7. プライベート・ネットワークとネットワーク仮想化

要点

- インターネットからきりはなされたプライベート・ネットワークではセキュリティが確保しやすく，従来のプロトコルにしばられない．
  - プライベート・ネットワークには物理的なものと仮想化されたもの (VPN) とがある．
- 仮想化されたネットワークとして VLAN があり，企業などでつかわれている．
- 仮想化にはつぎのような種類がある．
  - 質の仮想化と量の仮想化，分割型仮想化と融合型仮想化．
  - コンピュータの仮想化とネットワークの仮想化．
- ネットワーク仮想化の研究によって，プログラマブルで自由なプロトコルがつかえる仮想ネットワークが開発されつつある．
  - 仮想ネットワークにおいては，プライベート・ネットワークの利点をいかして IP やイーサネットとはことなる新プロトコルの実験が自由にできる．
  - 世界各地で新世代ネットワークの研究開発，とくにネットワーク仮想化の研究や実験がおこなわれている．(アメリカで GENI というプロジェクト，日本で AKARI や仮想化ノード (VNode) 開発・利用プロジェクトなど)．

プライベート・ネットワークとパブリック・ネットワーク

- インターネットはだれもが接続できるパブリックなネットワーク.
- 秘密情報などをあつかうため，特定の企業などだけが接続できるようにしたプライベート・ネットワークもある.
- パブリック・ネットワークからプライベート・ネットワークにアクセスできるようにするため，ゲートウェイが設置される.
  - プライベート・ネットワークは接続可能な場所が限定されるため，それをひろげるためにゲートウェイがつかわれる.
  - 不正使用をなくすため，ゲートウェイでは厳重な認証・制限が適用される.

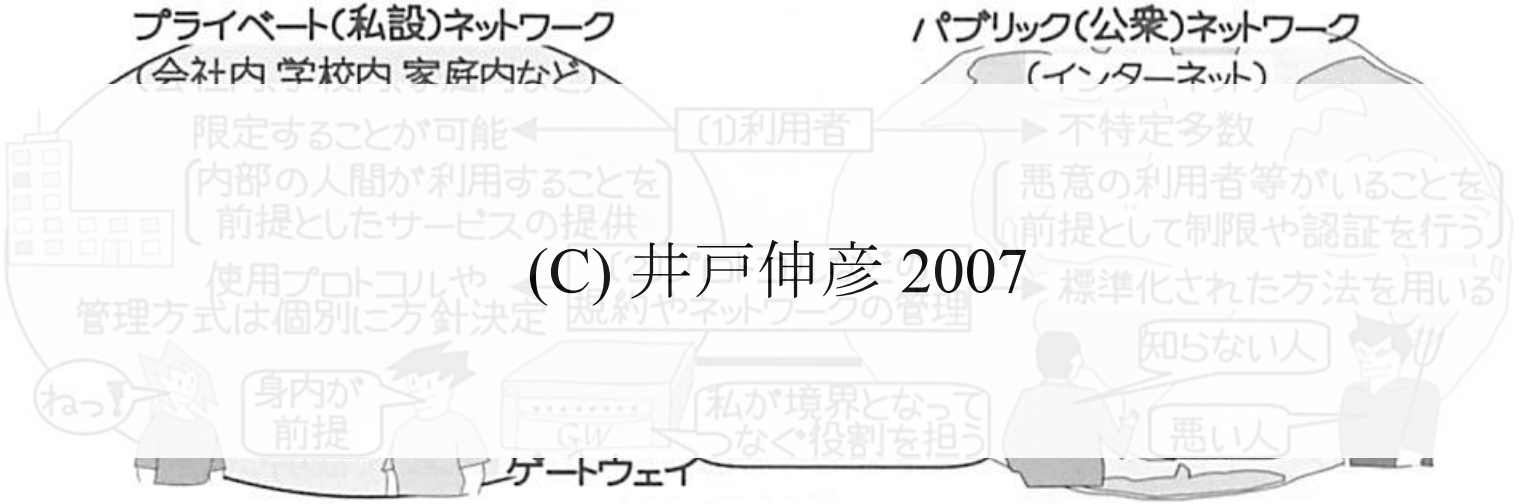


図9-1 プライベートネットワークとパブリックネットワーク、ゲートウェイ

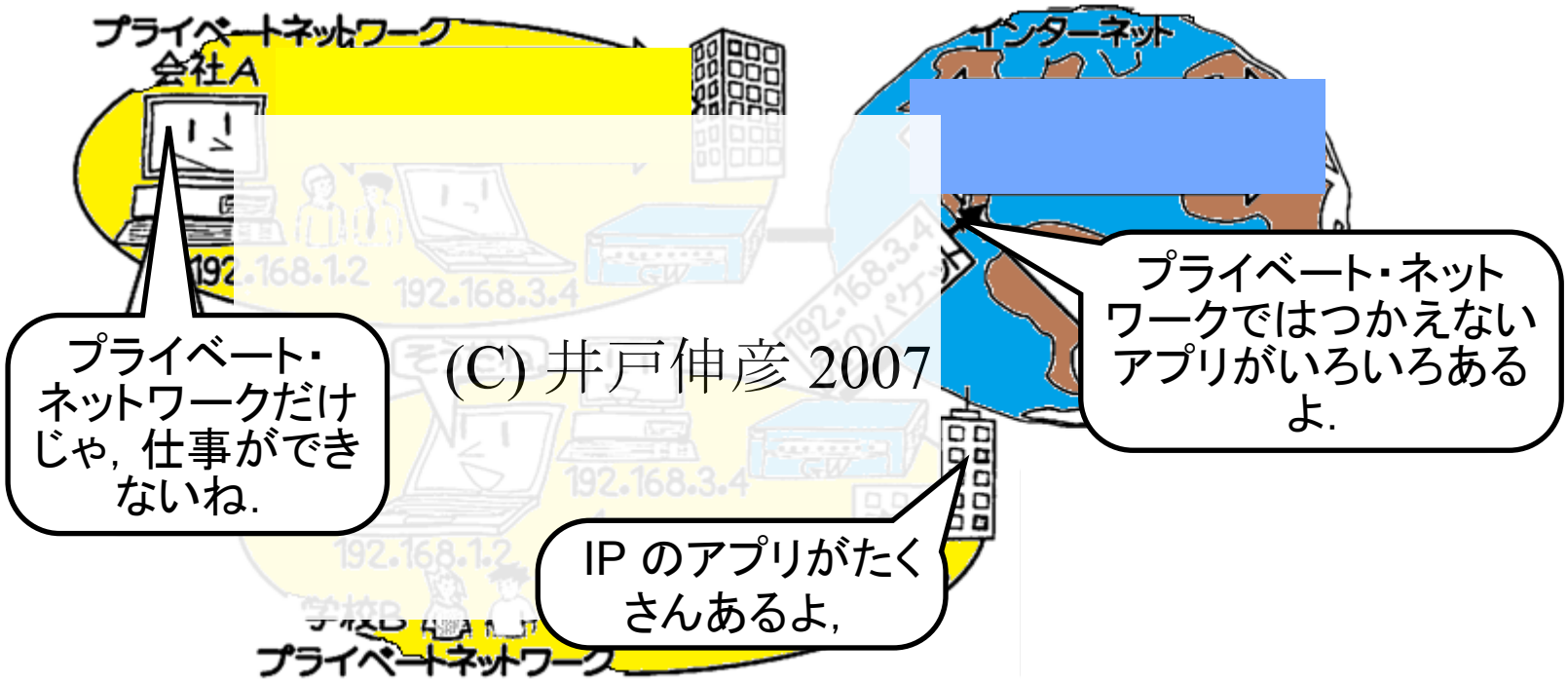
プライベート・ネットワークの利点

- 第 1 の利点はセキュリティ (秘密情報保護).
  - パブリック・ネットワークでは秘密が漏洩しやすい.
- パブリック・ネットワークの制約にしばられないことも利点.
  - パブリック・ネットワークは多数のユーザに共用されるので, 性能 (通信速度, QoS (サービス品質) など) も保証しにくい.
  - パブリック・ネットワークでつかわれるプロトコルにしばられない (たとえば, IP や TCP と共存できないプロトコルもつかえる).

IP によるプライベート・ネットワーク

■ プライベート・ネットワークでは IP 以外のプロトコルもつかえるが，現在はほとんどのネットワークで IP がつかわれている．

- ◆ 現在つかえるアプリケーションはほとんどすべて IP を使用するため．
- ◆ プライベート・ネットワークからパブリック・ネットワークにつなぐには IP が必要なため．



IP によるプライベート・ネットワークのアドレス

- プライベート・ネットワークからインターネット (パブリック・ネットワーク) につなぐなら，インターネットと重複するアドレスはつかえない.
- IETF できめたプライベート IP アドレス (下表の範囲) をつかう.

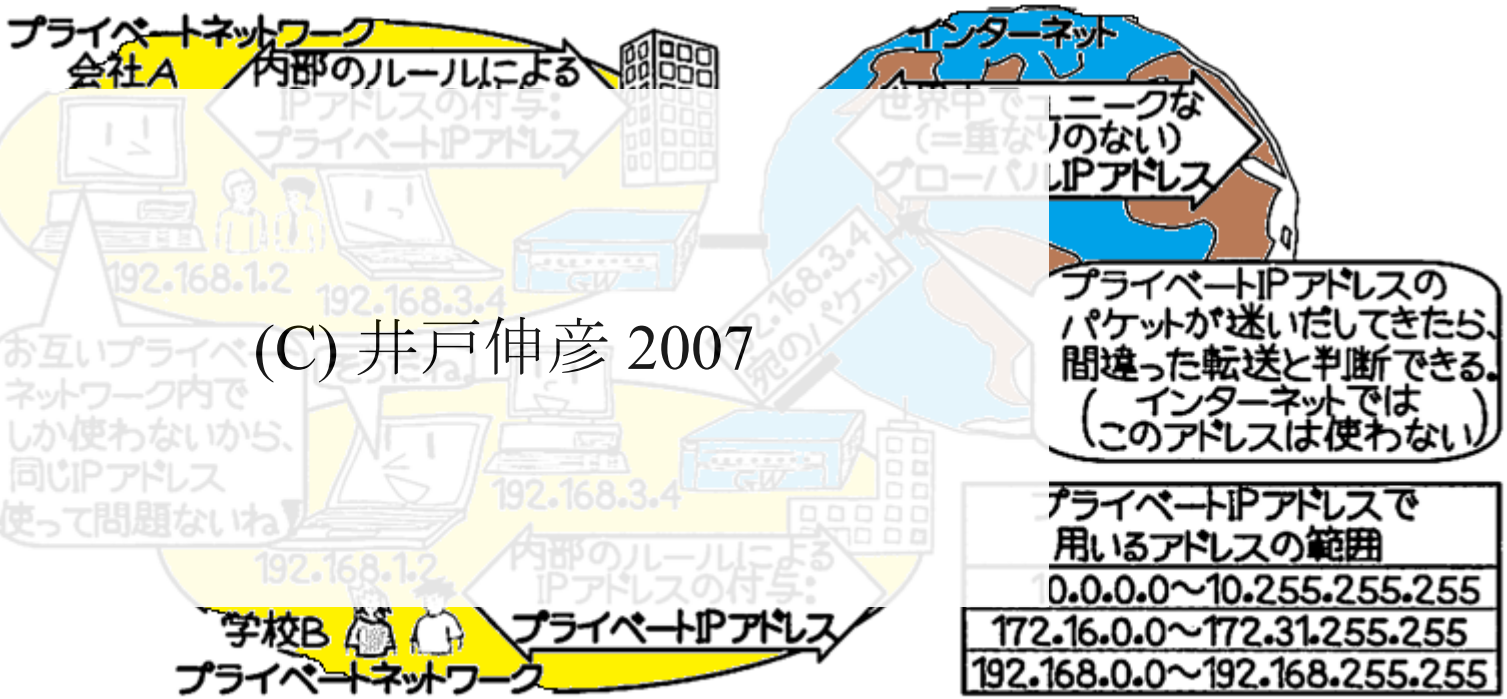


図9-3 プライベートIPアドレスとグローバルIPアドレス

ゲートウェイのやくわり

- ゲートウェイはプライベート・ネットワーク内のアドレスをかくす.
  - 同一のプライベート IP アドレスがほかでもつかわれているので，外部にみせてはいけない (アドレスの唯一性を保証するため).
  - プライベート IP アドレスが外部からわかると不正にアクセスされる可能性があるので，わからないようにする (セキュリティのため).
- ゲートウェイ (ファイアウォール) はプライベート・ネットワーク内への不正アクセスをふせぐ.
  - 特定のパターンの通信だけをゆるす (たとえば，通信開始は内側からにかぎるなど).

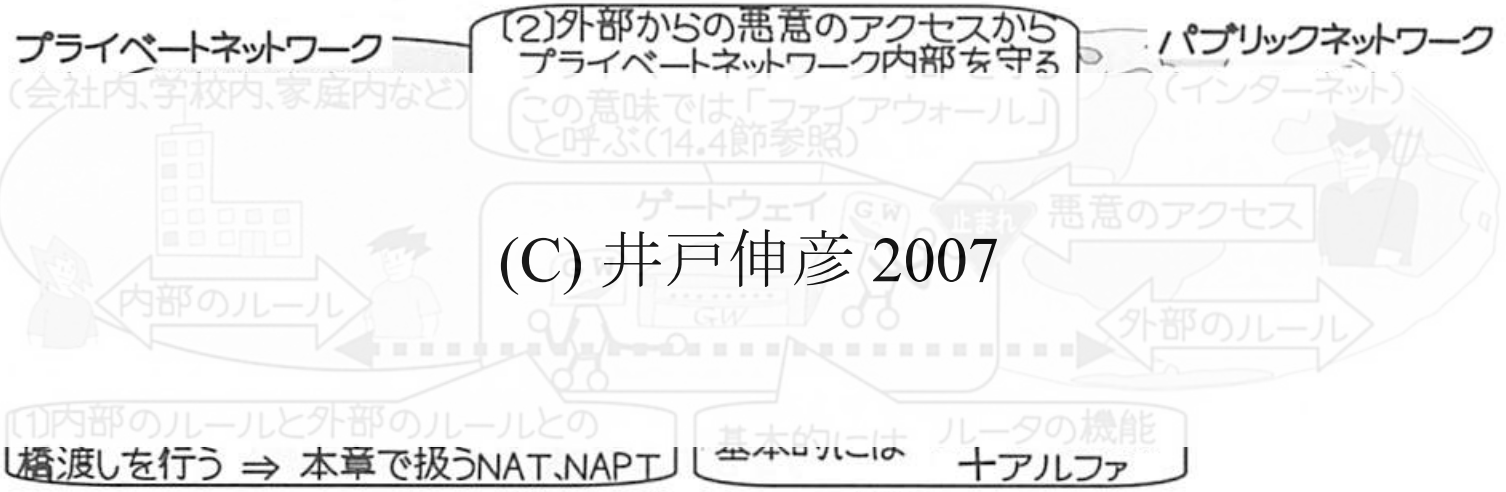
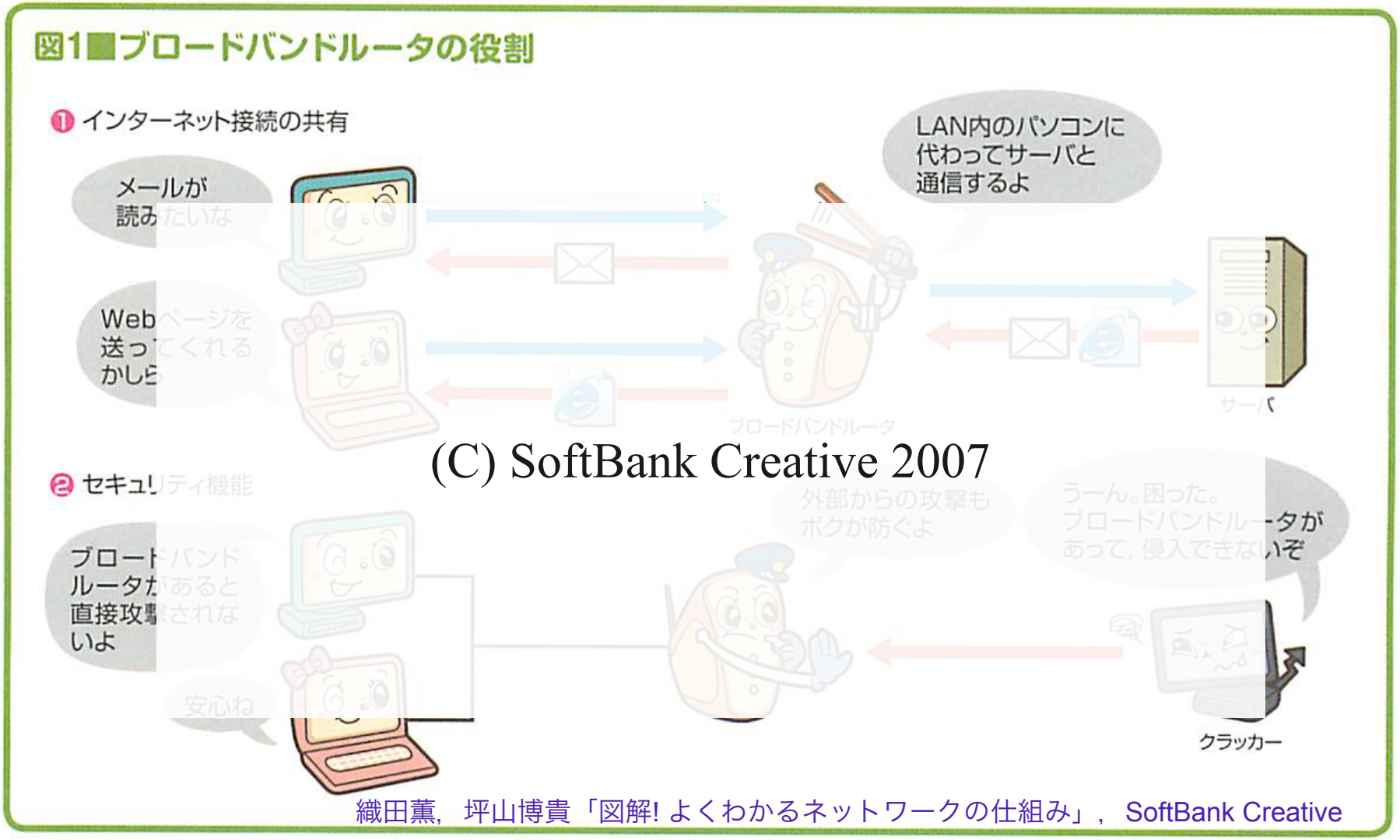


図9・2 ゲートウェイの役割

ゲートウェイとブロードバンドルータ

■ 一般家庭などでつかうためのゲートウェイはブロードバンド・ルータとよばれている.



