# 仮想化ノードを使用した
# 実験用非 IP プロトコルの開発

日立製作所 中央研究所
金田 泰

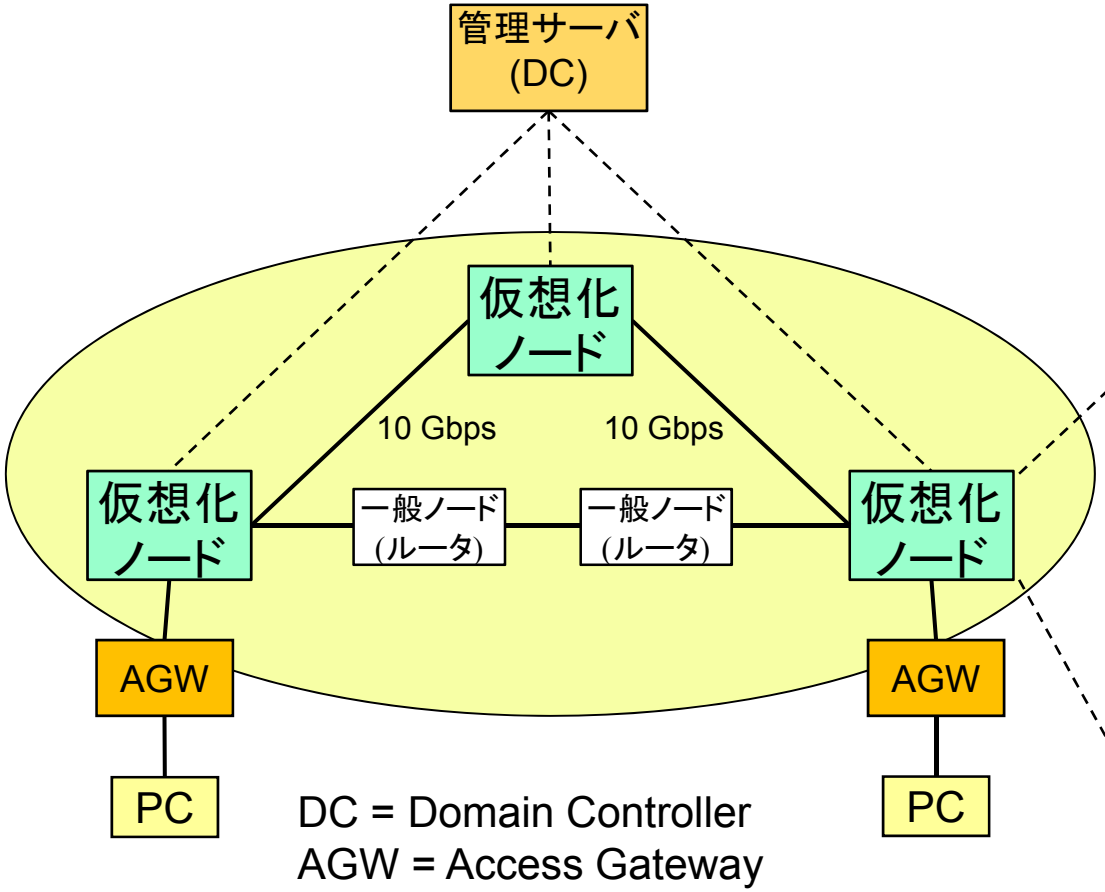
# 目次

- はじめに
- 仮想化ノードと仮想化基盤
- IPEC の開発目標
- IPEC の機能と実装
- 実験とデモ
- まとめ

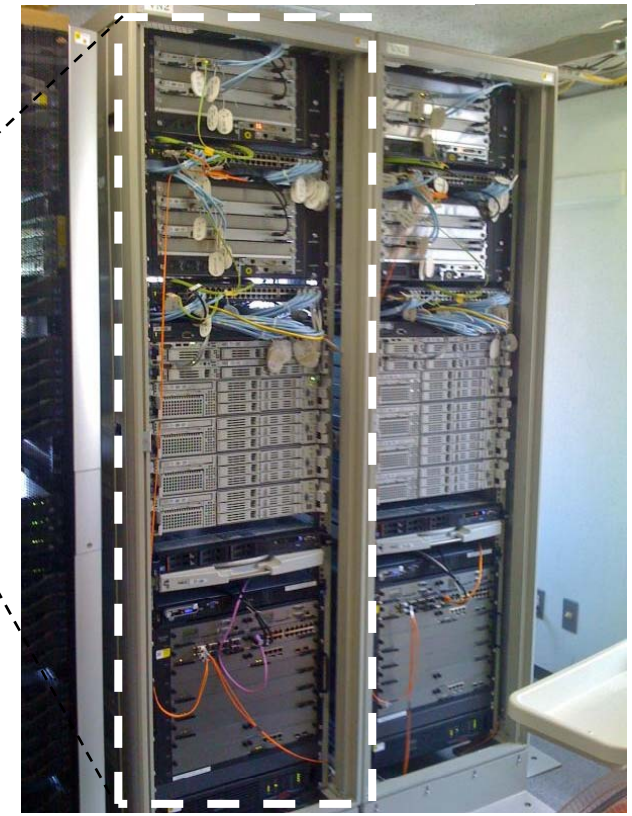
# はじめに

- この発表では “仮想化ノード” 試作機に実装した実験用非 IP プロトコル IPEC の機能と実装について報告する.
- 仮想化ノード・プロジェクトとネットワーク仮想化基盤について
  - 情報通信研究機構 (NICT) 中心に，東大と NTT，富士通，NEC，日立の各社が**共同開発・共同研究**している.
  - 既存のインフラを利用して新世代ネットワークを研究するための**ネットワーク仮想化基盤**を開発している.
  - ひとつの物理ネットワーク上で独立かつ自由に設計された**複数の仮想ネットワークが同時に動作**する環境を実現する.
