

---

# IPトレースバックとその応用

## JANOG 19

---

門林 雄基(奈良先端科学技術大学院大学)

許 先明(株式会社ブロードバンドセキュリティ)

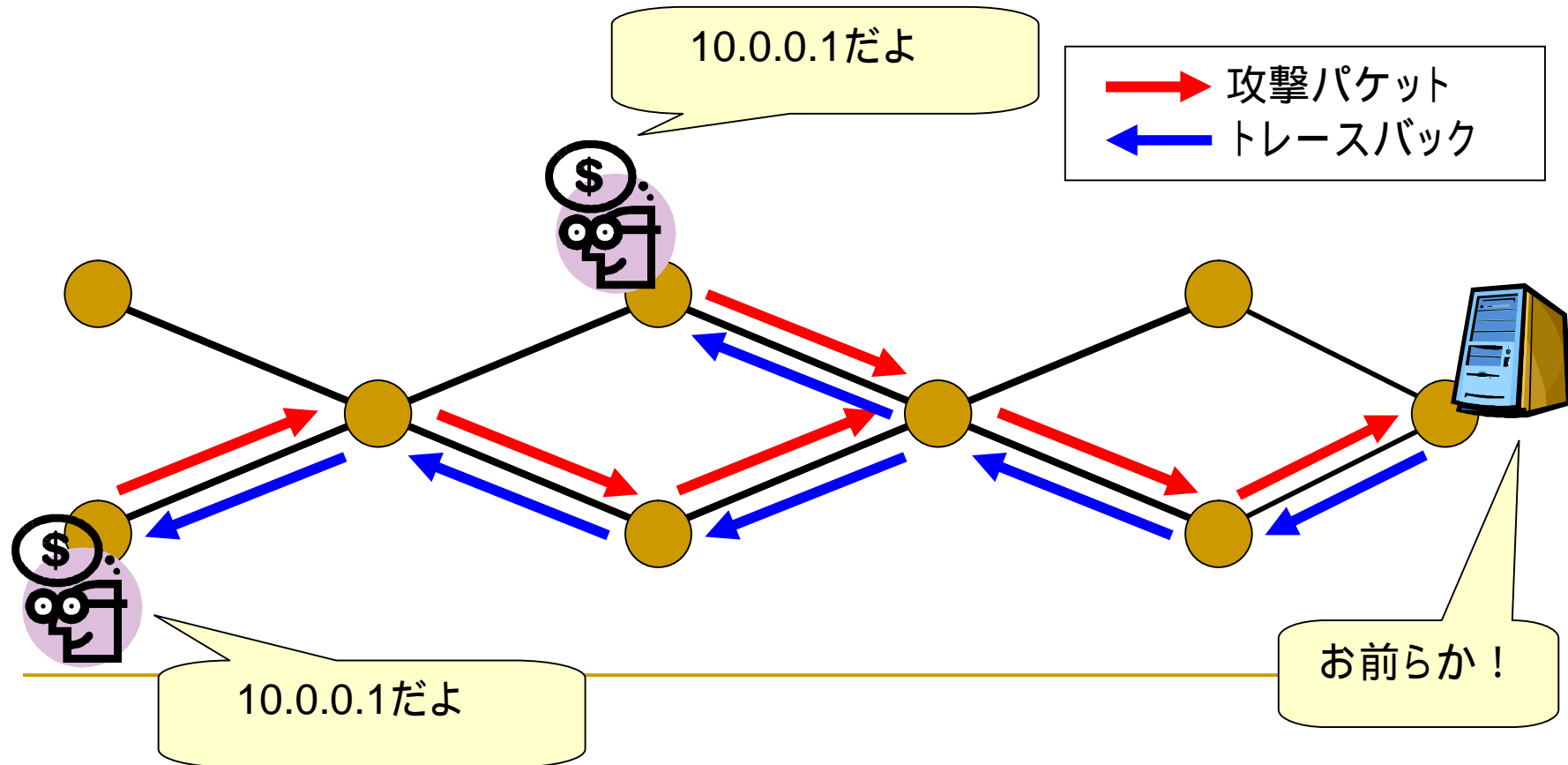
---

IPトレースバックの5W1Hを考える

---

# IPトレースバックとは

- 始点アドレスが詐称された通信の発信源をつきとめる技術
- 「IPパケットの発信源追跡」



---

## IPトレースバックが必要なわけ

- 始点アドレスが詐称されたDDoS
  - 始点アドレスが詐称されたUDP exploit
  - 始点アドレスが詐称されたDNSクエリ
  - ...
  - 発信源を突き止めて、何らかの対処をしたい
    - 遮断する、ワームを除去する、ユーザに警告する等
  - RTBH, input debugging
    - 手作業      手間がかかる
-

---

# 誰がIPトレースバックを使うのか

- ISPオペレータ
  - JPCERT?
  - SSP (セキュリティサービス事業者)?
  - 法執行機関?
- 
- 官憲の手を借りずに、できるだけ民間だけでインシデントを解決できるようにしたい
  - 「悪いことをしたら足がつく」可能性をつくってDDoS攻撃などの抑止力にしたい
-

---

## IPトレースバックと時間軸

- DDoSの最中や直後にIPトレースバックを実施
  - 自動判定がほぼ前提
  - 時間がたってからだと記録が残っていない可能性あり
  - 遡及能力はメモリオーバーヘッドに跳ね返る
-

