カプセル化パケットのフラグメント処理の検討

渡邉 悠雅 †*，尾久 史弥 †，鈴木 秀和 †, 内藤 克浩 ‡, 渡邊 晃 († 名城大学，‡ 愛知工業大学)

Study on Fragment Processing of the Capsulated Packet

Yuga Watanabe†, Fumiya Ogyu†, Hidekazu Suzuki†,Katsuhiro Naito‡, Akira Watanabe† (†Meijo University, ‡Aichi Institute of Technology)

1 はじめに

インターネットは自由な情報発信と情報収集が可能で、我々の生活に欠かせないインフラの一つとなっている。このような環境でネットワークセキュリティ技術が重要になっている。

カプセル化は VPN の構築などに利用されネットワークセキュリティを実現する上で重要な技術である。しかし、カプセル化を行うことでパケットサイズが増加し、フラグメントが発生して通信の性能低下を引き起こす可能性がある。そこで本稿ではエンド端末でカプセル化する場合と、ネットワークでカプセル化する場合のそれぞれに対してフラグメントにどう対処すべきかを検討した。

2 カプセル化の実現方法

IPv6 では、MTU は PMTUD(Path MTU Discovery) で調整される。本稿では IPv4 のフラグメント対策について述べる。

カプセル化処理を行う装置はエンド端末の場合 (以下エンド型) と、途中のネットワーク装置が行う場合 (以下ネットワーク型) がある。Fig. 1 にエンド型の、Fig. 2 はネットワーク型の構成を示す。実線はデータフローを表しカプセル化パケットが生成されるまでの流れを表す。

エンド型では一般に TUN/TAP インターフェースを利用し、カーネルが生成した IP パケットをカプセル化アプリケーションに引き渡す手法がとられる。カプセル化アプリケーションが所定の処理を行った後、実インターフェースに対してパケットを送信するとカプセル化パケットが生成される。一方、ネットワーク型は一般のエンド装置から送信された IP パケットを中継装置の raw ソケットで受信する。受信された IP パケットは中継機器内で所定の処理を行った後、別のインターフェースからカプセル化したパケットとして送信される。

一般に性能を確保するために、OS では MTU をネットワークの最大サイズとしている。カプセル化をするとパケットサイズが増加するため、MTU を上回ってしまう可能性がある。その際はカプセル化アプリケーションがフラグメント処理を行う必要があるが、この処理によって性能の低下を引き起こしてしまう。

3 フラグメント処理の方法

カプセル化パケットに対する処理方法はエンド型とネットワーク型でそれぞれ異なる。エンド型ではカプセル化アプリケーションの初期化モジュールで MTU を設定することができる。TCP の MSS は、MTU が決まると自動的に定まる。この処理を行うために、Fig. 1 で斜線で示される初期化モジュールのプログラムを変更し、点線で示される初期化処理にて MTU 設定を行っている。これに対し、ネットワーク型では一般にエンド装置の設定を変えられないので別のフラグメント対策が必要である。そこで TCP と UDP でそれぞれ以下の対策を取る。TCP では、3way handshake コネクション確立時で行われるオプションにて MSS 値を書き換える。通信経路上のカプセル化装置がエンド装置に代わってネゴシエーションを実行することにより MSS 値が適切に調節され、TCP の最大のパケット効率を引き出すことができる。一方、UDP はネゴシエーションがないため OS でフラグメント処理を行う。ネットワーク型では以上の処理を行うために、Fig. 2 に斜線で示される MSS 調整モジュールを実装する。

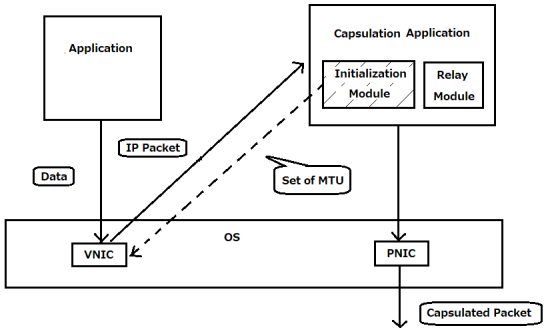


Fig. 1 End type configuration and fragment measure

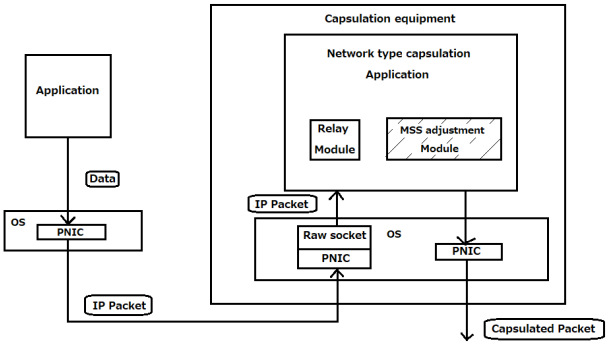


Fig. 2 Network type configuration and fragment measure

4 まとめ

パケット通信時にカプセル化処理を行う場合、パケットサイズ増加によりフラグメントが発生する．これに対し、エンド型ではアプリケーション内で MTU を書き換えることにより対処する．ネットワーク型は、TCP と UDP で処理を分ける。TCP では中継装置が 3way handshake の内容を書き換え、UDP では OS でフラグメント処理を行う．