  - ネットワーク仮想化基盤は，現在，研究開発用テストベッド・ネットワーク **JGN2plus** に導入されつつある.
- 非 IP プロトコル IPEC について
  - 仮想化基盤上で単純で汎用性のある非 IP プロトコルを確立するための第 1 歩として開発した.
  - Ethernet と IP の利点をあわせもつ実験用プロトコルの開発をめざしている.

# 仮想化基盤におけるネットワークの物理構成



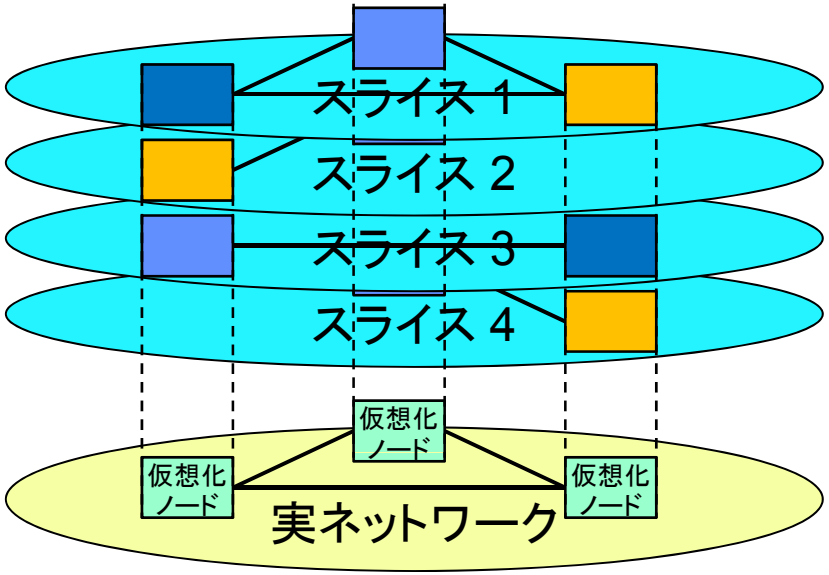

仮想化ノード



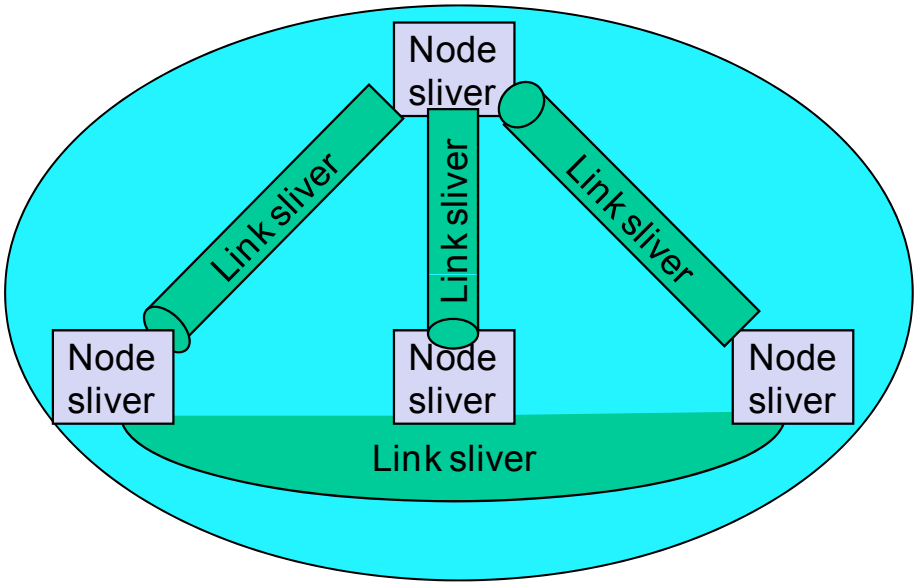
# 仮想化基盤におけるネットワークの論理構成

- **スライス (slice)**: 仮想化基盤上につくられる仮想ネットワーク.
- **スライスの主要な構成要素**
  - ◆ノードスリバー (node sliver): 仮想化ノード中に存在するプログラマブルな計算資源. プロトコル処理, ノード制御などに使用する.
  - ◆リンクスリバー (link sliver): ノードスリバー間を結合する仮想リンク. 物理ノード間を point-to-point でつなぐ.