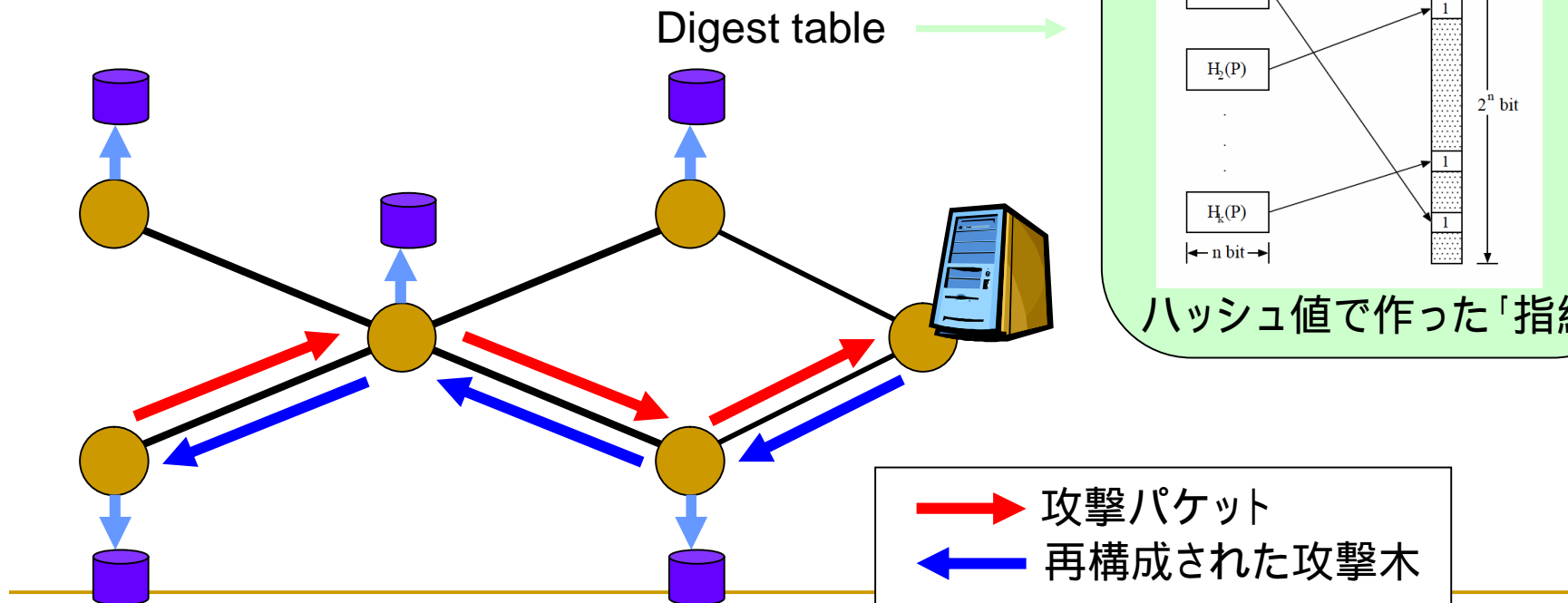
---

# トレースバックの起点と範囲

- 起点はパケットが通過したどこか
    - ルータ、スイッチ、IDSなど
  - 流入口がどこか分かれば...
  - 自分のプロバイダが発信源かどうか分かれば...
  - どのエッジの先か分かれば...
  - 国境まで切り分けられれば...
-

