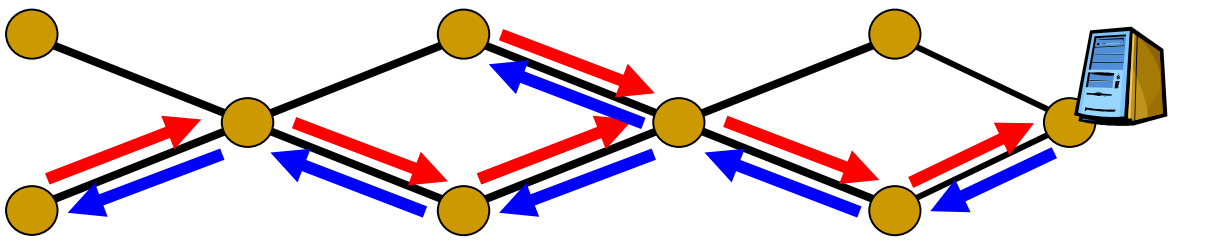
IPトレースバック・オペレーション
～H20実証実験にむけて～

Internet Week 2007
門林 雄基
奈良先端科学技術大学院大学

Part I: IPトレースバック事始め

IPトレースバックが提供する機能

- 「このパケットはどこから来たのか？」
- 流れた経路を復元
- 用途：
  - 巨大フローの流入口を特定
  - 大口顧客の経路監視
  - 経路のデバッグ
- AS間の協調も可能

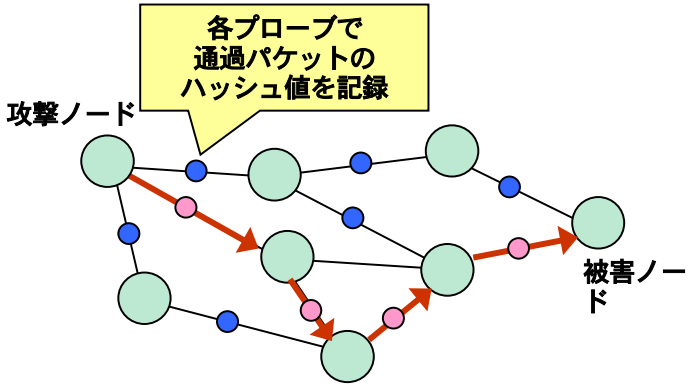


IPトレースバックの方式

本プロジェクトでは、ダイジェスト方式を改良した手法について研究開発を行なっている

【ダイジェスト方式　IPトレースバック】

- A. C. Snoelen, et al, “Hash-based IP Traceback”, SIGCOMM ’01 (2001)
- IPパケットから算出したハッシュ値を特徴として通信経路を追跡
- パケット観測による受動的手法　→ 導入リスクが低い
- 通信パケットの内容を変更する必要がない → 通信への影響が無い
- 単一パケットのトレースも可能　→ 多彩な応用が可能
- 高速なハッシュ演算処理が必要　→ ハードウェアで対応



平常時：プローブで通過パケットのハッシュ値を記録

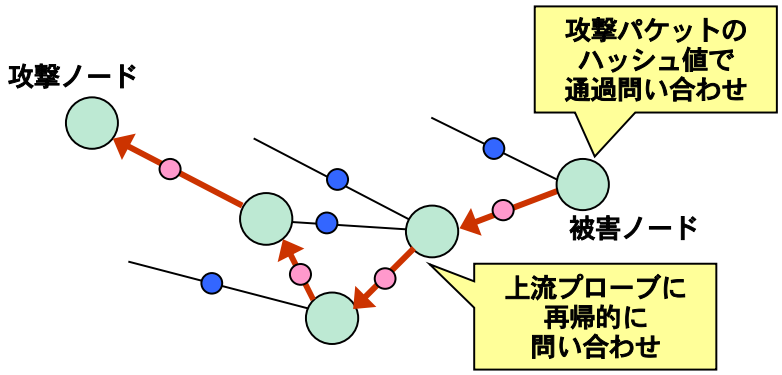
【処理手順】

(平常時)

1. 通信経路上に観測点(プローブ)を配置
2. 通過パケットからハッシュ値を算出
3. ハッシュ値を検索可能な形でキャッシュ

(トレース時)

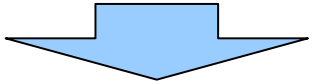
1. 追跡対象パケットのハッシュ値を算出
2. ハッシュ値を用いて上流のプローブに通過/非通過を問い合わせ
3. 通過ノードの上流のプローブに通過/非通過を問い合わせ
4. 問い合わせを再帰的に繰り返すことで通信経路/発信元を特定