仮想化基盤




スライスの構成

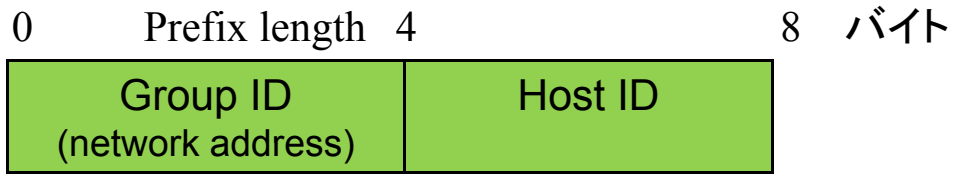


# IPEC の開発目標

- **新プロトコルの研究:** 単純で汎用性のある非 IP プロトコルの確立をめざした研究への第 1 歩とすること
  - IP 上で実現すると多層化し複雑化する機能を，**Ethernet や IP の長所をあわせもつ 1 層の単純な非 IP プロトコル**により実現する.
  - **Ethernet スイッチの学習アルゴリズムを拡張**し，ループをふくむ任意の構造のネットワークにおいて使用可能な転送アルゴリズムを実現する.
- **仮想化ノードの開発:** 仮想化ノード使用のネットワーク上で非 IP の新プロトコルが開発でき動作するのを実証すること
  - スライスの動作検証
  - ユーザビリティの検証
  - 仮想化ノードが新プロトコルの実験に適していることを確認
  - 今後の仮想化ノード使用による**新プロトコル開発のテストケース**をつくる (開発者にノウハウを提供する)

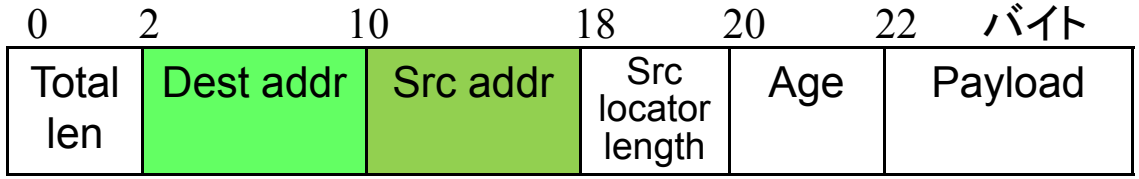
# IPEC のアドレスとパケットの形式

■ アドレス形式




◆ **Group ID**: ホストによって構成されるグループの ID

■ パケット形式




◆ **Age**: スイッチ間でパケットが転送されるごとに，1ずつ増加する．ループの存在により重複したパケットの廃棄に使用される．

# IPEC の実装法とパケット処理手順

- ノードスリバー上に IPEC を実装した.
  - 今回は VM (スローパス) 上に C 言語でプログラムを記述した.
- ノードスリバーに到着したパケットを，つぎの 2 ステップで処理する.
  - 学習
  - 転送

# IPEC の学習アルゴリズム

Ethernet スイッチと同様の学習だが，グループだけ学習する (→ スケールする)

if 到着パケットの src **group** が転送テーブルに登録されていない then
　転送テーブルに group, group length, input port, age を登録 (学習);

過去に到着したパケットよりみじかい経路をたどっているときと，過去の登録を忘却するべきときは学習する (→ 最短経路を学習する)

else if **登録要素の age > 到着パケットの age** or
　　**登録要素が「登録タイムアウト」している** then
　登録要素の age, port = 到着パケットの age, port;
　登録要素の タイムスタンプ = 現在の時刻 (ns);

ダブって到着したパケットは廃棄する (→ ループを許容する)

else if **登録要素の age < 到着パケットの age** or
　　**登録要素の port != 到着パケットの port** then
　パケットを廃棄 (転送アルゴリズムを実行しない);

else 登録要素の タイムスタンプ = 現在の時刻 (ns);

# IPEC の転送アルゴリズム

学習していない (あるいは忘却した) ときは “ブロードキャスト”