# IPトレースバックの動作原理

- パケットの一部のハッシュ値をいくつか計算し、「指紋」としてコンパクトに記録

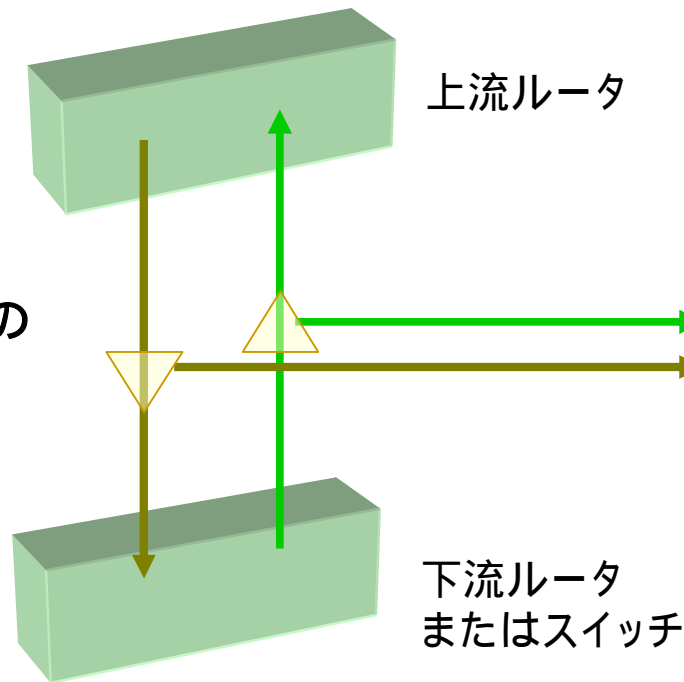




# IPトレースバックの運用形態その1

いまのところルータにトレース機能がないので、

光スプリッタ(あるいはメタルのスプリッタ)で信号を分岐して  
トレースバックノードへ。



peering

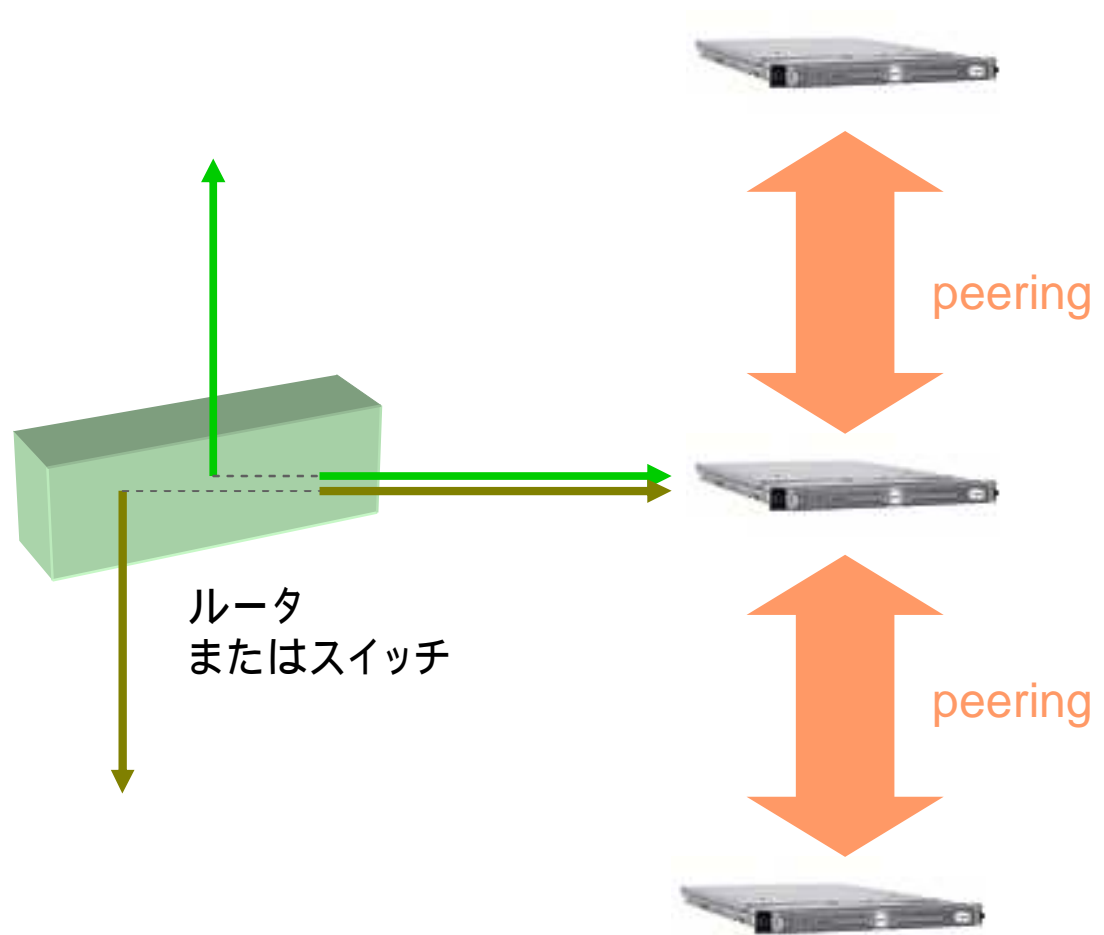


peering

隣接トレースバックノード

## 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。



---

IPトレースバックの効力を見極める

---

---

# IPトレースバックのスケールビリティ

- 18ノードを実験的に設置(東京、奈良など)
  - 100ノード程度はテストベッド検証済
  - 今後1000ノード以上の検証が必要
    - テストベッド、VM併用
-

# IPトレースバックの精度

- 誤検知率
  - メモリオーバーヘッド、計算オーバーヘッド次第
  - (例えば)0.3%

Bloom filters are typically described in terms of the number of digesting functions used and the ratio of data items to be stored to memory capacity. The effective false-positive rate for a Bloom filter that uses  $m$  bits of memory to store  $n$  packets with  $k$  digest functions can be expressed as

$$P = \left( 1 - \left( 1 - \frac{1}{m} \right)^{kn} \right)^k \approx \left( 1 - e^{-kn/m} \right)^k .$$