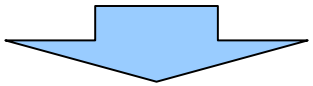
トレース時：攻撃パケットのハッシュ値で通過問い合わせ

動作の流れ

sFlow または IDS, IPS 等でトリガー



IPトレースバック（ドメイン内）
IPトレースバック（ドメイン間）



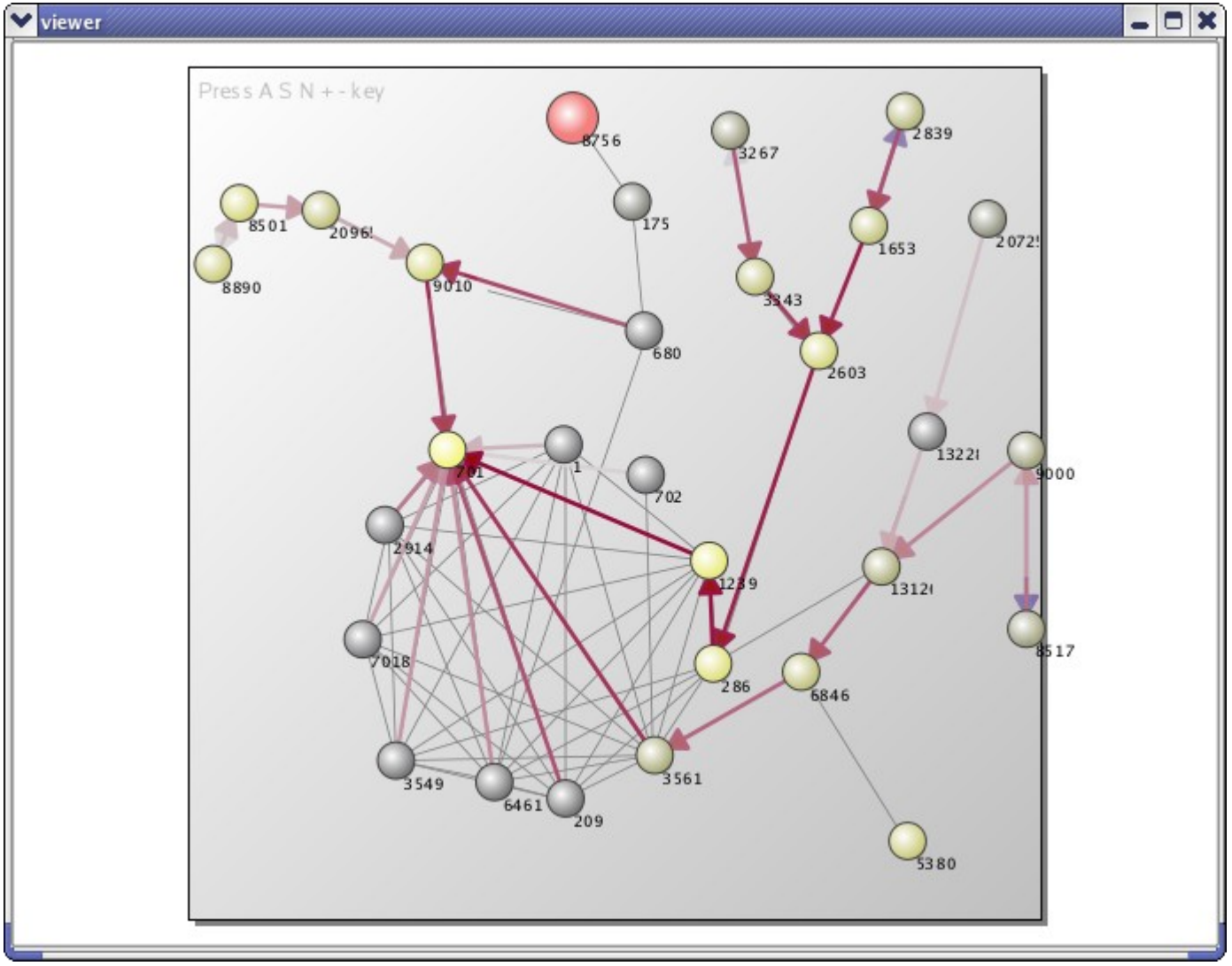
パケットの流れた経路を復元

トレースバックの開始

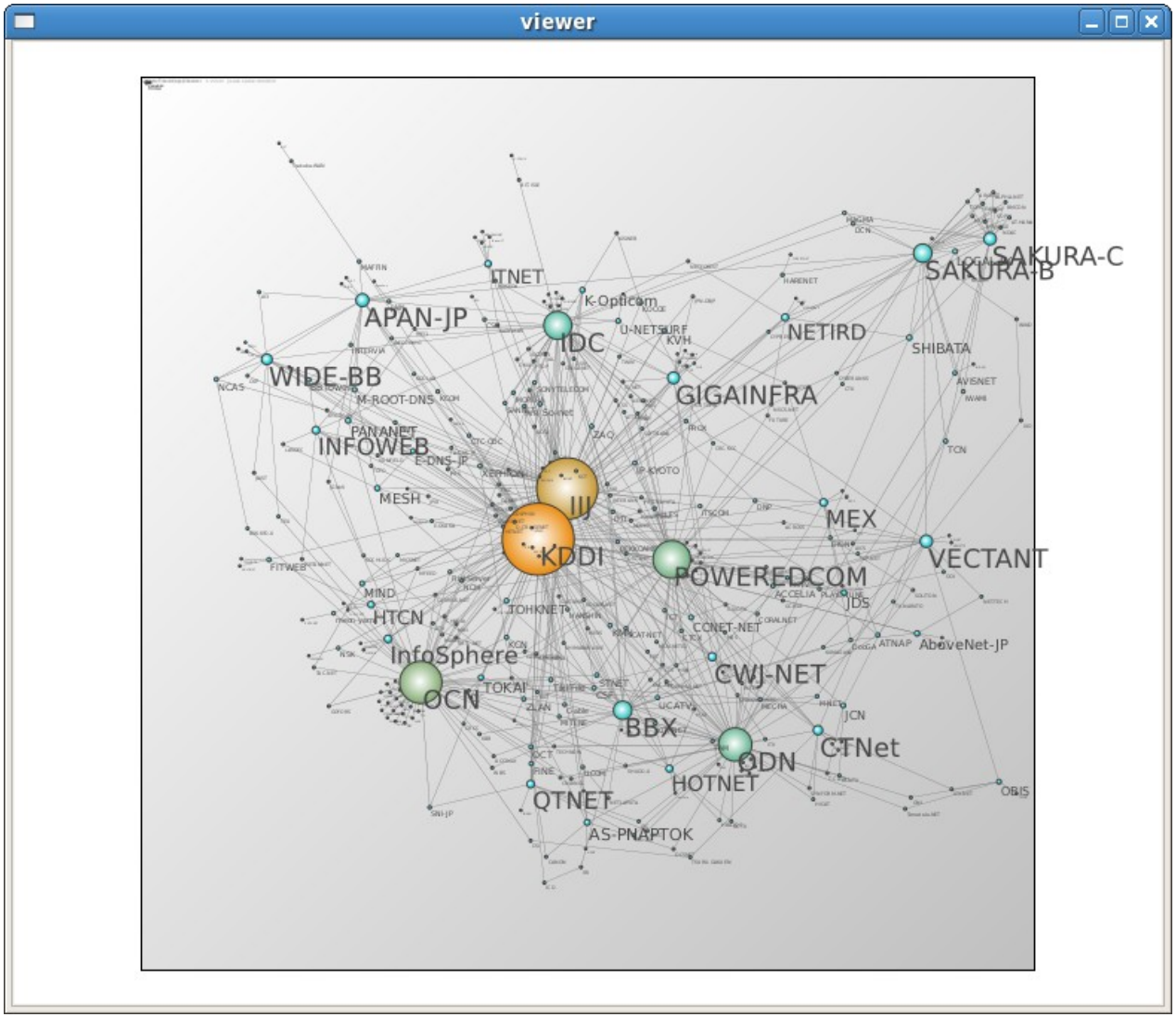
- 巨大フローの流入口を特定
  例）
  - Sflow → top N flow → トレースバック
  - IDS, IPS アラート → 条件指定 → トレースバック

- 大口顧客の経路監視
- 経路のデバッグ
  - ICMP等プローブパケット → トレースバック

トレースバックによる経路復元の様子



JP AS connectivity

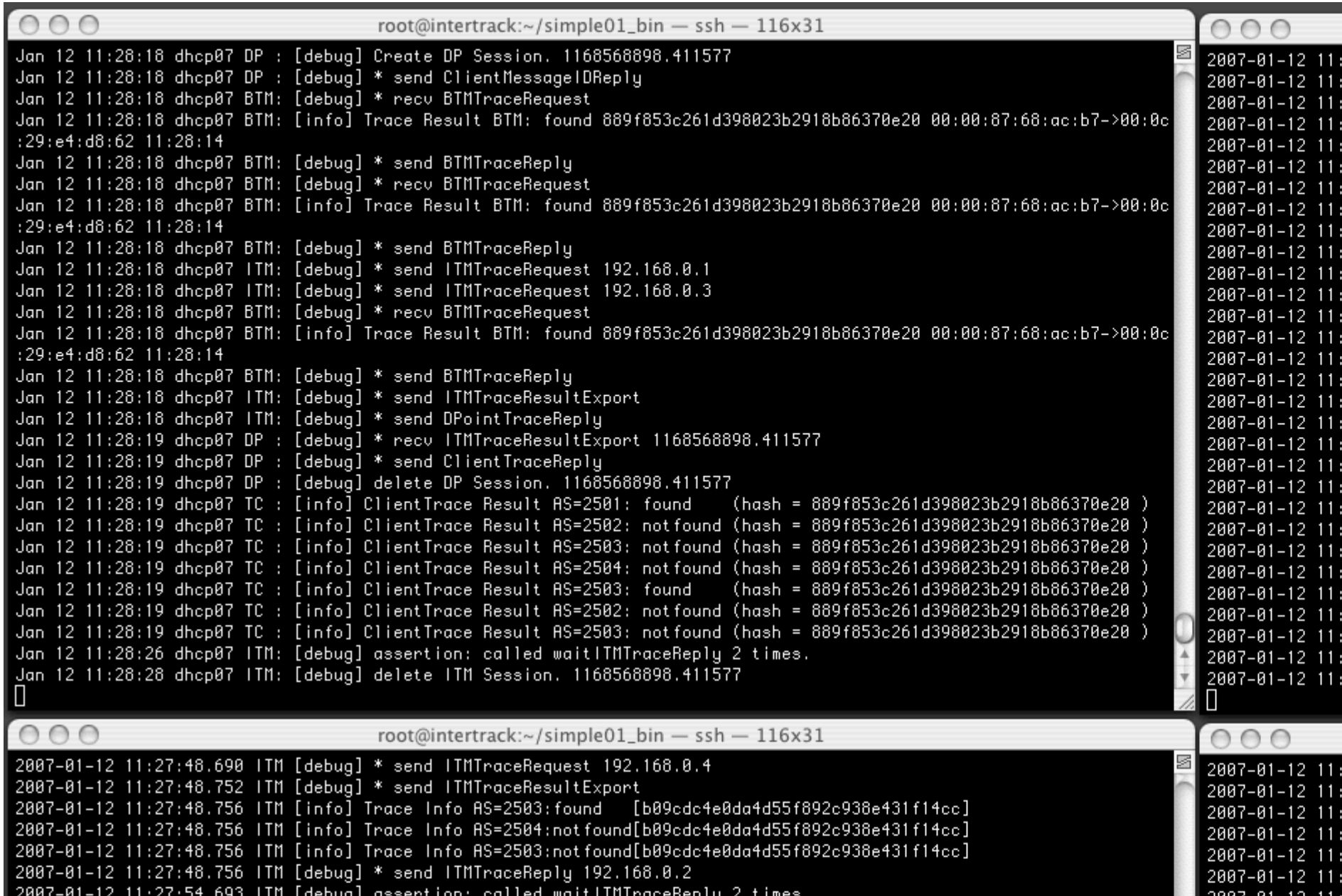


動作監視用コマンドライン・インターフェース

matsu@intertrack:~/simple01/intertrack/src/c++/libintertrack/session — ssh — 154x61

