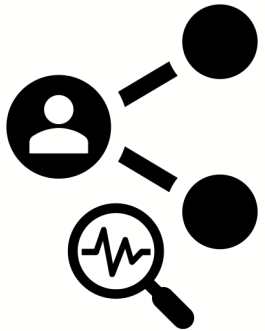
スキトオル。トラフィック
−monitoring your traffic−

Shishio Tsuchiya
shtsuchi@arista.com

ARISTA

はじめに

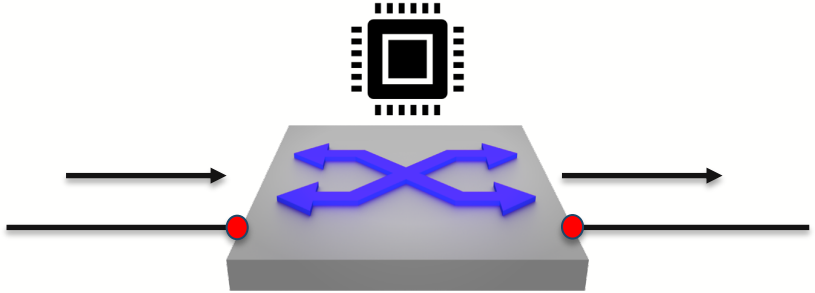
- ネットワークに流れるトラフィックを収集することは収容設計の計画を立てる意味でも、トラブル時に解析する意味でも非常に重要
- トラフィックをスキトオル。手法にフォーカスして説明し議論を行いたい



話すようなこと

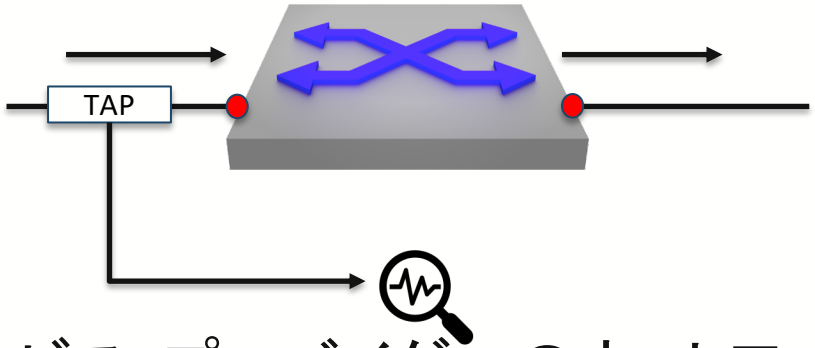
- ポートミラーリング
- Tap Aggregation
- DPI
- Telemetry など

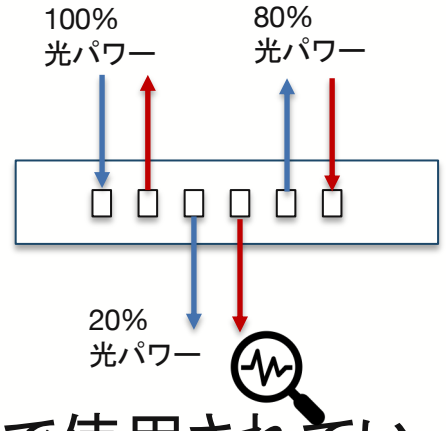
パケット処理



- ルーティングなどのプロトコルを処理するCPUコントロールプレーンとデータをフォワーディングするデータプレーン
- 通常ルータのCPUではコントロールプレーン以外はデータプレーンのカウンター情報のみが渡されるため、詳細パケットに関しては認識出来ない
- コントロールプレーンはTCPDUMPなど使用できる機器も最近は多い

