# Mintオペレーティングシステムにおける仮想ネットワークインタフェースによるOSノード間通信の実現

平成30年2月16日

岡山大学 工学部 情報系学科
小倉 伊織

# 研究背景

＜Mint＞

- 1台の計算機上で複数のLinuxを走行
- 計算機資源を各Linuxで分割/占有
- メモリの一部はすべてのLinuxで共有可能（共有メモリ）

＜Mintにおける既存のOSノード間通信＞

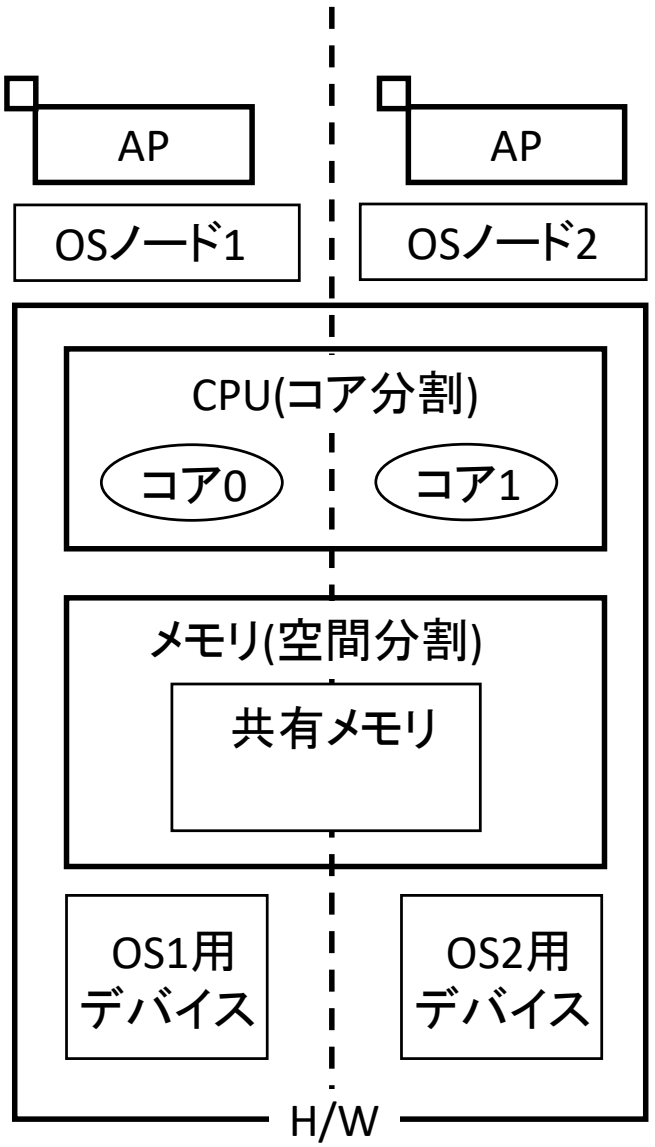
（手法1）NICハードウェアを用いた通信

問題: 通信可能なOSノード数に限界有

（手法2）共有メモリを介した通信

問題: 既存APの改変が必要

共有メモリを介してEthernet互換で通信する仮想ネットワークインタフェース(VNI)を実現



# 研究背景

＜Mint＞

- 1台の計算機上で複数のLinuxを走行
- 計算機資源を各Linuxで分割/占有
- メモリの一部はすべてのLinuxで共有可能（共有メモリ）

＜Mintにおける既存のOSノード間通信＞

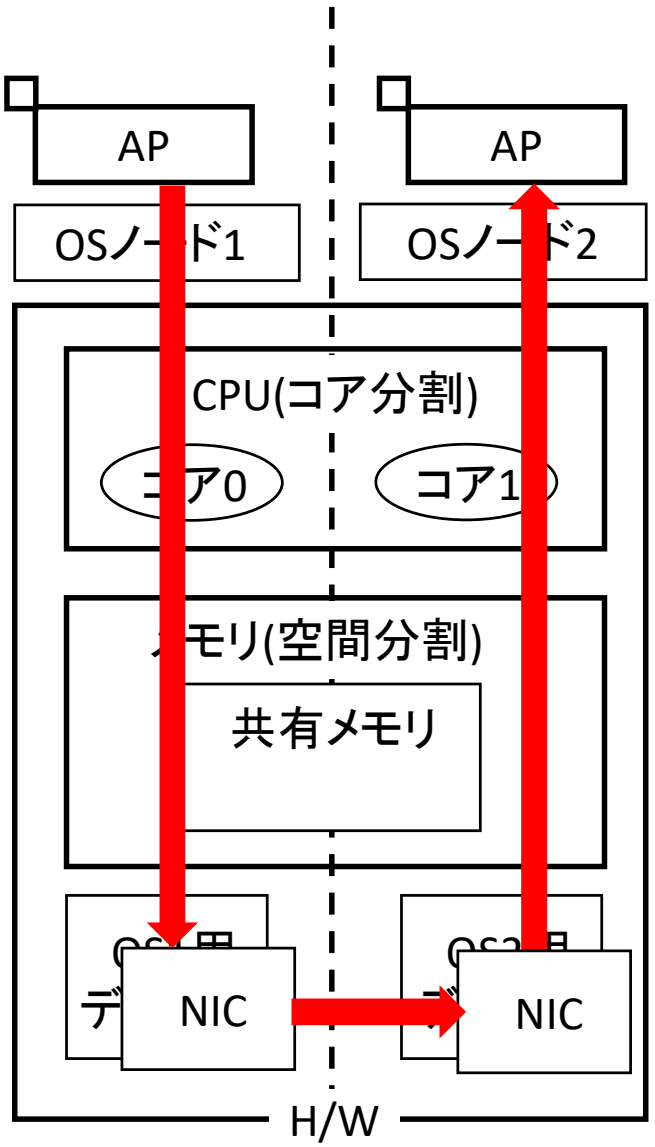
（手法1）NICハードウェアを用いた通信

