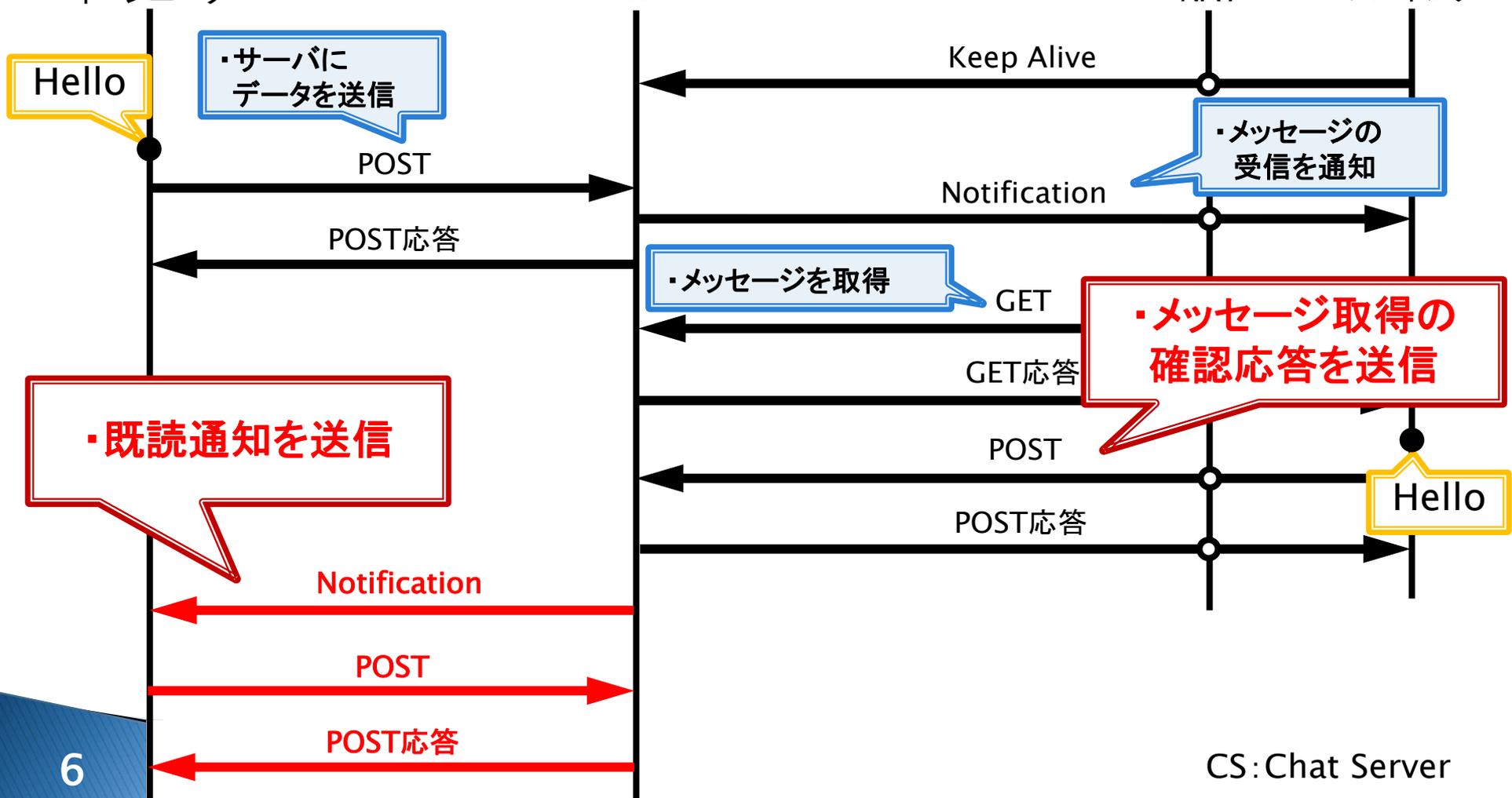
CS



NAT



レスポнда



従来方式の課題と解決策

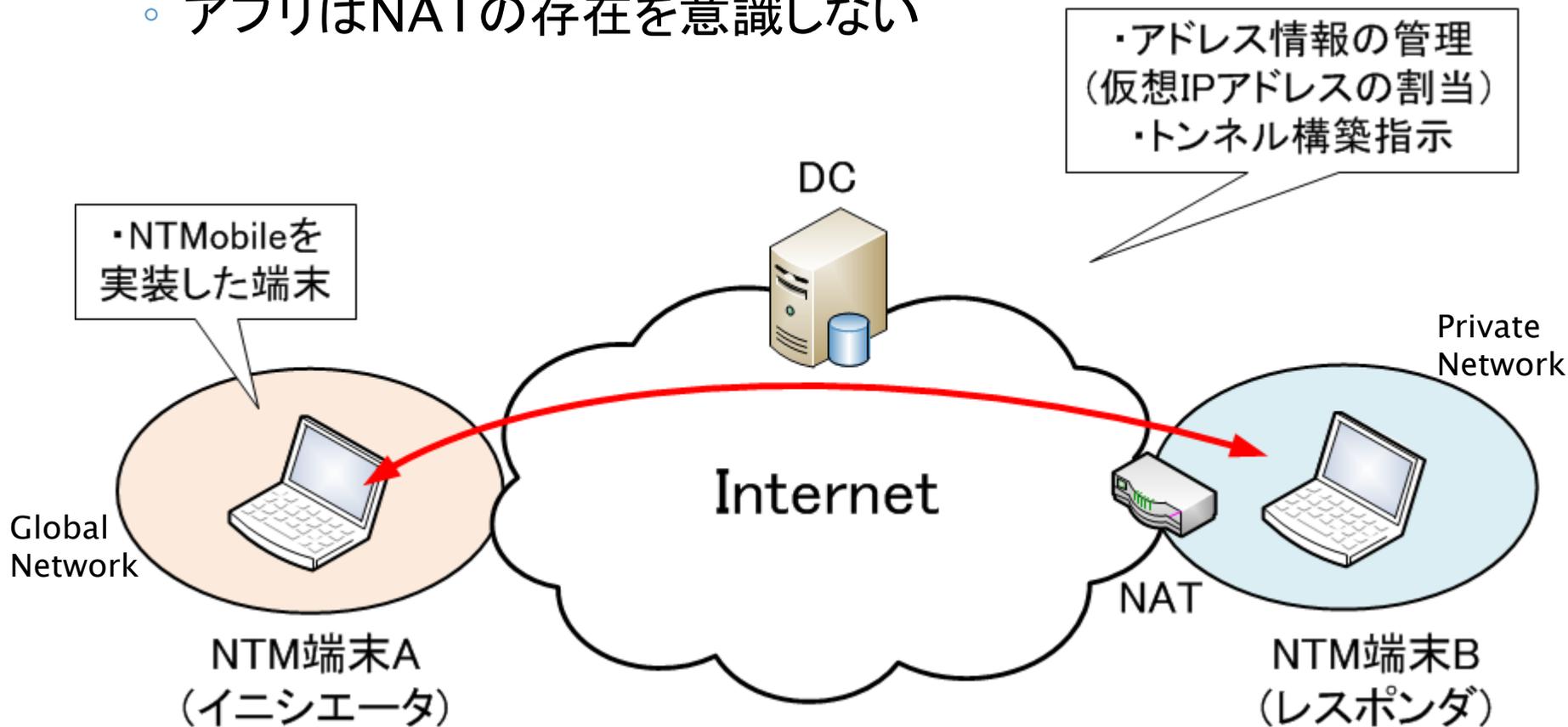
- ▶ サーバから情報漏洩する懸念
 - 管理者が情報を取得
 - 業務での利用は難しい
- ▶ サーバの管理負荷
 - サーバの障害・二重化等に対する管理負荷が大きい
- ▶ トラフィックが大きい
 - シーケンスが複雑
 - メッセージ送信毎、すべての処理を実行



- チャットをエンドツーエンドで実現
- NTMobile上でチャットを実現

NTMobile(Network Traversal with Mobility)