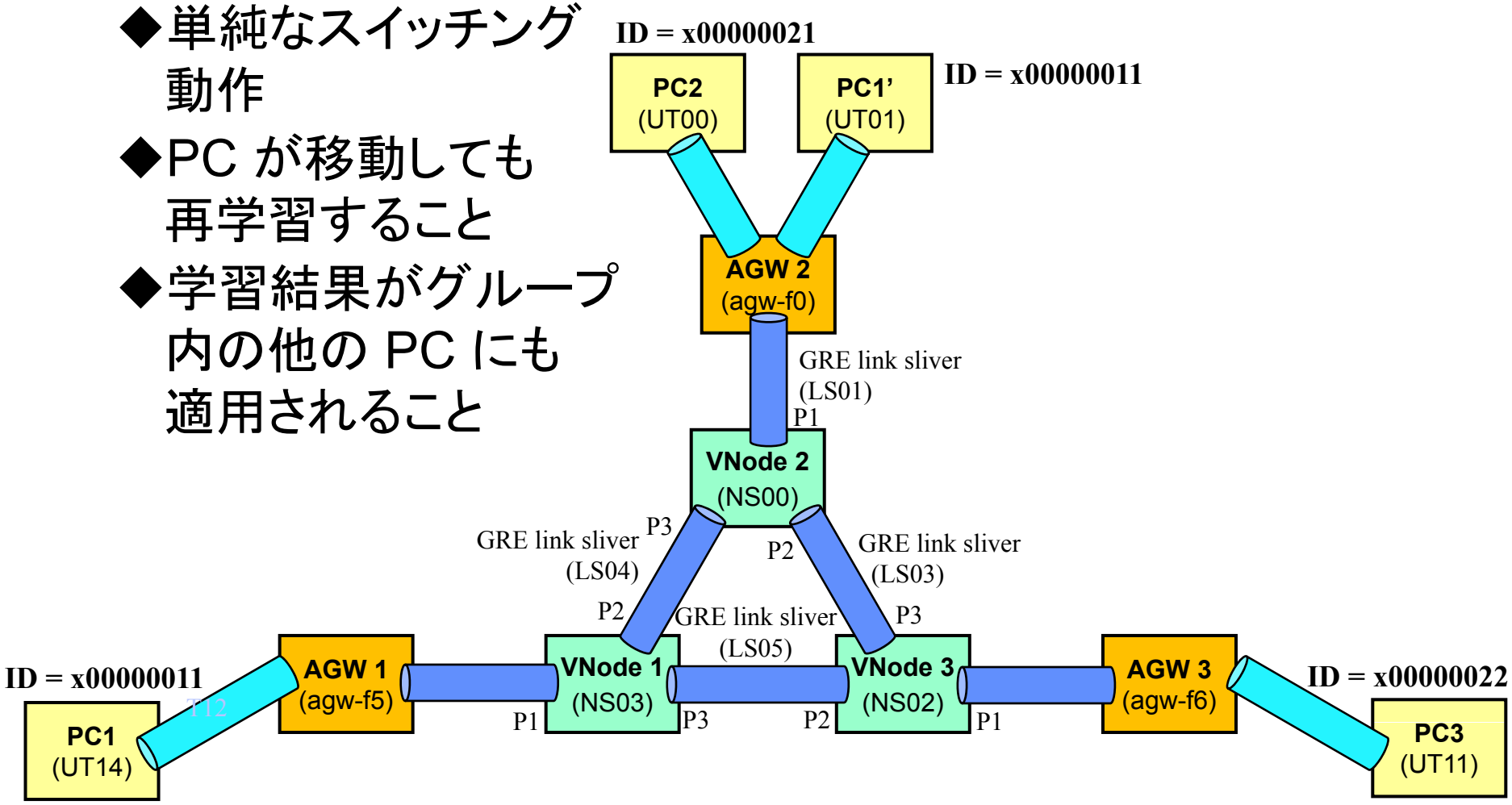
if 到着パケット の dest **group** が転送テーブルに登録されていない
　　or 登録要素が「**参照タイムアウト**」している then
　到着パケットの age を増加したものをフラッドする;

学習ずみのときは特定のポートだけに出力

else
　登録要素の port にだけ，到着パケットの **age を増加**したものを
　　出力する;