織田薫, 坪山博貴「図解! よくわかるネットワークの仕組み」, SoftBank Creative

アドレス変換と通信

- パブリック・ネットワークに対して内部のアドレスをかくすための機能がアドレス変換.
- 変換法には NAT と NAPT とがある.
  - NAT はアドレスを 1 対 1 に変換する.
  - NAPT (IP マスカレード) ならグローバル・アドレスが 1 個ですむため，通常は NAPT がつかわれる.

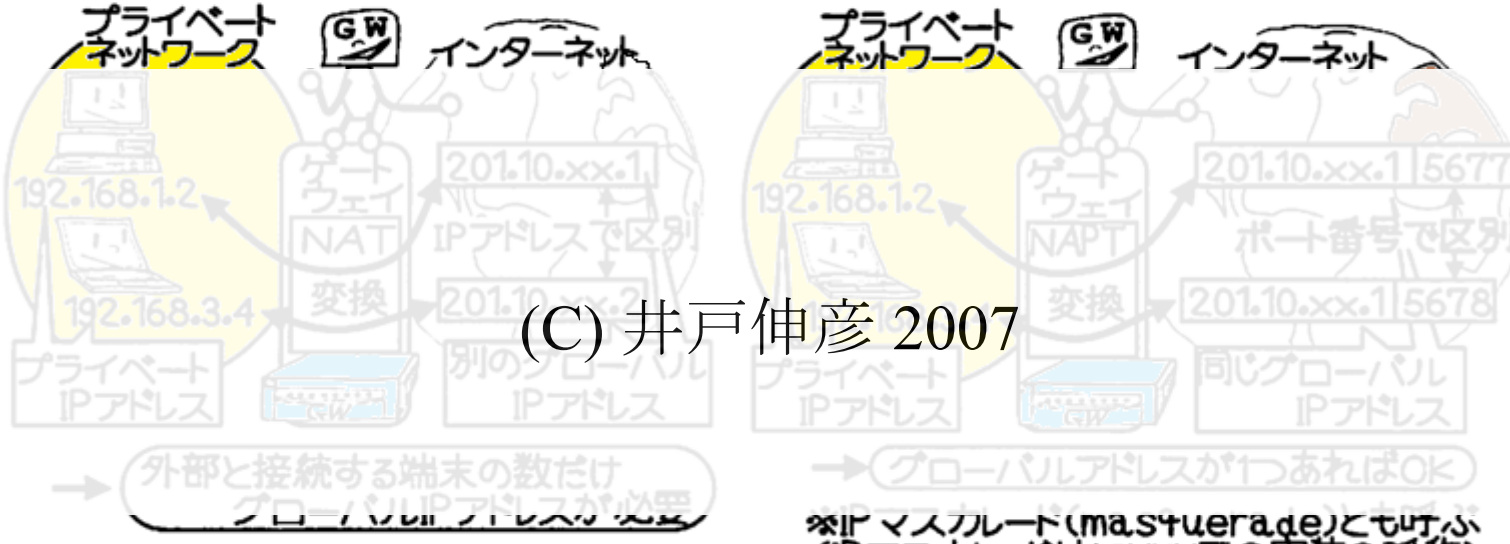
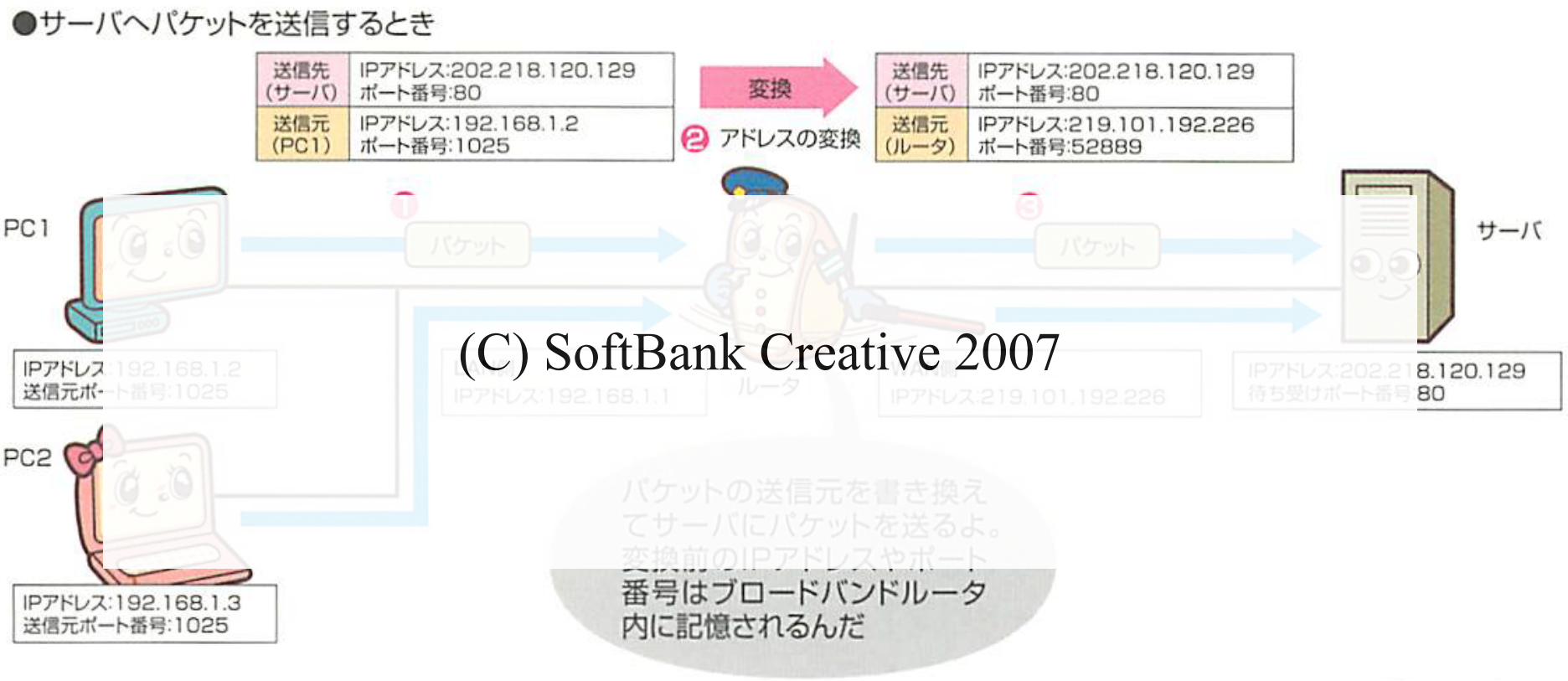


図9-4 NATとNAPT(IPアドレスのプライベートとグローバルとの変換)

NAPT のしくみ

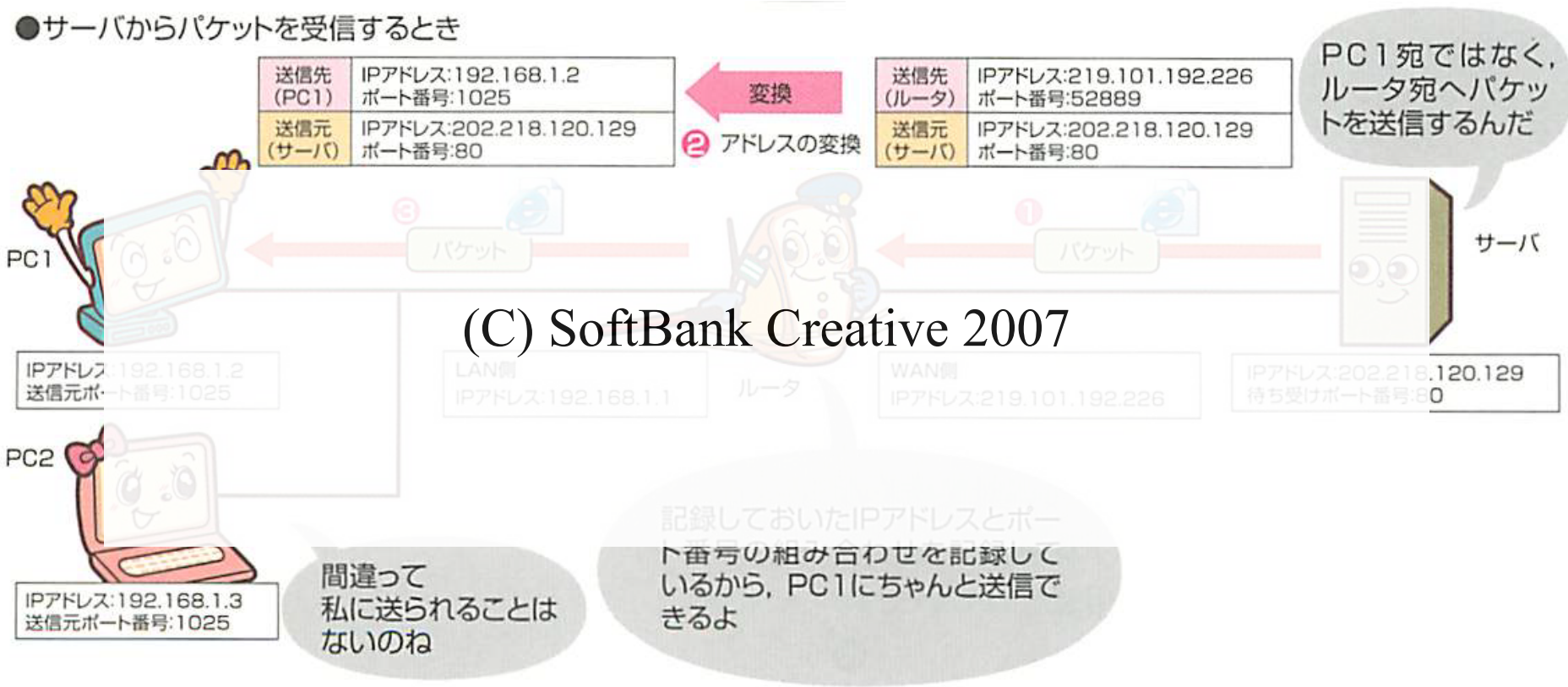
■ プライベート・ネットワークからの送信



織田薫, 坪山博貴「図解! よくわかるネットワークの仕組み」, SoftBank Creative

NAPT のしくみ（つづき）

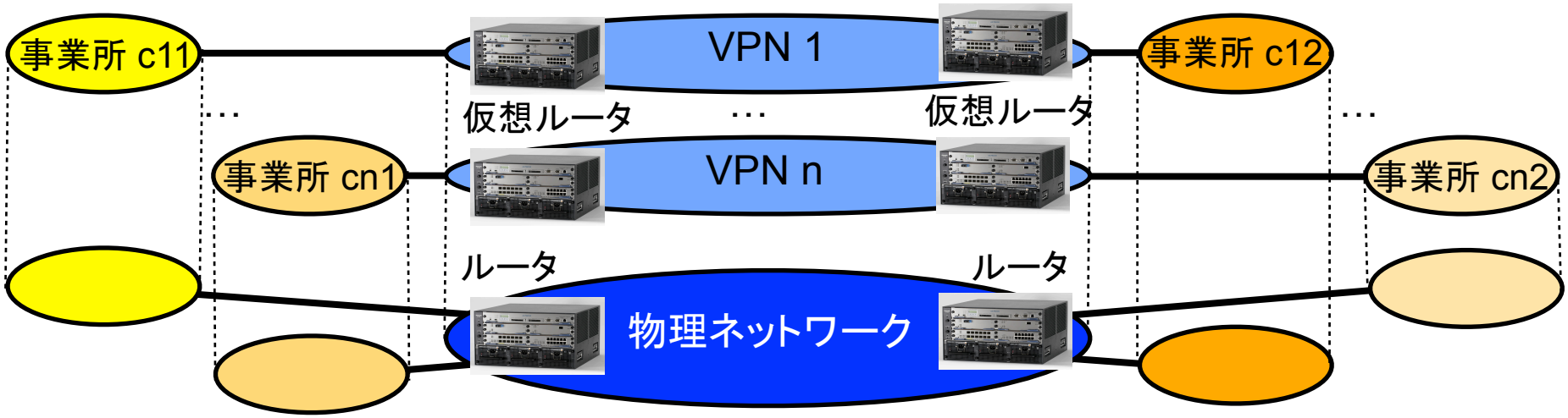
■ プライベート・ネットワークへの受信



織田薫，坪山博貴「図解! よくわかるネットワークの仕組み」，SoftBank Creative

プライベート・ネットワーク -- 物理的なものと仮想的なもの

- 広域に物理的なプライベート・ネットワークをつくるのは高コスト
  - ルータやスイッチを各地に設置して, その間をケーブルで接続する必要がある. 設置場所も用意する必要がある. (それぞれ設置コストがかかる)
  - ルータ / スイッチ, ケーブル, 設置場所などの管理コストがかかる.
- インターネットや共用 IP ネットワーク (電話会社などのもの) 城に仮想的なプライベート・ネットワークをつくれば低コスト
  - このようなネットワークを VPN (virtual private network) という.
  - 物理ネットワークの設置コストや管理コストはわけて負担すればよい.



仮想プライベート・ネットワーク (VPN)

- 複数の拠点をもつ企業のニーズ
  - 拠点間をむすぶプライベート・ネットワーク (専用線によるネットワーク) は高価なので, さけたい.
  - 拠点間はプライベート・ネットワークと同様に自由な通信が可能にしたいが, 同時にセキュアに通信したい.
- このようなニーズをみたす方法が VPN である.
- VPN における物理と論理
  - VPN は複数拠点間を物理的には共用ネットワークでむすぶ.
  - VPN は他のネットワークとは論理的に独立な (たがいに干渉しない) ネットワークを実現する.

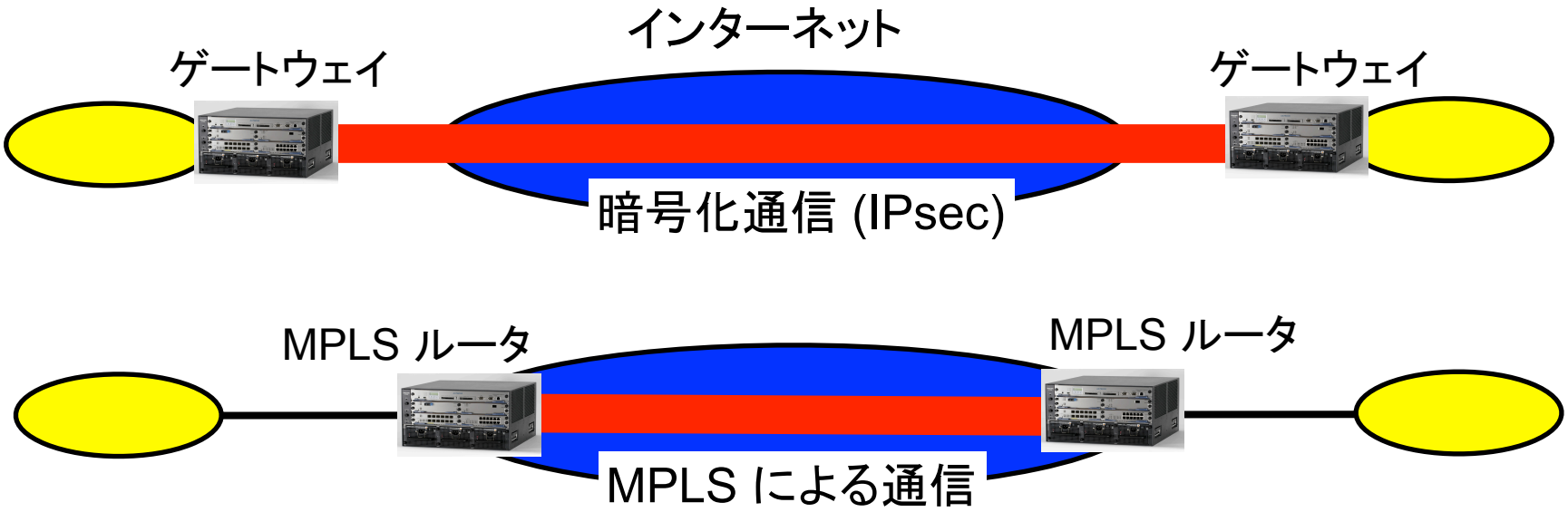
※ 認証:偽物の通信相手でないことの確認
IPsec:IP security protocol

図9-28 VPN(Virtual Private Network)

IP 用 VPN の種類

■ (IP をつかうための) 代表的な VPN には 2 種類ある.