- ▶ エンドツーエンドの通信を行える
 - ネットワーク環境(プライベート, グローバル)を意識せず通信
 - アプリはNATの存在を意識しない



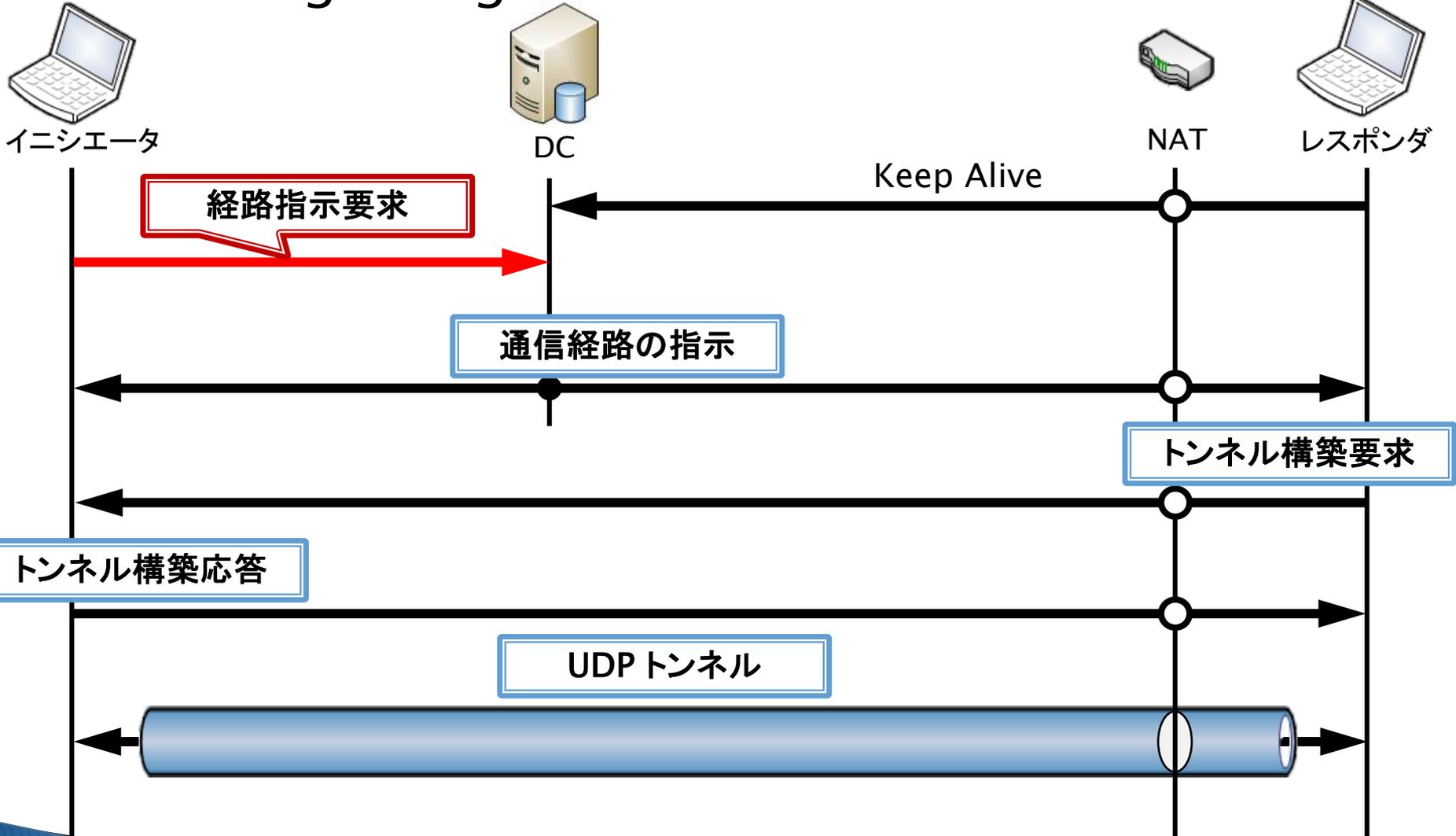
- ▶ アプリケーション間は**仮想IPアドレス**で通信
 - ネットワーク環境が変わっても変化しないIPアドレス
- ▶ 実際の通信は**実IPアドレス**でトンネル通信
 - 実IPアドレスで仮想IPアドレスをカプセル化
- ▶ イニシエータは通信開始時に**DCからの指示**に従ってレスポндаとの間に**トンネル**を構築