# IPトレースバックの動作速度

- 250ms以内に応答
- 最大2秒

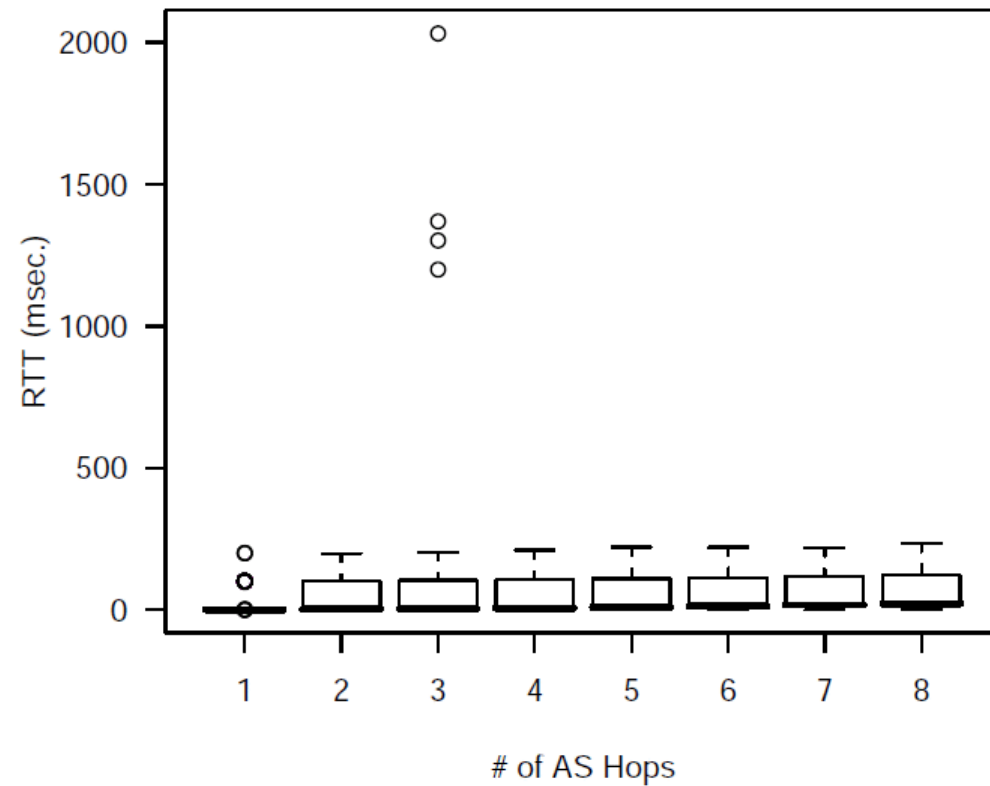


Figure 3.11: RTT of a ITM Traceback Request in a liner topology

---

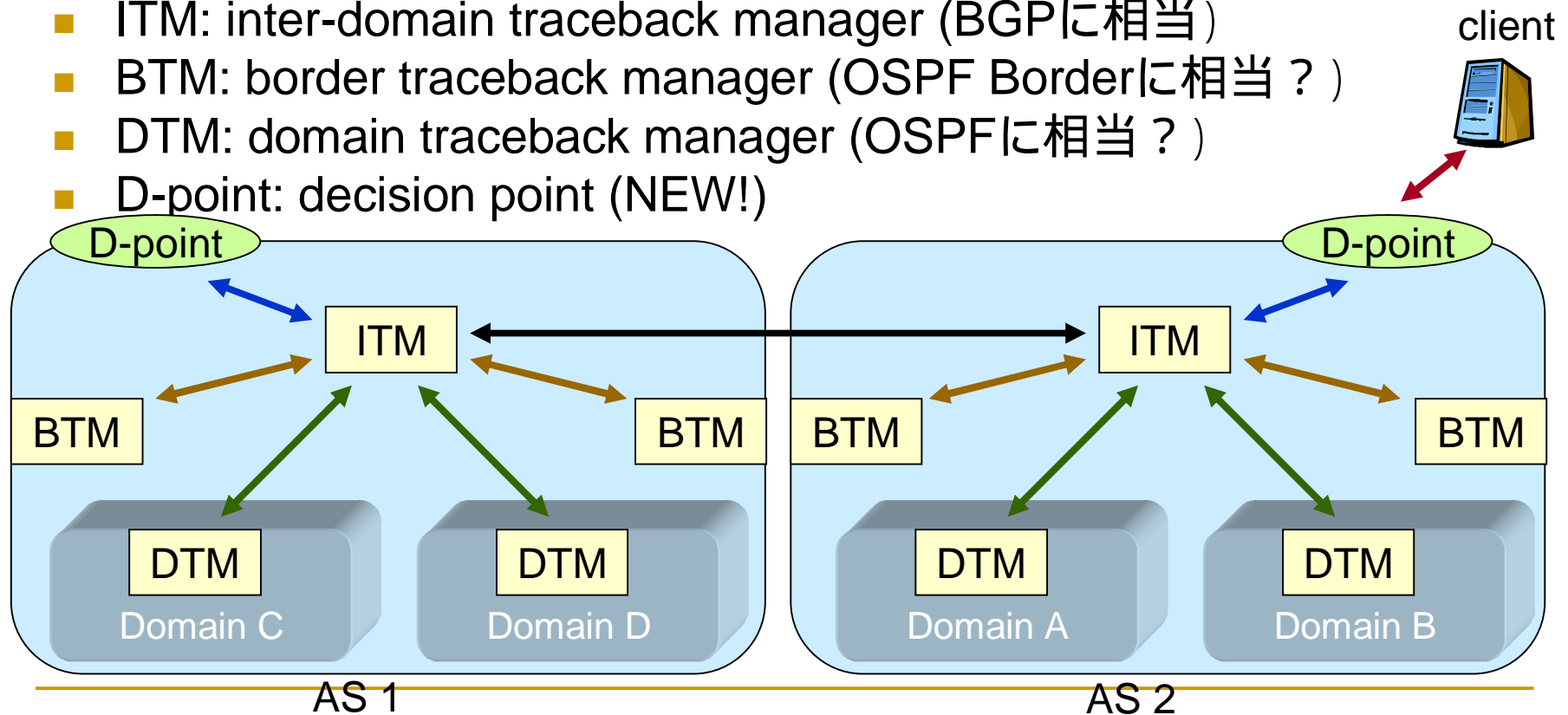
# 奈良先端での取り組み

- 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 アーキテクチャ

## ■ 特徴

- 多様なトレースバック方式をつかえる
- NAT, Firewallを超えた探知ができる(はず)
  - NAT, Firewallでの対応が前提
- ITM: inter-domain traceback manager (BGPに相当)
- BTM: border traceback manager (OSPF Borderに相当?)
- DTM: domain traceback manager (OSPFに相当?)
- D-point: decision point (NEW!)