- インターネット VPN (IPsec VPN)
- IP VPN (MPLS VPN)

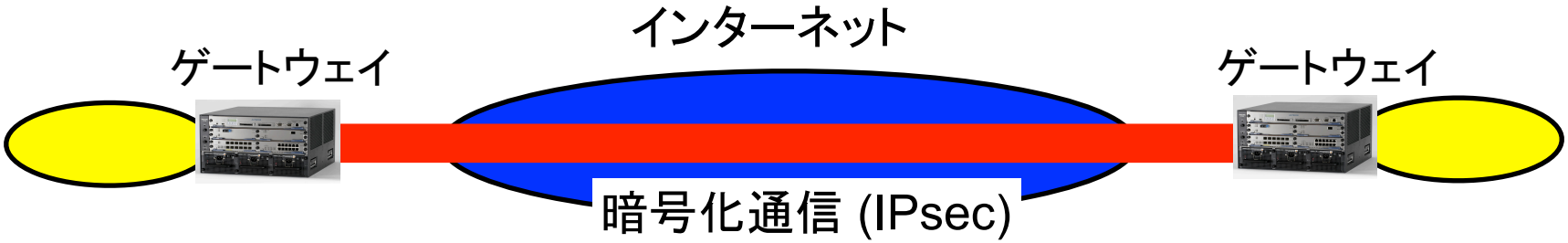


- 以下これらの VPN について説明する.

IP 用 VPN の種類 (つづき)

■ Internet VPN (IPsec VPN)

◆ インターネット上で暗号化通信をする.

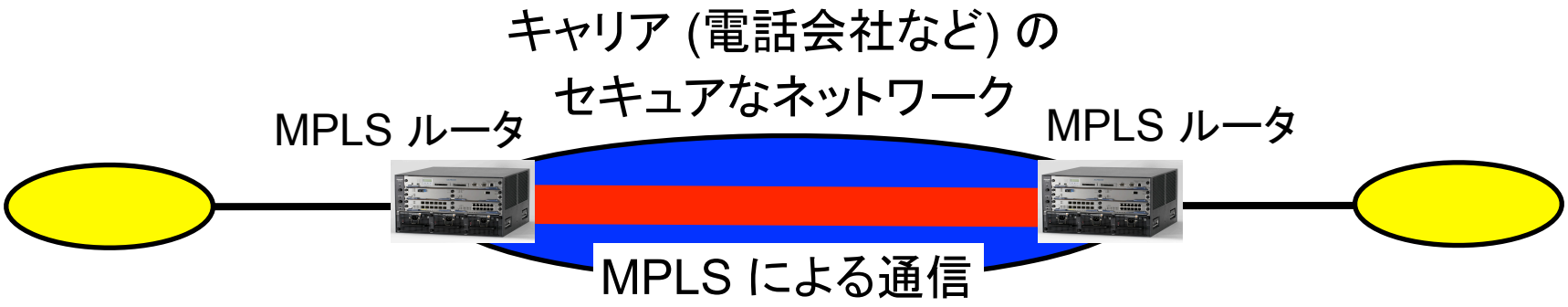


◆ IPsec というプロトコルでカプセル化する.

◆ 高性能な暗号化はコストがたかいことが欠点.

IP 用 VPN の種類 (つづき)

- IP VPN (MPLS VPN)
  - MPLS (Multi-Protocol Label Switching) というプロトコルを使用する.
  - インターネットではなく，あらかじめセキュリティが確保されたネットワークを使用する.
  - 暗号化はしない –– 暗号化は高コストなので，さける． 暗号化しなくてもインターネットのような危険はない.



プライベート・ネットワーク内のパケット

IP ヘッダ	ペイロード

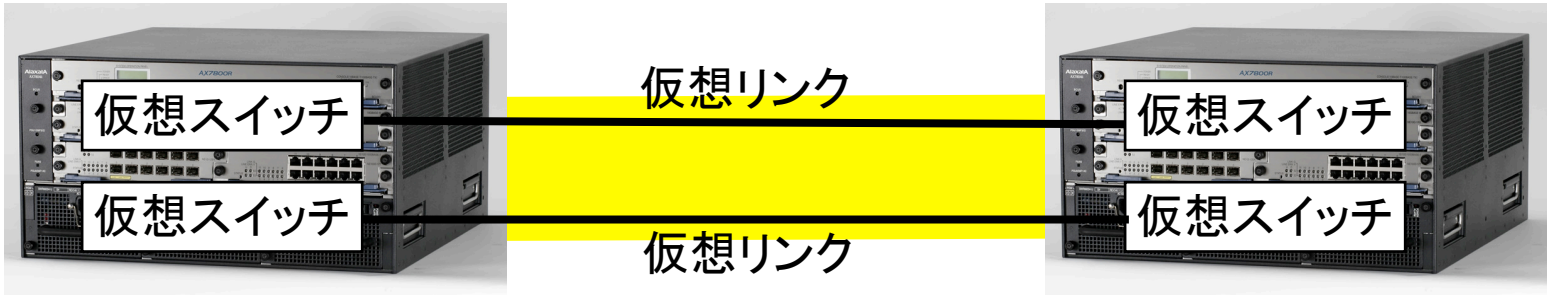
インターネット内のパケット

MPLS ヘッダ	IP ヘッダ	ペイロード

イーサネットの仮想化のためのしくみ: VLAN

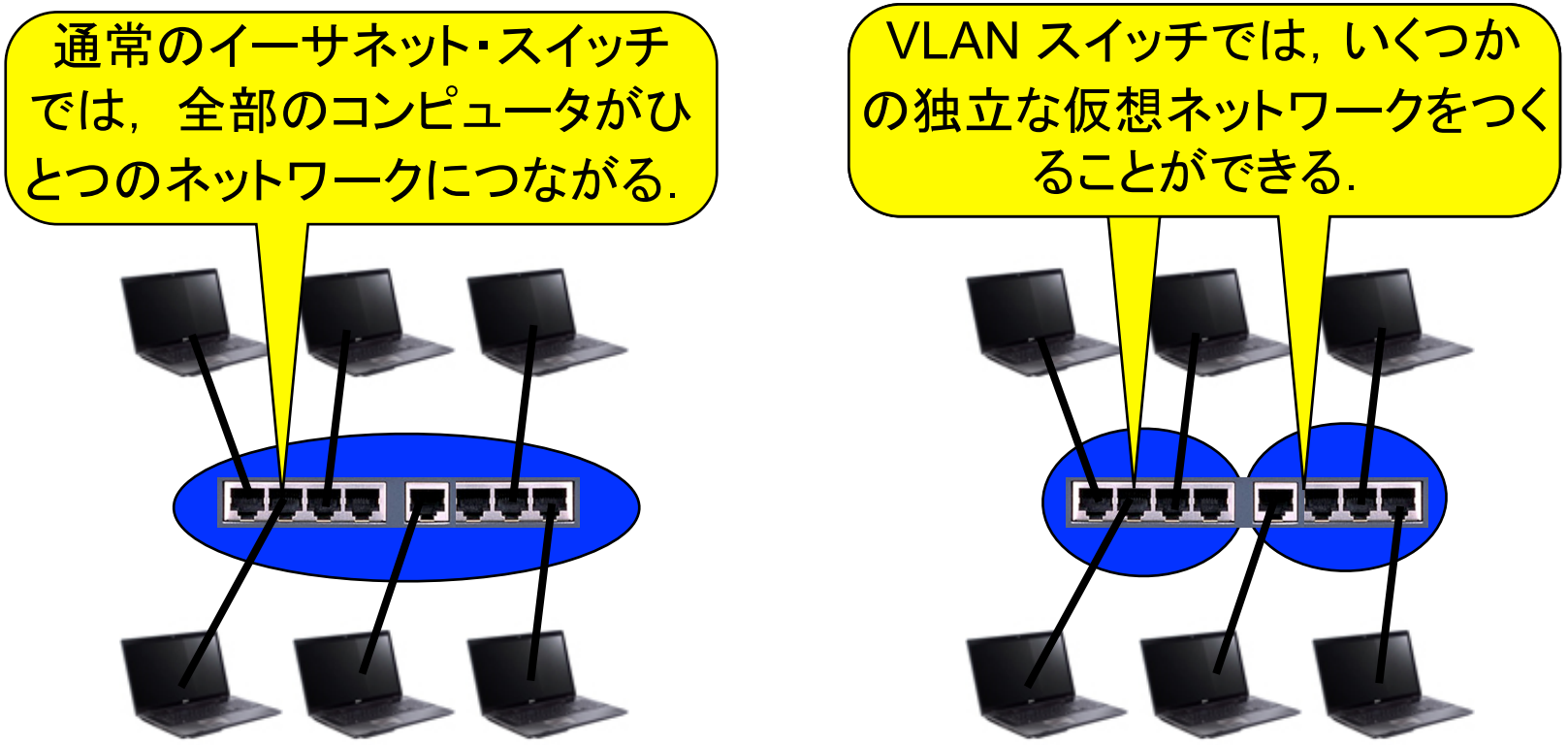
■ イーサネット上に仮想ネットワークをつくるためのしくみ VLAN が IEEE で標準化されている (IEEE802.1Q).

■ VLAN のポイントはスイッチの仮想化とリンクの仮想化



VLAN におけるスイッチの仮想化

■ VLAN をつかうと，1 個のスイッチを複数の独立なスイッチのようにつかうことができる．



VLAN におけるリンクの仮想化

■ VLAN では 1 本の物理リンクを仮想的に複数のネットワーク (イーサネット) で使用できる.

- ◆ このようなリンクをトランクリンクという.
- ◆ トランクリンクでは VLAN タグによって仮想ネットワークをくべつする.

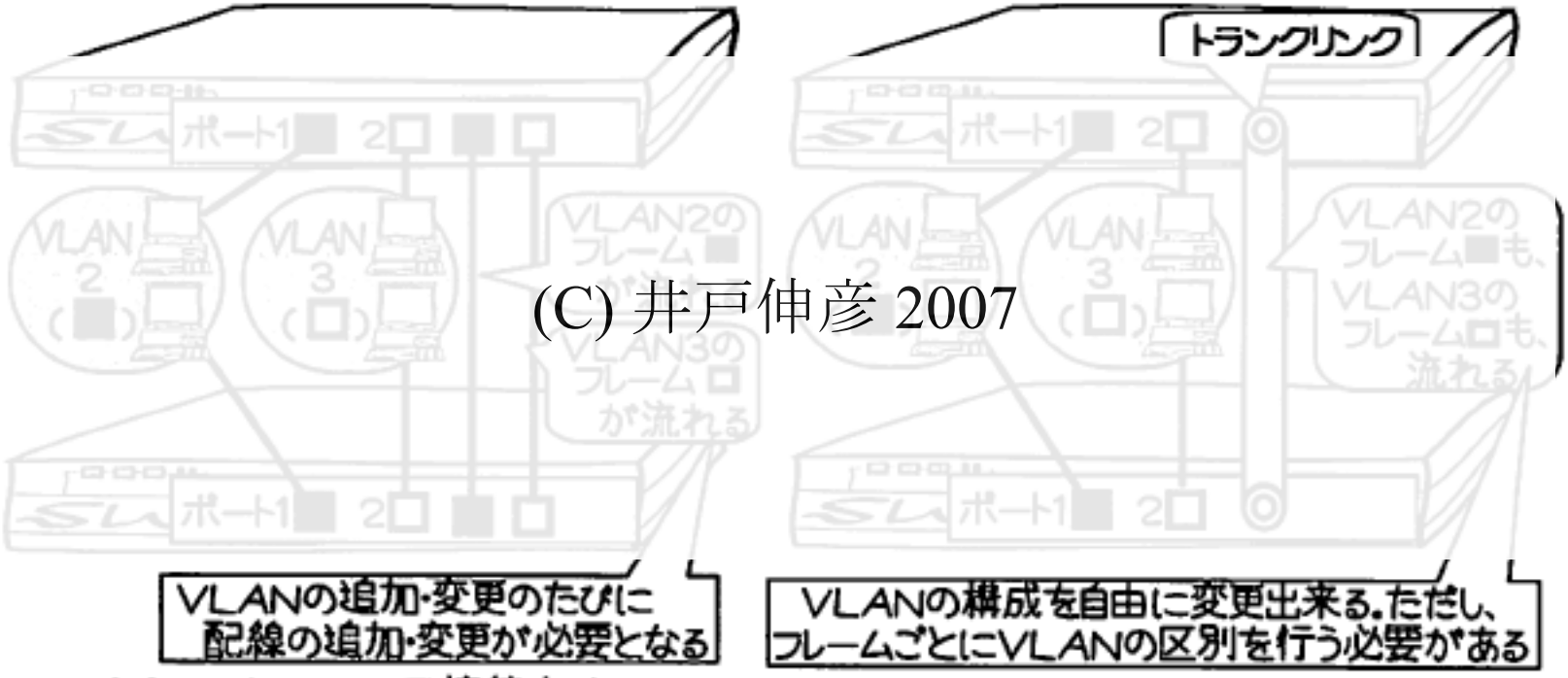


図9-23 トランクリンクの必要性

VLAN タグとパケット・フォーマット

■ VLAN タグは MAC ヘッダのなかにうめこまれる.

◆ VLAN パケットのフォーマットは IEEE802.1Q によって標準化されている.

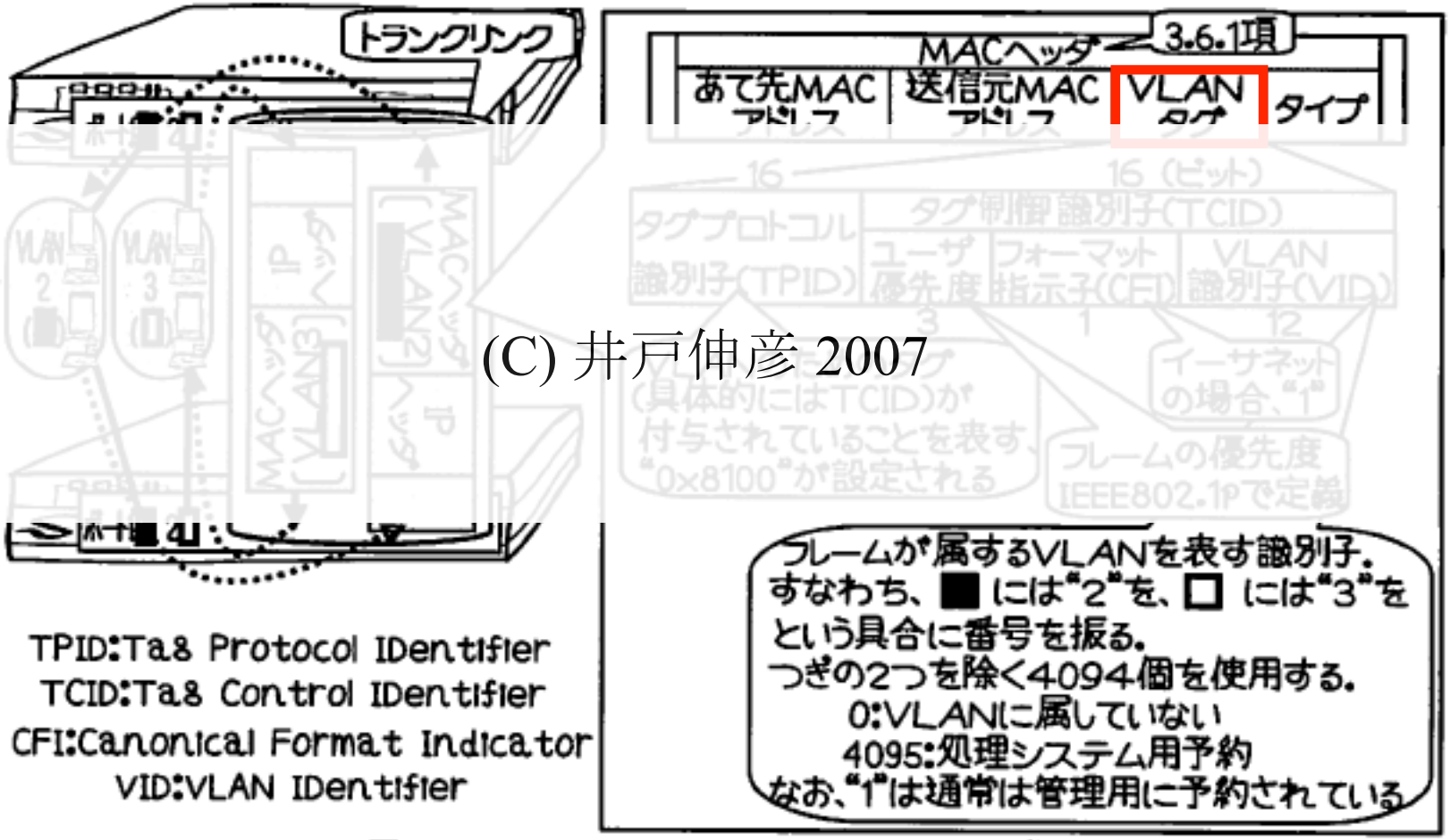


図9-24 IEEE802.1QによるVLANタグ形式

VLAN の利点 -- 企業などの組織の場合

■ VLAN は物理的にちらばった組織を容易に論理的にまとめることができる.