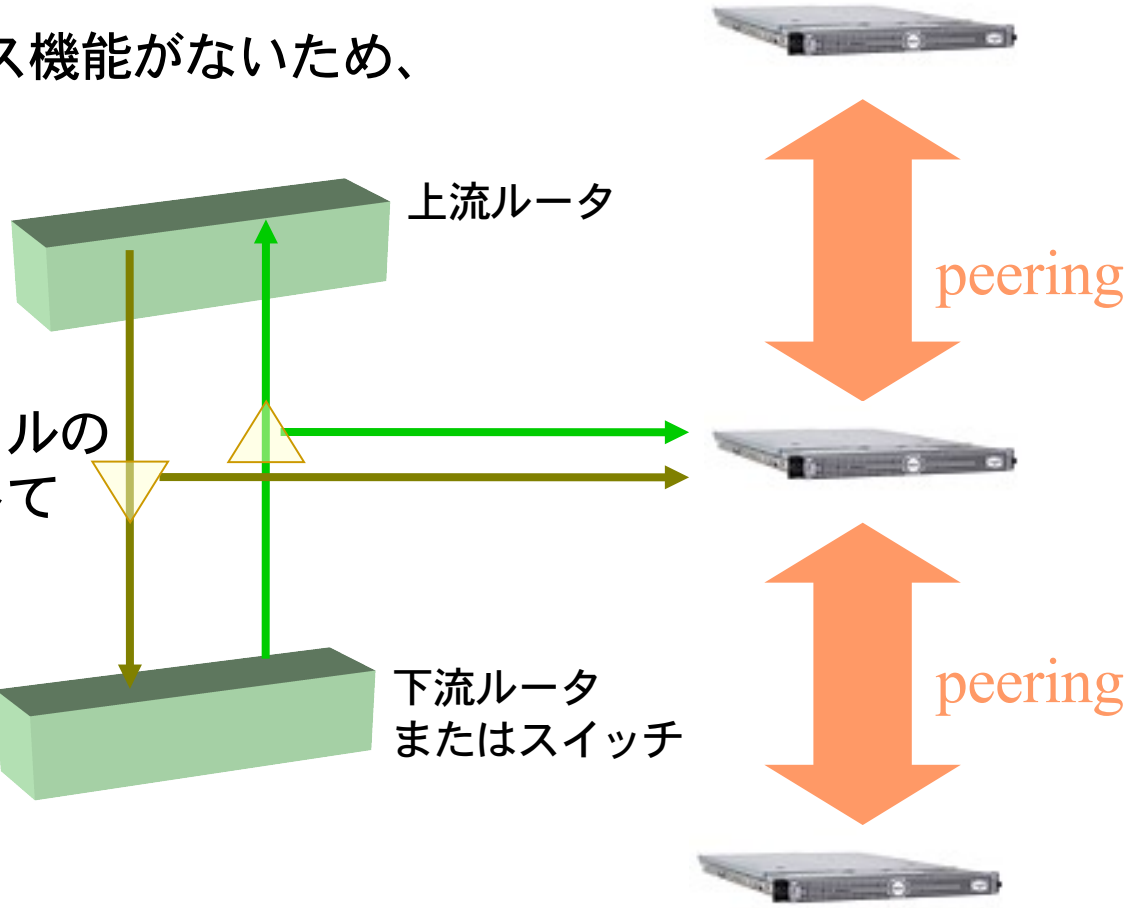
```
TC> show query
time_local  = 1168568988.260814
time_global = 1168568988.368585
hash = ac50d483d5163712d9bbbaf190e0364a

=== request ===
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<InterTrackMessage type="ClientTraceRequest" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="In

  <ClientTraceRequest>
    <DestinationNode>
      <NodeID idtype="IP">
        <IPAddress block="loopback" mask="32" version="4">127.0.0.1</IPAddress>
      </NodeID>
    </DestinationNode>
    <SourceNode>
      <NodeID idtype="IP">
        <IPAddress block="loopback" mask="24" version="4">127.0.0.1</IPAddress>
      </NodeID>
    </SourceNode>
    <TemporarySequenceNumber sec="1168568988" usec="260814"/>
    <PacketDump PayloadLength="32" encodetype="md5" header="ip" iftype="1">ac50d483d5163712d9bbbaf190e0364a</PacketDump>
    <Options>
      <Option type="type">content</Option>
    </Options>
  </ClientTraceRequest>

</InterTrackMessage>

=== reply   ===
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<InterTrackMessage type="ClientTraceReply" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="Inte

  <ClientTraceReply Export_type="AllResultExport">
    <SourceNode>
      <NodeID idtype="IP">
        <IPAddress block="loopback" mask="24" version="4">127.0.0.1</IPAddress>
      </NodeID>
    </SourceNode>
    <DestinationNode>
```

ログ


root@intertrack:~/simple01_bin — ssh — 116x31
Jan 12 11:28:18 dhcp07 DP : [debug] Create DP Session. 1168568898.411577
Jan 12 11:28:18 dhcp07 DP : [debug] * send ClientMessageIDReply
Jan 12 11:28:18 dhcp07 BTM: [debug] * recv BTMTraceRequest
Jan 12 11:28:18 dhcp07 BTM: [info] Trace Result BTM: found 889f853c261d398023b2918b86370e20 00:00:87:68:ac:b7->00:0c
:29:e4:d8:62 11:28:14
Jan 12 11:28:18 dhcp07 BTM: [debug] * send BTMTraceReply
Jan 12 11:28:18 dhcp07 BTM: [debug] * recv BTMTraceRequest
Jan 12 11:28:18 dhcp07 BTM: [info] Trace Result BTM: found 889f853c261d398023b2918b86370e20 00:00:87:68:ac:b7->00:0c
:29:e4:d8:62 11:28:14
Jan 12 11:28:18 dhcp07 BTM: [debug] * send BTMTraceReply
Jan 12 11:28:18 dhcp07 ITM: [debug] * send ITMTraceRequest 192.168.0.1
Jan 12 11:28:18 dhcp07 ITM: [debug] * send ITMTraceRequest 192.168.0.3
Jan 12 11:28:18 dhcp07 BTM: [debug] * recv BTMTraceRequest
Jan 12 11:28:18 dhcp07 BTM: [info] Trace Result BTM: found 889f853c261d398023b2918b86370e20 00:00:87:68:ac:b7->00:0c
:29:e4:d8:62 11:28:14
Jan 12 11:28:18 dhcp07 BTM: [debug] * send BTMTraceReply
Jan 12 11:28:18 dhcp07 ITM: [debug] * send ITMTraceResultExport
Jan 12 11:28:18 dhcp07 ITM: [debug] * send DPointTraceReply
Jan 12 11:28:19 dhcp07 DP : [debug] * recv ITMTraceResultExport 1168568898.411577
Jan 12 11:28:19 dhcp07 DP : [debug] * send ClientTraceReply
Jan 12 11:28:19 dhcp07 DP : [debug] delete DP Session. 1168568898.411577
Jan 12 11:28:19 dhcp07 TC : [info] ClientTrace Result AS=2501: found (hash = 889f853c261d398023b2918b86370e20)
Jan 12 11:28:19 dhcp07 TC : [info] ClientTrace Result AS=2502: notfound (hash = 889f853c261d398023b2918b86370e20)
Jan 12 11:28:19 dhcp07 TC : [info] ClientTrace Result AS=2503: notfound (hash = 889f853c261d398023b2918b86370e20)
Jan 12 11:28:19 dhcp07 TC : [info] ClientTrace Result AS=2504: notfound (hash = 889f853c261d398023b2918b86370e20)
Jan 12 11:28:19 dhcp07 TC : [info] ClientTrace Result AS=2503: found (hash = 889f853c261d398023b2918b86370e20)
Jan 12 11:28:19 dhcp07 TC : [info] ClientTrace Result AS=2502: notfound (hash = 889f853c261d398023b2918b86370e20)
Jan 12 11:28:19 dhcp07 TC : [info] ClientTrace Result AS=2503: notfound (hash = 889f853c261d398023b2918b86370e20)
Jan 12 11:28:26 dhcp07 ITM: [debug] assertion: called waitITMTraceReply 2 times.
Jan 12 11:28:28 dhcp07 ITM: [debug] delete ITM Session. 1168568898.411577
2007-01-12 11:
root@intertrack:~/simple01_bin — ssh — 116x31
2007-01-12 11:27:48.690 ITM [debug] * send ITMTraceRequest 192.168.0.4
2007-01-12 11:27:48.752 ITM [debug] * send ITMTraceResultExport
2007-01-12 11:27:48.756 ITM [info] Trace Info AS=2503:found [b09cdc4e0da4d55f892c938e431f14cc]
2007-01-12 11:27:48.756 ITM [info] Trace Info AS=2504:notfound[b09cdc4e0da4d55f892c938e431f14cc]
2007-01-12 11:27:48.756 ITM [info] Trace Info AS=2503:notfound[b09cdc4e0da4d55f892c938e431f14cc]
2007-01-12 11:27:48.756 ITM [debug] * send ITMTraceReply 192.168.0.2
2007-01-12 11:
2007-01-12 11:

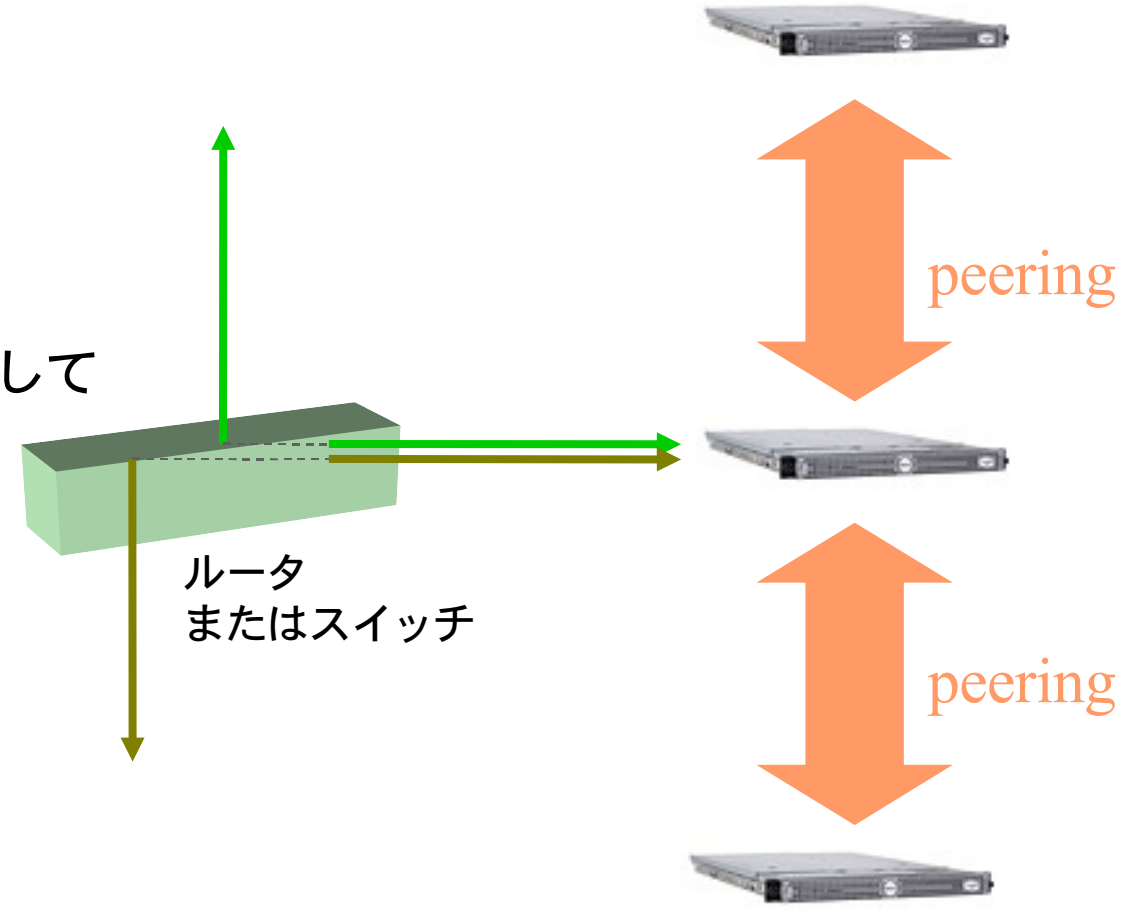
IPトレースバックの運用形態その１

現時点で、ルータにトレース機能がないため、



IPトレースバックの運用形態その２

ポートミラーリングを設定して
トレースバックノードへ。

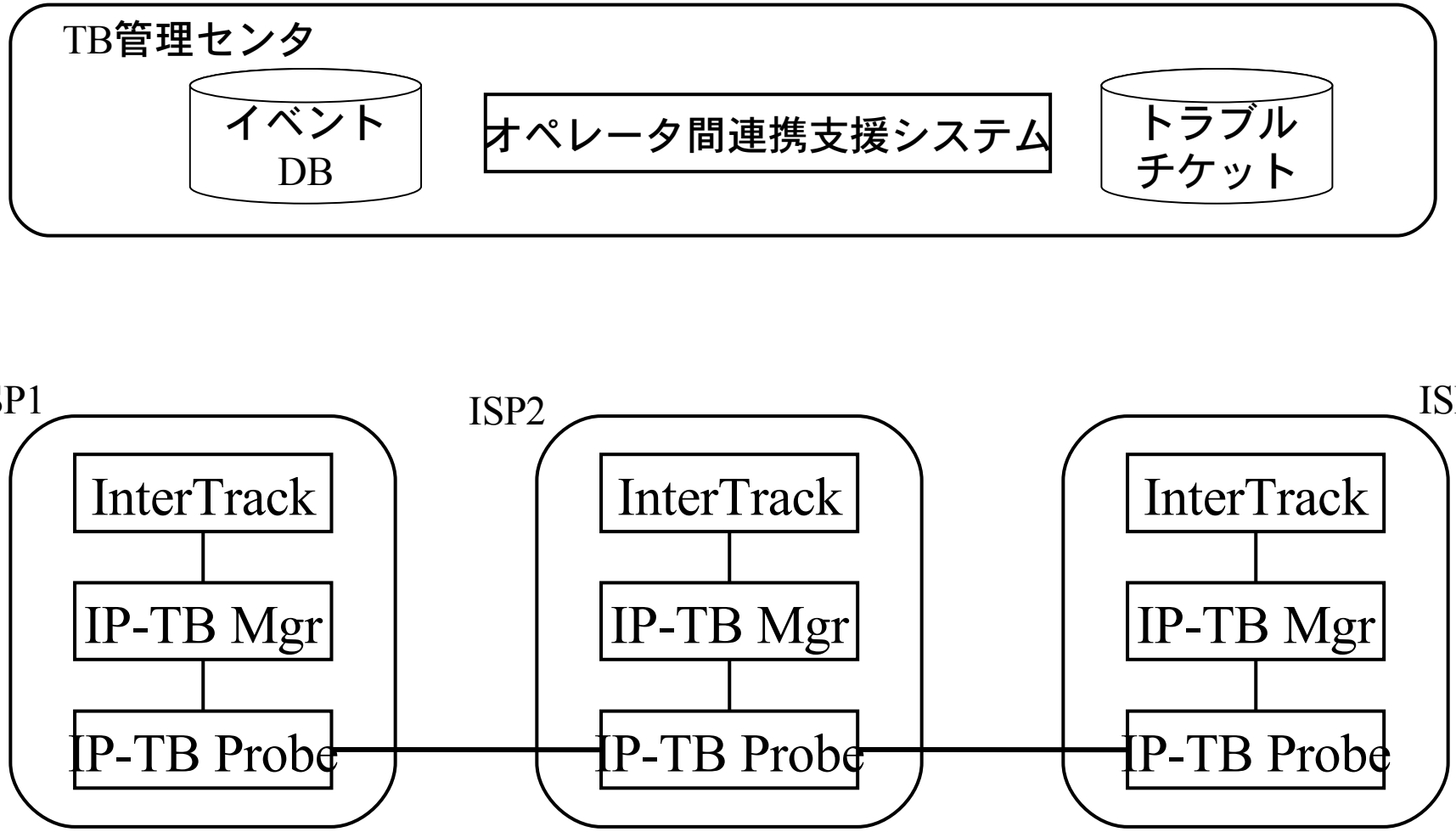


オペレータ間連携支援システム

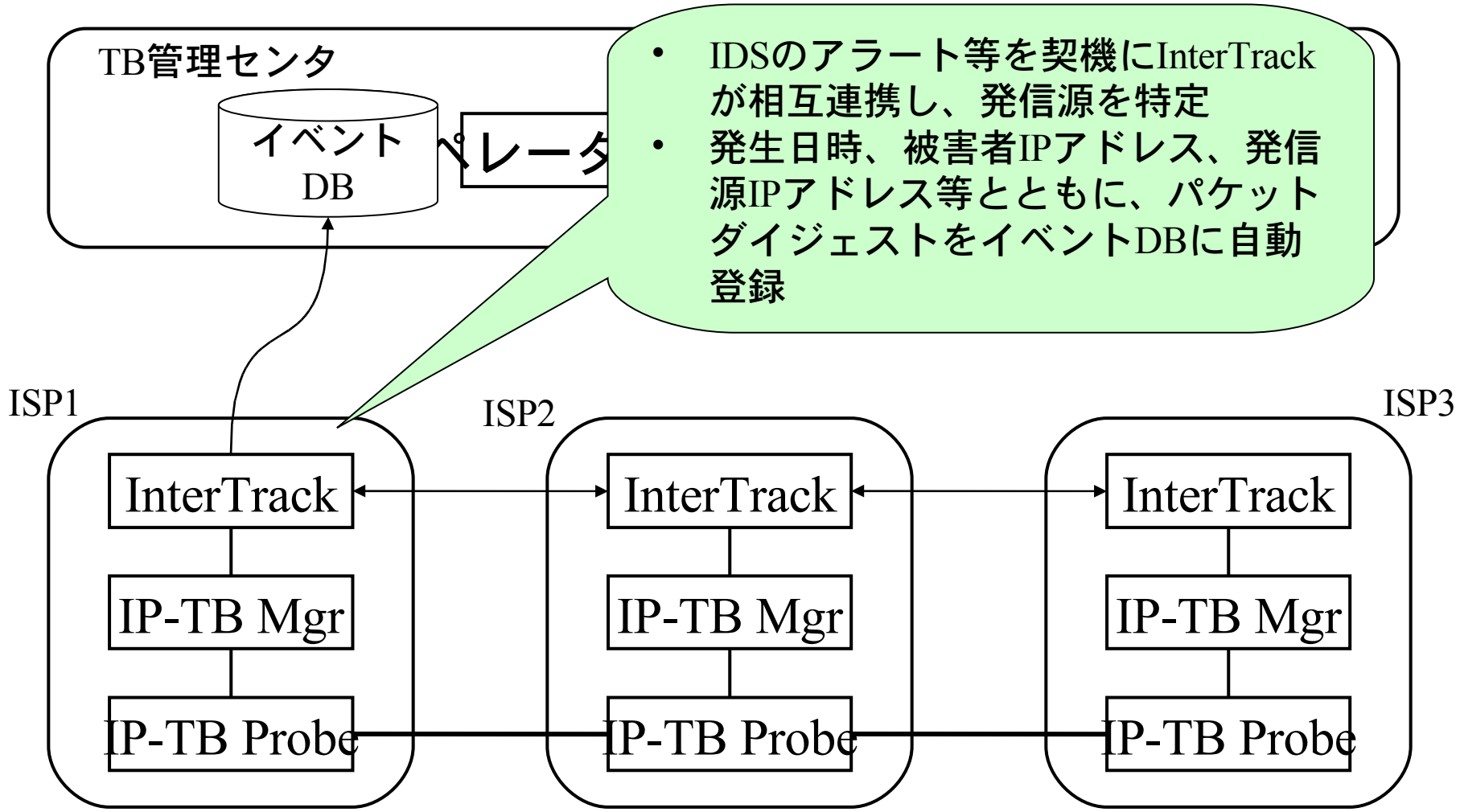
- 複数のISPにまたがるインシデントが発生した際に、ISPのオペレータ間で情報を共有し、迅速な解決を支援するソフトウェア