# 実験 – その目的とスライス構成

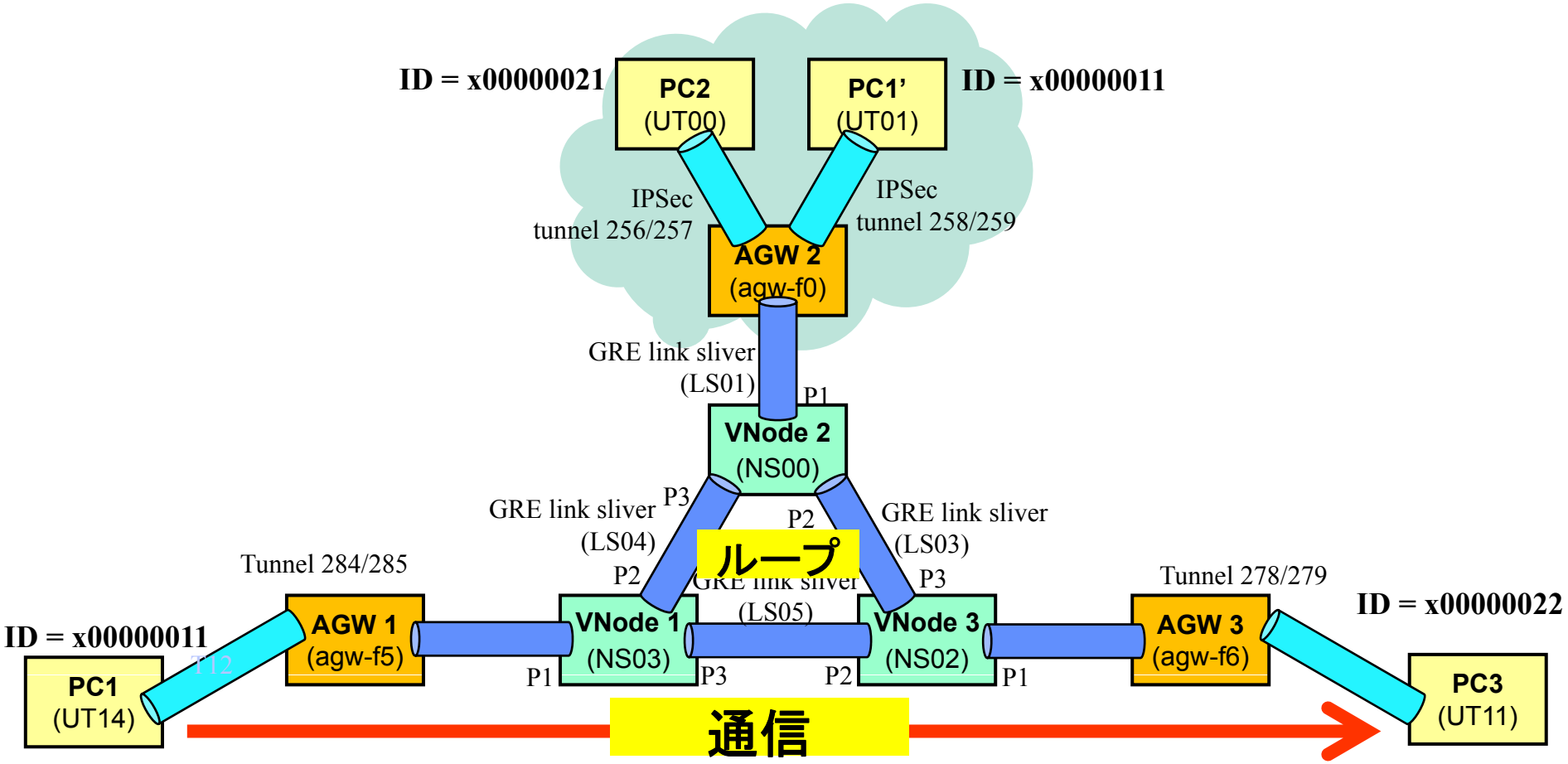
- IPEC を実装し，3 個の仮想化ノードを使用したループをふくむスライスでつぎの通信動作を確認した.
  - ◆単純なスイッチング動作
  - ◆PC が移動しても再学習すること
  - ◆学習結果がグループ内の他の PC にも適用されること



# 実験 / デモのシナリオ

## ■ 単純なスイッチング動作の確認

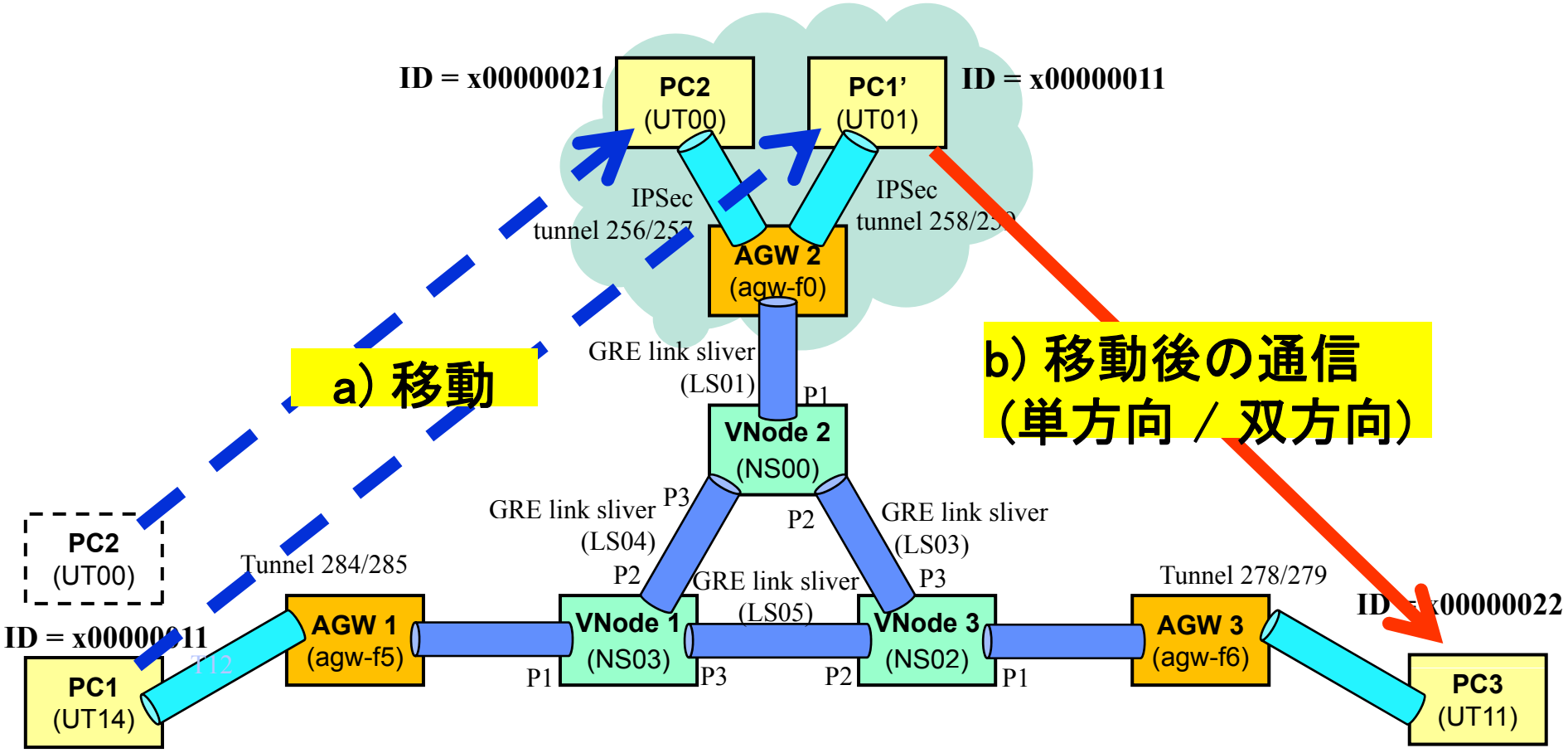
- ◆ PC1 から PC3 への通信を観察した.
- ◆ 学習過程をみるために, PC3 は最初の 40 秒間は応答しない.