◆ VLAN をつかわないと組織がかわるたびに物理配線を変更しなければならない.

◆ VLAN をつかえば物理配線はかえずに仮想リンクをかえることができる.

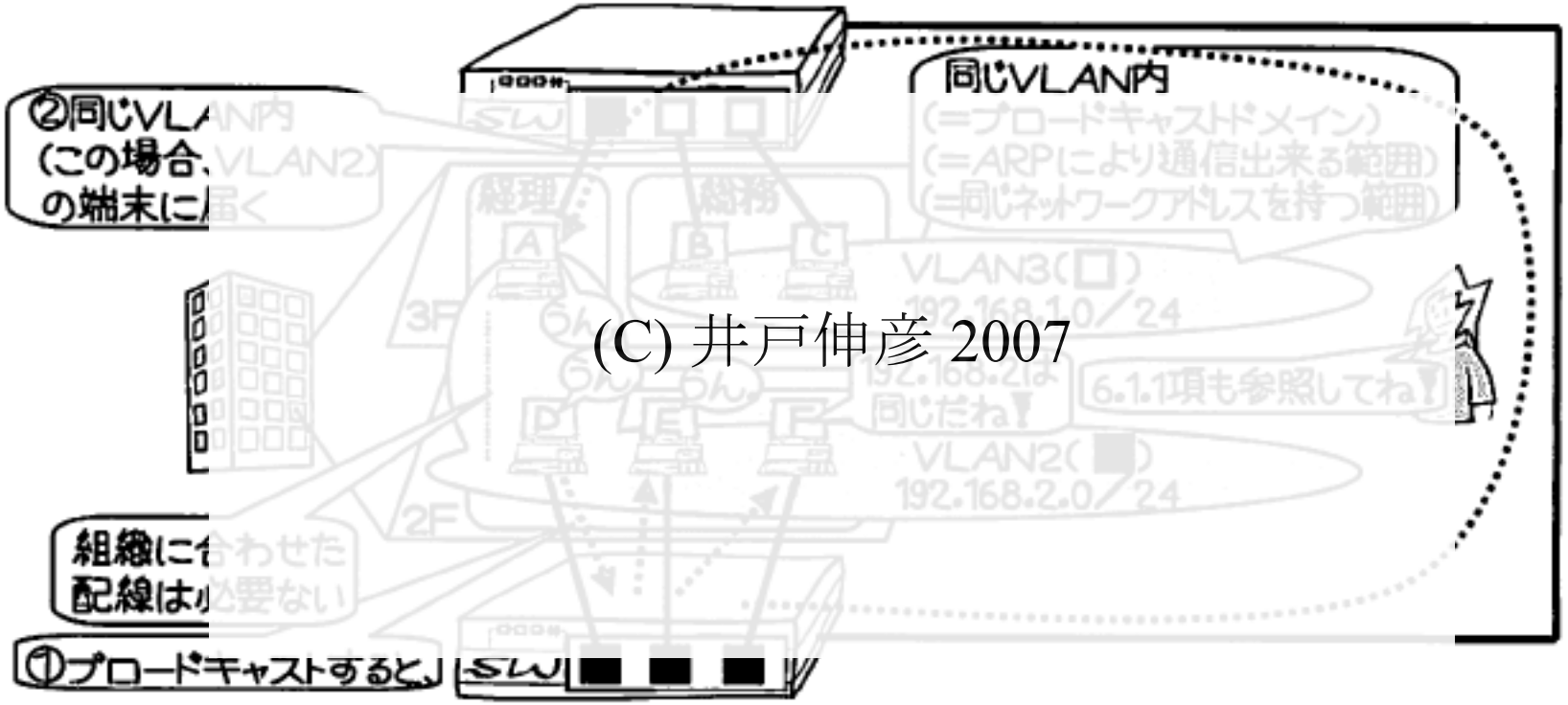


図9-21 VLANの構成例

仮想化とは?

■ 仮想化とは, 物理的なコンピュータやネットワークがもつ機能とは質や量においてことなる機能を実現することである.

■ 仮想化の分類

◆質の仮想化

◆量の仮想化

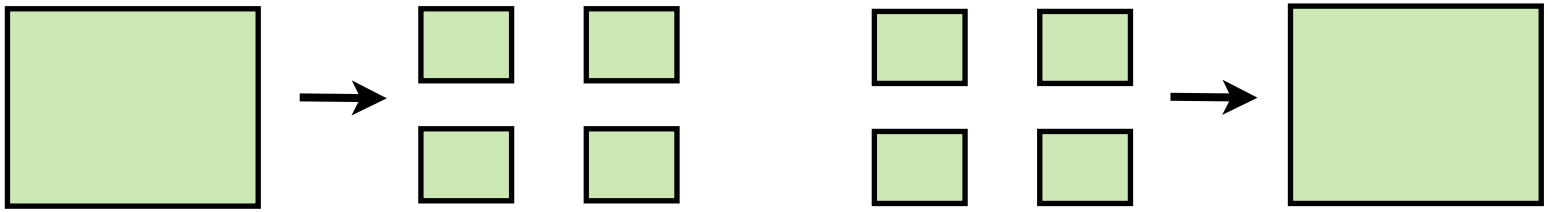
●分割型の仮想化

●融合型の仮想化

量の仮想化と質の仮想化

■ 量の仮想化

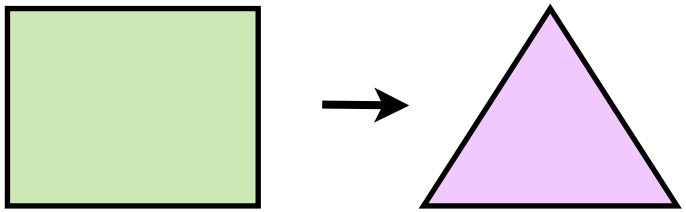
◆ 仮想化によって数量をふやす，またはへらす.



◆ 仮想化前と質はかわらない (かもしれない).

■ 質の仮想化

◆ 仮想化によって質をかえる.



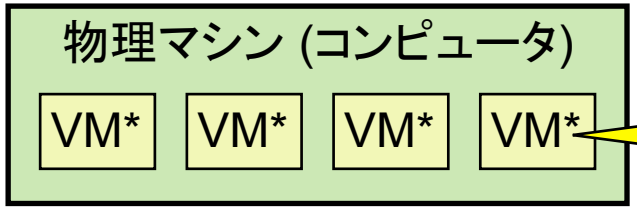
◆ 仮想化前と数量はかわらない (かもしれない).

分割型仮想化と融合型仮想化

■ 分割型仮想化

◆ 1 個のコンピュータやネットワークのなかに複数の仮想的なコンピュータやネットワークをつくる.

◆ 例:



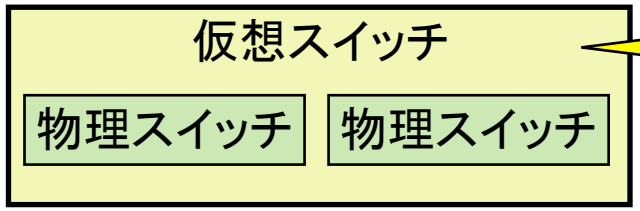
物理マシンとは質がことなる VM をつくることも可能

*VM: 仮想マシン

■ 融合型仮想化

◆ 複数のコンピュータやネットワークを仮想的に 1 個のコンピュータやネットワークにみせる.

◆ 例:

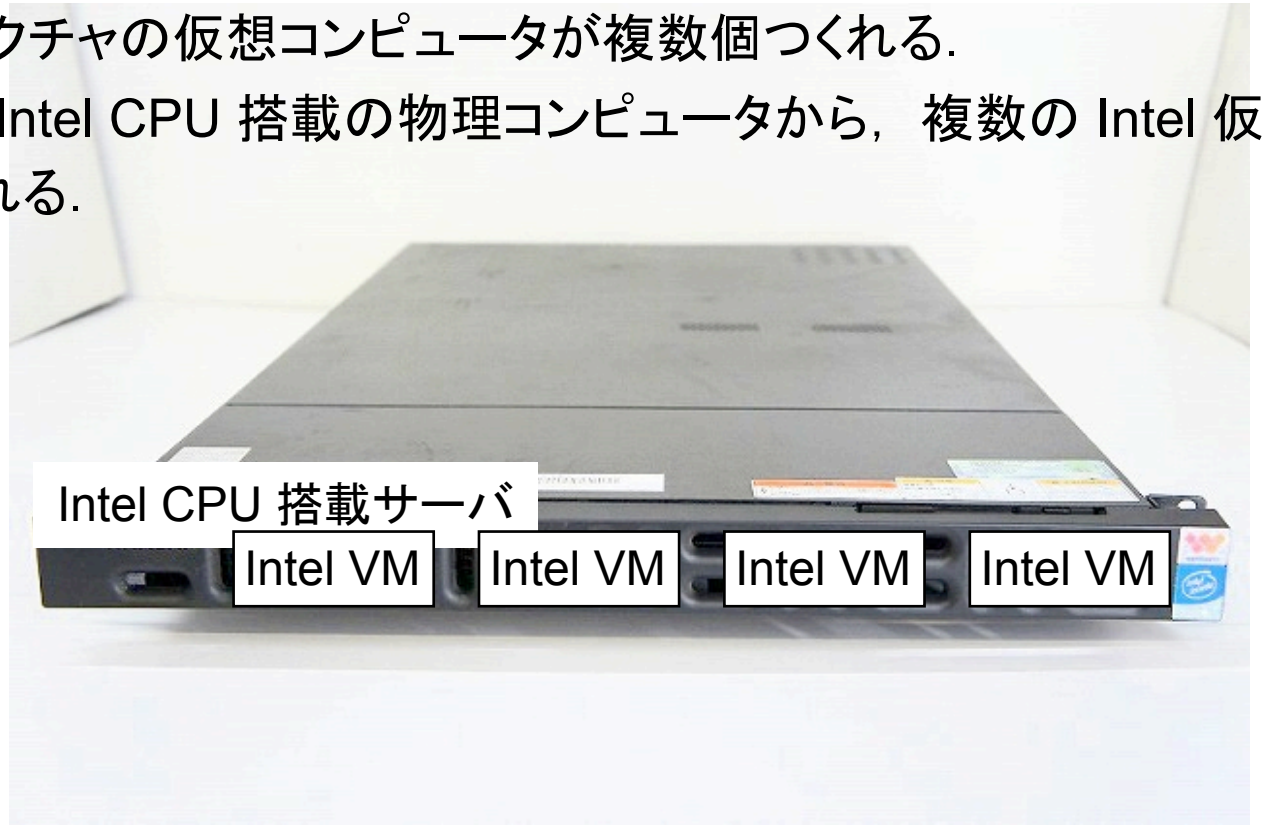


物理スイッチとは質がことなる仮想スイッチをつくることも可能

■ 現在, 実用化・実験されている仮想化技術のおおくは分割型仮想化を実現している.

コンピュータ (サーバ) の仮想化

- 融合型の仮想化はほとんどない (?)
- 量の仮想化 (分割型)
  - 最近話題になるサーバ仮想化は量の仮想化: 物理コンピュータとおなじアーキテクチャの仮想コンピュータが複数個つくれる.
  - たとえば Intel CPU 搭載の物理コンピュータから, 複数の Intel 仮想マシンがつくれる.



コンピュータ (サーバ) の仮想化 (つづき)

■ 質の仮想化 (分割型)

◆ 物理コンピュータとはことなるアーキテクチャのコンピュータをつくる.

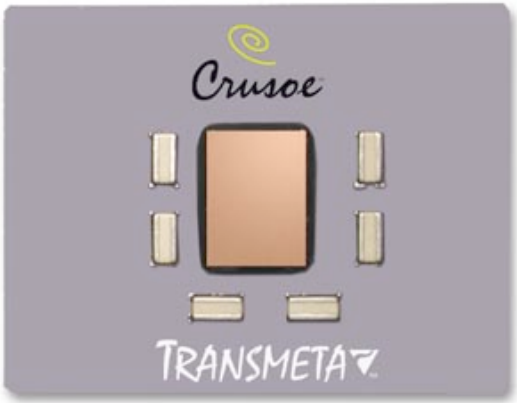
■ 質の仮想化の例

◆トランスメタ社の Crusoe, Efficeon:
Intel x86 の命令を独自の命令 (VLIW) に翻訳して実行する CPU (2002-2004 年ごろ).

◆Intel Pentium Pro (とそれ以降の CPU):
複雑な x86 命令を単純な RISC 風命令にハードウェアで翻訳して実行する.

◆P コード・マシン: コンパイラの仕事が容易になるような仮想マシンを定義して, シミュレータで実行する. N. ヴィルトの Pascal P が有名.

◆バイトコード・マシン: Smalltalk, Java などの言語はバイトコードとよばれる仮想的な機械語を実行する仮想マシン (シミュレータ) で実行される.



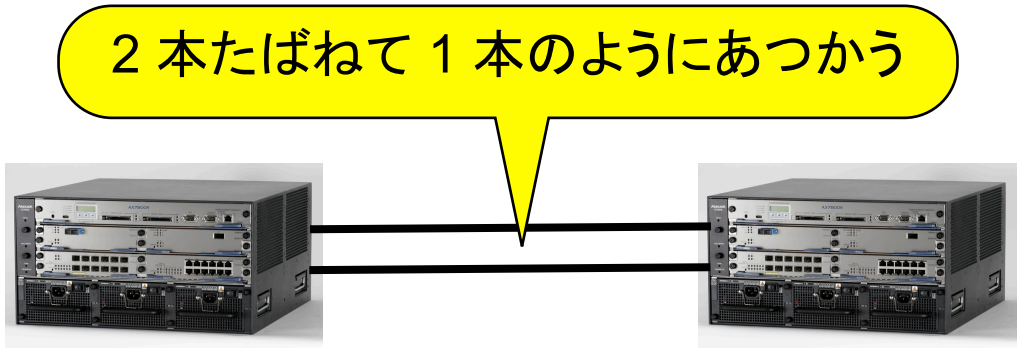
ネットワークの仮想化

■ 分割型仮想化の例

- ◆ VLAN
- ◆ VPN

■ 融合型仮想化の例

◆リンク･アグリゲーション: 複数の物理リンクをたばねて，1 個のリンクにみせる.



プライベート・ネットワークとネットワーク仮想化

- プライベート・ネットワークの利点のひとつは，パブリック・ネットワークではつかえないプロトコルが自由につかえること.
- ところが，従来の VPN では基本的に IP しかつかえない.
- 「ネットワーク仮想化」の研究とは?
  - 従来の VPN と同様のプライベートなネットワークをつくり，そのうえでさらに自由に新プロトコルが開発・使用できるようにする.
  - 新プロトコルがつかえるためにはネットワーク・ノード (スイッチ，ルータ) がプログラマブルであることが重要
    - 仮想ネットワークの所有者がネットワーク・ノードを自由にプログラムできる (C などでプログラム開発できる) ようにする.

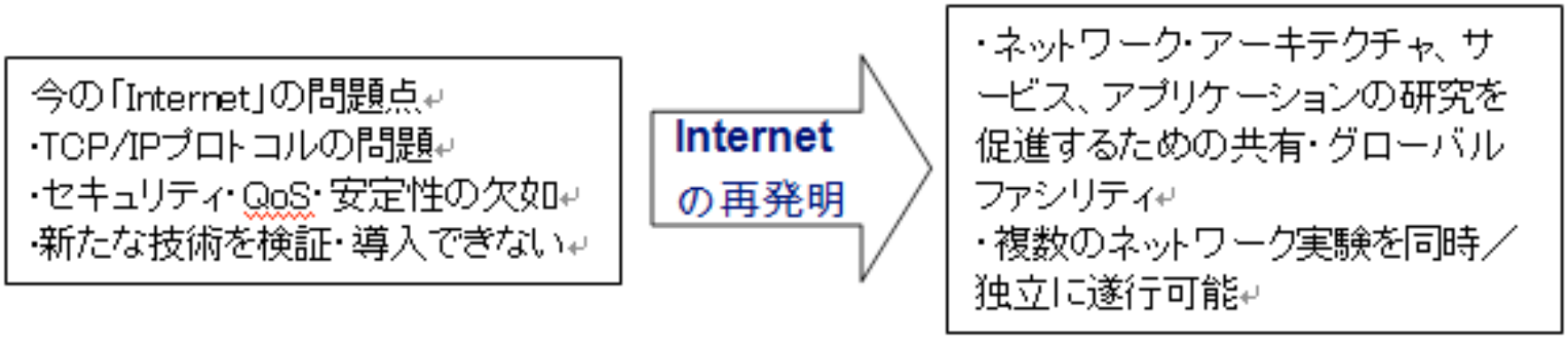
新世代ネットワーク研究とネットワーク仮想化

■ インターネットの限界

- ◆ インターネットが登場してから 30 年以上経過し，現在のニーズにはかならずしもあっていないところがある.
- ◆ 既存のインターネットに対してセキュリティ，QoS (Quality of Service)，安定性などの面で限界があることが指摘されている.

■ クリーンスレート構想

- ◆ 米国では 2000 年ごろから既存のインターネットを根本から見直して将来のネットワークを構築する「クリーン・スレート・インターネット構想」が議論されるようになった.