エンドツーエンド通信が可能

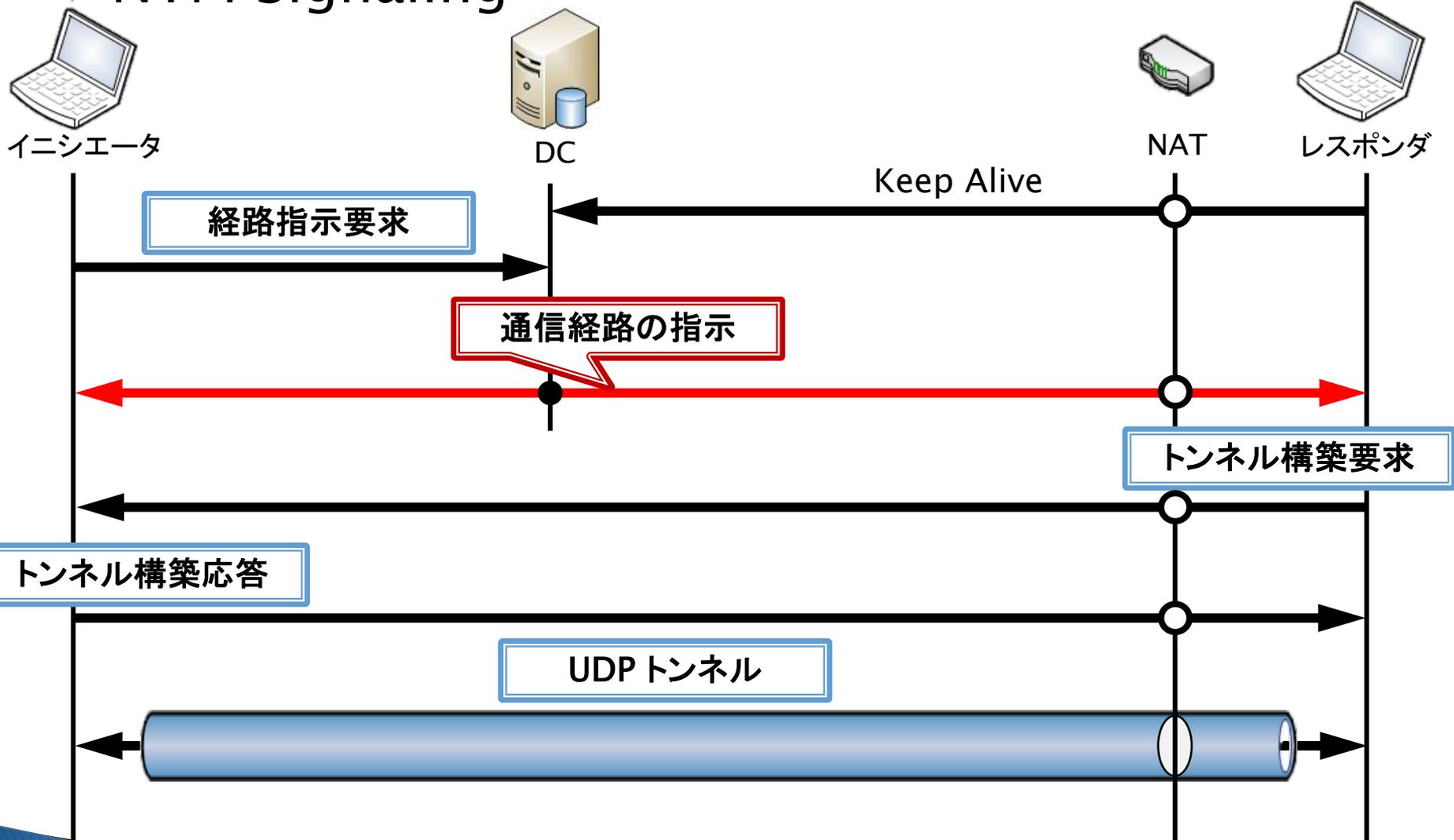
NTMobileを用いたチャット通信

NTM Signaling



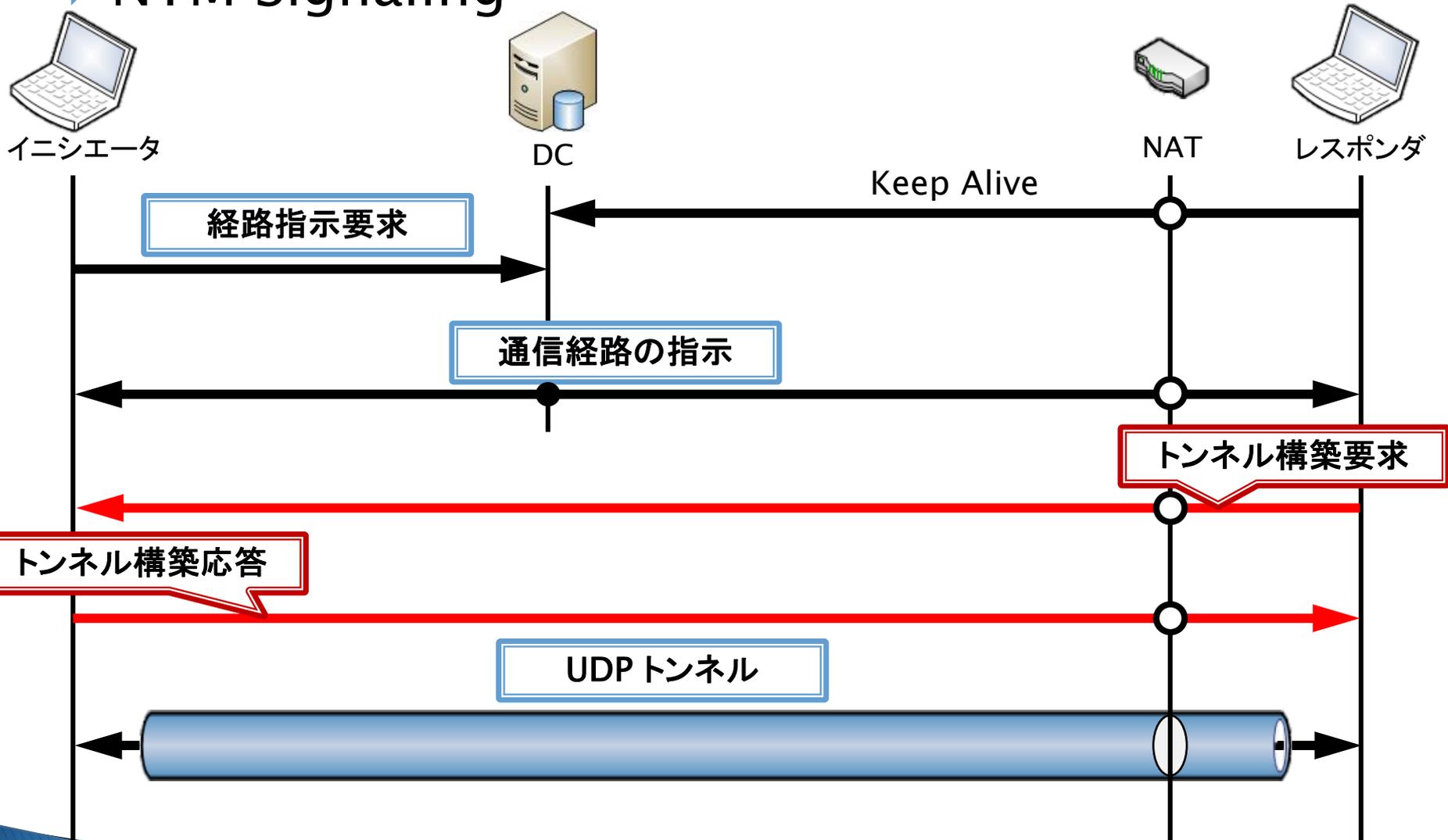
NTMobileを用いたチャット通信