# 実験 / デモのシナリオ (つづき)

■ PC が移動しても再学習することの確認

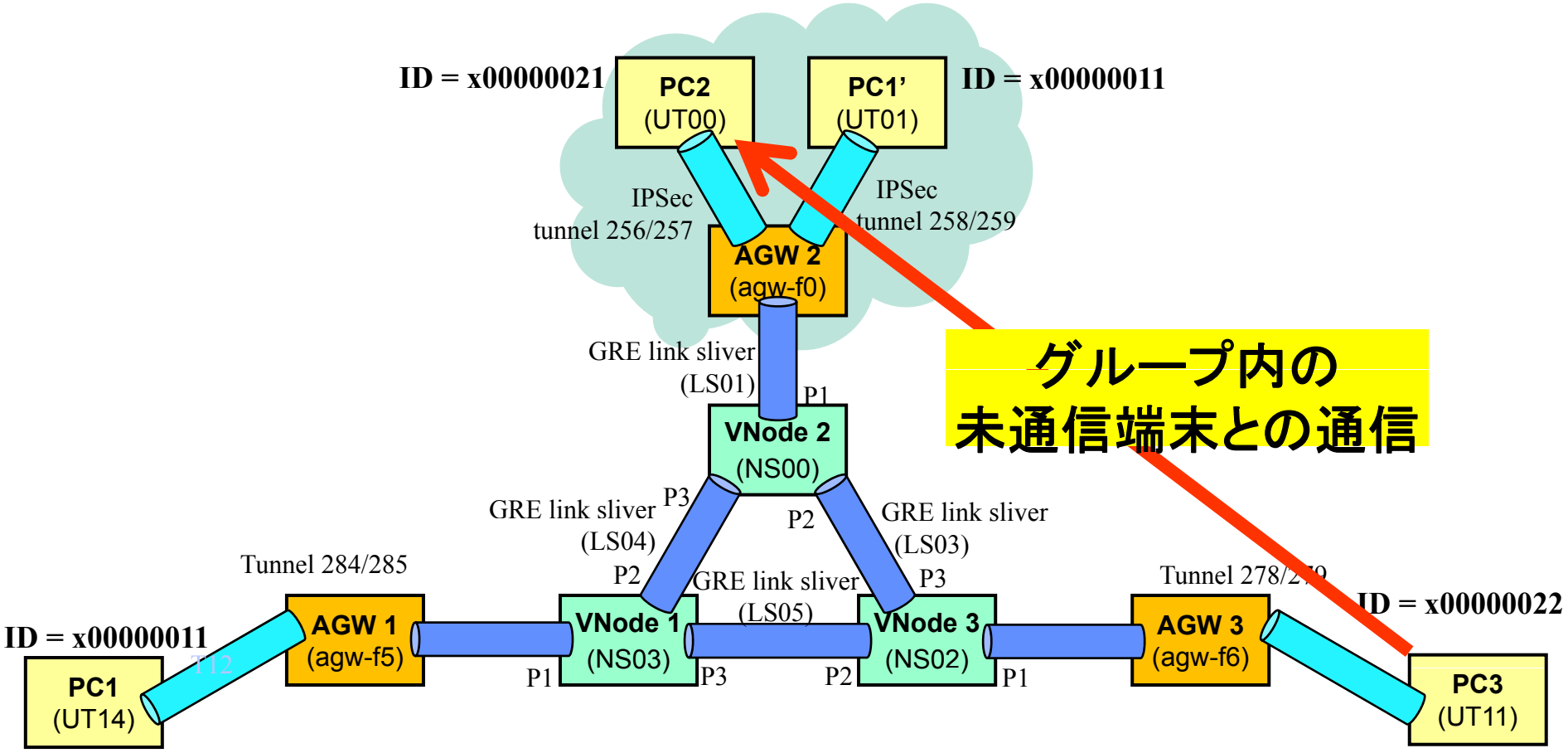
◆ PC1 と PC2 を仮想的に移動させてから，PC1-PC3 間で通信する.