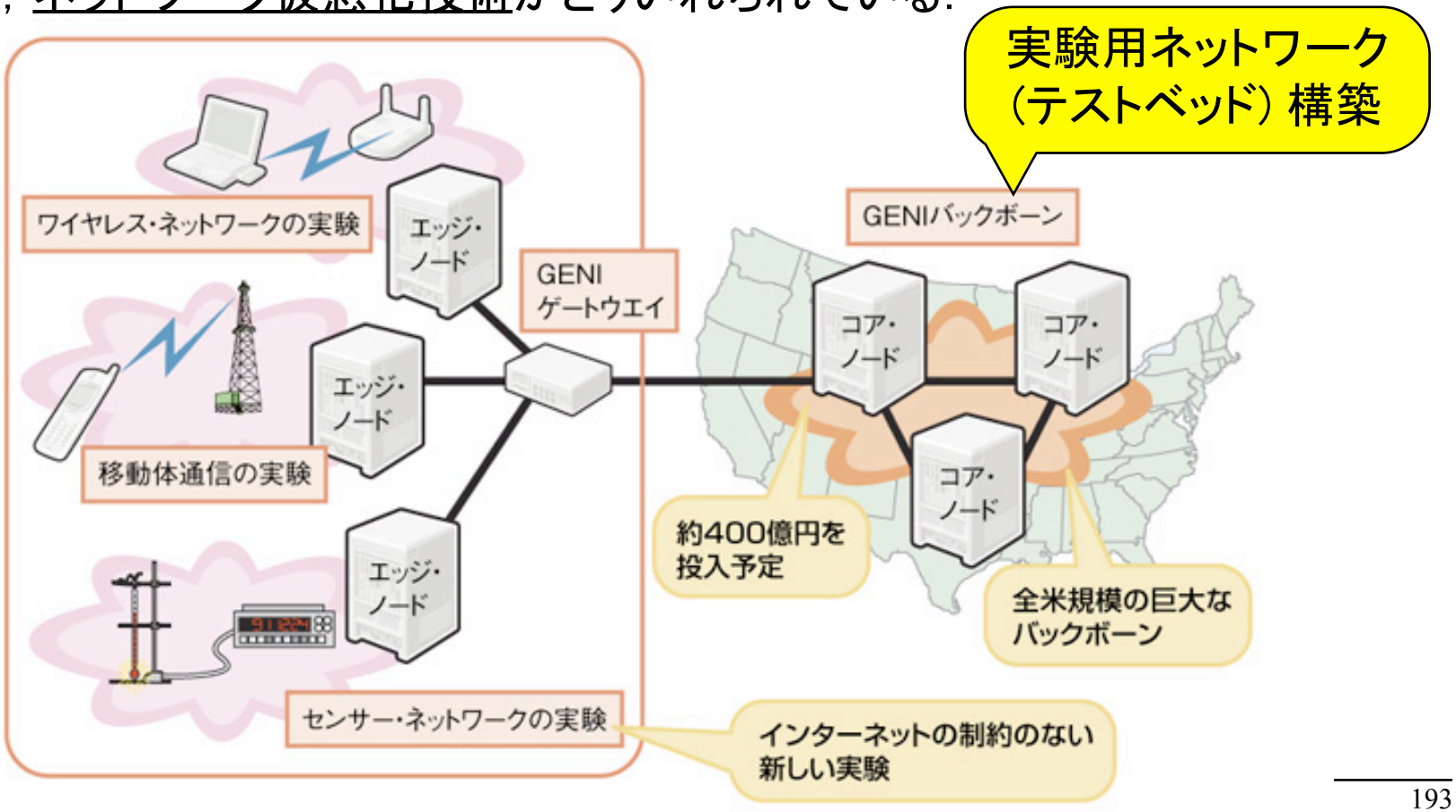
アメリカにおける新世代ネットワークとネットワーク仮想化

- インターネットはネットワークの進化をとめている ?!
  - アメリカなどでは，インターネットとはまったくちがう，あたらしいネットワークをつくる必要があることが合意された.
- FIND (Future Internet Design)
  - アメリカの科学財団 (NSF) ではあたらしいネットワークをつくるため，FIND という研究ファンド・プログラムを設置した.

EU (ヨーロッパ) では FP7 (7th Framework Programme) というプロジェクトをやっている

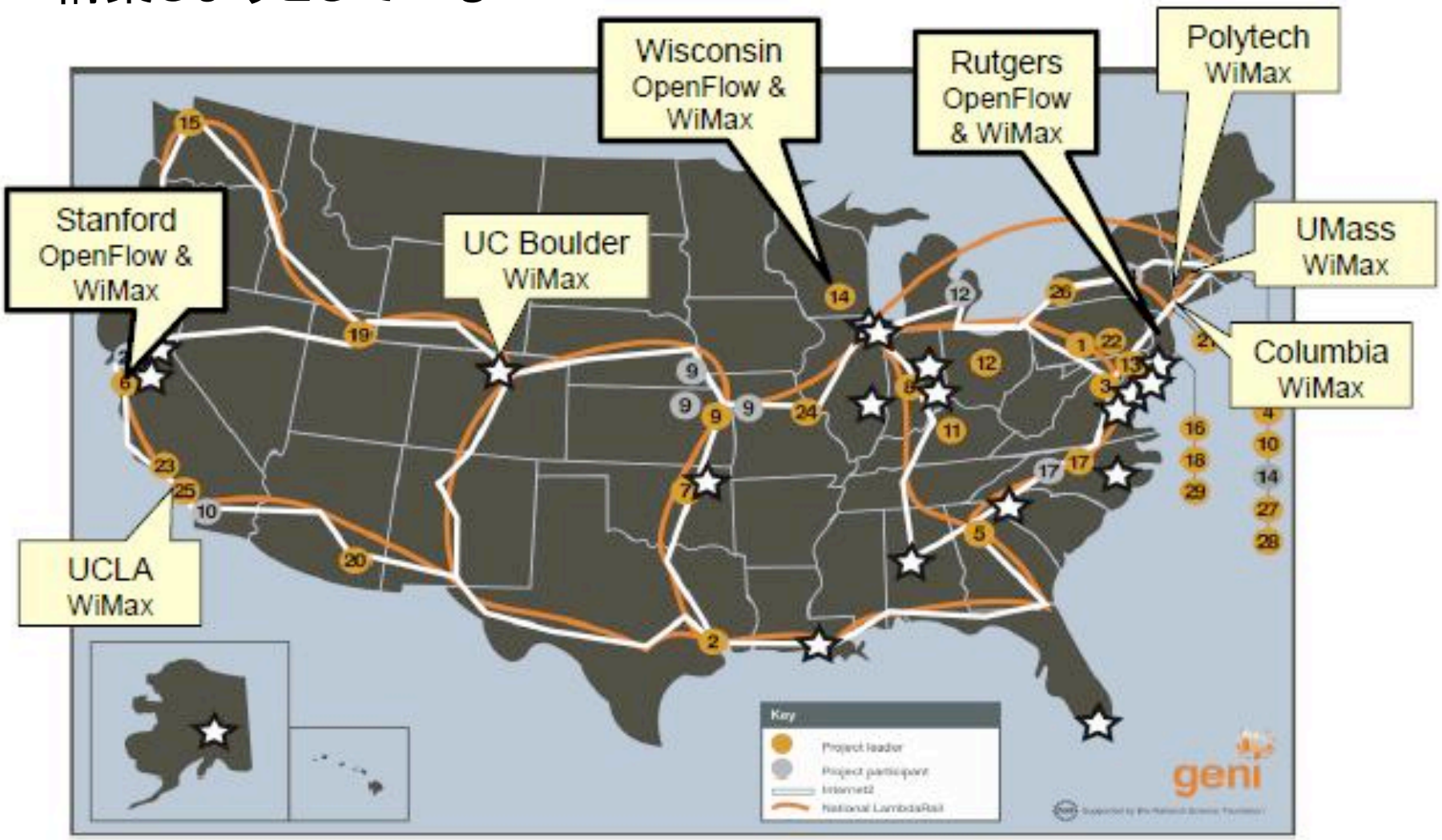
アメリカにおける新世代ネットワークとネットワーク仮想化（つづき）

- GENI (Global Environment for Network Innovations)
  - 現在のインターネットにはとりいれるのがむずかしい IP に依存しない新技術を開発するために GENI というプロジェクトが開始された.
  - GENI では，かぎられた物理ネットワーク上でさまざまな実験ができるように，ネットワーク仮想化技術がとりいれられている.



アメリカにおける新世代ネットワークとネットワーク仮想化（つづき）

■ GENI では全米規模のバックボーン (骨格となるネットワーク) を構築しようとしている.



ソフトウェア定義ネットワーク

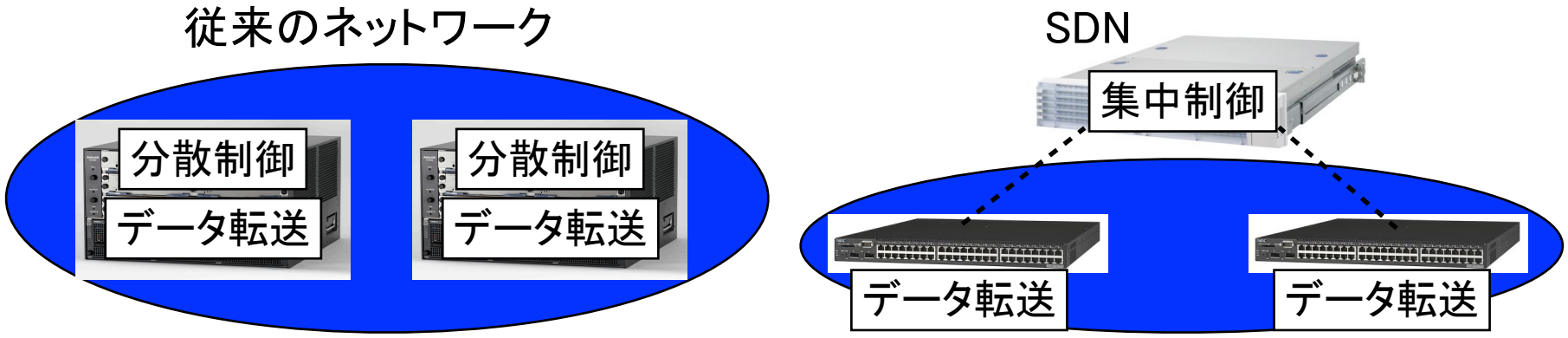
■ ソフトウェア定義ネットワーク (Software Defined Network: SDN)

◆ これまでは機種依存だったスイッチやルータの制御を, 外部のコントローラ から一元的におこなえるようにしたネットワーク.

◆ OpenFlow が代表的な制御方式.

■ SDN の基本思想: コントローラ-データ分離

◆ コントローラがスイッチに規則を配布し, 各スイッチはそれにもとづいて動作する.



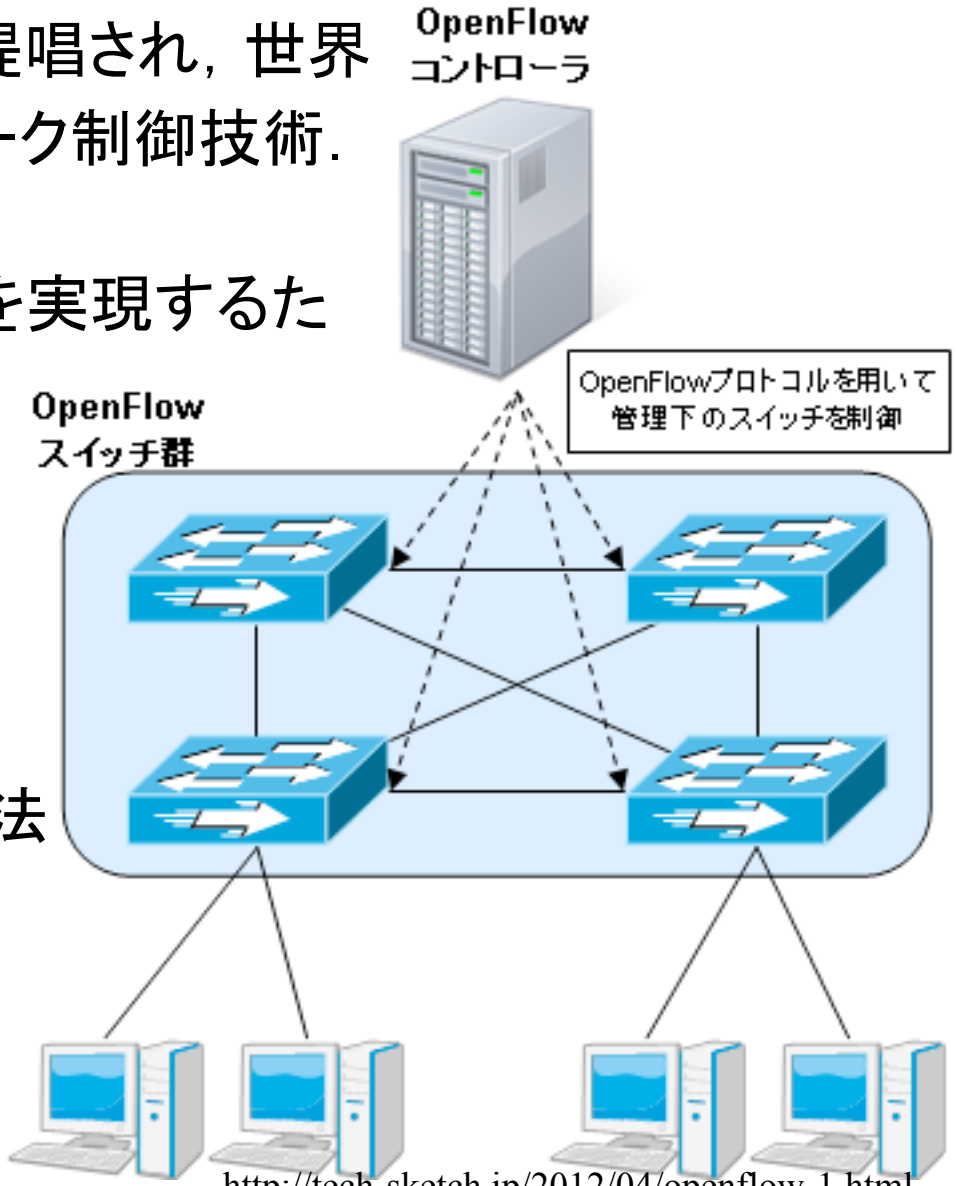
ソフトウェア定義ネットワーク（つづき）

■ ソフトウェア定義ネットワークの利点

- ◆ スイッチング，ルーティングの方法・方針 (ポリシー) などをかんたんにためすことができる.
- ◆ スイッチやルータの従来は外部から制御できなかった部分が，制御できるようになる.
- ◆ 従来は機種依存の方法でしか制御できなかったスイッチやルータの機能が統一的な方法で制御できるようになる.

OpenFlow 概論

- 2008 年に Stanford 大学で提唱され，世界中で注目されているネットワーク制御技術.
  - イーサネット上でつかわれる.
- ソフトウェア定義ネットワークを実現するための最有力な方法.
- OpenFlow ネットワークのハードウェア構成
  - OpenFlow スイッチ
  - OpenFlow コントローラ
- スイッチとコントローラの通信法
  - OpenFlow プロトコルによる.



http://tech-sketch.jp/2012/04/openflow-1.html

OpenFlow 概論（つづき）

- 通常のスイッチやルータで構成された既存のネットワーク上に OpenFlow スイッチをおいて使用することができる.
  - つまり，従来技術と共存できる.
- ベンダのなかでは NEC がもっとも積極的にとりくんでいる.
  - NEC の製品 →

OpenFlow のしくみ

- OpenFlow による制御は条件と動作との組 (規則) により指定される.
- 条件は処理の対象であるパケットを特定する.
  - MAC アドレス, IP アドレス, TCP/UDP ポート番号など, 物理層 (L1) からトランスポート層 (L4) まで, どの階層のデータもあつかえる.
  - 例: TCP ポート番号が 80 のパケット.
- 動作は条件に合致したパケットに対し行う動作を規定する.
  - 他のポートへの転送, ヘッダのかきかえ, パケットの破棄などが指定できる.
- 規則の例: TCP ポート番号が 80 のパケットは破棄する.

OpenFlow による制御の例

- OpenFlow で「ルータ」をつくることも可能.
- IP/Ethernet のルータは「とどいたパケットの IP アドレスの判定結果にもとづいて，そのパケットの MAC アドレスをかきかえて，しかるべきインタフェースに転送する」.
  - 受信者 MAC アドレスをネクストホップの MAC アドレスにかきかえる.
  - 送信者 MAC アドレスをそのルータの MAC アドレスにかきかえる.
  - ネクストホップにつながるネットワーク・インタフェースから出力する.
- これをサブネットごとに規則として記述して設定すれば，OpenFlow スイッチはルータとして機能する.
  - つまり，ルーティング・テーブルのかわりに規則のならびを使用する.

ルーティング・テーブルの例

あてさき	ネクストホップ
172.17.4.0/24	**192.168.2.251**
192.168.1.0/24	*（直接）
192.168.2.0/24	*（直接）

等価な OpenFlow の規則

if 受信者 IP アドレスの先頭 24 ビットが 172.17.4 **then**
  受信者 MAC アドレス =
    192.168.2.251 に対応する MAC アドレス
  送信者 MAC アドレス = 自 MAC アドレス
  受信者 MAC アドレスにつながるインタフェースに出力

OpenFlow による制御の例（つづき）

■ OpenFlow による制御の長所と短所

◆長所: IP やイーサネットの制御は一層 (IP, イーサネット) でとじていたが, OpenFlow では層にまたがる制御ができる.