文 献

[1] 尾久．他：第 79 回全国大会講演論文集,2017(1),367-368,2017.
[2] 稲垣．他：第 80 回全国大会講演論文集,2018(1),215-216,2018.

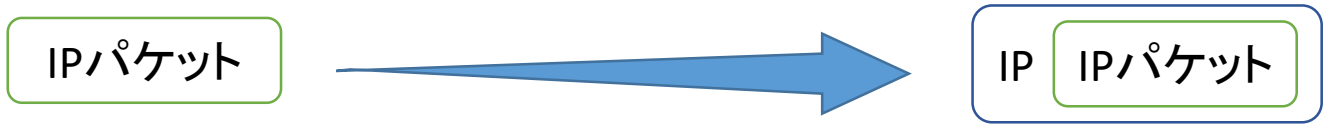
カプセル化パケットの フラグメント処理の検討

渡邉 悠雅†*, 尾久 史弥†, 鈴木 秀和†, 内藤 克浩‡, 渡邊 晃†
(†名城大学, ‡愛知工業大学)

研究背景

- インターネットの普及でIPネットワークは生活必須に
  - ➢高度なセキュリティ技術が重要
    - ➢**カプセル化**は重要なネットワークセキュリティ技術の一つ

- カプセル化
  - オブジェクト指向でオブジェクト内部のデータや型、振る舞いを隠蔽する技術
    - ➢例えば、IPパケットを1つのデータ捉えパケットを生成する



研究背景

- カプセル化が引き起こすフラグメント
  - ➢パケットサイズ増加でMTUを超えたパケットが生成される

フラグメントがスループットの低下を引き起こす可能性がある

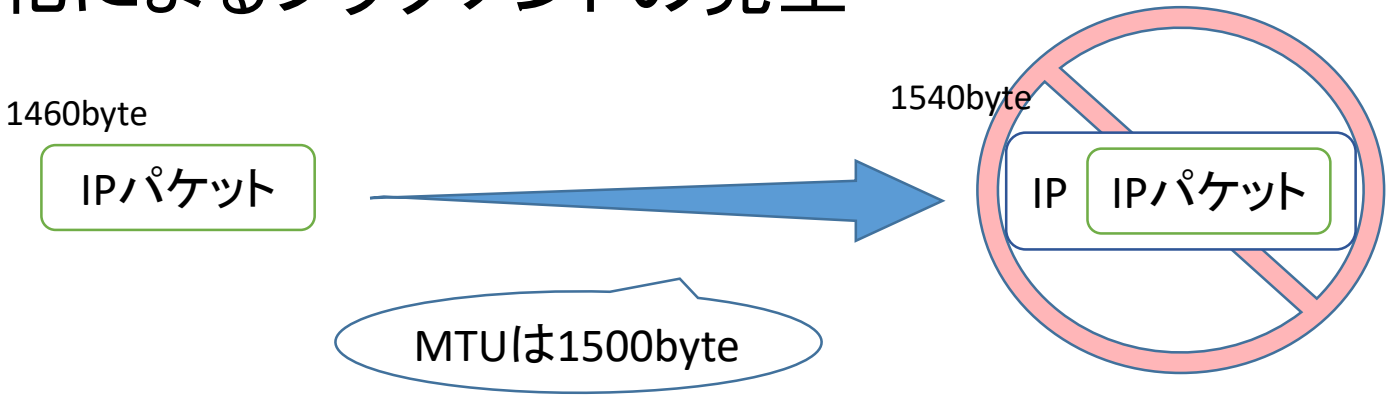
- MTU
  - ➢ネットワーク装置やホストが送受信できるIPヘッダを含めた最大サイズ