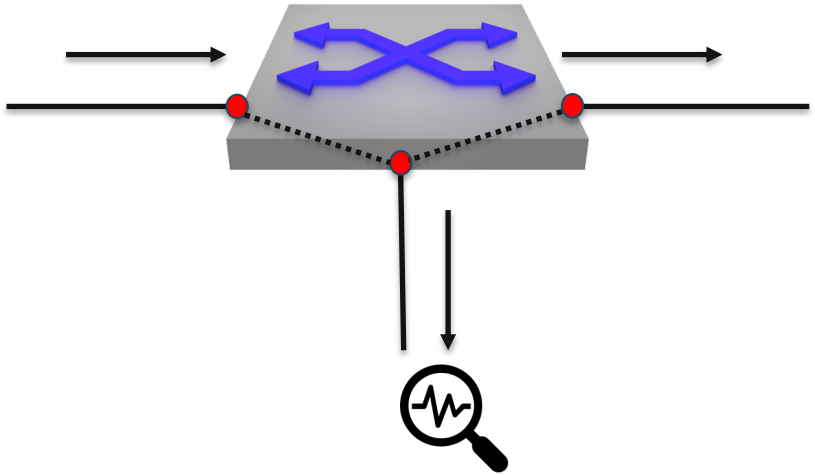
A Network TAP (Terminal Access Point)





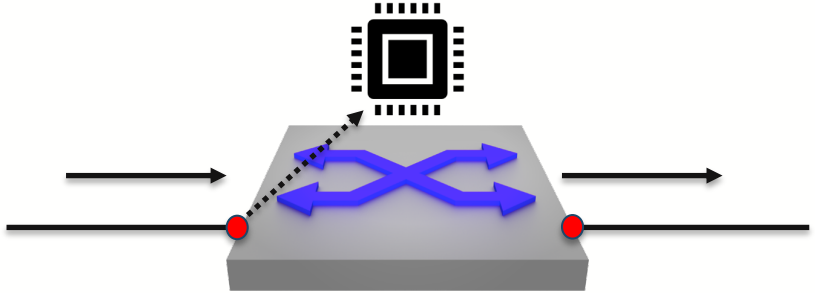
- 多くのサービス・プロバイダーのネットワークで使用されている
- 事前に用意されていると良いが後ほどの導入は難しい
- 設定などはいらない

ポートミラーリング



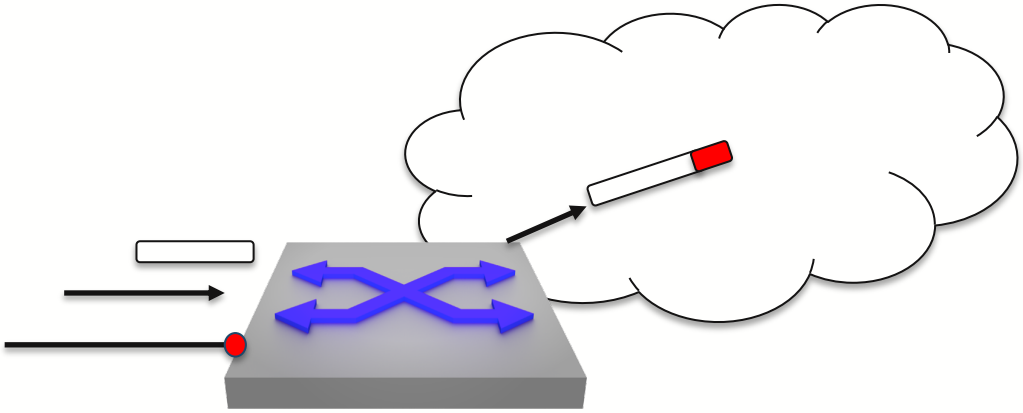
- 受信ポート/送信ポートの通過するトラフィックを他のポートに流す
- ポートミラーリング/SPAN(Switched Port Analyzer)ともいう
- ACLを使って、必要なものだけにフィルタリングする事も可能