問題: 通信可能なOSノード数に限界有

（手法2）共有メモリを介した通信

問題: 既存APの改変が必要

共有メモリを介してEthernet互換で通信する仮想ネットワークインタフェース(VNI)を実現



# 研究背景

＜Mint＞

- 1台の計算機上で複数のLinuxを走行
- 計算機資源を各Linuxで分割/占有
- メモリの一部はすべてのLinuxで共有可能（共有メモリ）

＜Mintにおける既存のOSノード間通信＞

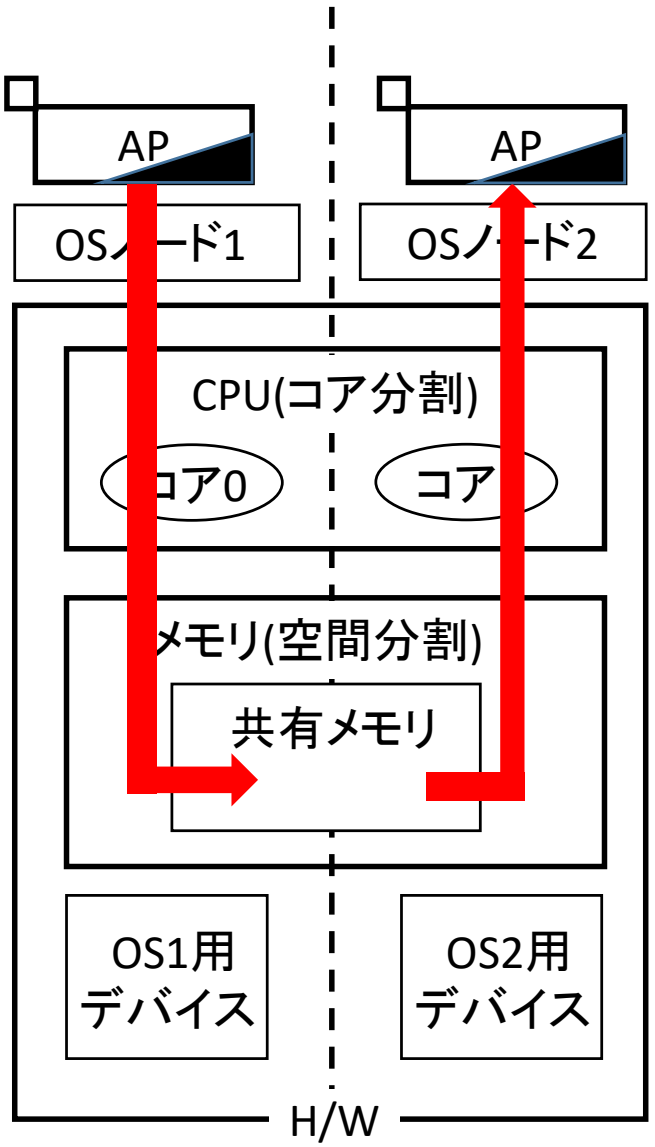
（手法1）NICハードウェアを用いた通信

問題: 通信可能なOSノード数に限界有

（手法2）共有メモリを介した通信

問題: 既存APの改変が必要

共有メモリを介してEthernet互換で通信する仮想ネットワークインタフェース(VNI)を実現



# 研究背景

＜Mint＞

- 1台の計算機上で複数のLinuxを走行
- 計算機資源を各Linuxで分割/占有
- メモリの一部はすべてのLinuxで共有可能（共有メモリ）

＜Mintにおける既存のOSノード間通信＞

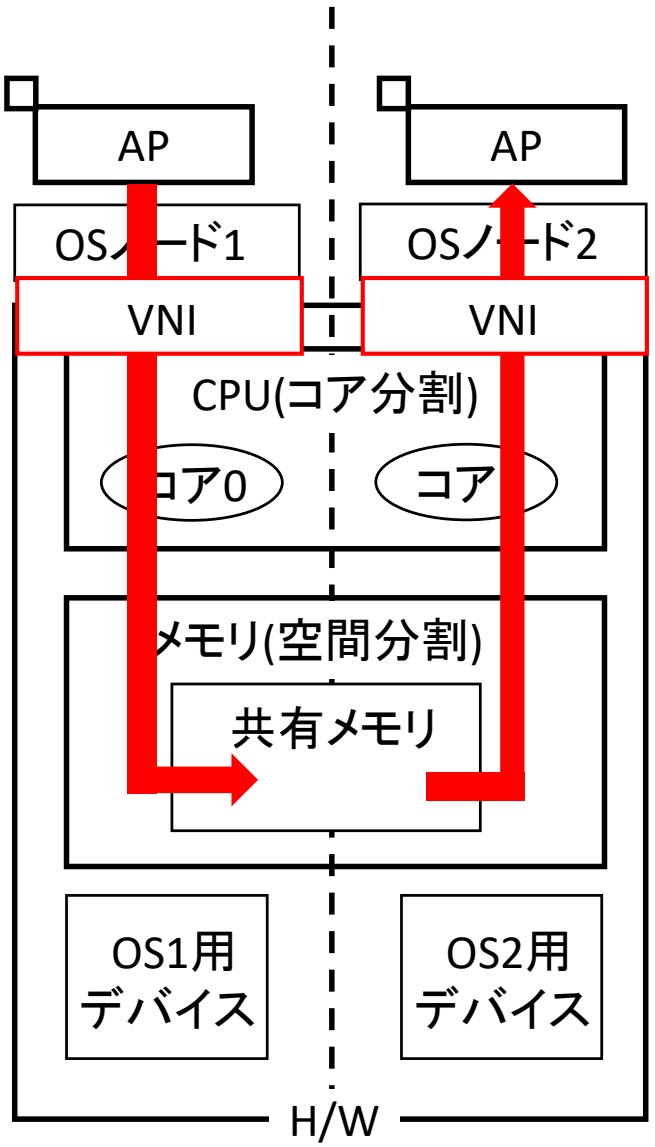
（手法1）NICハードウェアを用いた通信

問題: 通信可能なOSノード数に限界有

（手法2）共有メモリを介した通信

問題: 既存APの改変が必要

共有メモリを介してEthernet互換で通信する仮想ネットワークインタフェース(VNI)を実現



# 課題

(1) 送受信バッファの構成の検討

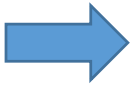
(構成1) パケットを送信先毎に別々の送受信バッファで管理する構成

(構成2) すべてのパケットを同一の送受信バッファで管理する構成

どちらの構成が提案手法に適しているか検討

(2) 排他制御すべき操作の検討

不要な操作を排他制御すると，ロックを保持する時間が長くなる



ロック獲得のための待ち時間の長大化

排他制御する操作が最小となるように検討

# 送受信バッファの構成の検討

(構成1) パケットを送信先毎に別々の送受信バッファで管理する構成

(構成2) すべてのパケットを同一の送受信バッファで管理する構成

| | 利点 | 欠点 |
|---|---|---|
| 構成1 | 処理工数が小さい | 送受信バッファの利用率が低い |
| 構成2 | 送受信バッファの利用率が高い | 処理工数が大きい |