NTM Signaling



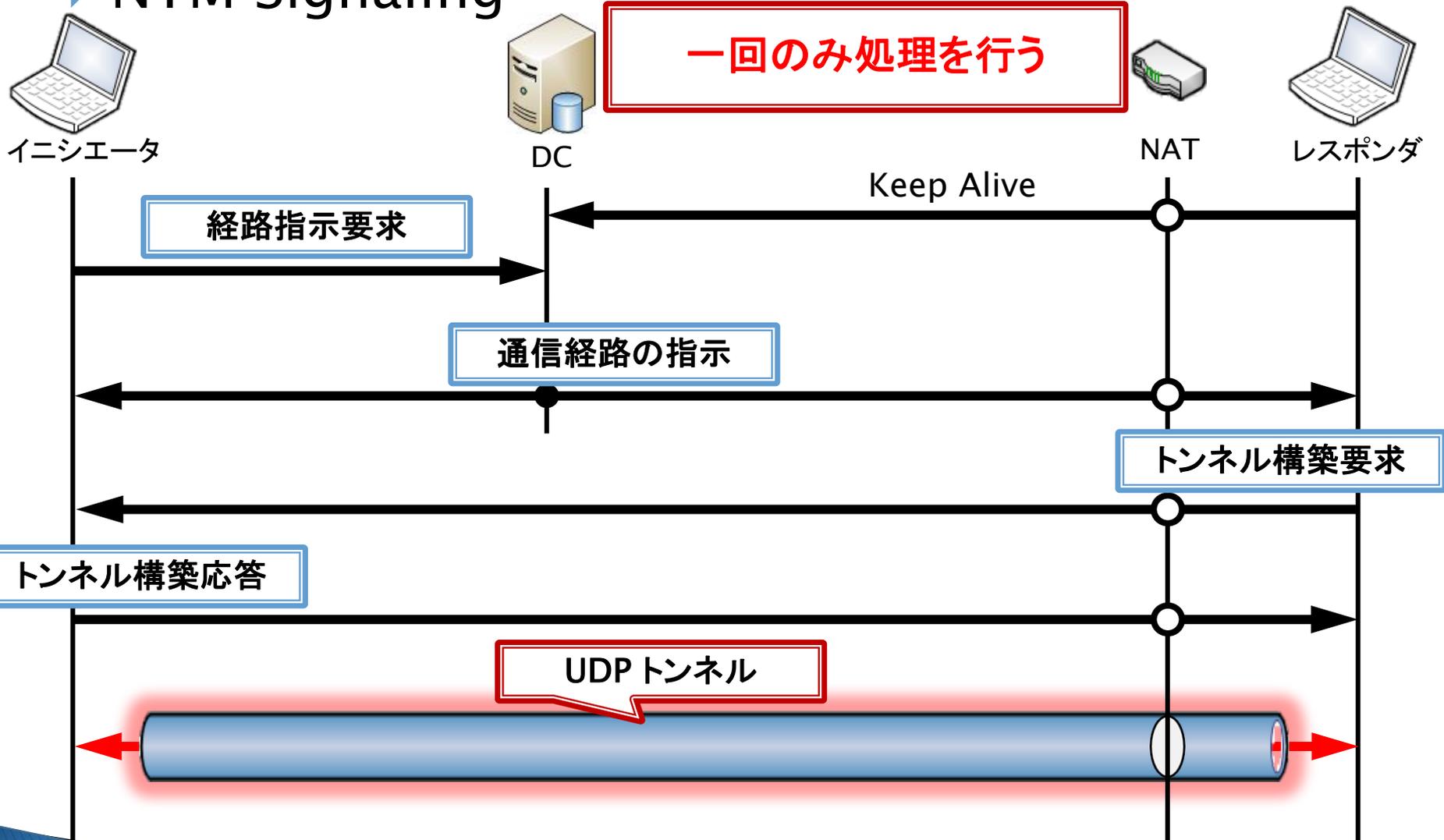
NTMobileを用いたチャット通信

NTM Signaling



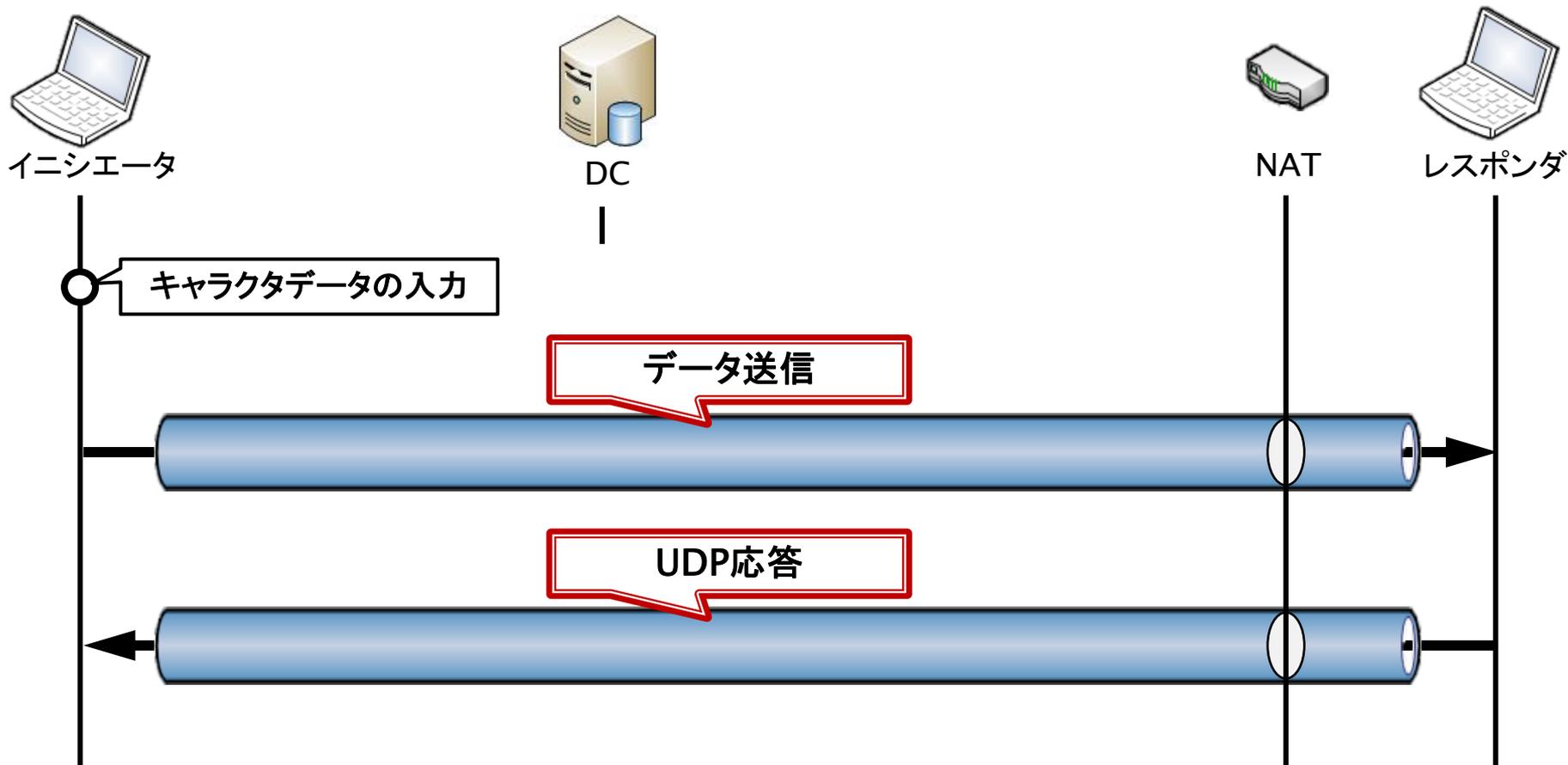