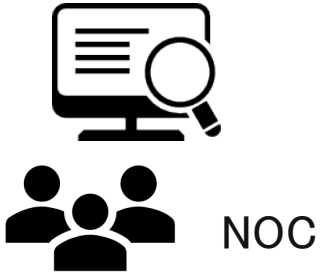
Mirroring to CPU



- ミラーリング先をポートではなく、CPUにする事でデータプレーントラフィックもtcpdump可能に
- CoPPなどのCPU保護をする機能は重要に

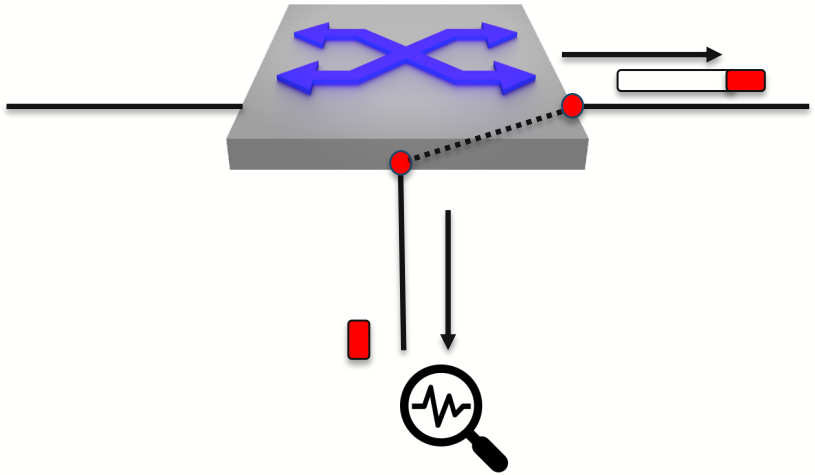
ポートミラーリング to Tunnel





- NOCなどで即座にデータプレーントラフィックを解析する必要がある際にはトンネルをし、パケットを遠隔地に送る
- 基本的にGREはどんなパケットでもカプセル化する事が出来るので使用される
- ただしベンダーの実装によってはいくつかの種類もあるため注意も必要
  - https://tools.ietf.org/html/draft-foschiano-erspan