- IPトレースバックで復元された経路をもとに複数ASで連携

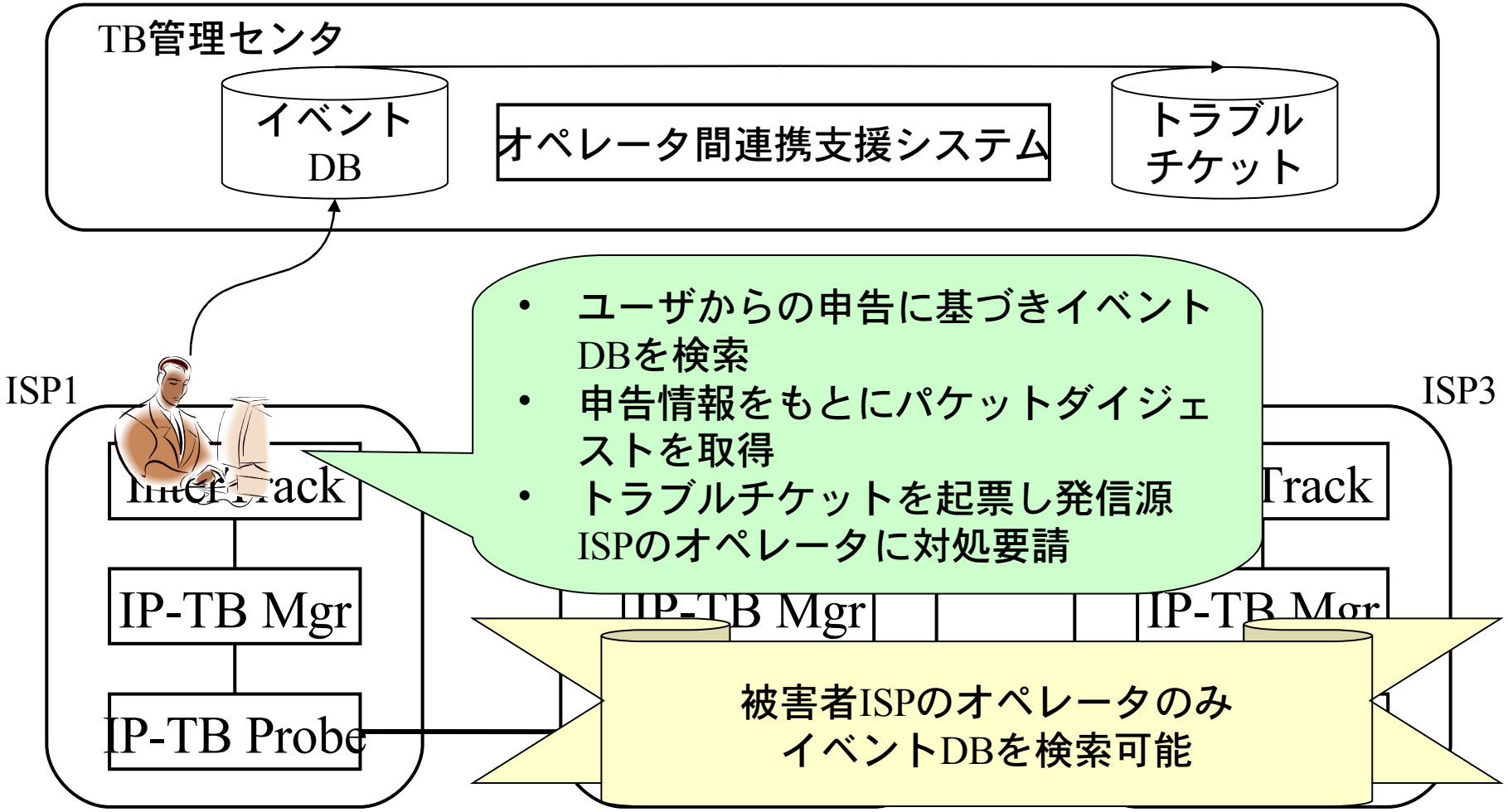
全体アーキテクチャ (IP-TB)



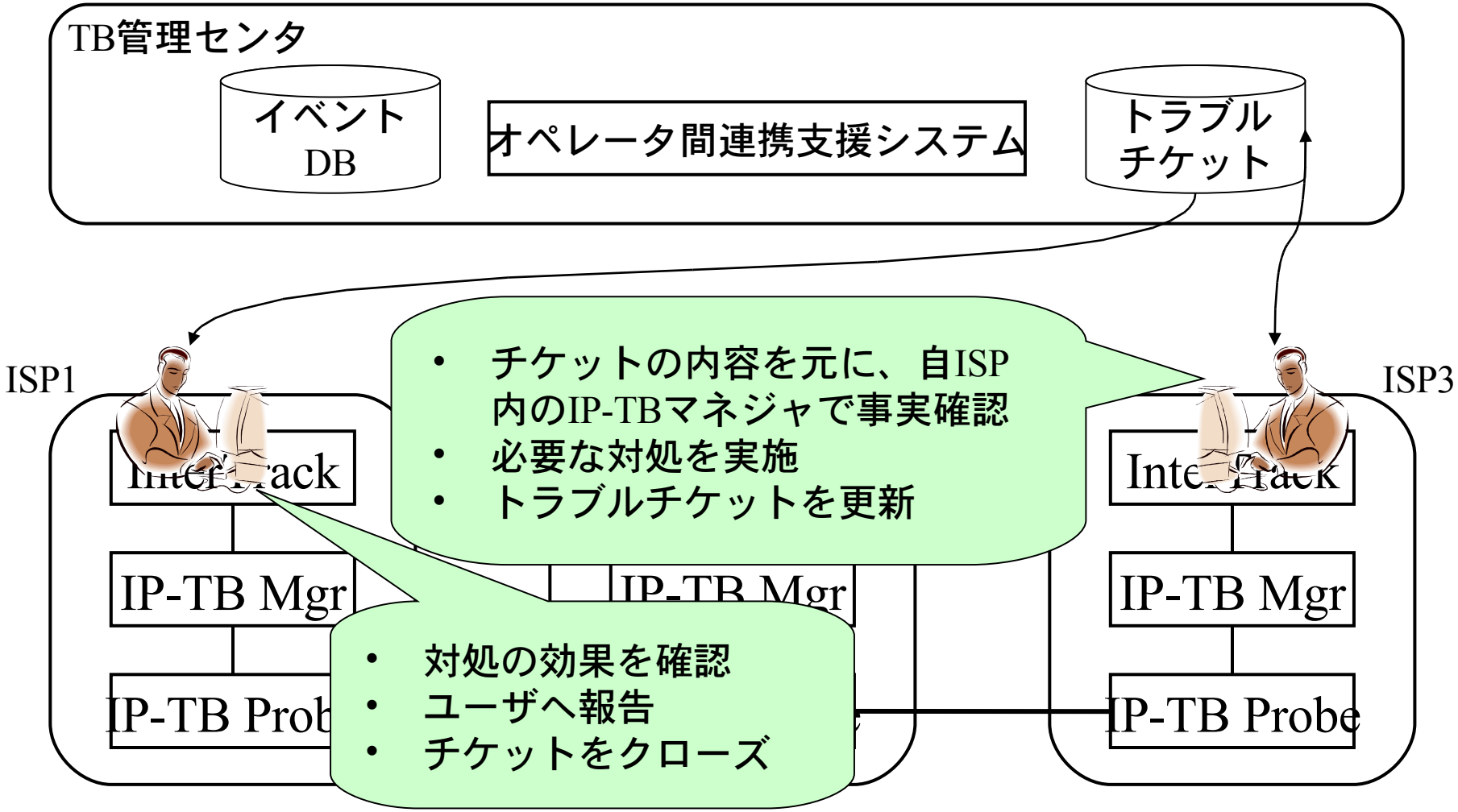
動作概要 – 平常時



動作概要 – インシデント発生時



動作概要 – インシデント対応時



Q&A

Part II: IPトレースバックの仕組み

リンク検査方式

- 受信者の最寄りのルータから、フローの上流となるリンクを特定し、隣接ルータへと順にたどっていくことで発信源を特定することができる。
- そのための手法としては input debugging が知られている。input debugging はルータのデバッグ機能を利用したものである。
- 受信者はパケット群を分析し、特徴を抽出する必要がある。
- ネットワーク管理者が、ルータの出力ポートにおいて対象パケットを判別するフィルタを設定することで、対象パケットの入力ポートを知ることができる。
- 入力ポートが分かれば隣接ルータを割り出すことができる。これを順に繰り返すことで発信源を特定することができる。

逆探知パケット方式

- ルータを通過するパケットに何らかの方法で逆探知のための情報を付与できれば、被害者の近傍でそれを収集し発信源を特定することができる。
- 逆探知のための情報を別のパケットにおさめ、被害者へ届ける手法として、ICMPに新たなメッセージ種別 traceback を定義し、ICMP traceback メッセージ内に各リンクのIPアドレスやインターフェース名、公開鍵などをおさめる手法が提案されている。

入力フィルタ方式

- もし、すべてのネットワークにおいて始点アドレスの偽造を許さないように設定することができれば、IPトレースバック技術は必要ないといえる。
- 入力フィルタの設定を自動化するため、Unicast RPFと呼ばれる機能がいくつかのルータに実装されている。

マーキング方式

- 逆探知のための情報をIPv4ヘッダ内の未使用ビットを用いて伝える方式が研究されている。
- これらの方式は追加のパケットを用いることなく、元のパケットにマークを付けることからマーキング方式と呼ばれる。
- これらの方式ではいずれもフラグメント時にのみ用いられるIP identification フィールドを流用し、ルータにおいてある確率でマークをつける。
- マーキングにIP identificationフィールドを用いた場合、IPsec AHによるヘッダの認証が機能しなくなってしまう。
- マーキング方式はVPN, ストリーミング・メディアなどの比較的新しいアプリケーションとの親和性が低い。