＜比較＞

- (構成2)の方が処理工数が大きい

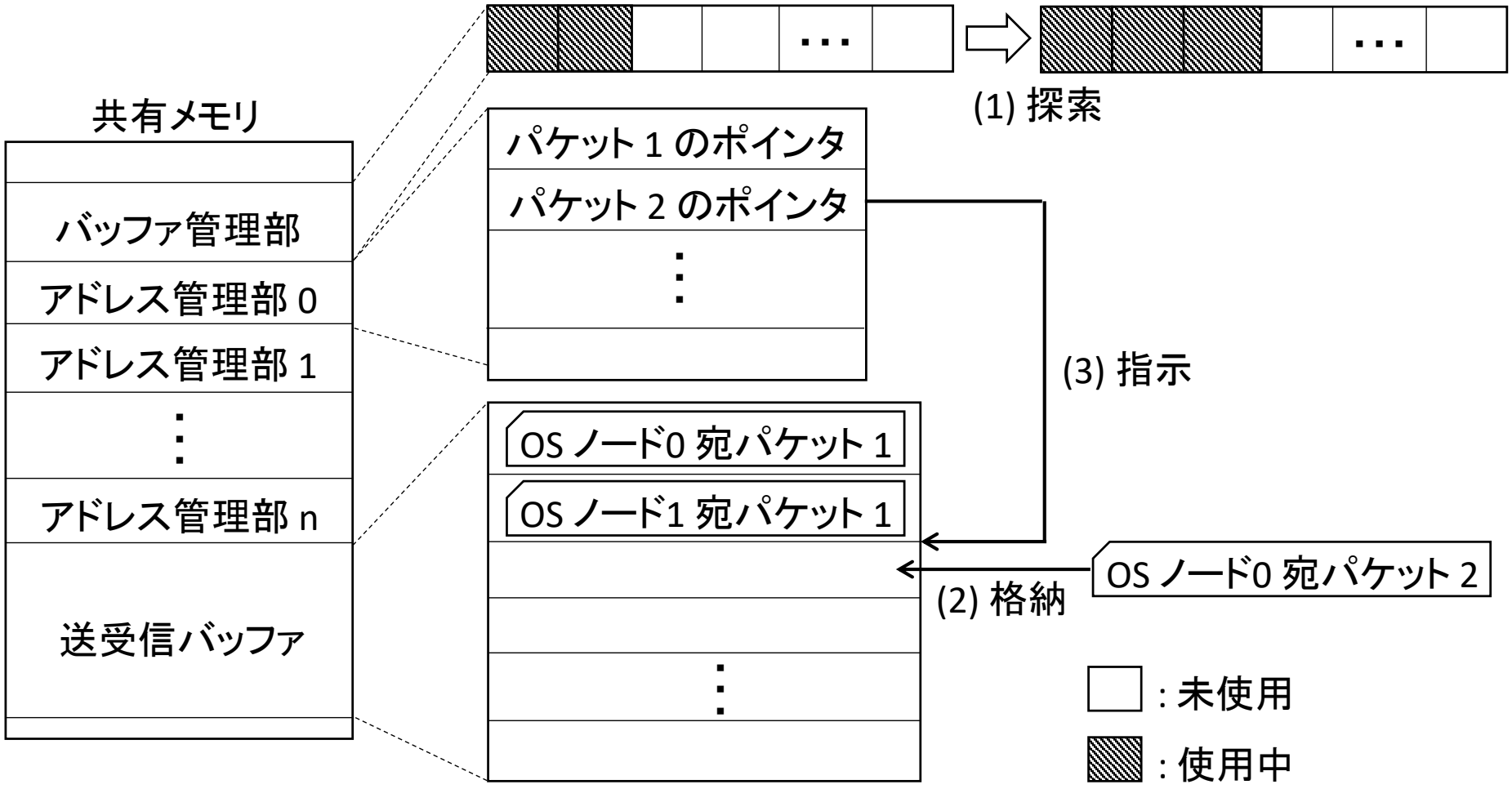
  ただし，処理時間の増加は全体に比べて極めて小さい

- (構成2)の方が送受信バッファの利用率が高い

提案手法には(構成2)を採用する

# (構成2)の処理流れ

(構成2) すべてのパケットを同一の送受信バッファで管理する構成



# 排他制御すべき操作の検討

＜共有資源を参照/更新する操作＞

(1) バッファ管理部の操作
(2) アドレス管理部の操作
} 排他制御が必要

∵ 複数のOSノードが同時に操作する

(3) 送受信バッファの操作 } 排他制御は不要

∵ (1)と(2)が排他制御されている場合，
複数のOSノードが同じ領域を同時に操作しない