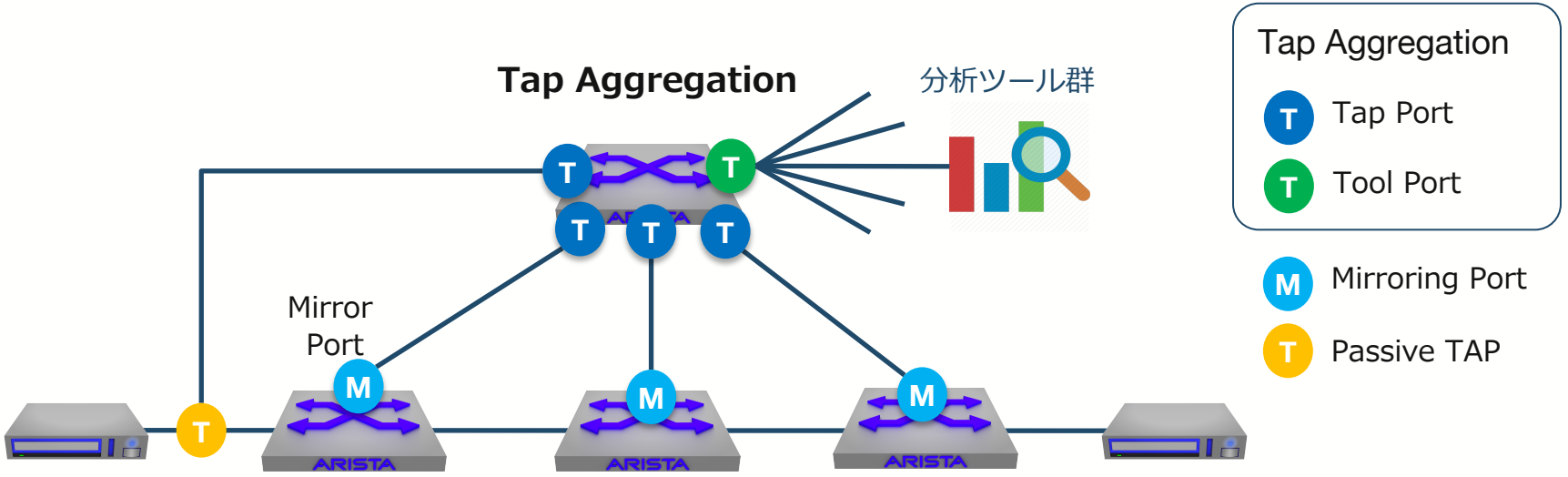
ミラーリングTruncation

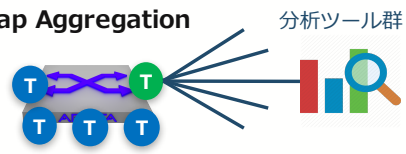


- トラフィックトレンドの調査などはパケット全体が必要になるケースは少ない
- ヘッダー情報のみで十分なケースも多く、truncationは有効な手段となる

Tap Aggregation

- タップまたはミラーリングされたデータの集約と分析ツールへのデータ転送
- 分析ツールへの投資を最小限に





Tap Aggregation Configuration

```
switch(config)#tap aggregation
switch(config-tap-agg)#mode exclusive
```

- Tap Aggregationを有効化

```
switch(config)#interface ethernet 41-43
switch(config-if-Et41-43)#switchport mode tap
switch(config-if-Et41-43)#show interface ethernet 41-43 tap
Port      Configured     Status         Native    Id   Truncation Default
          Mode                          Vlan      Vlan            Group
-------------------------------------------------------------------------
Et41      tap            tap            1         1    0          ---