ダイジェスト方式

- パケットの記録を効率よく保持することができれば、IPヘッダを流用あるいは拡張することなく逆探知が可能となる。
- パケットを記録する代わりに、パケットの先頭部分から計算した複数のハッシュ値を記録する方式が提案されている。
- 具体的には、IPヘッダのうち経路中で不変である部分とペイロード先頭の8byte（計28byte）を対象とし、k個の独立なハッシュ関数を適用した結果を2^n bitのビットマップとして保持するというものである。
- あるパケットが通過したかどうかは、パケットの先頭部分を用いてk個のハッシュ関数を計算することで検査することができる。計算結果と2^n bitのビットマップを比較し、すべてのビットが1であれば高い確率でパケットが通過したと言える。

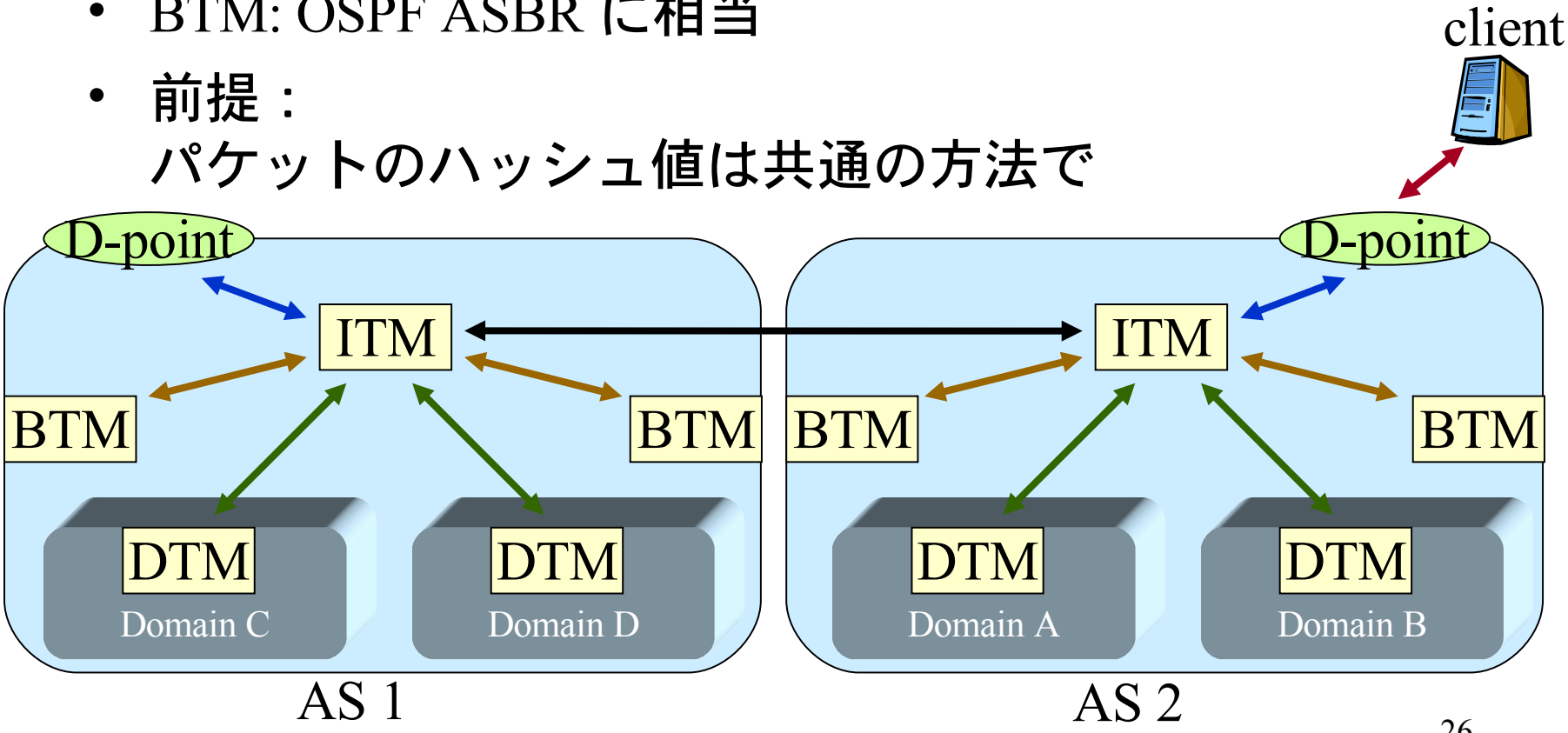
ドメインをまたぐトレースバック

- さまざまな方式
- インターネット全体での運用：
  - 単一の方式をグローバルで、という前提
- 上記前提には無理がある

- IGP, EGP に相当するドメイン内外での分離
- → InterTrack

InterTrack

- ITM: BGP router に相当
- DTM: OSPF router に相当
- BTM: OSPF ASBR に相当
- 前提：
  パケットのハッシュ値は共通の方法で



InterTrack の特徴

- 複数の事業者からなるネットワークにおいて、事業者をまたがる発信源追跡を行うためのソフトウエアを実現
- 異なるトレースバックシステムをXMLを用いて相互接続
- 多様なトレースバック方式を内包可能
- NAT, Firewallを超えた追跡が可能
  - NAT, Firewallでの対応が前提
- トレースバック方式実装のためのハーネスを用意
  - ステータス管理
  - ディスパッチャ
  - コネクション管理等

InterTrack 取り組み概要

- トレースバックシステムの相互接続アーキテクチャを**設計**
  - InterTrack アーキテクチャ、プロトコル
- InterTrack フレームワークを**実装**
  - トレースバック方式のためのソフトウェア・バス
- テストベッドで**動作検証**
  - クラスタ、実ネットワーク
- 目標：
  - 実動システムを ISP/IRTオペレータに提供すること

InterTrackの動作速度

- 250ms以内に応答
- 最大２秒

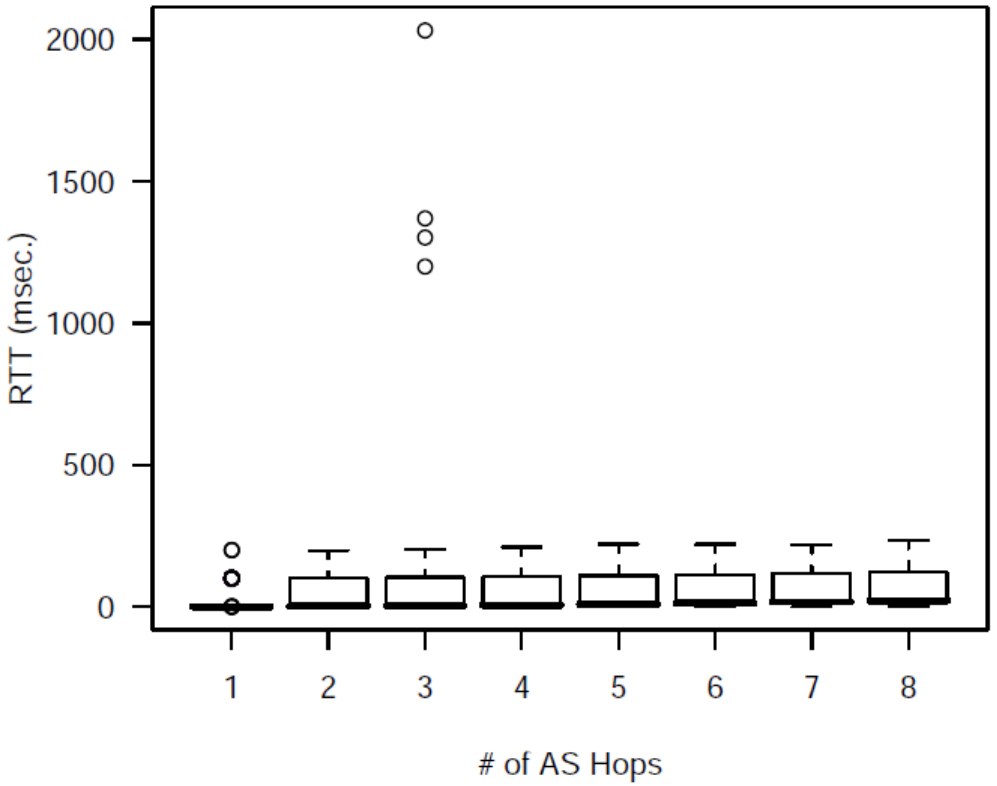


Figure 3.11: RTT of a ITM Traceback Request in a liner topology

これまでの取り組み

- 2000: IP traceback (Savage et al., SIGCOMM)
- 2003: Hierarchical IP traceback (Oe et al., SAINT)

  Layer-2 トレースバック (Hazeyama et al., J. IEICE)
- 2004: InterTrack 設計実装開始
- 2005: InterTrack (Hazeyama et al., ACSAC)

  InterTrack実動

  NICT委託研究開始、８社コンソーシアム
- 2007: InterTrackを基盤として開発

  KDDI, 沖電気、松下電工

活動概要 - 研究開発目標

本研究の最終目標

- トレースバックアルゴリズムの開発とトレースバックプラットフォームの実装、および運用体制の検討を経て、実ネットワークに近い環境を用いて、開発したトレースバックシステムの有効性の検証のために実証試験を実施する。
- これら活動を通し、攻撃がどこから実行されようとしているのかを探索する能動的な警戒手段としての実用的なトレースバックの技術確立を本研究の最終目的とする。