NTMobileを用いたチャット通信

NTM Signaling



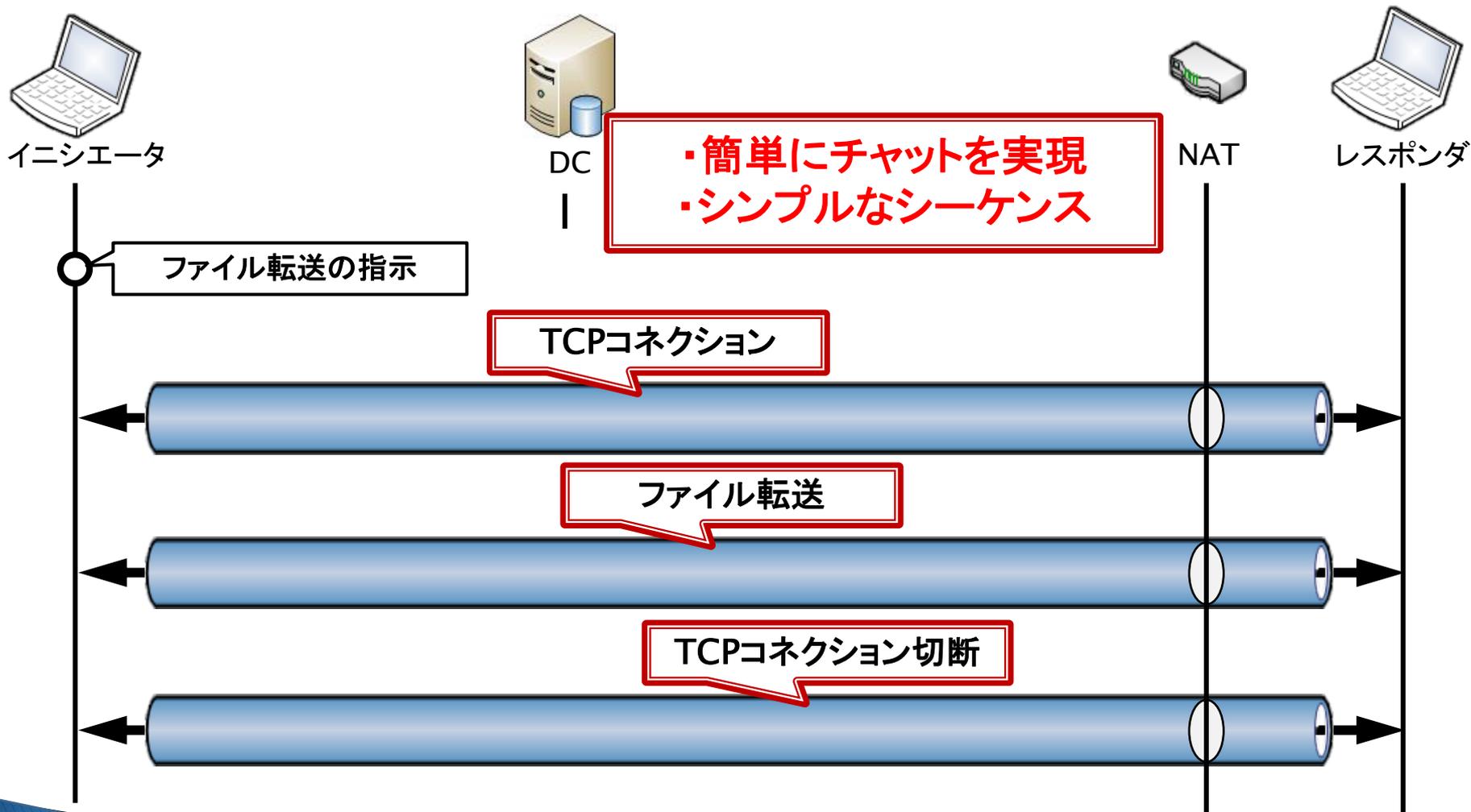
NTMobileを用いたチャット通信

▶ キャラクタデータ



NTMobileを用いたチャット通信

▶ ファイル転送



	従来方式 (CS経由)	提案方式
セキュリティ	× (管理者が情報を取得)	○ (情報漏洩の心配がない)
サーバ管理	× (サーバの障害、 二重化等の管理)	△ (CS不要→DC)
トラフィック	× (シーケンスが複雑、 毎回全ての処理を実行)	○ (シグナリング処理は 初回のみ)