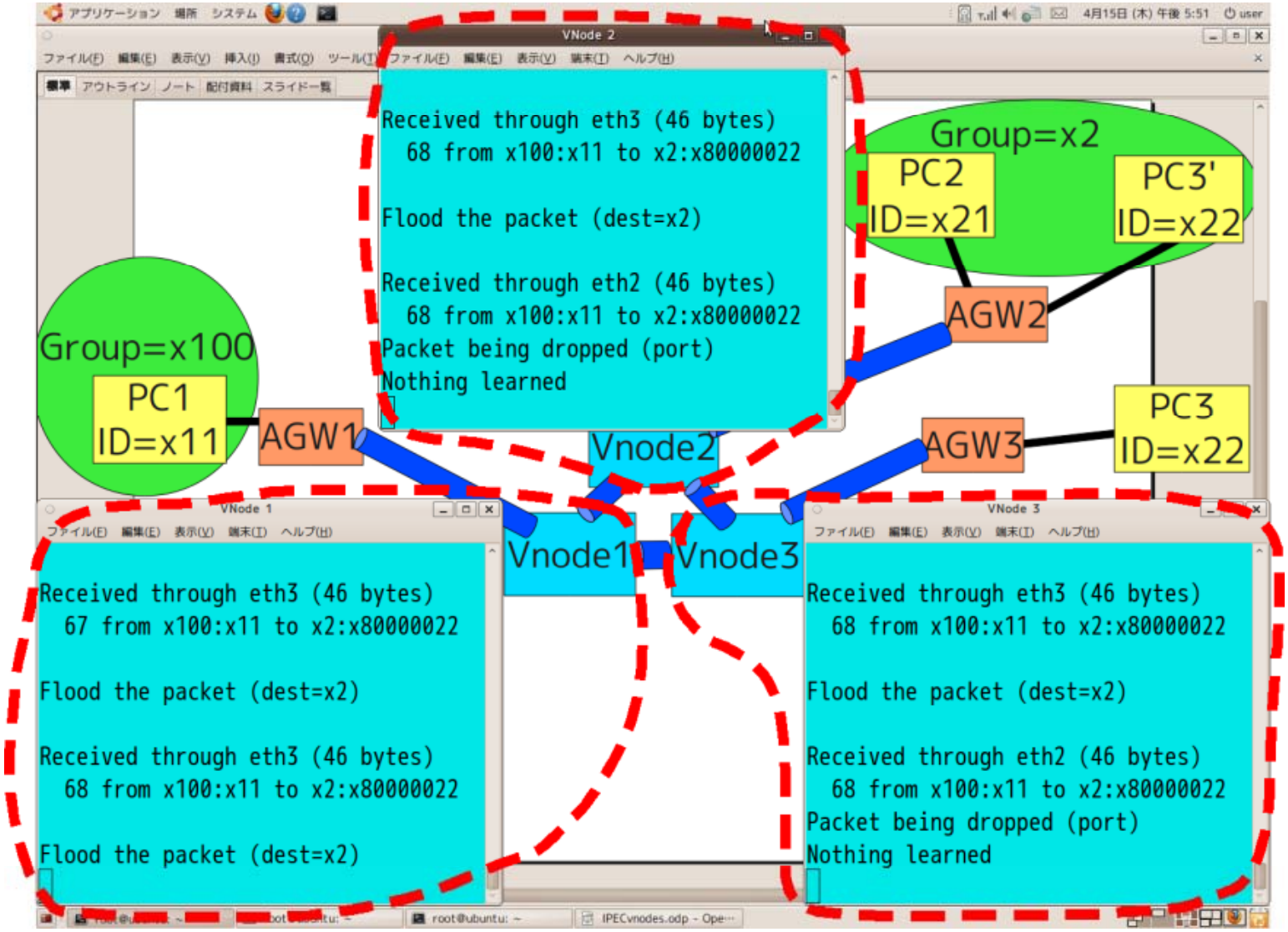
# 実験 / デモのシナリオ (つづき)