MSS
TCPベースの考えで、一度に受信できるデータ(セグメント)の最大長

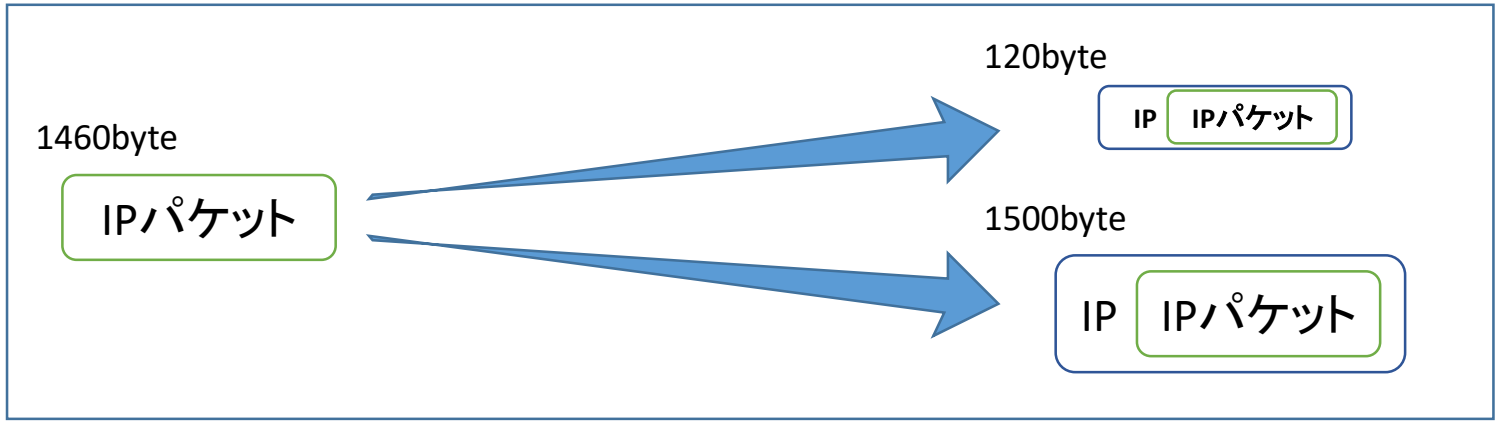
- フラグメント
  - MTUを上回るパケットが送信されるときパケットを分割する処理

研究背景

- カプセル化によるフラグメントの発生



フラグメント発生



研究目的

- フラグメントによるスループット低下を抑えることを検討
  - エンド端末がカプセル化処理する場合（以下エンド型とする）
  - ネットワーク装置がカプセル化処理する場合（以下ネットワーク型とする）

- 実際にフラグメント対処処理を行い性能測定
  - 今回の検証はエンド型を中心に行った
  - カプセル化処理システムとしてNTMobileを使って検証する
    - 他のカプセル化システムでも検証可能だが今回はNTMobileを使用した

NTMobile

- IPv4・IPv6混在環境で移動透過性と通信接続性を実現する技術
  - NTMobile通信はアプリケーションレベルで実現可能
  - NTMobileライブラリによるカプセル化

    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

- TUN型NTMobile（エンド型）
  - エンド端末でカプセル化処理をするNTMobile

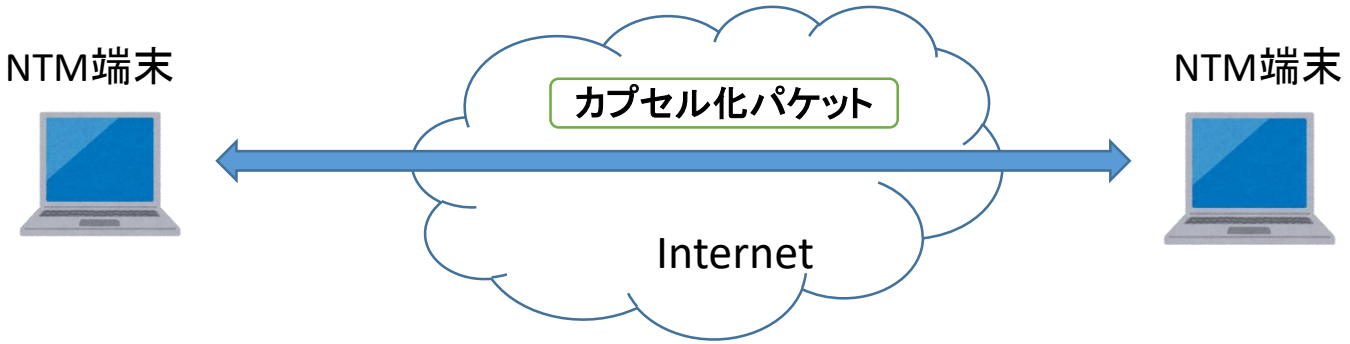
    稲垣.他:第 80 回全国大会講演論文集,2018(1),215-216,2018.