IP アドレスだけをみている

IP アドレスと MAC アドレスをあわせて読み書きしている

ルーティング・テーブルの例

あてさき	ネクストホップ
172.17.4.0/24	**192.168.2.251**
192.168.1.0/24	*（直接）
192.168.2.0/24	*（直接）

等価な OpenFlow の規則

if 受信者 IP アドレスの先頭 24 ビットが 172.17.4 **then**
受信者 MAC アドレス =
192.168.2.251 に対応する MAC アドレス
送信者 MAC アドレス = 自 MAC アドレス
受信者 MAC アドレスにつながるインタフェースに出力

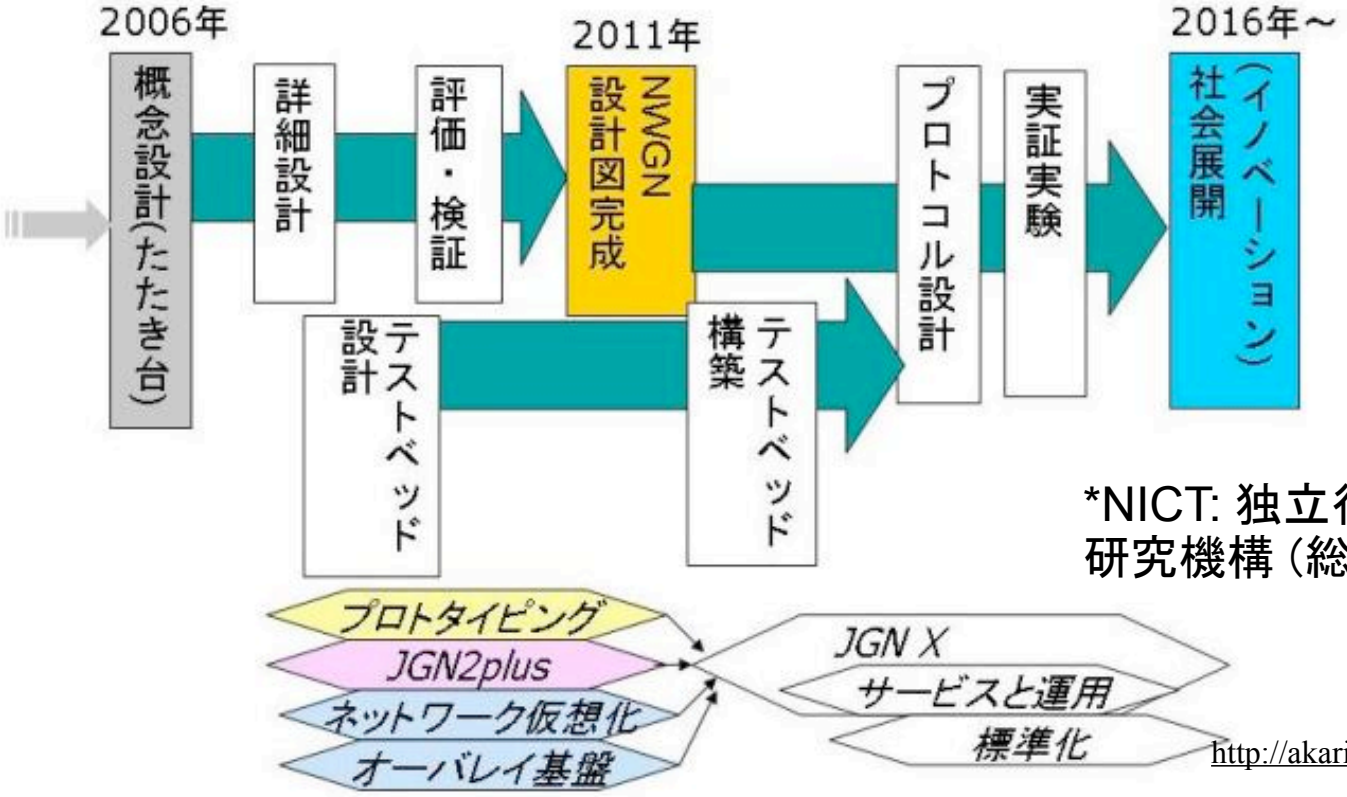
◆短所: IP/Ethernet だけでしかつかえない.

●IP, Ethernet 以外のプロトコルではつかえない.

●IP とイーサネット以外のリンク層, イーサネットと IP 以外のネットワーク層のくみあわせでもつかえない.

AKARI -- 日本における新世代ネットワーク研究

- 2015 年に新世代ネットワークの基礎技術を実現することをめざして, NICT* 中心に 2006 年からつづけられてきたプロジェクト.
- まったくあたらしいネットワークアーキテクチャを確立し, それにもとづいたネットワーク設計図を作成することを目的としている.
- クリーンスレートをめざしている -- インターネットにとらわれない.



*NICT: 独立行政法人 情報通信研究機構 (総務省の外郭組織)

http://akari-project.nict.go.jp/index2.htm

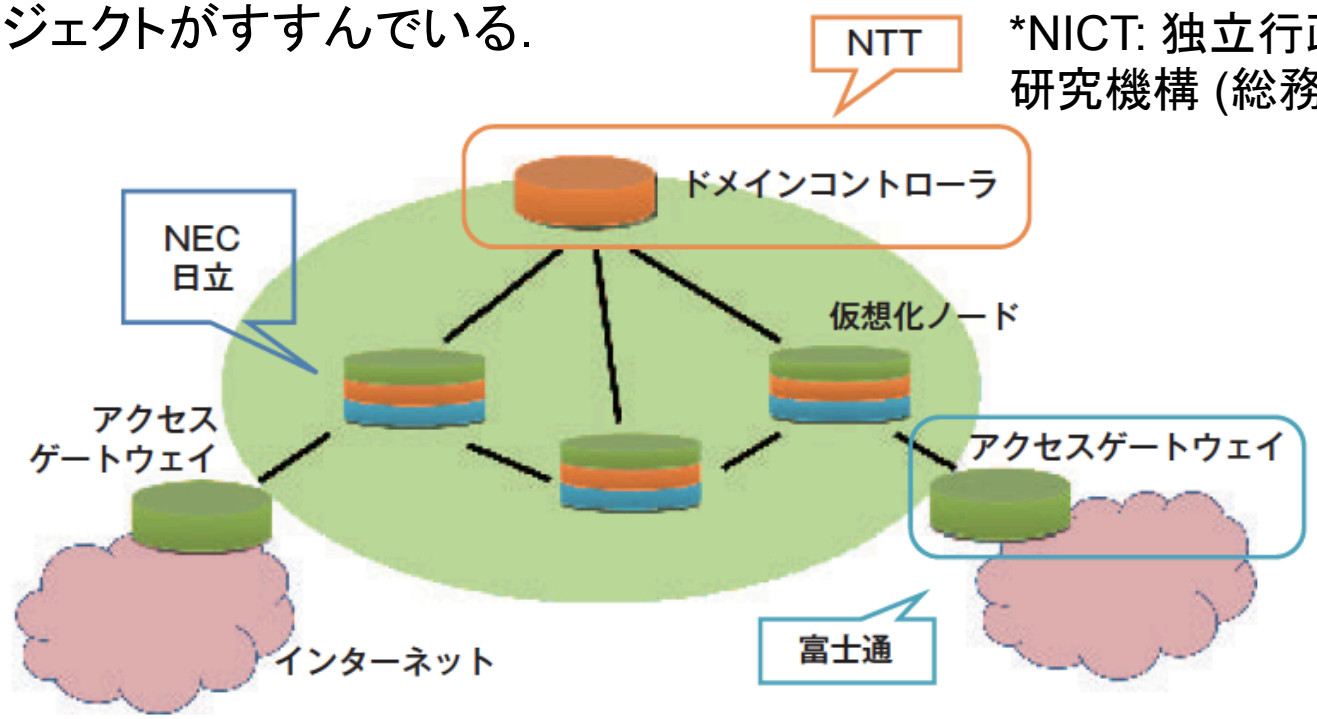
VNode -- 日本におけるネットワーク仮想化研究

■ 日本の代表的なネットワーク仮想化研究プロジェクトとして「仮想化ノード (VNode) プロジェクト」(とその後継プロジェクト) がある.

◆ VNode プロジェクト (2009-2010) では, NICT という場で東大, NTT, 富士通研究所, NEC, 日立が共同研究してきた.

◆ 現在 (2011-2014) はその後継プロジェクトとして NICT の委託研究プロジェクトがすすんでいる.

*NICT: 独立行政法人 情報通信研究機構 (総務省の外郭組織)



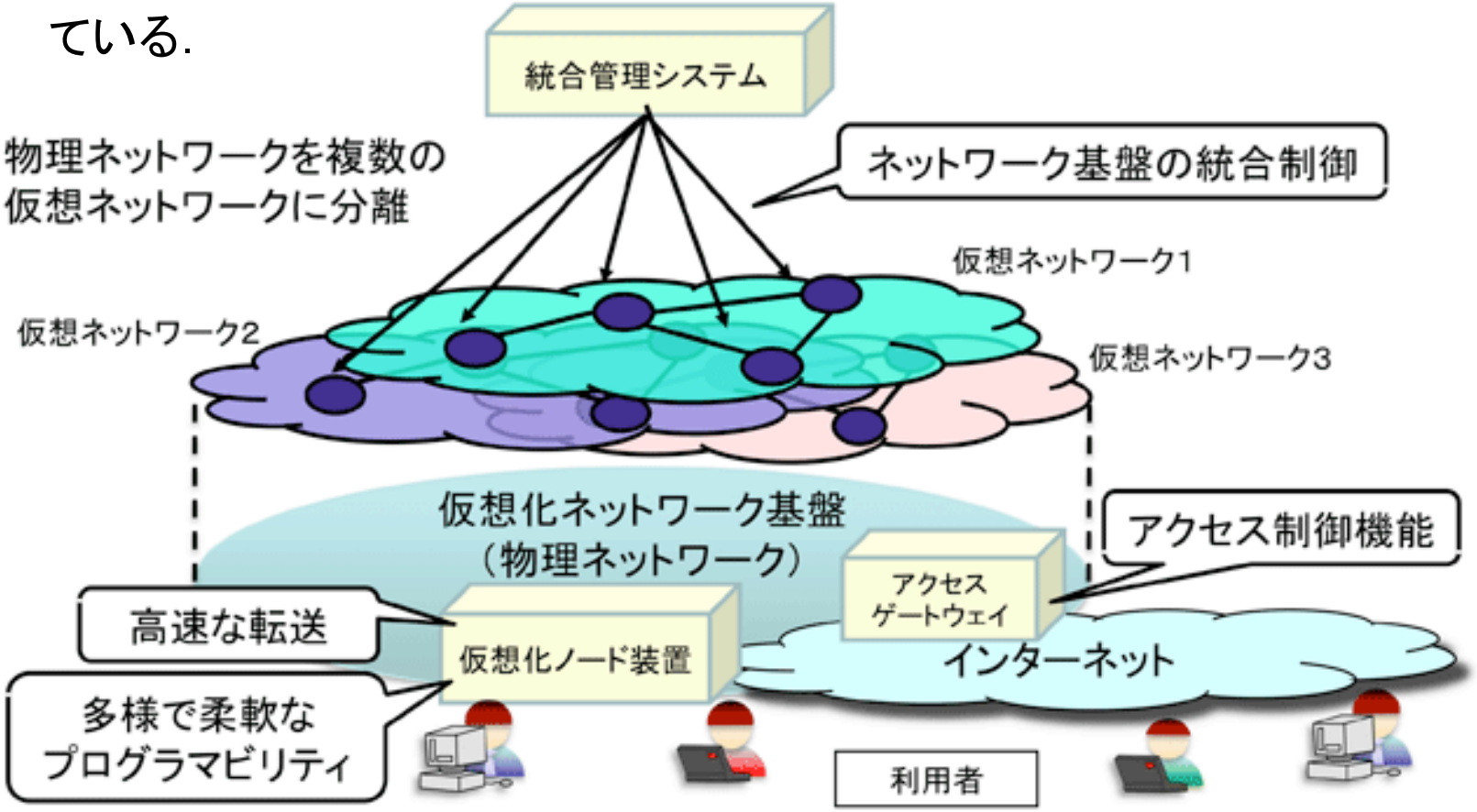
http://www.nict.go.jp/publication/NICT-News/.../NICT_NEWS_1006_J.pdf

図6●仮想化ノード・プロジェクトでの各社の分担

VNode -- 日本におけるネットワーク仮想化研究（つづき）

■ VNode プロジェクトの成果

- ◆ 仮想化ノード (VNode) という装置によって構成される物理ネットワーク (ネットワーク仮想化基盤という) 上に仮想ネットワーク (スライスという) が生成できるようにした.
- ◆ JGN-X という研究ネットワーク上でこのネットワーク仮想化機能を提供している.

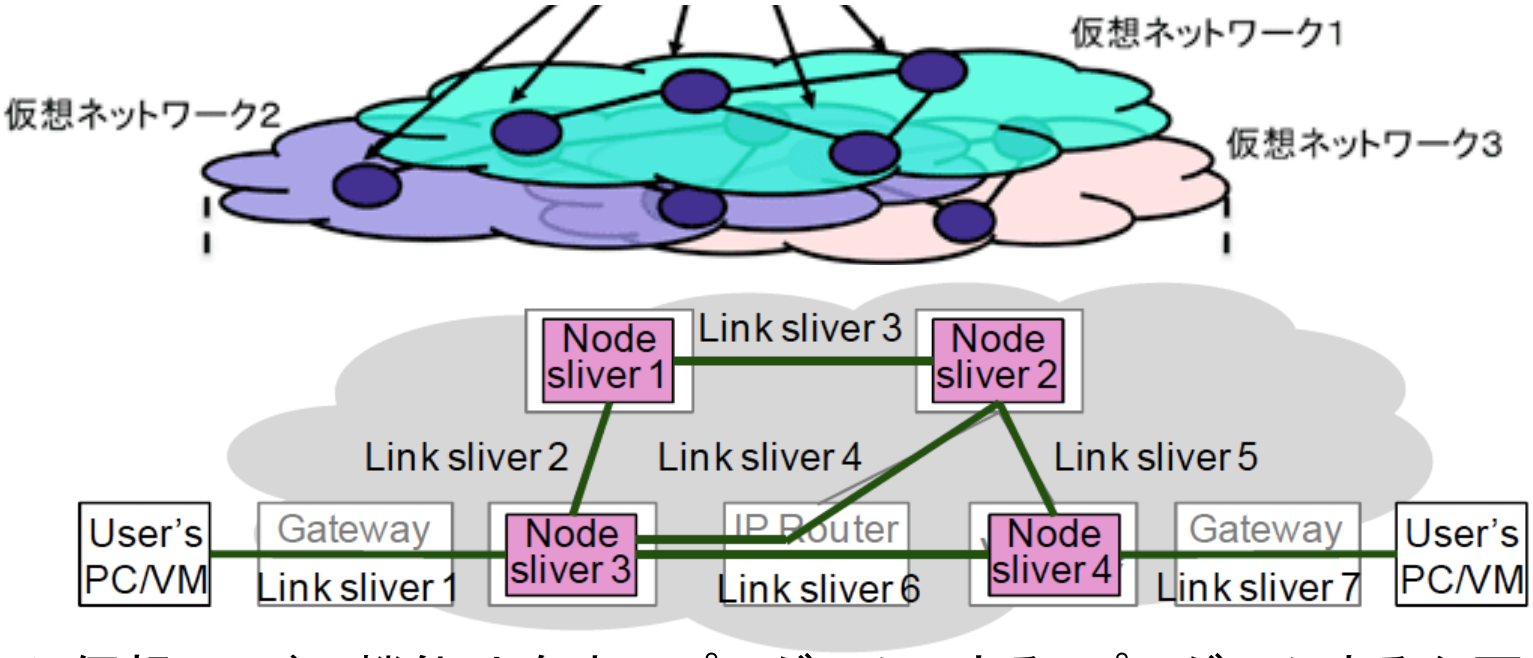


VNode における仮想ネットワークのモデル

■ スライス (仮想ネットワーク)

◆自由な構造のスライスをつくることができる: 任意の個数の仮想ネットワーク・ノードをつくり, 物理的なリンクに制約されない仮想リンクでむすぶことができる.

◆「スライス」とよばれる理由: 物理的なネットワークを仮想的に分割して (スライスして) 仮想ネットワークをつくることができるから.