■ 学習結果がグループ内の他の PC にも適用されることの確認

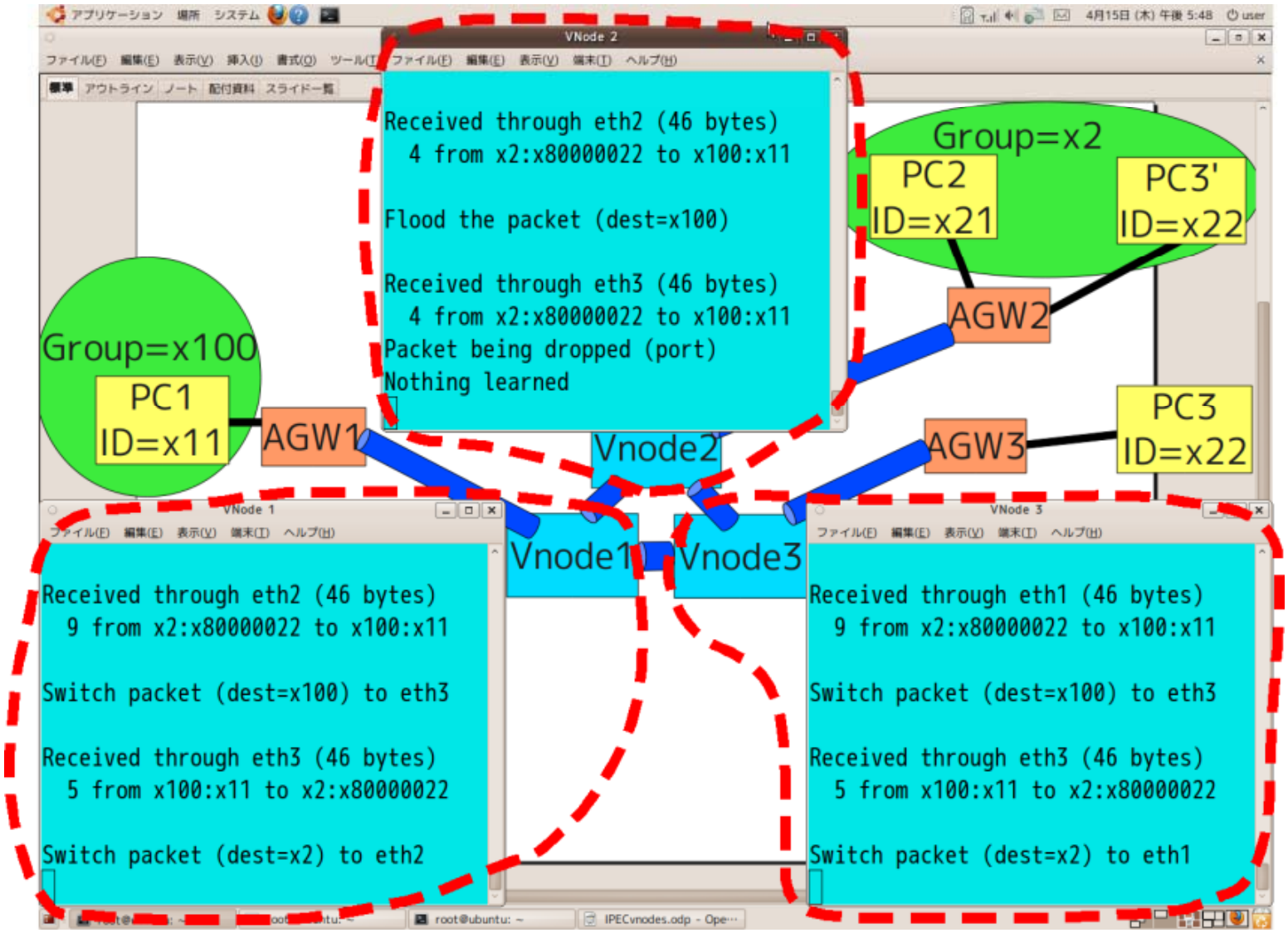
◆ PC3 と，PC1 と同一グループに属する，まだ通信していない PC2 とを通信させる.



# ノードスリバーの出力表示例: フラディング時



# ノードスリバーの出力表示例: スイッチング時



# 広域での実験とデモ等

- 広域での実験・デモを Interop Tokyo 2010 (幕張，6/7-11) において実施して，おなじ実験結果がえられた.



- 8th GENI Engineering Conference (GEC8) において IPEC を紹介しデモビデオを Web に掲載している.
  - ◆ GEC は米国における新世代ネットワーク・プロジェクトである GENI の会議

# まとめ

- **つぎのような特徴をもつ非 IP プロトコル IPEC を開発した.**
  - Ethernet, IP それぞれの特徴的な機能の一部を 1 層の単純な非 IP プロトコルによって実現した.
  - Ethernet スイッチの学習アルゴリズムを拡張して, ループをふくむネットワークで使用でき障害にも対応できる方法を実現した.
  - **学習をグループ単位でおこなう**ため, Ethernet よりスケールする. また, グループ単位の移動が効率的に学習できる.
- **IPEC を VNode 上に実装して, グループ単位の学習や端末の移動に実際に対応できることを実験により確認した.**
- **今後の課題: ネットワーク・プロセッサ (ファストパス) 使用の実装・実験**