- アダプタ型NTMobile（ネットワーク型）
  - ネットワーク中継装置でカプセル化処理をするNTMobile

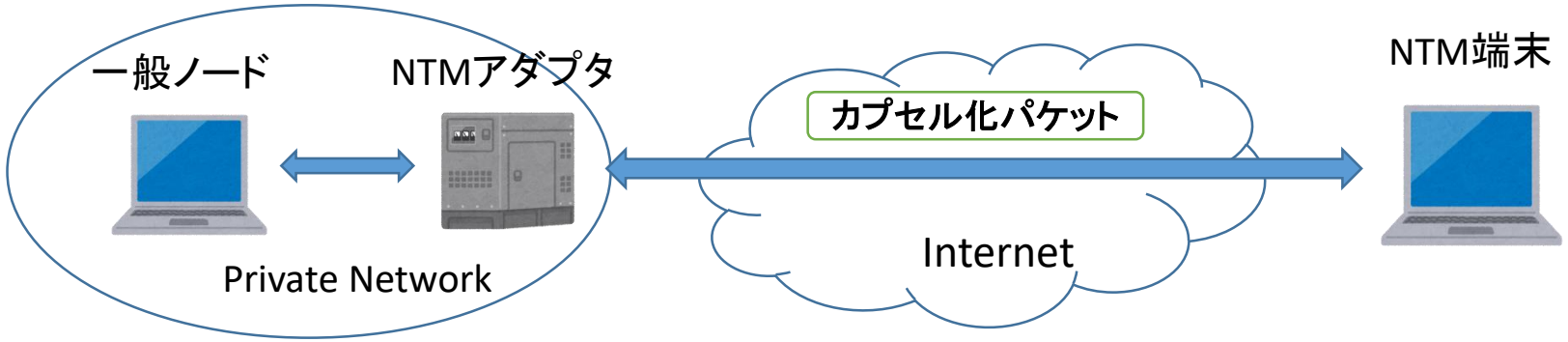
    尾久.他:第 79 回全国大会講演論文集,2017(1),367-368,2017.