# 評価

＜評価項目＞

(1) アプリケーション改変の有無

(2) 提案手法の実装によるコード量

(3) 性能評価

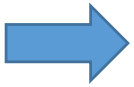
(A) 通信におけるレイテンシ

(B) 単位時間あたりの通信量

(C) パケットロス率

＜評価結果＞

(1) 既存APの改変なしでTCP/IPで通信可能であることを確認



既存APの改変なしで通信可能

(2) 追加したコード量は4ファイルで計474行であり，
Mint カーネル全体の約0.004%と極めて小さい

# まとめ

＜実績＞

(1) 送受信バッファの構成の検討

(2) 排他制御すべき操作の検討

(3) 提案手法の実装

(4) 提案手法の評価

　(A) アプリケーション改変の有無

　(B) 提案手法の実装によるコード量

＜今後の課題＞

(1) 性能評価

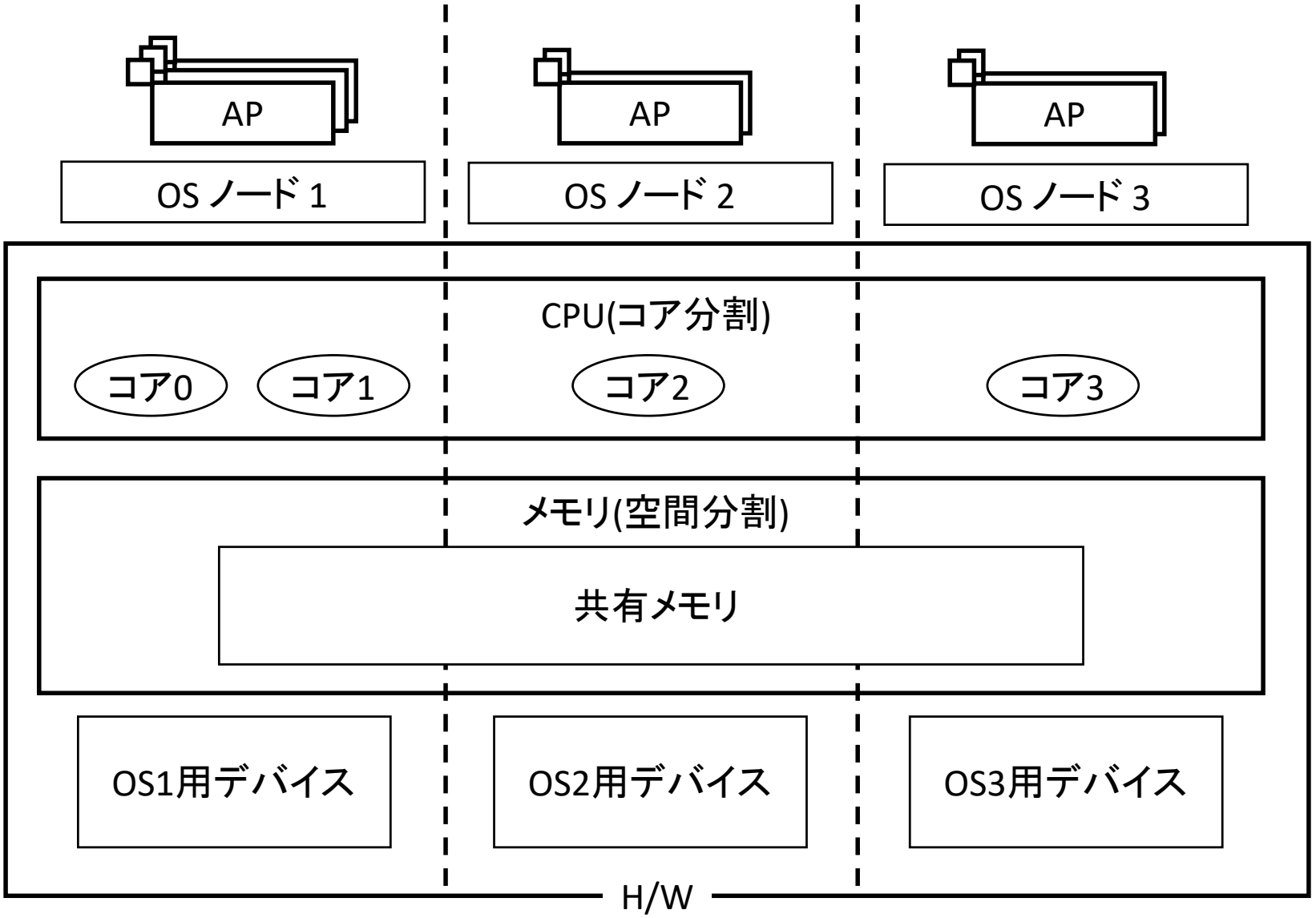
　(A) 通信におけるレイテンシ

　(B) 単位時間あたりの通信量

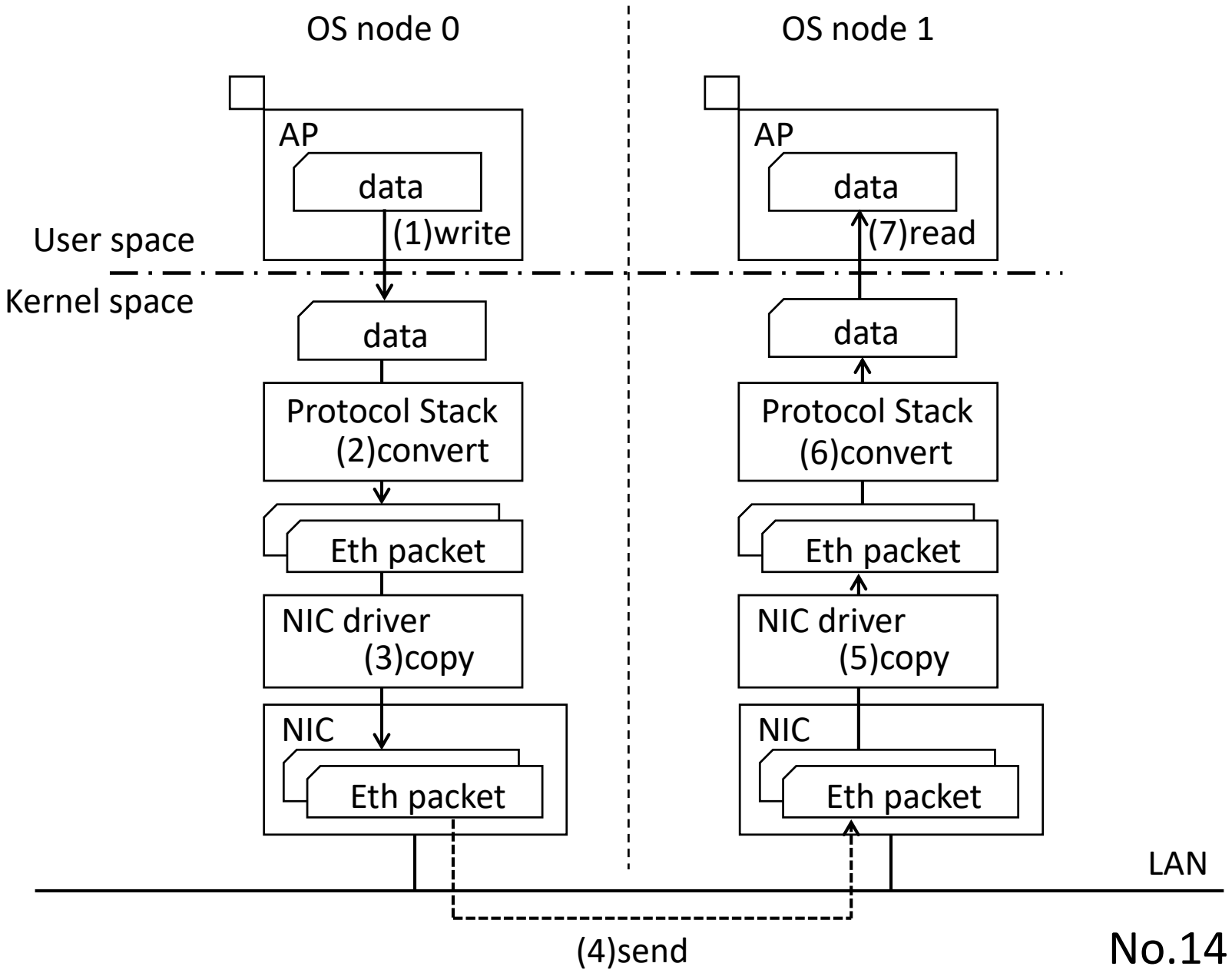
　(C) パケットロス率

# 予備スライド

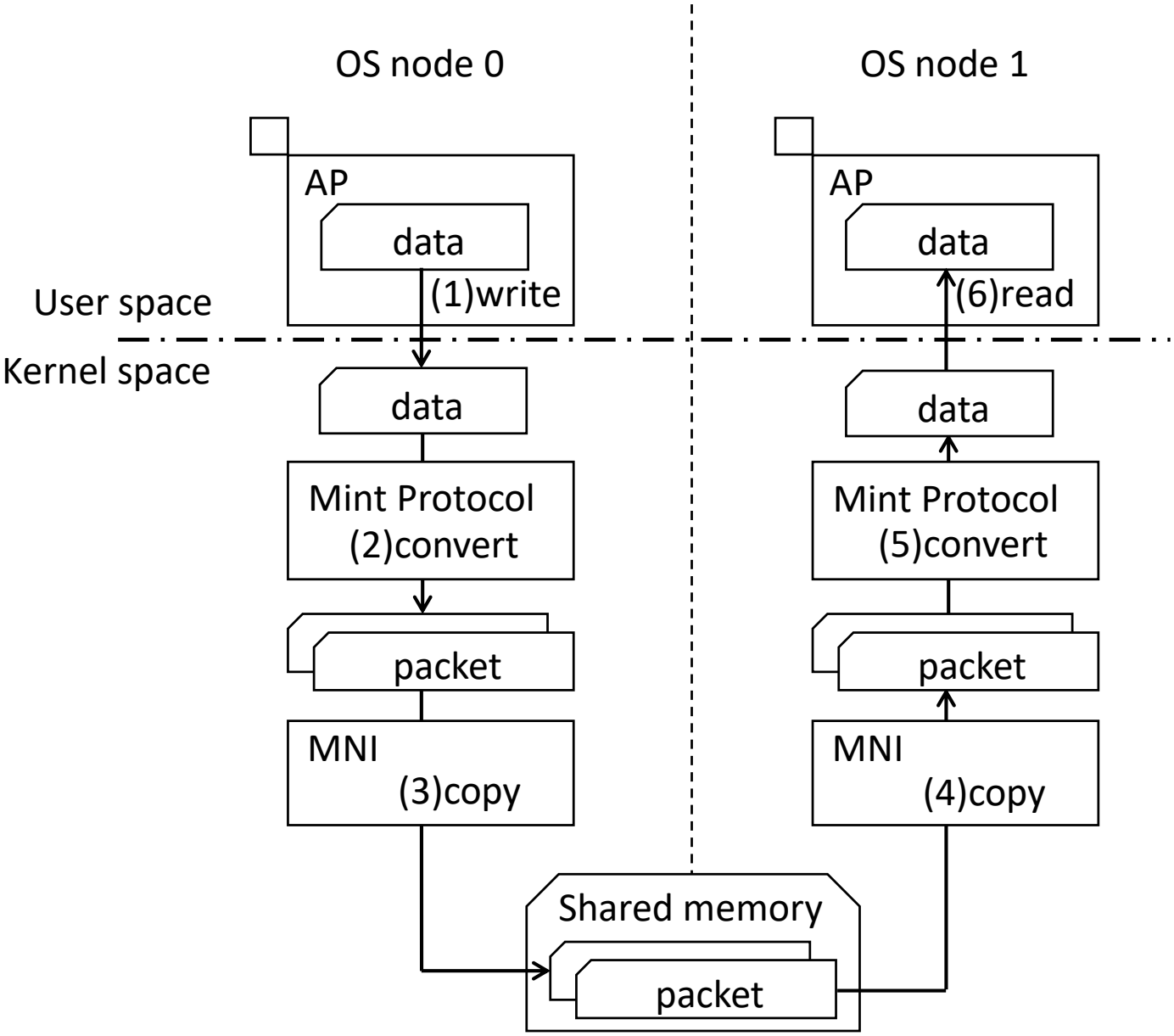
# Mint の構成例


AP
AP
AP
OS ノード 1
OS ノード 2
OS ノード 3
CPU(コア分割)
コア0
コア1
コア2
コア3
メモリ(空間分割)
共有メモリ
OS1用デバイス
OS2用デバイス
OS3用デバイス
H/W