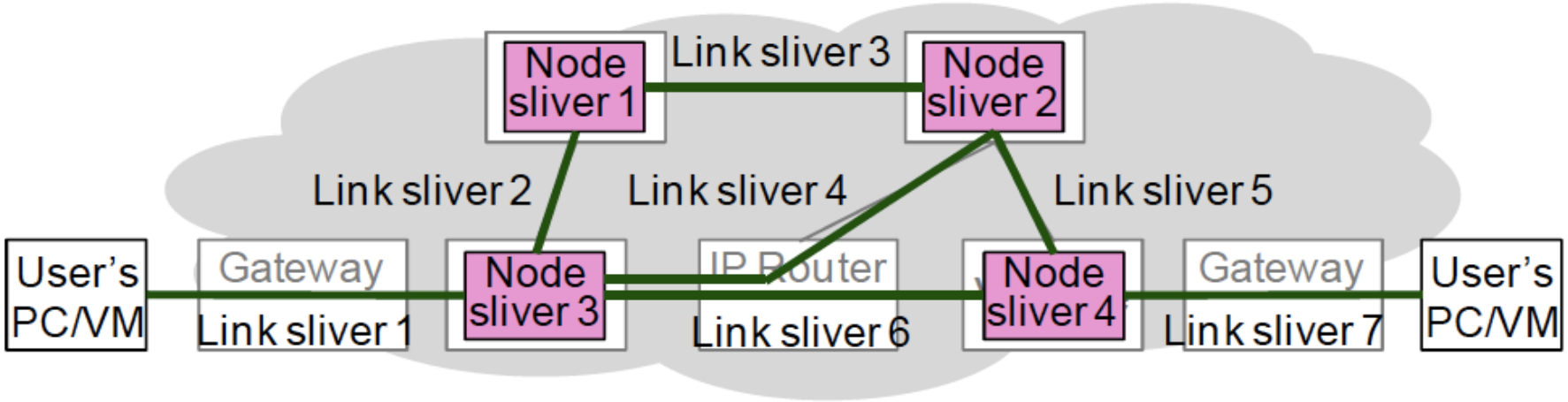


◆仮想ノードの機能は自由にプログラムできる (プログラムする必要がある).

VNode における仮想ネットワークのモデル（つづき）

■ 仮想ノードと仮想リンク

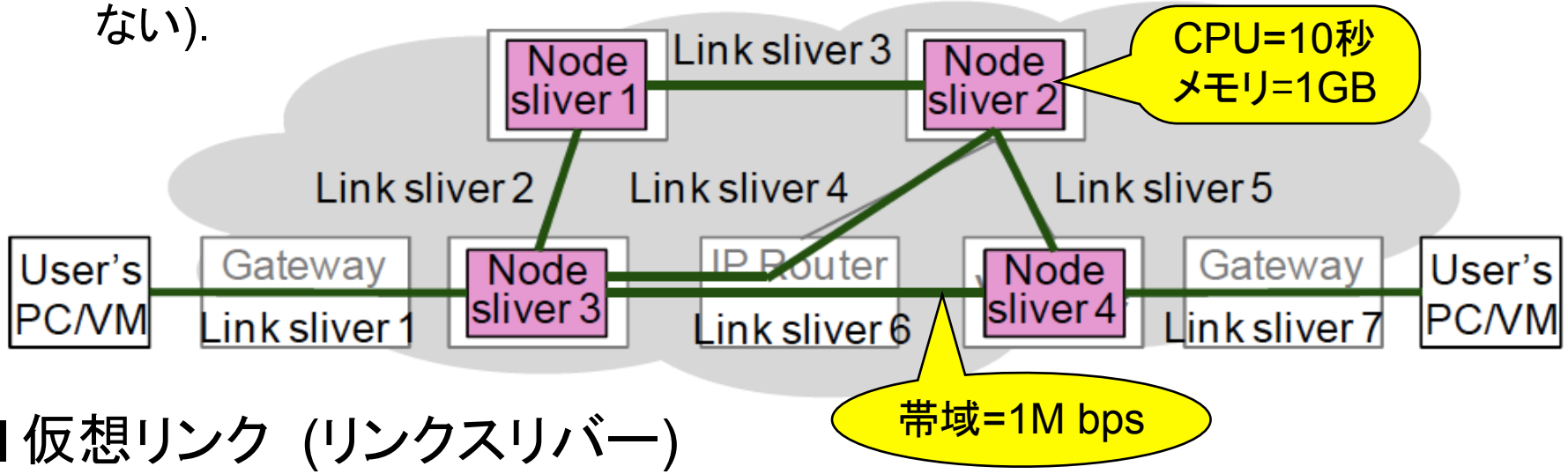
- ◆ 仮想化されたネットワーク資源はスリバー (sliver) とよばれる.
- ◆ VNode プロジェクトではとくに仮想ノードをノードスリバー，仮想リンクをリンクスリバーと呼んでいる．(が，ここではより一般的な用語である仮想ノードと仮想リンクということばをつかう.)



VNode における仮想ネットワークのモデル (つづき)

■ 仮想ノード (ノードスリバー)

- ◆ VNode (プログラマ) 内の計算資源 (CPU 時間, メモリなど) をあらわす.
- ◆ おもにプロトコルの処理や制御などにつかう -- その機能は自由にプログラムできる.
- ◆ どんなパケット・フォーマットでもあつかえる (IP やイーサネットにしばられない).



■ 仮想リンク (リンクスリバー)

- ◆ ことなる VNode 内の 2 つの仮想ノードをつなぐ仮想リンクのネットワーク資源 (帯域など) をあらわす.
- ◆ どんなパケット・フォーマットでもあつかえる.

VNode の構造: プログラマとリダイレクタ

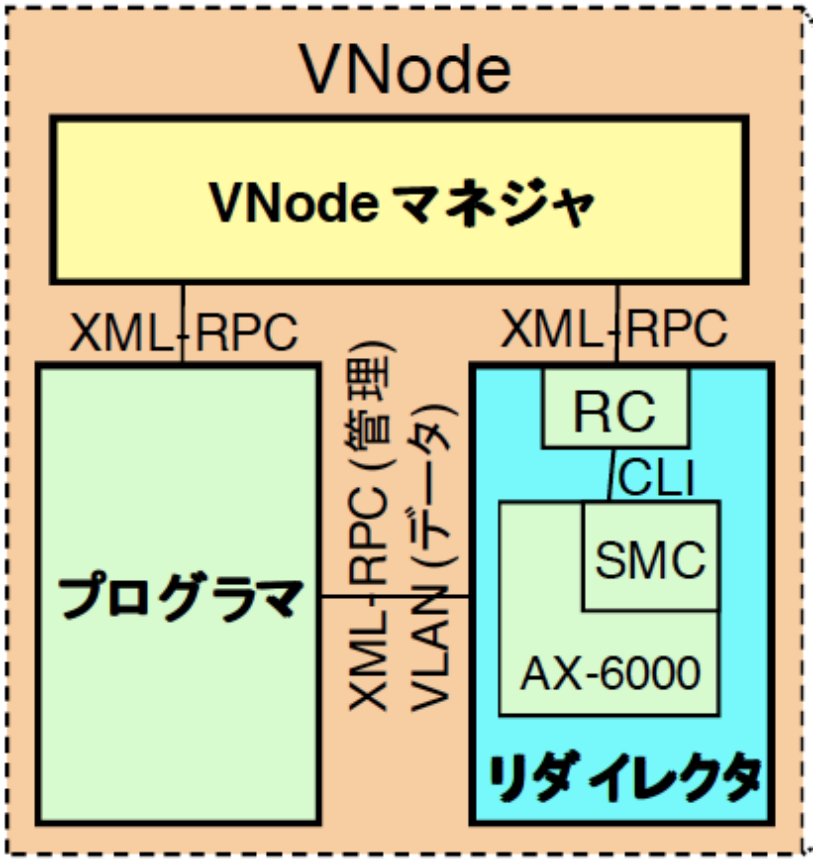
■ プログラマ (Programmer)

◆ パケットの加工や転送先の決定などの処理をおこなう (計算 / ストレージ・リソースをもつ).

◆ プログラマブルなので「プログラマ」とよぶ.

■ リダイレクタ (Redirector)

◆ パケットを他の VNode 等から受信してプログラマにリダイレクトし, プログラマからのパケットを他の VNode などに送信する (ネットワーク・リソースをもつ).



VNode への仮想ネットワーク定義

- 管理サーバ (ドメイン・コントローラ) に XML で記述した「スライス定義」をあたえると，スライスが生成される．
  - ネットワーク配線や機器の設置などの手間をいっさいかけずに，のぞみのネットワークをつくることができる．
- 現在は人手で XML を記述する必要があるが，GUI などでさらに容易にスライスが記述できるようにすることをめざしている．

```
<?xml version="1.0" encoding="UTF-8"?>
<slice-design>
 <slicespec name="IPEC_Slice_000">
  <sliverdef>
   <linkSlivers><!-- 以下はリンクスリバーの定義 -->
    ...
    <linkSliver name="LS01" subtype="GRE" type="link">
      <vports><vport name="e1"/><vport name="e2"/></vports>
    </linkSliver>
    ...
   </linkSlivers>
   <nodeSlivers><!-- 以下はノードスリバーの定義 -->
    ...
    <nodeSliver name="Node2" type="proe">
     <vports><vport name="vp1"/><vport name="vp2"/>
       <vport name="vp3"/></vports>
     <hierarchy>
      <sliverdef>
       <nodeSlivers>
        <nodeSliver name="SP00">
         <vports><vport name="vip1"/><vport name="vip2"/>
           <vport name="vip3"/></vports>
         <instance subtype="KVM" type="SlowPath_VM">
          <resources><!-- ノードスリバーの計算資源 -->
           <resource keyword="cpu" value="1"/><!-- CPUの個数 -->
           <resource keyword="arch" value="x86_64"/>
           <resource keyword="memory" value="2048"/>
          </resources>
          <params><!-- 以下はあらかじめ用意された VM イメージ -->
           <param keyword="bootImage"
            value="http://.../KVM_Ubuntu910Server32.img"/>
          </params>
         </instance>
        </nodeSliver>
       </nodeSlivers>
       <linkSlivers>...</linkSlivers>
      </sliverdef>
      <structure>... </structure>
      <params>...</params>
     </hierarchy>
    </nodeSliver>
    ...
    <nodeSliver name="AGW2" type="agw">
     <vports><vport name="vp1"/></vports>
    </nodeSliver>
   </nodeSlivers>
  </sliverdef>
  <structure><!-- 以下はノードスリバーとリンクスリバーとの結合の定義-->
   ...
   <bind name="w11"><!-- w11 は AGW2 と LS01 を結合する -->
    <vport portname="vp1" slivername="AGW2"/>
    <vport portname="e1" slivername="LS01"/>
   </bind>
   ...
  </structure>
 </slicespec>
 <!-- 以下はノードスリバーの物理ノードへの配置 (マッピング) -->
 <mapping slice="IPEC_Slice_000" vnetwork="NICTtestbed">
  <amap node="AGW2" vnode="agw-f0"/>
  ...
  <amap node="Node2" vnode="rp-nh0"/>
 </mapping>
</slice-design>
```

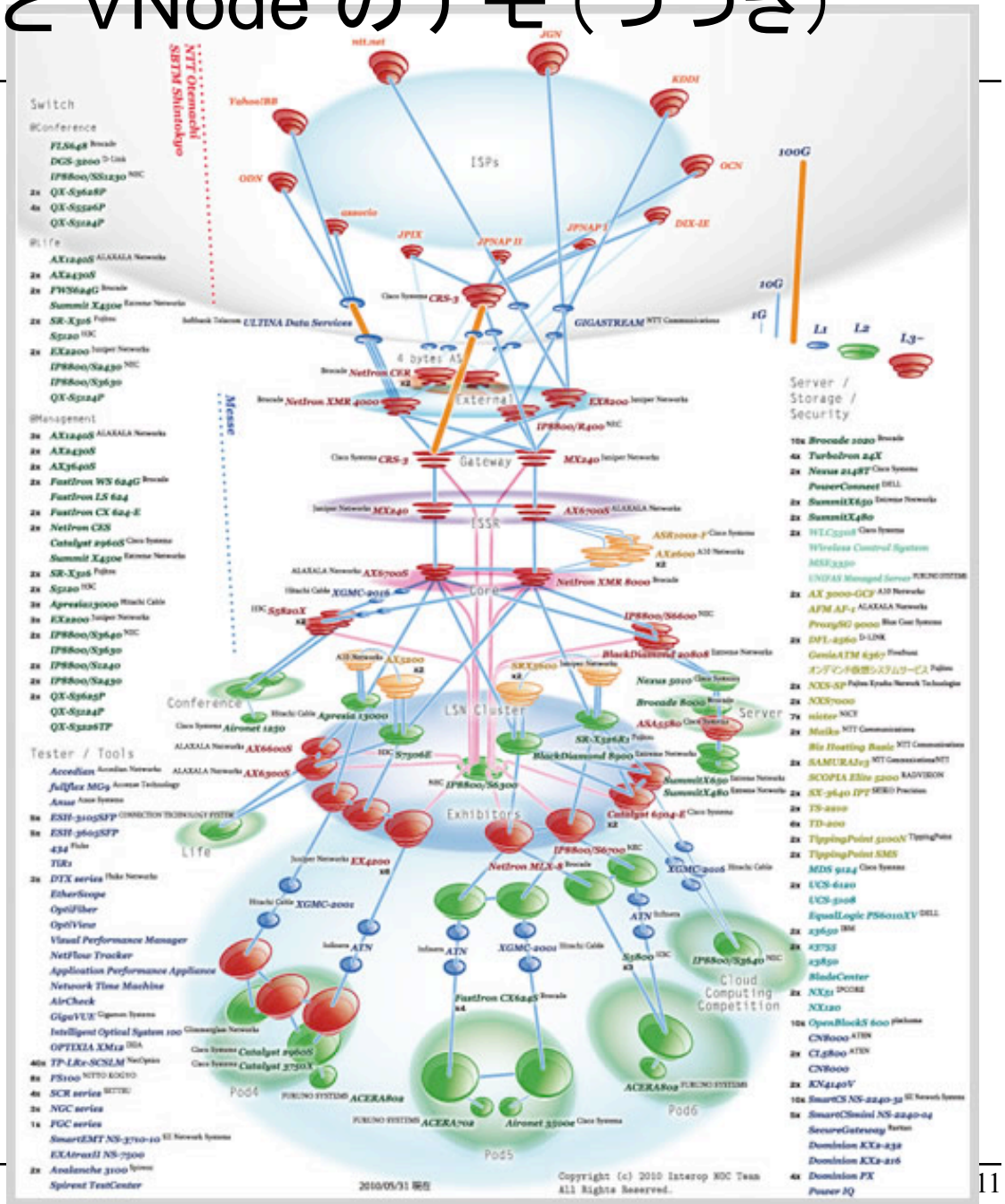
Interop Tokyo と VNode のデモ

- 毎年 6 月にひらかれるインターネット関連の展示会 Interop Tokyo で展示・デモしている.

Interop 2010 の会場のようす

Interop Tokyo と VNode のデモ（つづき）

- Interop Tokyo では毎年, ネットワーク構成にいろいろくふうをこらしている.
  - Cisco, Alaxala をはじめ, 多数の会社のネットワーク機器が つかわれ, 展示されている.
  - 右図は 2010 年の構成.



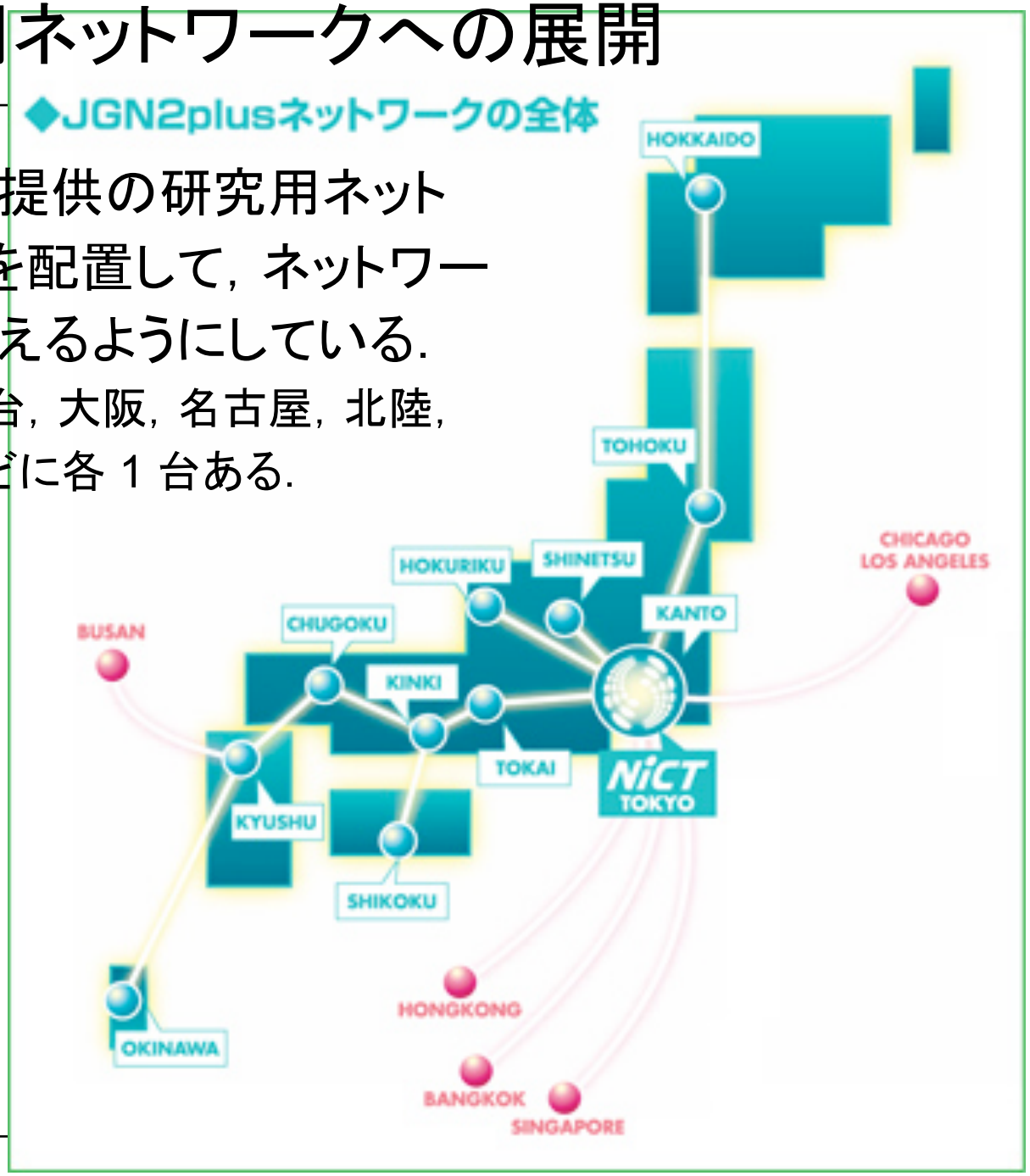
Interop Tokyo と VNode のデモ (つづき)

- NICT ではブースをもうけて多数の研究展示をしている.
- そのなかに VNode の展示もある (写真は 2011 年).