各課題の最終目標

課題　ア　「全体アーキテクチャの設計」

トレースバック技術に対する誤解や過度の期待を解消 。
攻撃フローが存続する間にトレースバックを容易に行う。
相互接続アーキテクチャを設計・実装 する。

課題　イ　「トレースバック・アルゴリズムの開発」

踏み台攻撃を検知することのできる実用的なトレースバックアルゴリズムを開発する。また、プライバシの保護についても既存方法についての検討を行い、盛り込むものとする。

課題ウ　「トレースバック用データ収集装置（プローブ装置）の開発」

プローブ装置を利用する実用的なトレースバック用のソフトウェアを開発する。また、プライバシの保護機能も動作するものとする。

課題　エ　「トレースバックプラットフォームの実証実験」

課題ア、イ、ウで開発された成果についてトレースバックプラットフォームの実証実験を実施する。

活動概要 - 体制

代表研究責任者　：日本電気株式会社

- **課題 ア　「全体アーキテクチャの設計」**
  - ア－１　トレースバック機構を構築する上で考慮するべき事項の網羅　（国立大学法人奈良先端科学技術大学院大学）
  - ア－２　基本的なトレースバック方式の開発　（株式会社ＫＤＤＩ研究所）
  - ア－３　トレースバックシステムの相互接続アーキテクチャの開発　（国立大学法人奈良先端科学技術大学院大学）
- **課題　イ　「トレースバック・アルゴリズムの開発」**
  - イ－１　ＩＰパケットトレースバックアルゴリズムの開発　（松下電工株式会社）
  - イ－２　アプリケーショントレースバックアルゴリズムの開発　（株式会社クルウィット）
  - イ－３　異なるレイヤ由来の情報からトレースバック能力を向上させるアルゴリズムの開発（株式会社クルウィット）
- **課題　ウ　「トレースバック用データ収集装置（プローブ装置）の開発」**
  - ウ－１　IPトレースバック用データ収集装置の開発　（株式会社ＫＤＤＩ研究所）
  - ウ－２　アプリケーショントレースバック用データ収集装置の開発　（日本電気株式会社）
- **課題　エ　「トレースバックプラットフォームの実証実験**
  - エ－１　実装および運用体制の検討　（財団法人日本データ通信協会）
  - エ－２　攻撃パターンの想定　（財団法人日本データ通信協会）
  - エ－３　動作検証　（財団法人日本データ通信協会）
- **課題　オ　「テーマ全体管理」**　（日本電気株式会社）

通信の秘密について

- 通信の秘密
  - 明らかな攻撃の場合には、通信の秘密よりも緊急対応のほうが優先する
  - そもそも通信の秘密とは、言論の自由を守るもので、攻撃や悪意のある通信を守るものではない。
  - 通信の秘密は手紙の話であり、電気通信を想定していなかった。電話からインターネットに適用するかどうかは大きな分かれ道である。
- ヘッダ、ペイロード
  - 事業者間でヘッダやペイロードを交換するのは通信の秘密を侵していることになると考えられる。
- ハッシュ値
  - 一方向性を前提とすれば問題ないと思われる。
- すべては、総務省の解釈次第

ハッシュ関数の安全性について

- 一方向性
  - H(M) → M の導出が困難
- 衝突困難性
  - $H(M_1) = H(M_2)$ となる M_1, M_2 の発見が困難
- 第2原像計算困難性
  - H(M') = H(M) となる M' の発見が困難

- MD5
  - 衝突困難性は弱い
- SHA-1
  - 衝突困難性が弱くなってきている
  - ただしCollision が１つ計算できたとしても、現実のアプケーションに影響はない（IW2006 松井氏の講演資料を参照）
- 一方向性については問題なし

Q&A

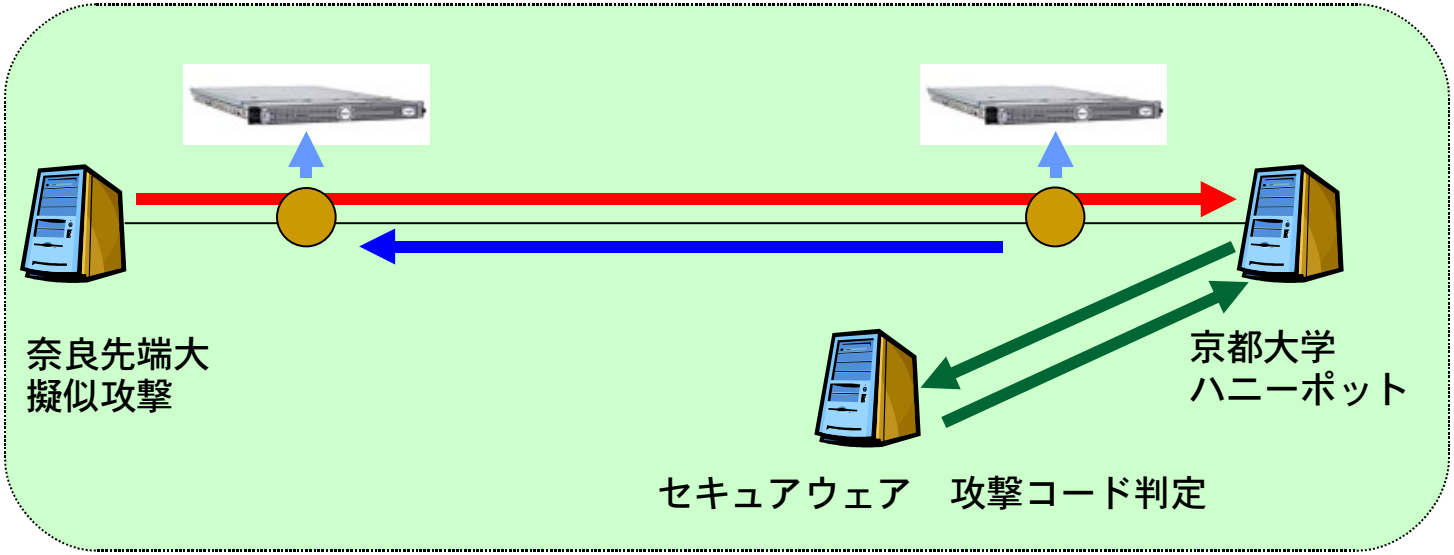
Part III: IPトレースバックの実証実験

H16実証実験

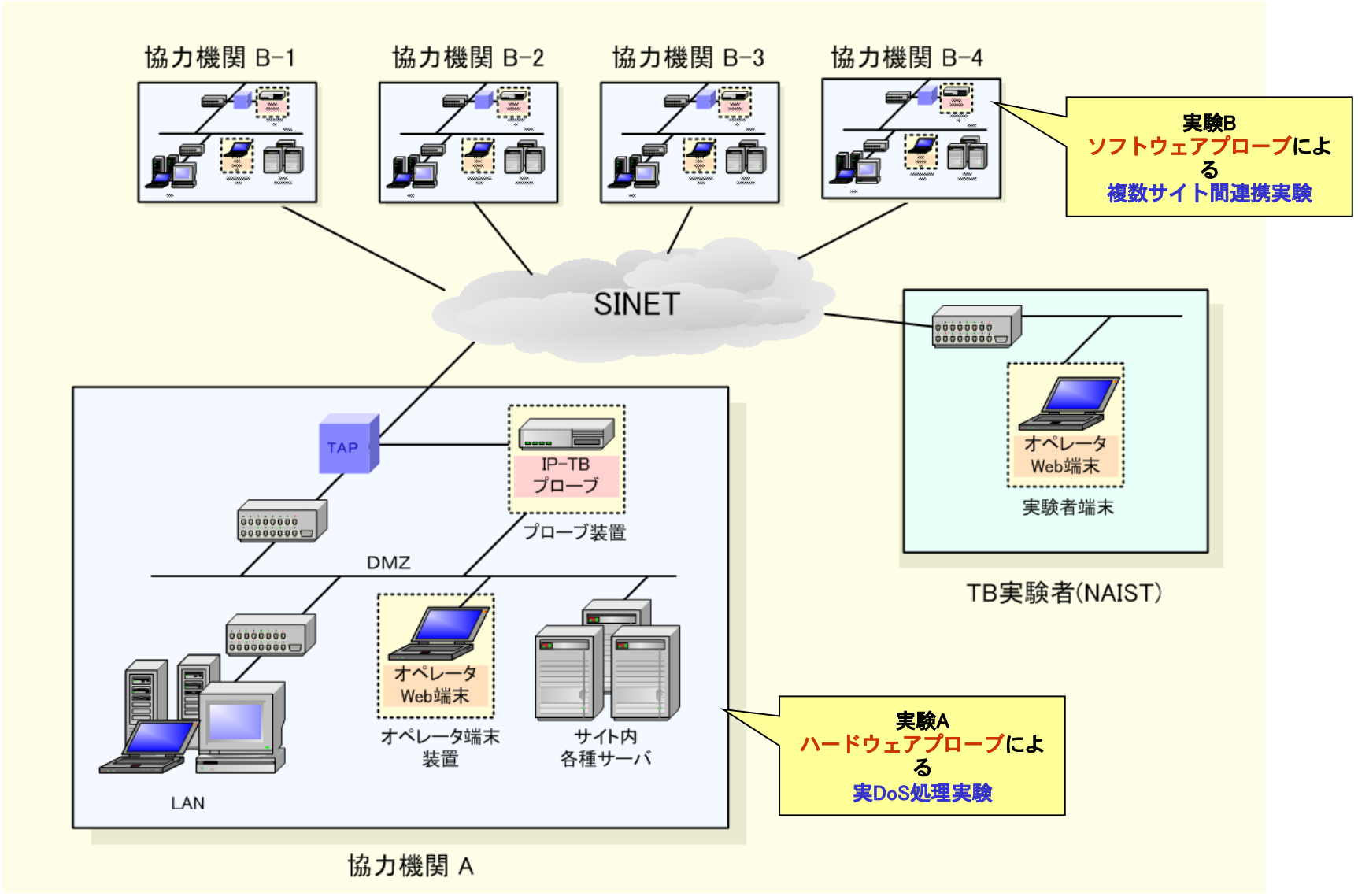
- テストベッド（40ノード）を構築し機能・動作検証
- NICT北陸リサーチセンター(Starbed) において200台規模での動作を検証
- 追跡可能性や追跡精度の向上に関する実験を実施