NTMobile

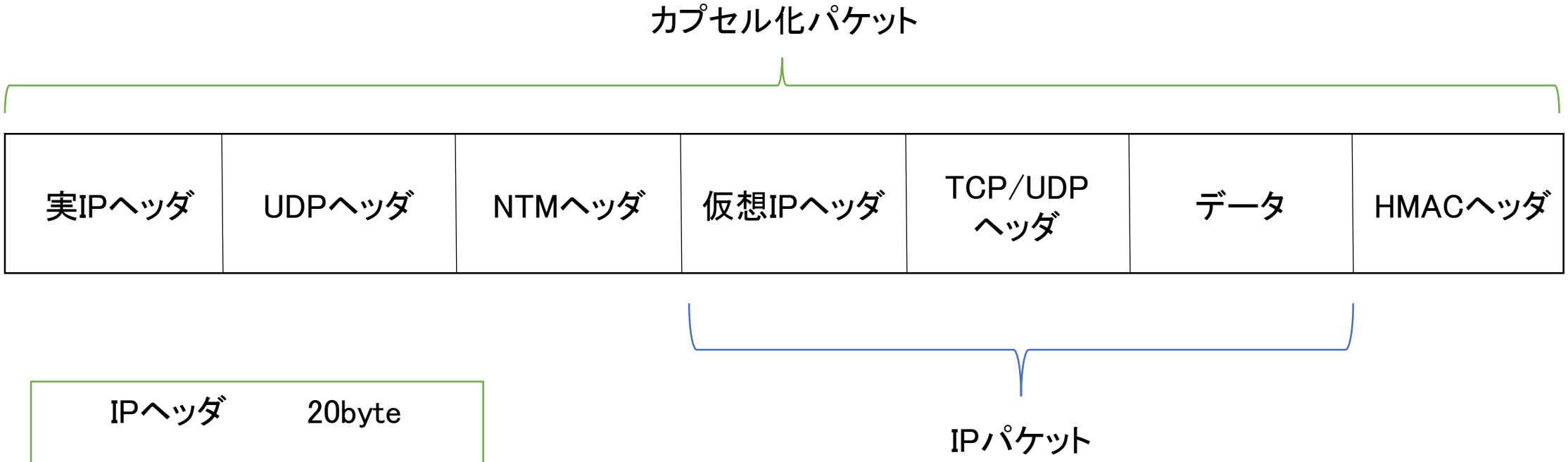
- TUN型NTMobile（エンド型）



- アダプタ型NTMobile（ネットワーク型）



NTMobile

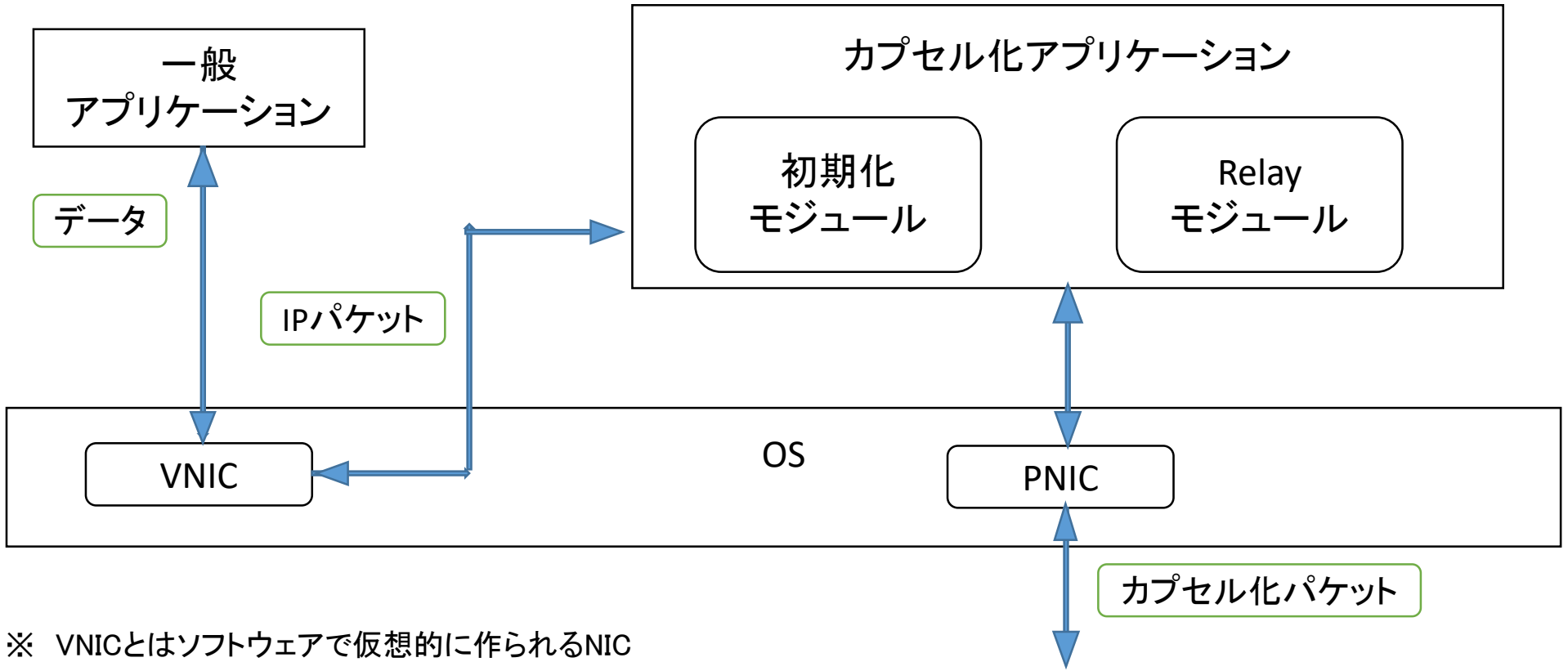


IPヘッダ	20byte
UDPヘッダ	8byte