Et42      tap            tap            1         1    0          ---
Et43      tap            tap            1         1    0          ---
switch(config-if-Et41-43)#
```

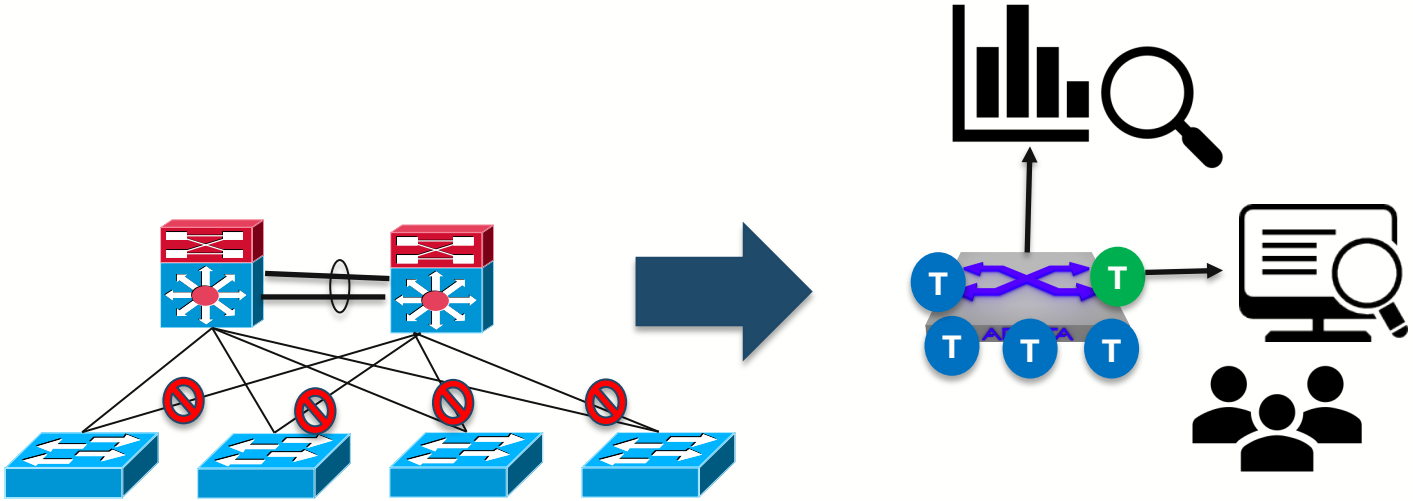
- tapポート設定

```
switch(config)#interface port-channel 101-103
switch(config-if-Po101-103)#switchport mode tool
switch(config-if-Po101-103)#show interface port-channel 101-103 tool
Port      Configured     Status         Allowed              Id   Timestamp
          Mode                          Vlans                Tag  Mode
-------------------------------------------------------------------------
Po101     tool           tool           All                  Off  ---
Po102     tool           tool           All                  Off  ---
Po103     tool           tool           All                  Off  ---
switch(config-if-Po101-103)#
```