H17実証実験

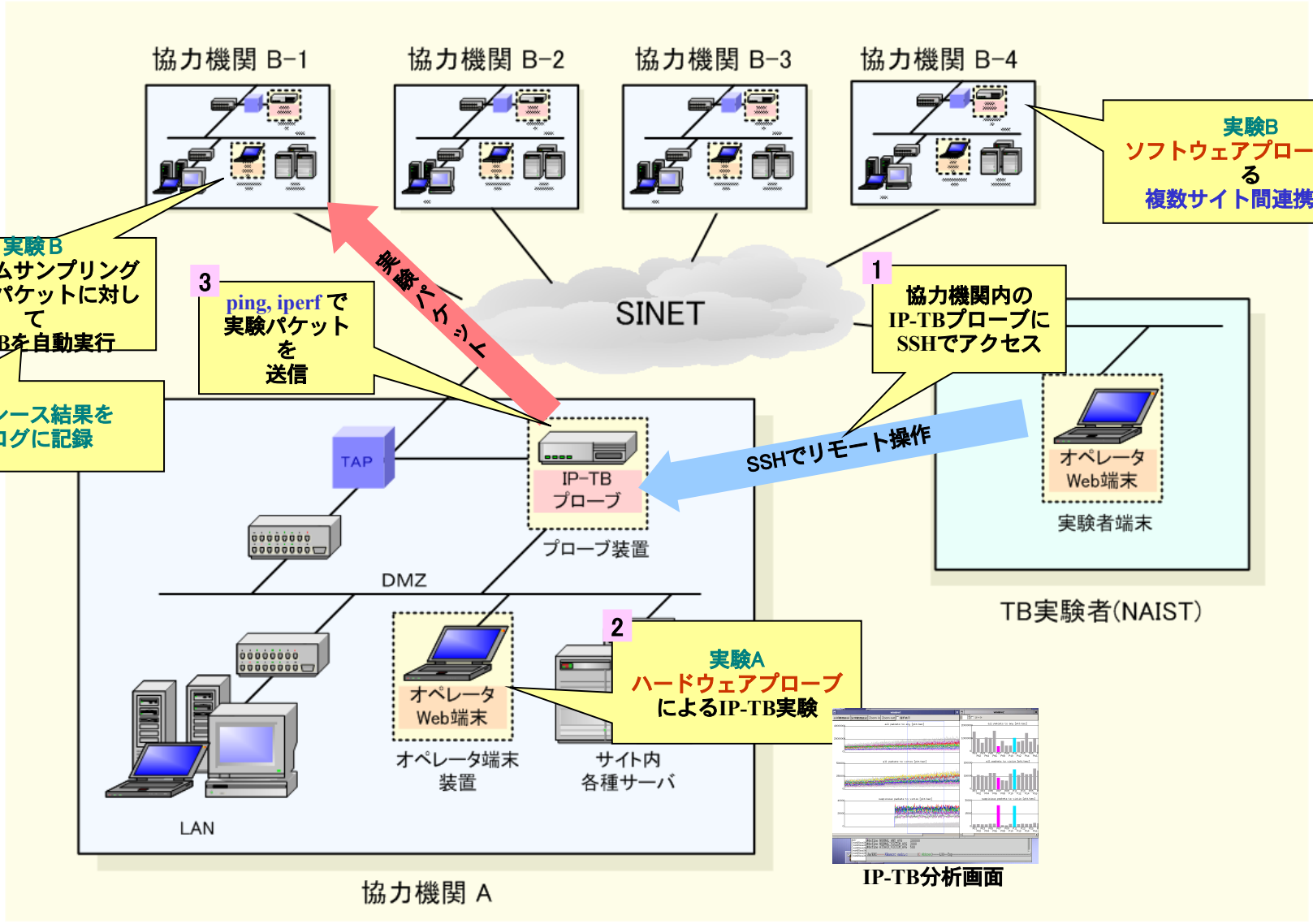
- 京都大学ハニーポットにて攻撃ベクタ収集
- セキュアウェア・攻撃コード判定システムにて攻撃判定
- トレースバックシステムにて攻撃元特定



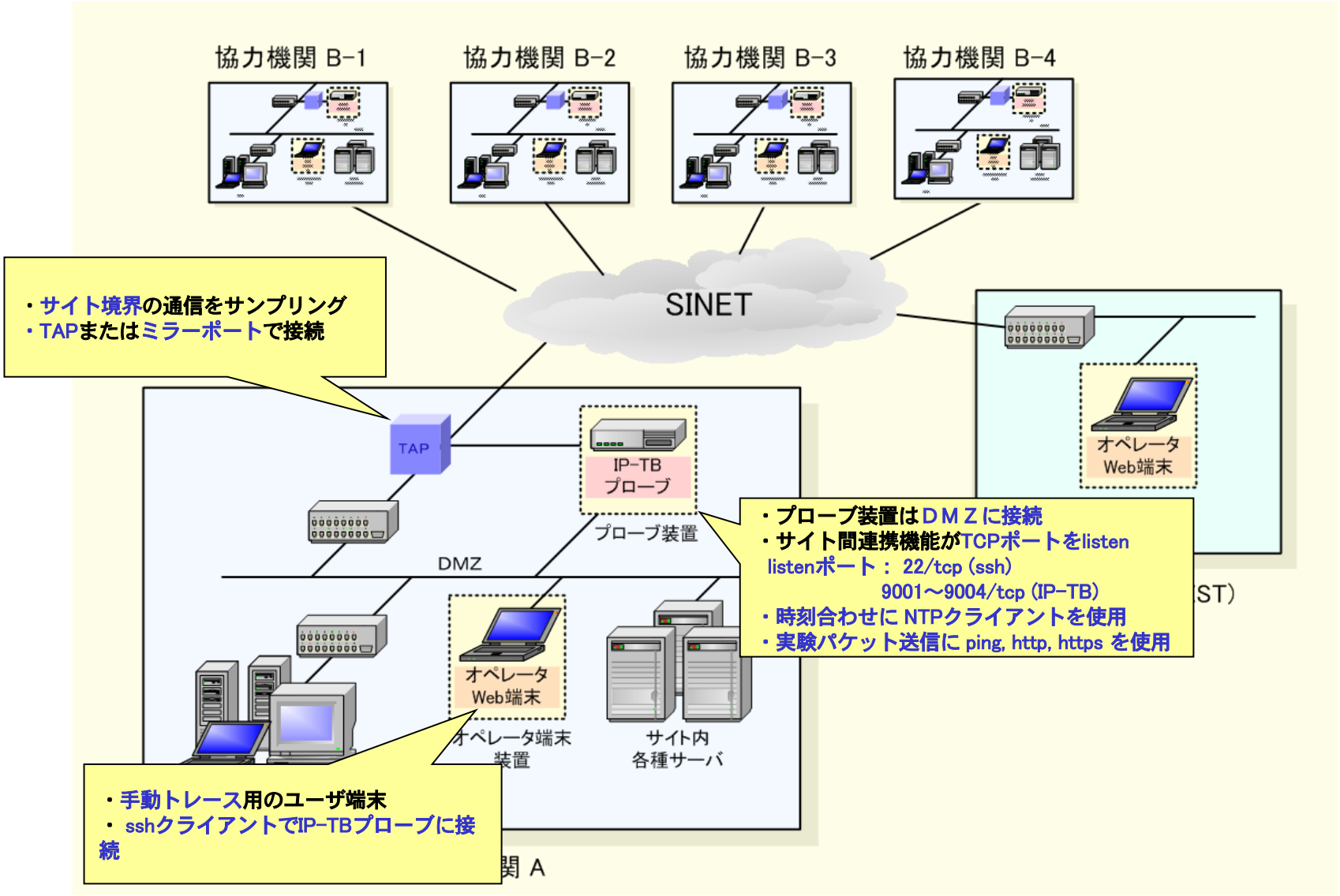
H19 実証実験 - システム構成



H19 実証実験



H19 実証実験 – 機器構成



H20実証実験にむけて

- 実証実験にご参加いただけるISPを募集しています
- Telecom-ISA 若狭さん secretariat <at> telecom-isac.jp までご連絡ください

Q&A

どうもありがとうございました