NTMヘッダ	36byte
HMACヘッダ	16byte

※ カプセル化パケットはIPパケットより80byte大きい

カプセル化処理 -エンド型-

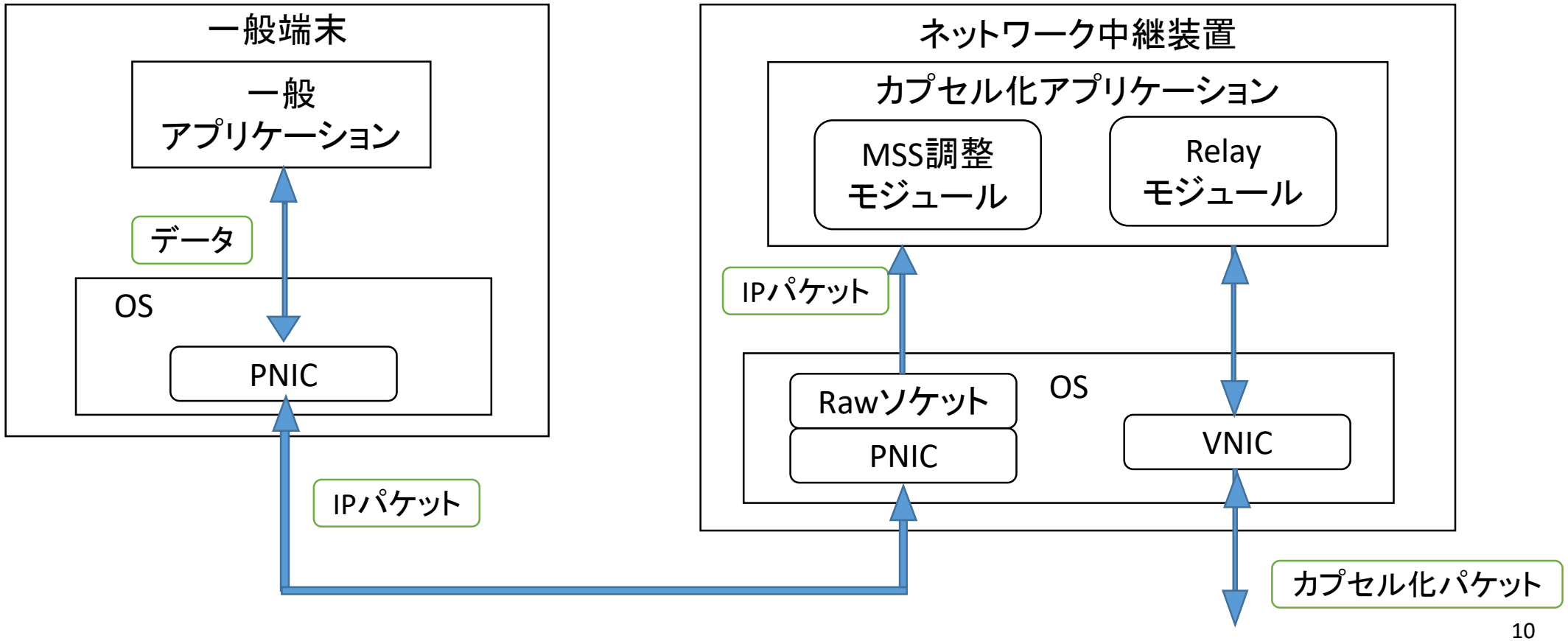
- エンド端末のアプリケーションでカプセル化を行う
  - ➢今回検証使用するのはTUN型NTMobile