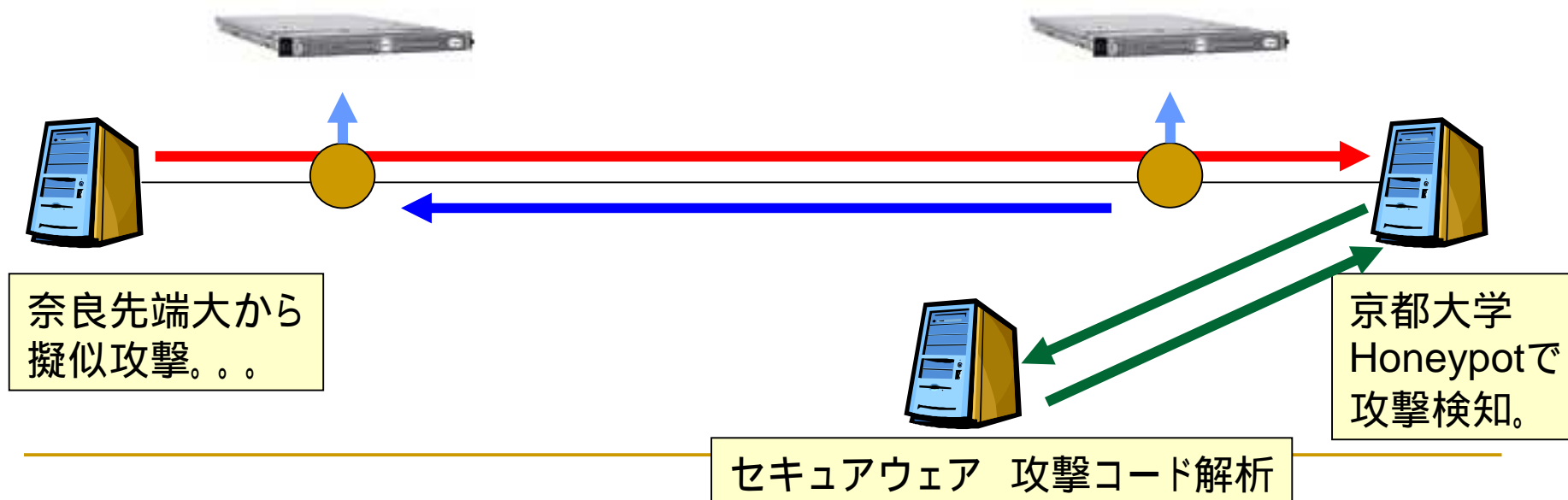




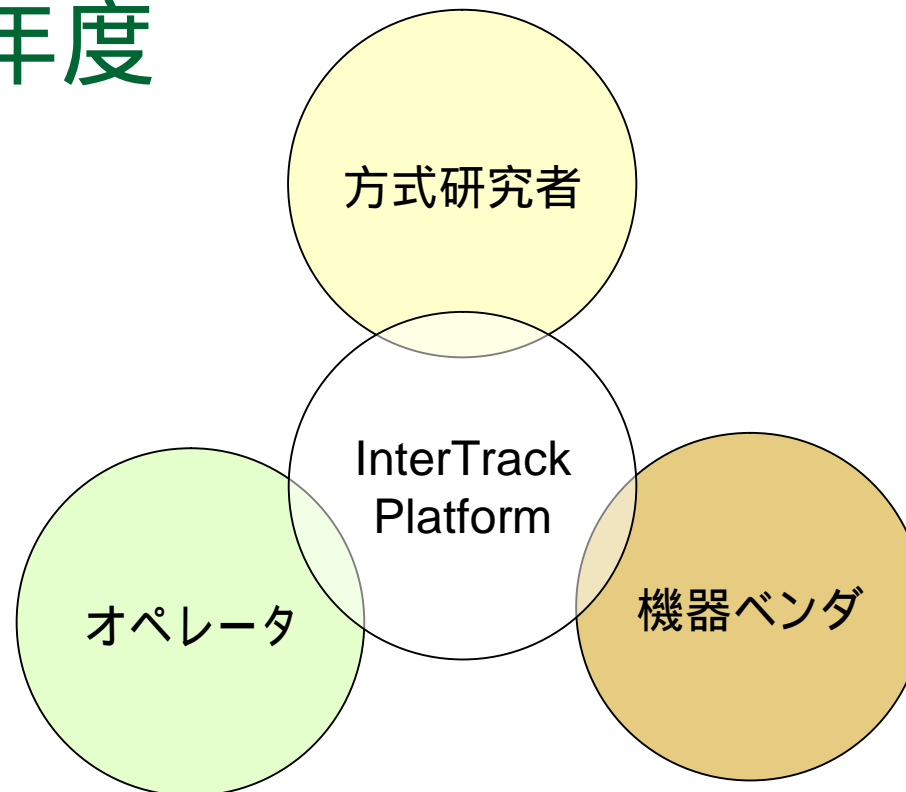


# つながる・ひろがる・ひびきあう (平成17年度)

- 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・システムにて攻撃コード解析
- トレースバックシステムにて攻撃元特定



# 平成18年度



- 通信の秘密に関する取り組み
- オペレータとの協働
  - 運用系ネットワークへのテスト導入
- 各種機器ベンダとの協働

来年度以降あそんでくれる人、募集中！

# InterTrack のCLI

```
matsu@intertrack:~/simple01/intertrack/src/c++/libintertrack/session — ssh — 154x61
TC> show query
time_local = 1168568988.260814
time_global = 1168568988.368585
hash = ac50d483d5163712d9bbbf190e0364a

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

  <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">ac50d483d5163712d9bbbf190e0364a</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="Int

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

# InterTrack のログ

```
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: not found (hash = 889f853c261d398023b2918b86370e20 )
Jan 12 11:28:19 dhcp07 TC : [info] ClientTrace Result AS=2503: not found (hash = 889f853c261d398023b2918b86370e20 )
Jan 12 11:28:19 dhcp07 TC : [info] ClientTrace Result AS=2504: not found (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: not found (hash = 889f853c261d398023b2918b86370e20 )
Jan 12 11:28:19 dhcp07 TC : [info] ClientTrace Result AS=2503: not found (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
[]