研究用ネットワークへの展開

■ JGN-X という NICT 提供の研究用ネットワーク上に VNode を配置して，ネットワーク研究者などがつかえるようにしている．

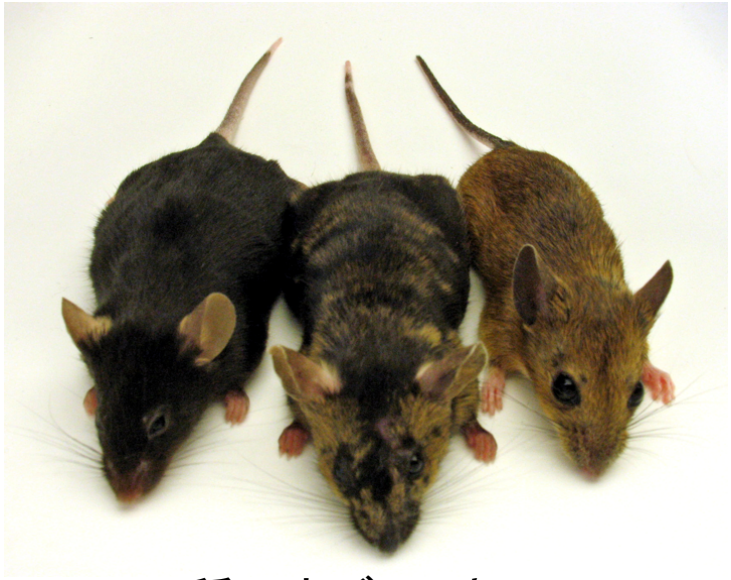
◆ VNode は東京に 2 台，大阪，名古屋，北陸，けいはんな，岡山などに各 1 台ある．



IP にしばられないプロトコルの研究例 -- IPEC

■ IPEC (IP Ether Chimera) は VNode プロジェクトのなかで金田が開発した実験的なプロトコル.

ギリシャ神話のキマイラ



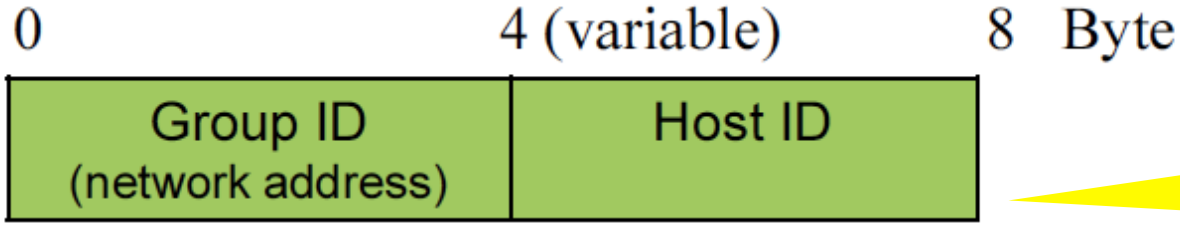
2 種のネズミのキメラ

■ IPEC の開発目的は，IP と Ethernet の利点をかねそなえた単純なプロトコルをつくること.

- ◆ Ethernet の利点は単純さ
- ◆ IP の利点はネットワークにループを許容すること
- ◆ IP と Ethernet とをくみあわせる (IP/Ethernet) と，両者のアドレスを対応づけるために複雑になる (ARP が必要になる).

IP にしばられないプロトコルの研究例 -- IPEC (つづき)

■ IPEC のアドレス・フォーマット

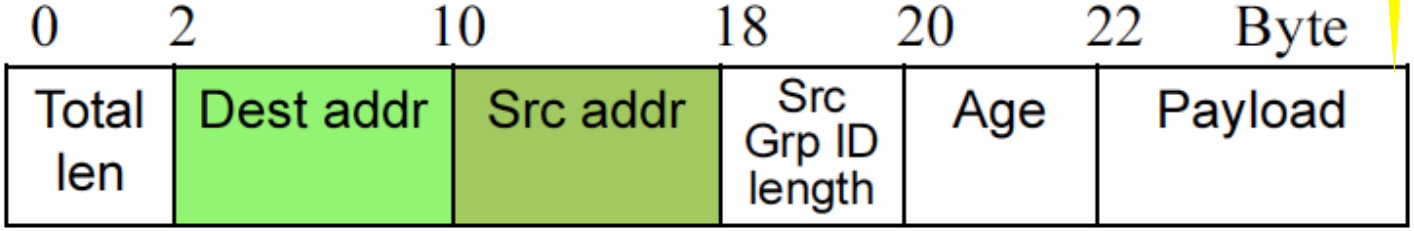


おぼえる必要はない

◆ ホスト ID: コンピュータごとにことなる識別子.

◆ グループ ID: コンピュータのグループごとにことなる識別子.

■ IPEC のパケット・フォーマット



◆ 受信者アドレス (dest addr)

◆ 送信者アドレス (src addr)

◆ Age: 転送されるごとに +1 される (IP の TTL に対応)
-- これをつかってループをふせぐ.

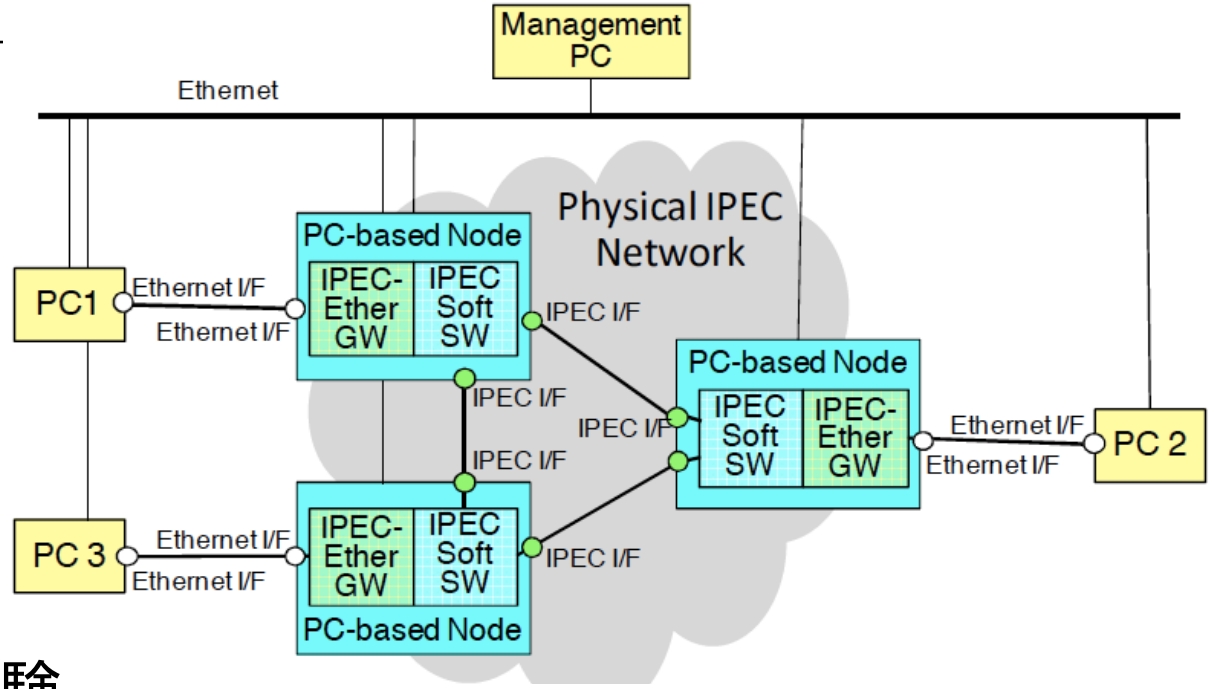
IP にしばられないプロトコルの研究例 -- IPEC (つづき)

- IPEC は IP と Ethernet の両方をおきかえる.
  - 基本的にはネットワーク層 (IP のかわり) のプロトコルだが, Ethernet の転送機能もおきかえる.
- IPEC の学習アルゴリズム
  - 個々のアドレス (ホスト ID) を学習するのでなく, グループ ID を学習することで, 大規模なネットワークに適用できる.
  - パケットの Age フィールドを利用して, おなじパケットが複数個とどいたときは最短経路を選択する.
    - それ以降は最短経路だけがつかわれる.
    - ネットワークにループがあってもよい.
- IPEC の転送アルゴリズム (基本的にイーサネットとおなじ)
  - 学習していないあいだはブロードキャスト (フラディング) する
    -- パケットをコピーする.
  - 学習したあとはその結果にしたがってスイッチする
    -- パケットをコピーしない).

IP にしばられないプロトコルの研究例 -- IPEC (つづき)

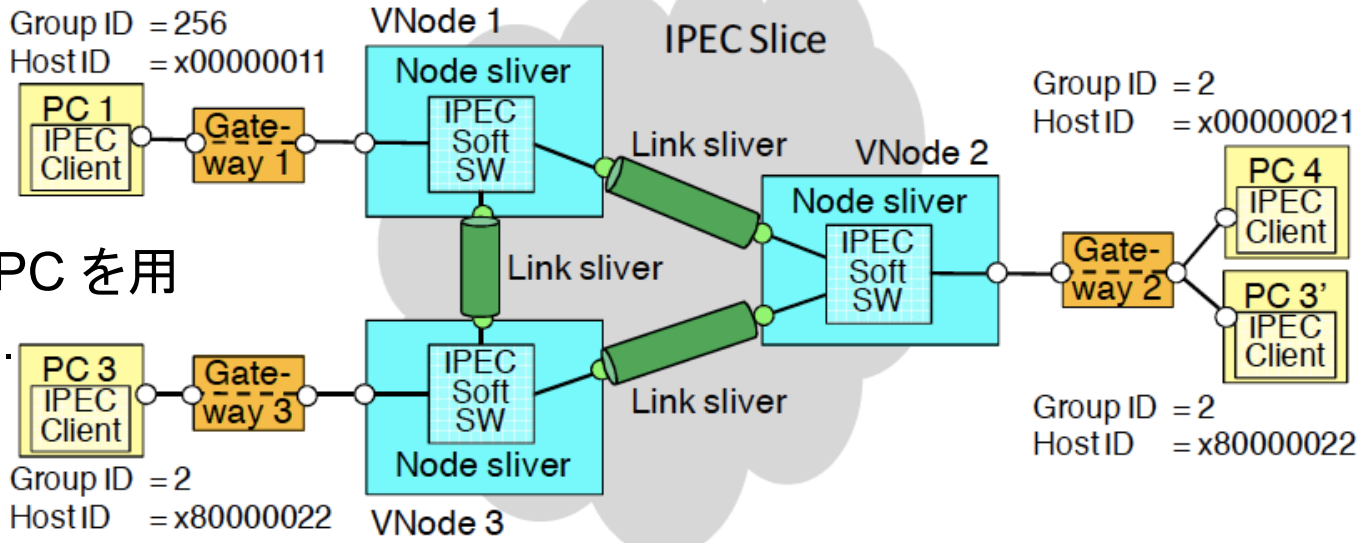
■ PC をつないでの IPEC の実験

◆ IPEC スイッチ用に 3 台，端末用に 3 台の PC を用意して通信する.



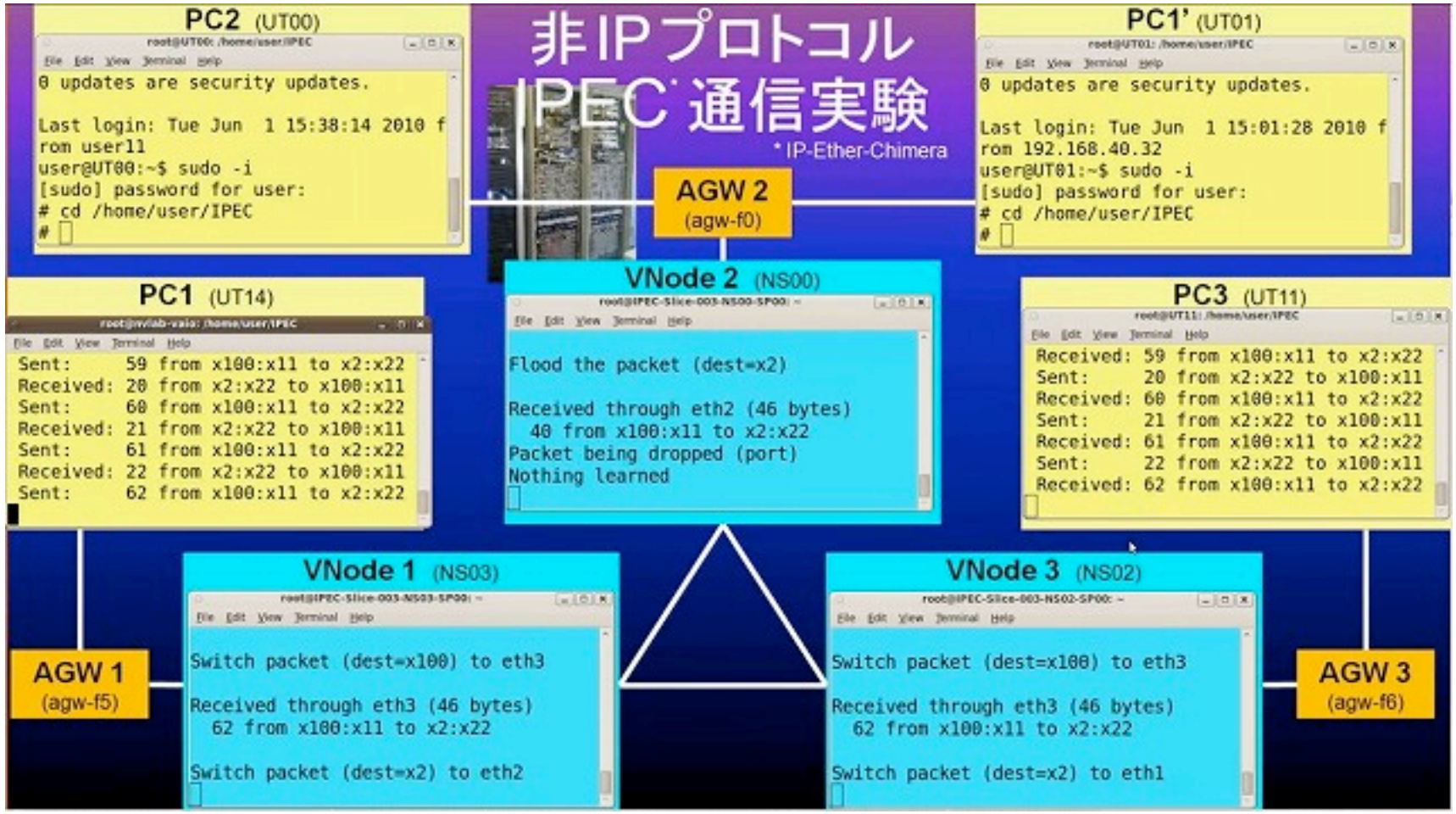
■ VNode を使用した実験

◆ IPEC スイッチ用に 3 台の VNode，端末用には 3 台の PC を用意して通信する.



IP にしばられないプロトコルの研究例 -- IPEC (つづき)

■ IPEC のデモ・ビデオ (GENI Engineering Conference むけ)



プライベート・ネットワークとネットワーク仮想化のまとめ

- インターネットからきりはなされたプライベート・ネットワークではセキュリティが確保しやすく，従来のプロトコルにしばられない．
  - プライベート・ネットワークには物理的なものと仮想化されたもの (VPN) とがある．
- 仮想化されたネットワークとして VLAN があり，企業などでつかわれている．
- 仮想化にはつぎのような種類がある．
  - 質の仮想化と量の仮想化，分割型仮想化と融合型仮想化．
  - コンピュータの仮想化とネットワークの仮想化．
- ネットワーク仮想化の研究によって，プログラマブルで自由なプロトコルがつかえる仮想ネットワークが開発されつつある．
  - 仮想ネットワークにおいては，プライベート・ネットワークの利点をいかして IP やイーサネットとはことなる新プロトコルの実験が自由にできる．
  - 世界各地で新世代ネットワークの研究開発，とくにネットワーク仮想化の研究や実験がおこなわれている．(アメリカで GENI というプロジェクト，日本で AKARI や仮想化ノード (VNode) 開発・利用プロジェクトなど)．