- toolポート設定

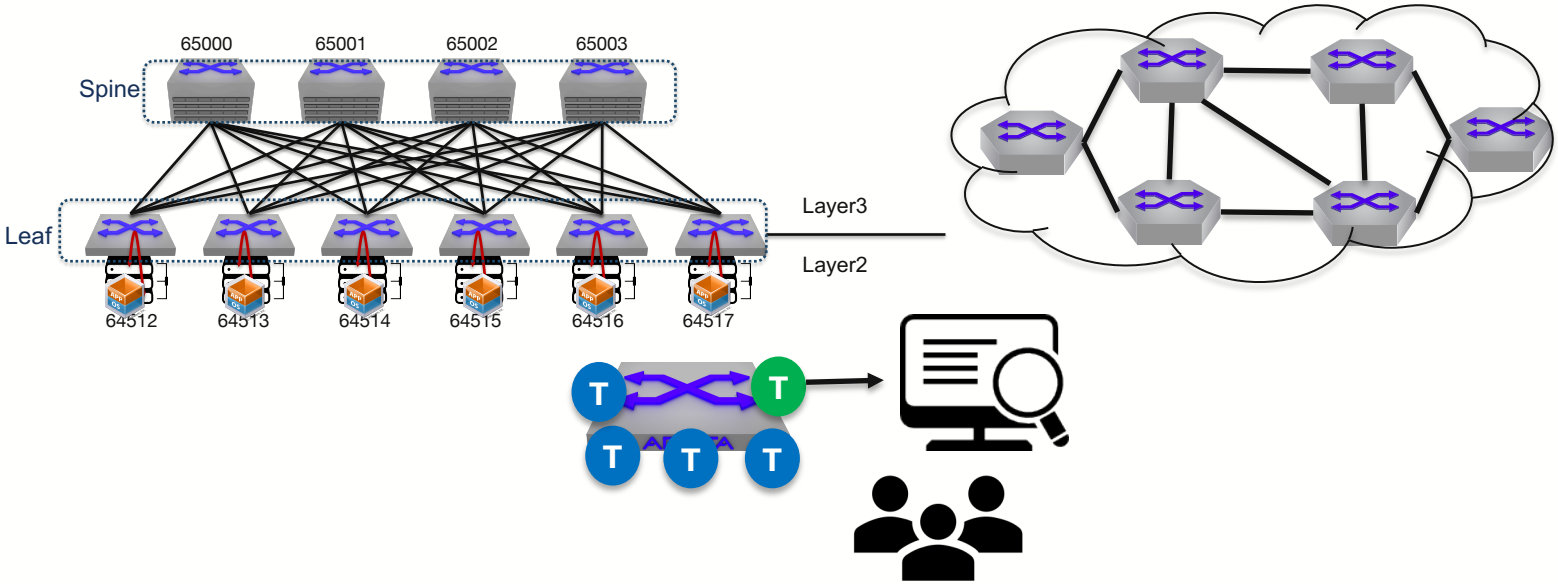
ARISTA

レガシーネットワークのためのフローデータ



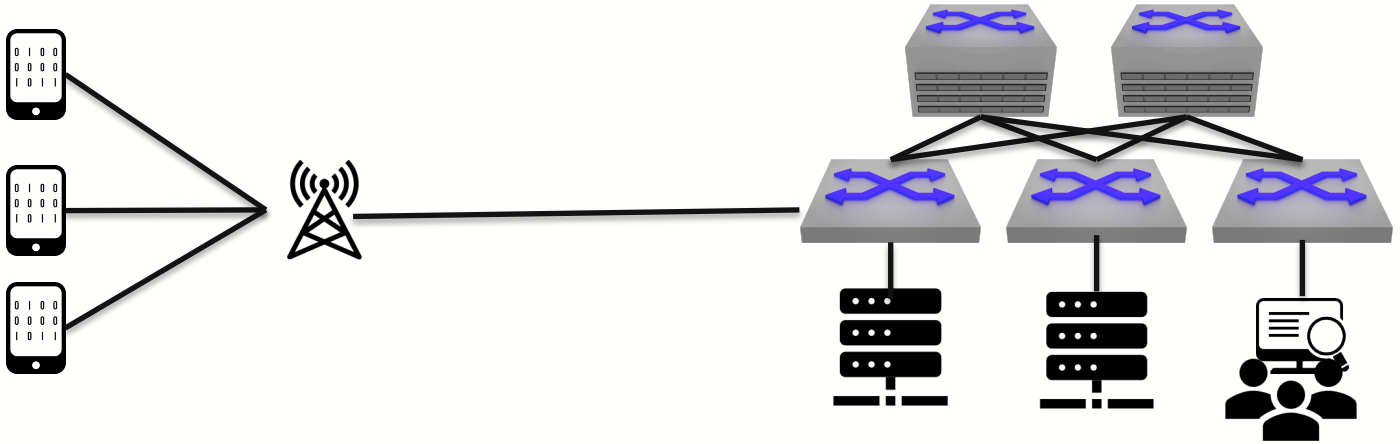
- レガシーのネットワーク機器ではsFlow/Netflowなどのフロー機能をサポートしてないケースも多い
- 一方フロー情報はセキュリティなどのためにも必要になる
- ミラーリングなどはサポートしてるため、TapAgg機器に情報を集め、フロー情報とパケット情報の両方を抜き出す

ヘッダーリムーバル



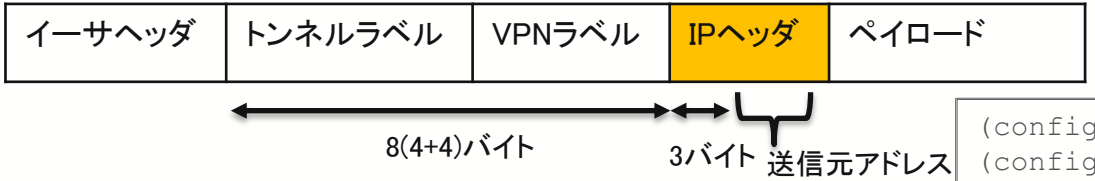
- サービス・プロバイダーでのMPLSやデータセンターでのVXLANなどコアネットワークではトラフィック解析や分析も少し難しくなる
- ヘッダーを除去して、トラフィック装置へ転送