※ VNICとはソフトウェアで仮想的に作られるNIC

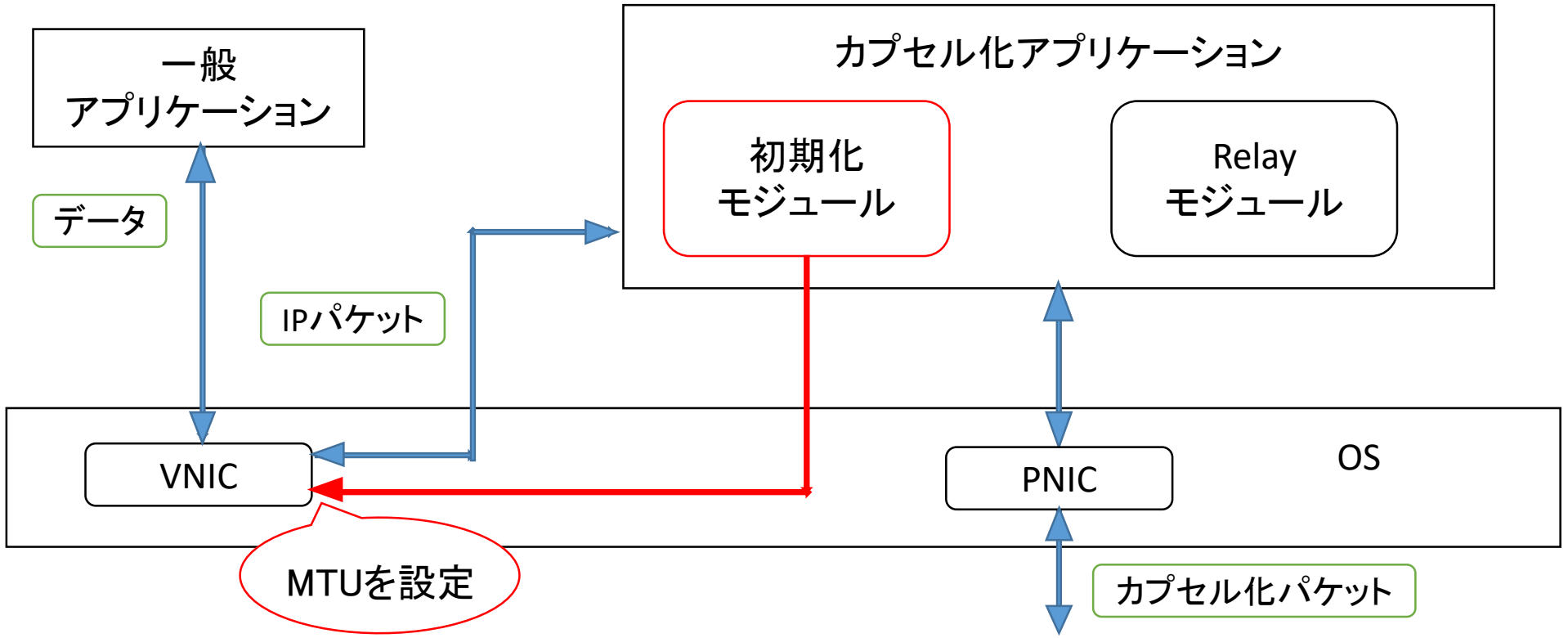
カプセル化処理 –ネットワーク型–

- ネットワーク装置がカプセル化処理



フラグメント対処　–エンド型–

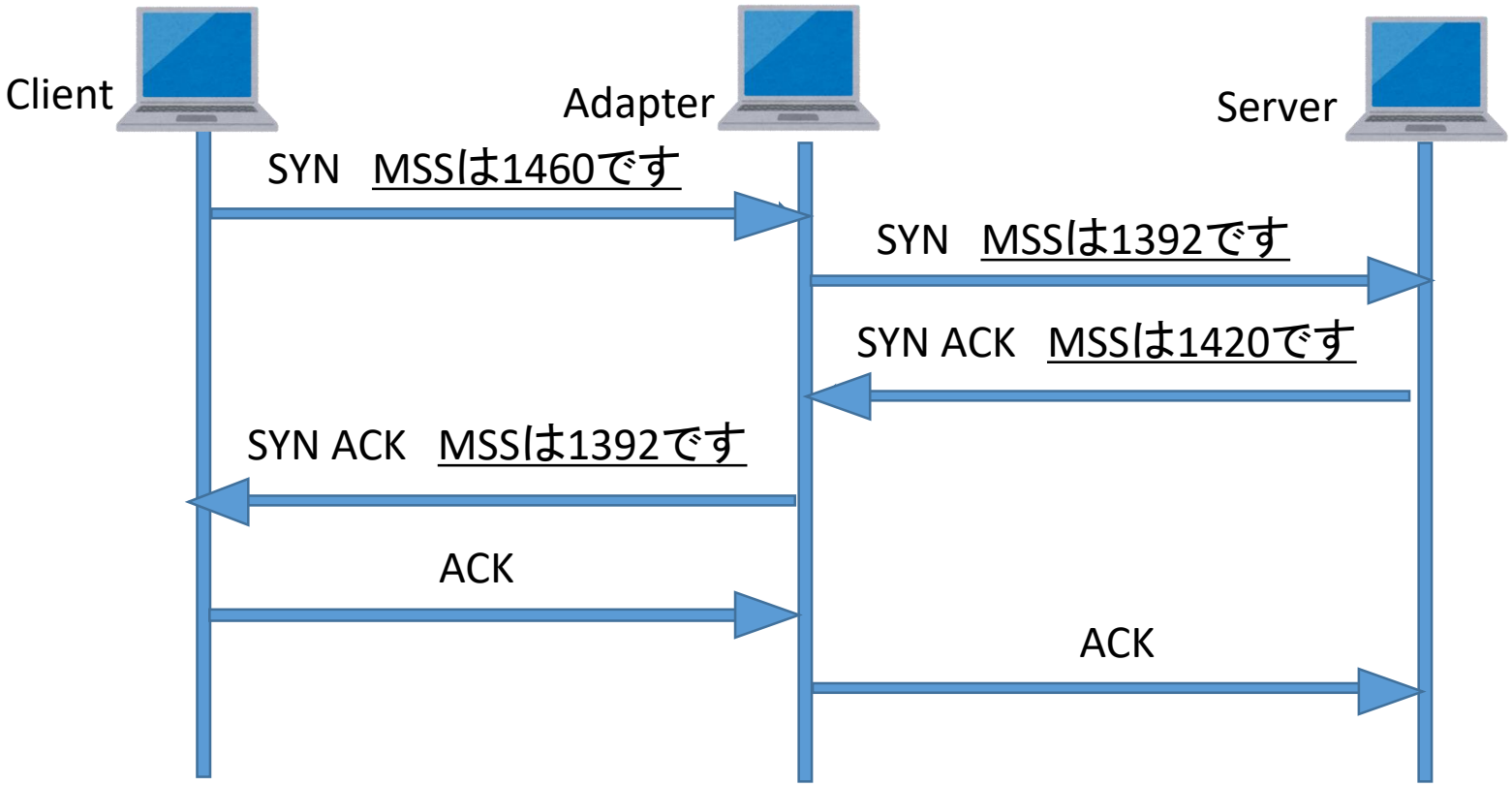
- カプセル化アプリケーションの初期化モジュールを変更
  - ➢初期化モジュールでMTUを設定すると自動的にMSSが定まる



[RFC 6691](informational) D. Borman, "TCP Options and Maximum Segment Size (MSS)", https://tools.ietf.org/html/rfc6691, (IETF),updated by RFCs 879, 2385 , July 2012.