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:not found[b09cdc4e0da4d55f892c938e431f14cc]
2007-01-12 11:27:48.756 ITM [info] Trace Info AS=2503:not found[b09cdc4e0da4d55f892c938e431f14cc]
```

---

# IPトレースバックの諸問題を 点検する

---

---

# 運用面の課題を点検する

- Peering
    - 隣接ASが対応していない場合は  
すっとばして peer することも可能
  - トレース結果
    - 経路ASが分かったあとの運用フローは  
電話、IM, Jabberなど  
オペレータ支援システムも開発中
  - IDSとの連動
    - pcap フォーマット等でのパケット記録が前提
-

# コスト面の課題を点検する

- 初期導入コスト
  - 光スプリッタの場合...
    - 上り、下りに1つずつ、
  - ポートミラーリングの場合...
    - 機材コストはゼロ？、構成に制約あり
  - ノード
    - PC, OS, peering 設定
- 定常運用コスト
  - Resolverなみ？





---

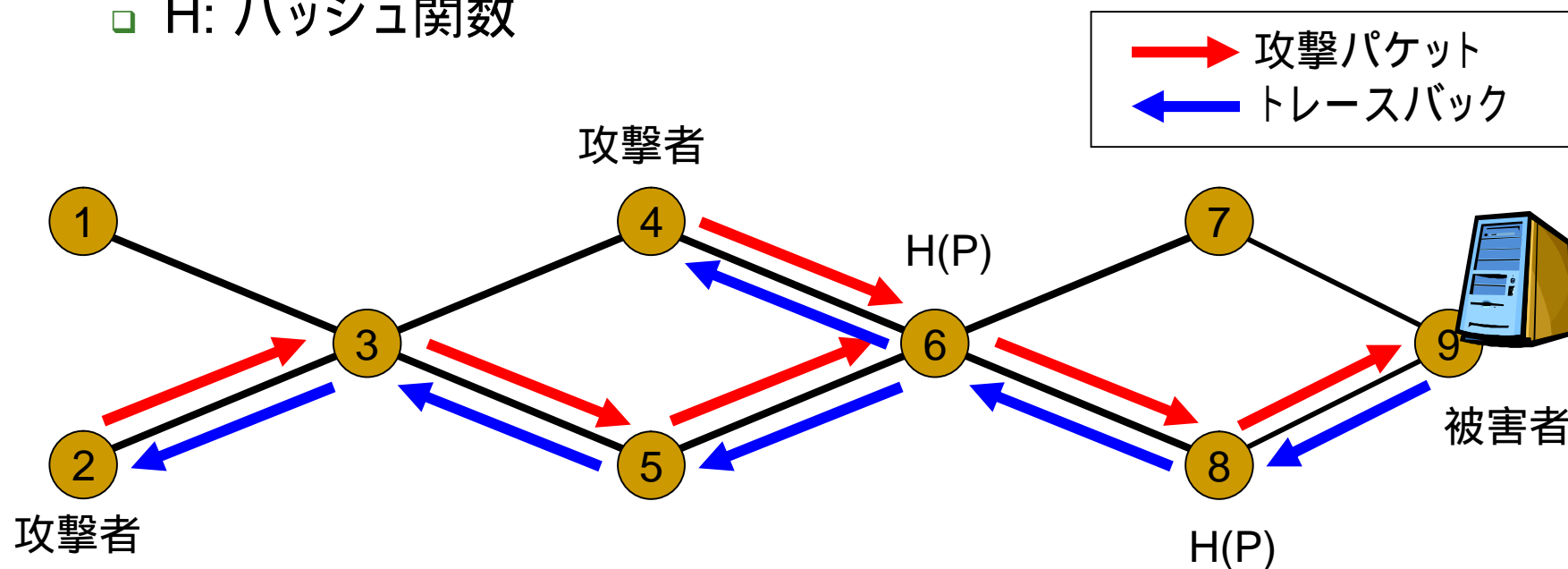
# 法律面の課題を点検する

## ■ 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。
  - 電気通信事業分野におけるプライバシー情報に関する懇談会「電気通信事業における個人情報保護に関するガイドライン」：**最終報告書まだ**
  - 総務省の勉強会どうなった？
-

# IPトレースバックにおける通信の秘密

- IPヘッダ、ペイロードは通信の秘密にあたる
- パケットPについて $H(P)$ を用いて問い合わせ
  - H: ハッシュ関数



(注) 6 と 8 は独立した通信事業者であり、ヘッダやペイロードを互いに漏洩してはならない。