なにで振り分けたいか？

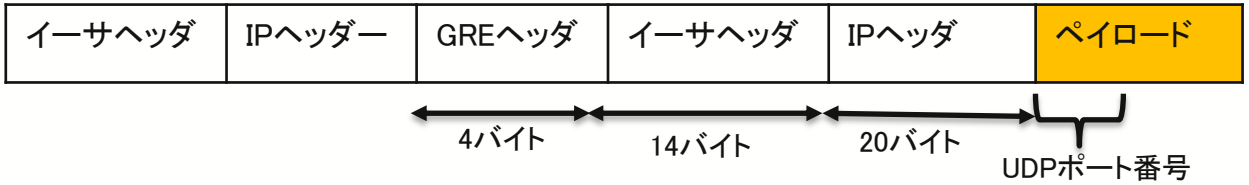


- カプセル化されたパケットの内部ヘッダー
- HTTPクライアントのOS種別など(UserAgent)
- シリアルナンバーやIMSI/GTP TEIDなど

DPIを利用したトラフィックステアリング



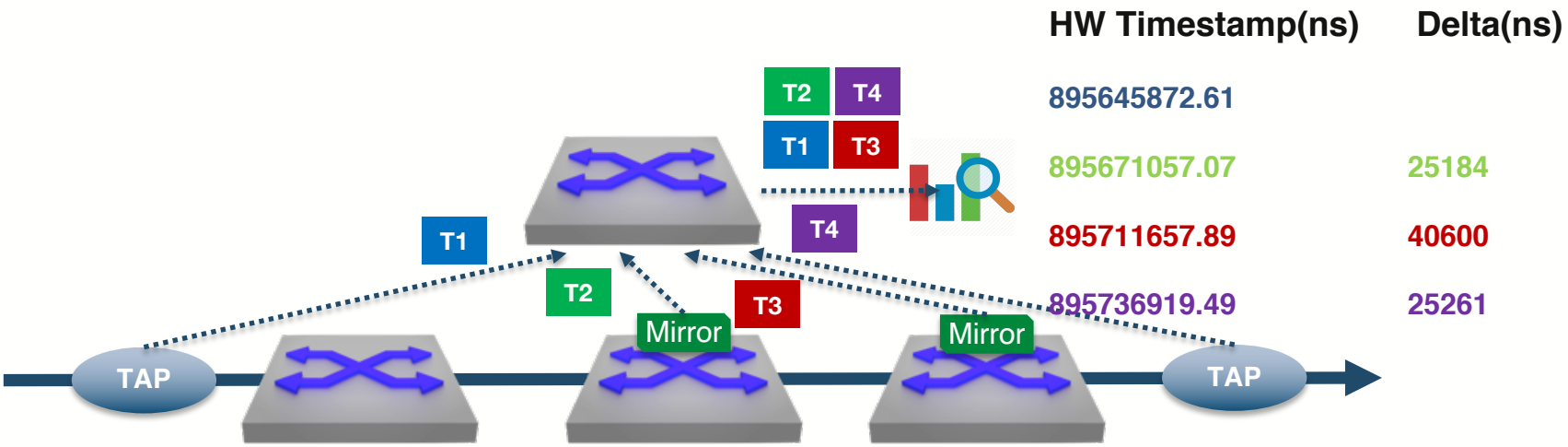

```
(config)# mac access-list <name>
(config-acl-<name>)# permit mac <mac-header-
options>
payload offset <offset> pattern <pattern> mask
<mask>
```




```
(config)# ip access-list <name>
(config-acl-<name>)# permit ip <ip-header-options>
payload header <start|end> offset <offset>
pattern <pattern> mask <mask>
```