フラグメント処理 –ネットワーク型–

- TCPでは3way hand shakeのコネクション確立時にMSSを変更
  - ➢SYN/SYN ACKパケットでオプション部にあるMSSを変更し実現



フラグメント処理 –ネットワーク型–

- UDPではネットワーク装置のOS側でフラグメント対処
  - アプリケーションレイヤで作ったパケットをOSでフラグメントする

※ NTMobileはNTMmobileフレームワークというカプセル化アプリケーションを使用

  - 1500バイトのパケットをそのままカプセル化アプリケーションでカプセル化
  - パケット送信時にOS側でフラグメント対処処理
  - 受信側はOSでフラグメント再構築

動作環境　–エンド型–

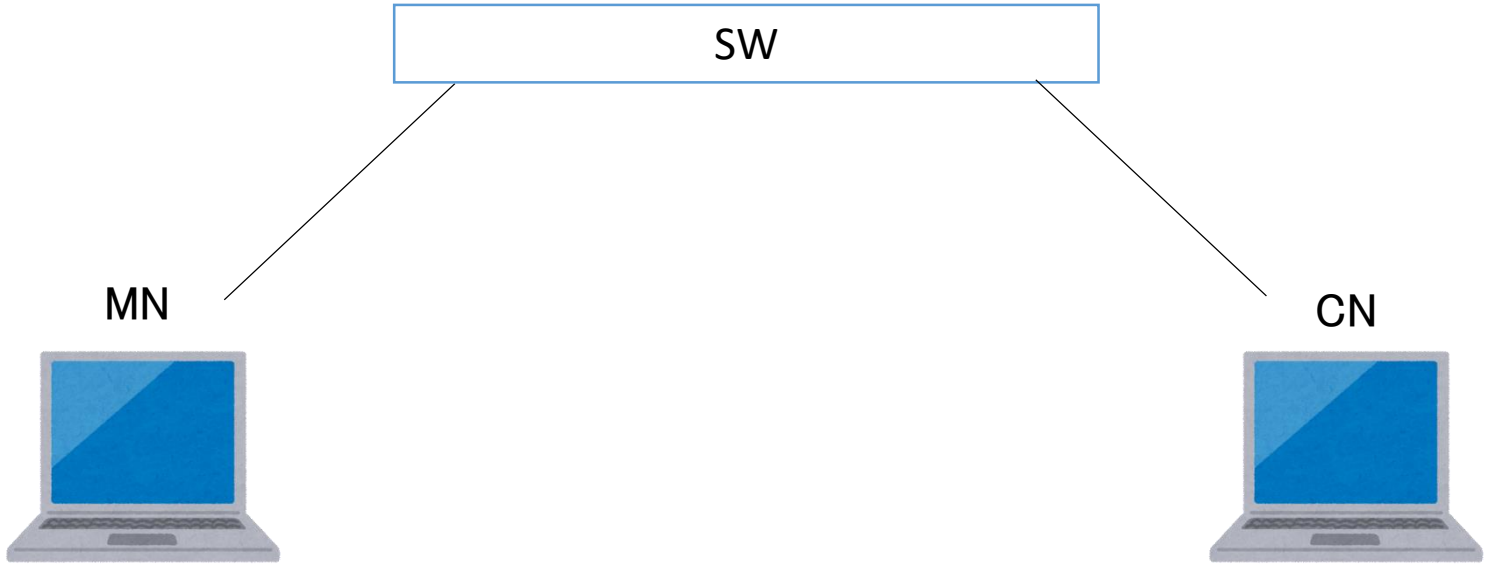
- 検証で使用する装置のスペック
- スループットの計測はフリーソフトであるiperfを用いた

MN、CN	
OS	ubuntu 14.04 LTS (32bit)
プロセッサ	Intel Core i5-2520M CPU @2.50GHz ×4
メモリ	1.9GiB

※　MNは送信側のノード、CNは受信側のノードとする

動作環境　–エンド型–

- 検証を行ったネットワーク構成
- インターネット通信等を想定している
  - 性能測定のためスイッチングハブを使いローカルで検証



性能評価 –エンド型–

- TCPとUDPそれぞれで5回測定した平均のスループット

	MTUを設定	OSでフラグメンテーション
TCP	342Mbit/s	×
UDP	296Mbit/s	143Mbit/s

TCPをOSでフラグメンテーションする結果は今回は計測できなかった

考察

- UDPはフラグメンテーションすることで倍近くスループットが低下
  - 分割したパケットの分送信する回数が増えたからだと考えられる

  → 分割するのでTCPでも同様な結果がみられるのではないかと予想される

まとめ

- カプセル化をするとパケットサイズが増加しフラグメント発生
  - ➢エンド端末でカプセル化
    - アプリケーション内でMTUを書き換える
  - ➢ネットワーク装置がカプセル化
    - TCPは3way hand shakeのオプションを書き換え
    - UDPは中継装置のOS側でフラグメント対処

補足

- IPv6ではPass MTU Discoveryがあるためフラグメントはない