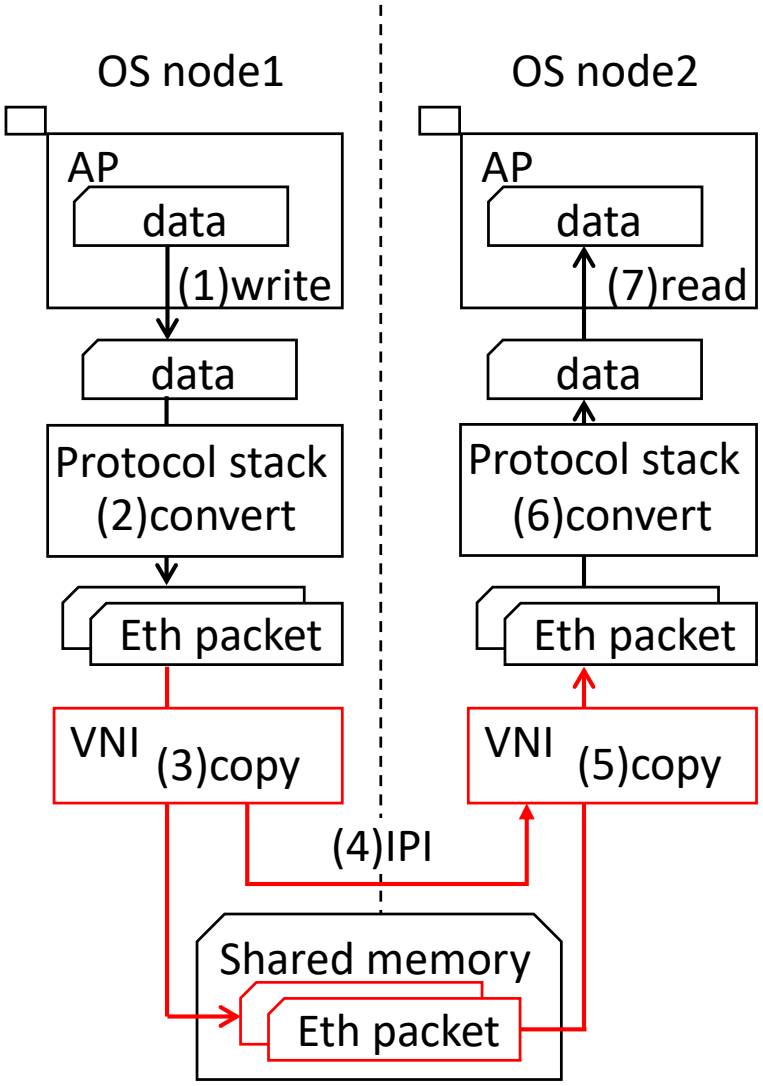
# NICハードウェアを用いた通信



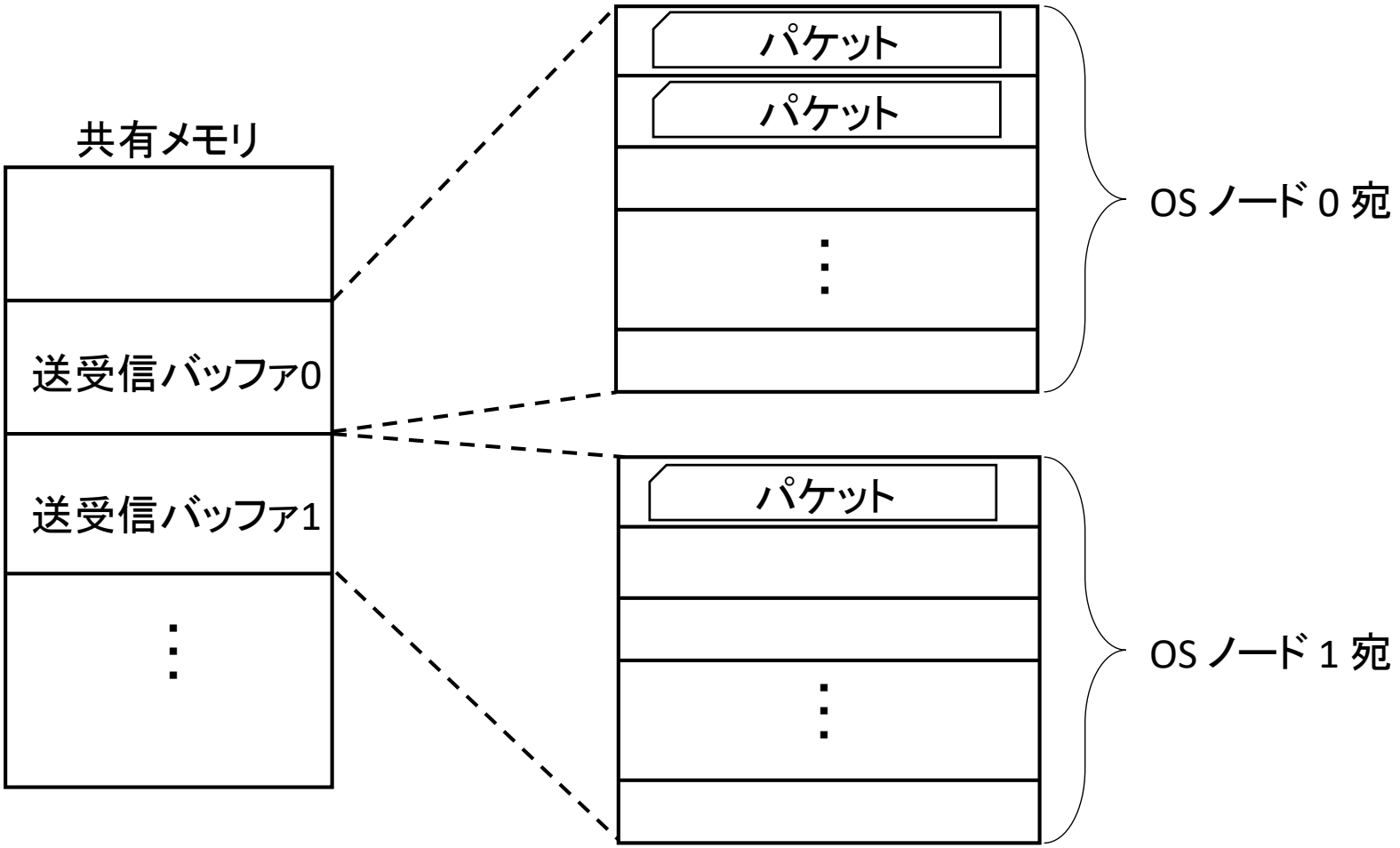
# 共有メモリを介した通信



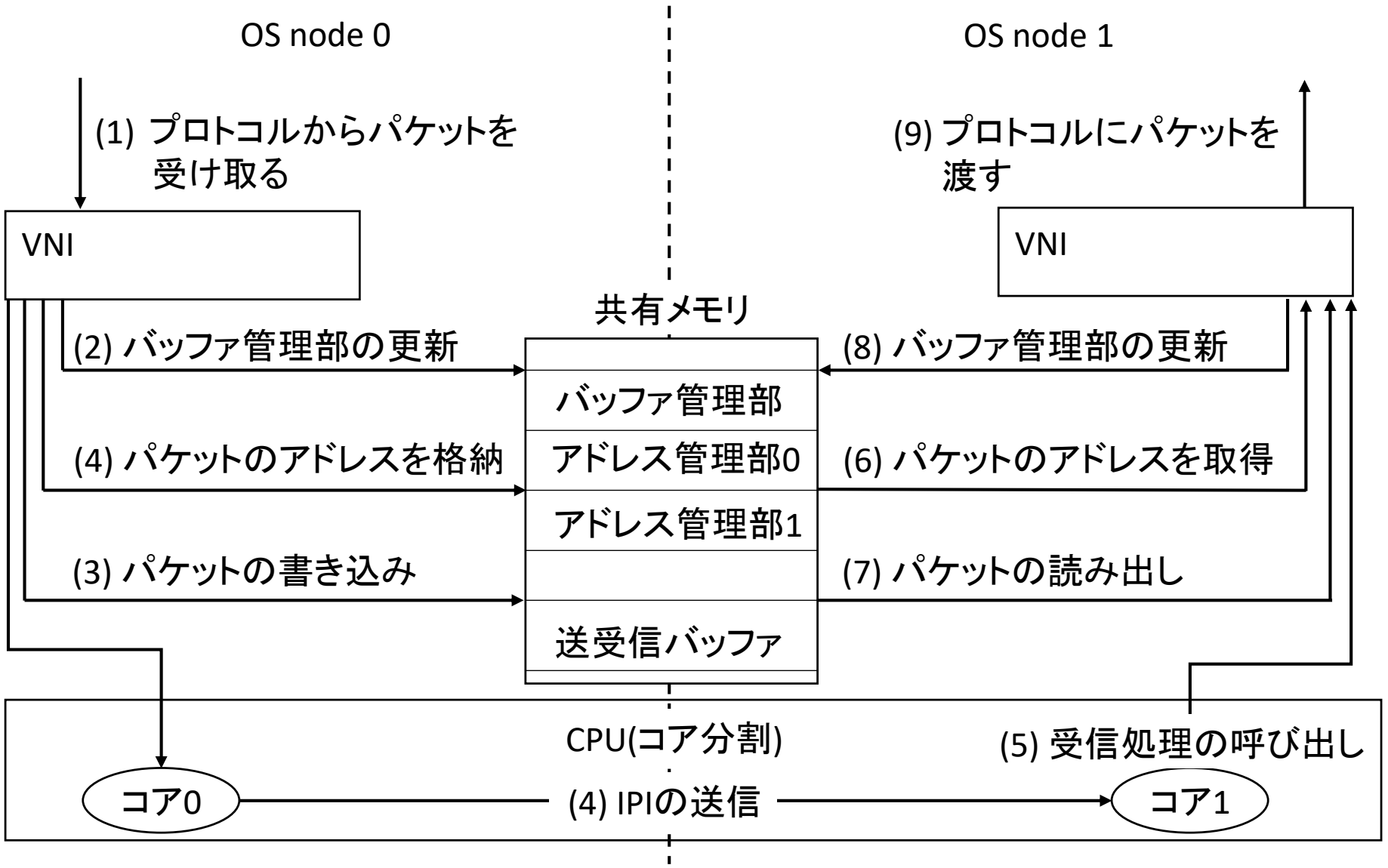
# VNIを用いたOSノード間通信



# 構成1



# VNIの処理流れ


OS node 0
OS node 1
(1) プロトコルからパケットを受け取る
(9) プロトコルにパケットを渡す
VNI
VNI
共有メモリ
(2) バッファ管理部の更新
(8) バッファ管理部の更新
バッファ管理部
(4) パケットのアドレスを格納
アドレス管理部0
(6) パケットのアドレスを取得
アドレス管理部1
(3) パケットの書き込み
(7) パケットの読み出し
送受信バッファ
CPU(コア分割)
(5) 受信処理の呼び出し
コア0
(4) IPIの送信
コア1

# アプリケーション改変の有無

＜テスト＞

（テスト1）TCP/IPプロトコルを用いた通信

送信側で特定の文字列を送信し，受信側で受信した文字列を標準出力する

（テスト2）ベンチマークツールを用いた通信

Linux上で動作するベンチマークツールを用いて通信できることを確認する

（テスト1）から，提案手法において，既存APを改変せずにNICハードウェアを用いた通信と同様の結果を確認した

（テスト2）から，iperf を用いて実効スループット802 μ msで通信可能であることを確認した

# 提案手法の実装により追加したコード量

| 通番 | ファイル | コード量 |
|---|---|---|
| 1 | driver/net/vni/vni.c | 456 |
| 2 | driver/net/vni/vni.h | 16 |
| 3 | driver/net/vni/Makefile | 1 |
| 4 | driver/net/Makefile | 1 |
| 合計 | 4ファイル | 474 |