まとめ

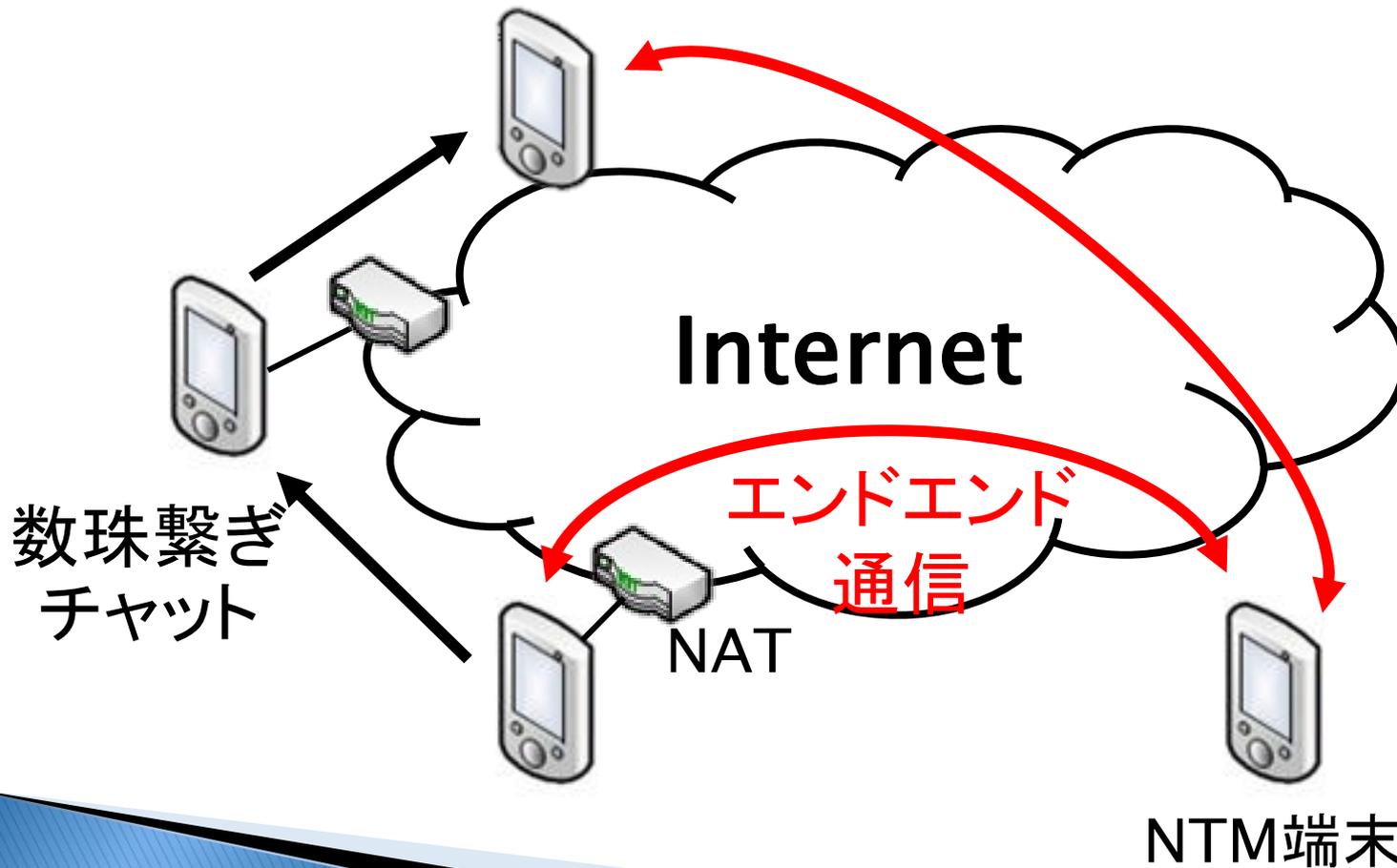
- ▶ NTMobileを用いたエンドツーエンド通信によるチャット通信方式
 - トンネルを構築・経由してキャラクタデータやファイル転送を直接実行
 - 提案方式の有用性を確認
 - サーバの管理が不要
 - サーバから情報漏洩の心配がない
 - トラフィックの軽減

- ▶ 今後の予定
 - 提案手法の実装・性能評価
 - 相手端末が起動していない場合、大規模なチャットを行う場合の処理について検討

補足資料

今後の改善点

- ▶ 相手端末が起動していない場合
 - 従来の方式で通信
- ▶ 複数人での大規模なチャットを行う場合



RS(Relay Server)

- ▶ イニシエータ・レスポндаが共にNAT配下に存在する場合の通信の中継
- ▶ エンドツーエンドで通信を行える
 - CSと異なりRSは経路するのみ