- ヘッダだけでは無くペイロードの中身や内部にあるIPヘッダやUDPヘッダを見てのトラフィックステアリングが必要
- オフセットとパターンの組み合わせで認識する

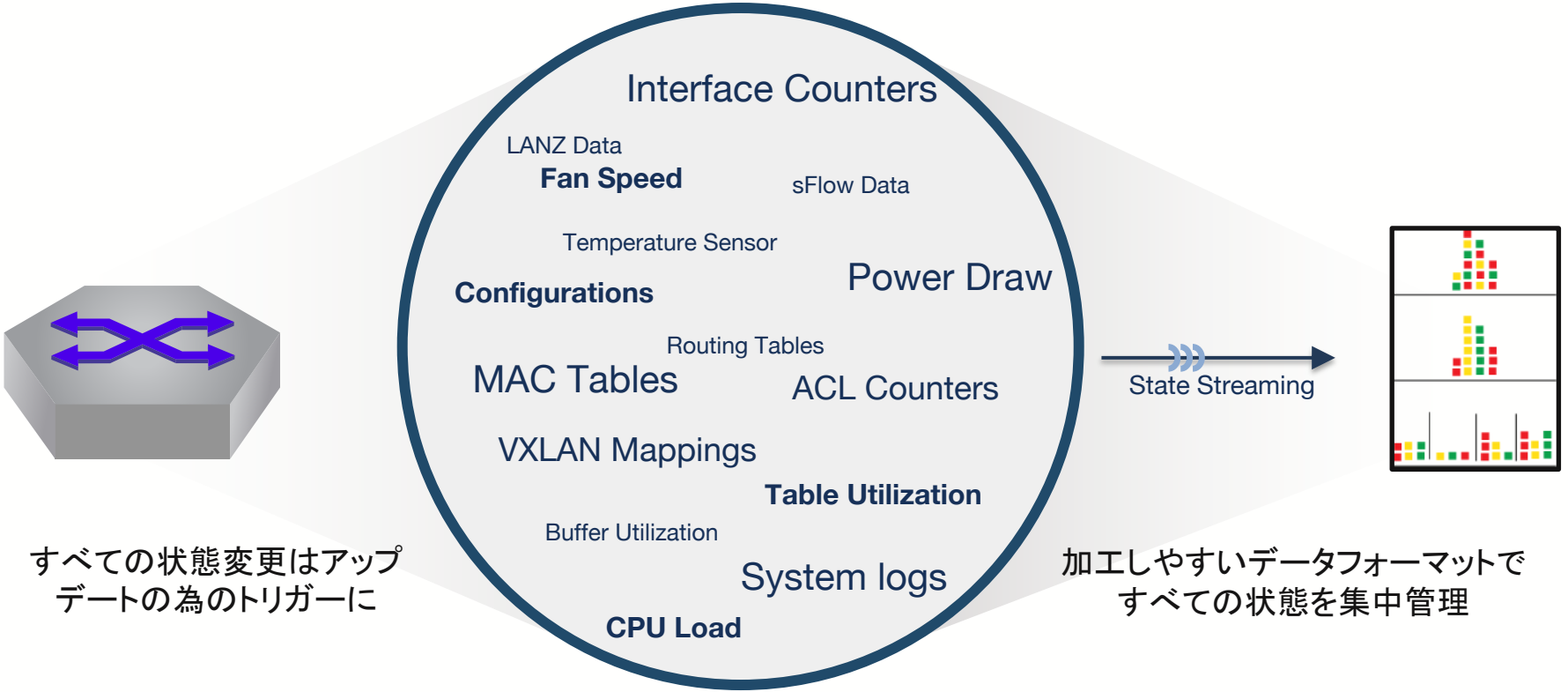
ネットワークの遅延情報のモニタ
パケットタイムスタンプ



- 各ポートでタイムスタンプ情報を付加し、TAPアグリゲーションでモニタリングする

ARISTA

ステートストリーム



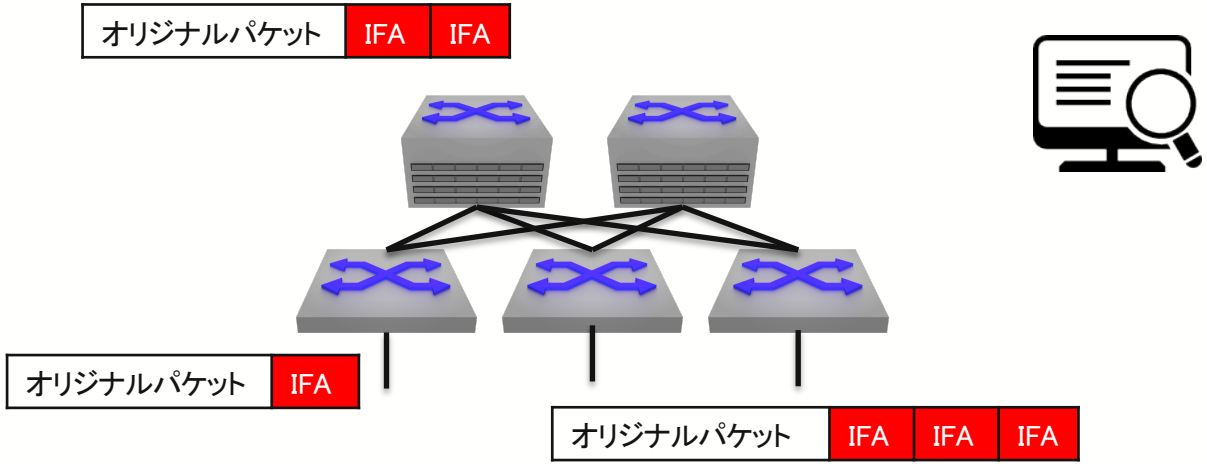
すべての状態変更はアップデートの為のトリガーに

加工しやすいデータフォーマットですべての状態を集中管理

すべての状態変化をあらゆるデバイスから即座に

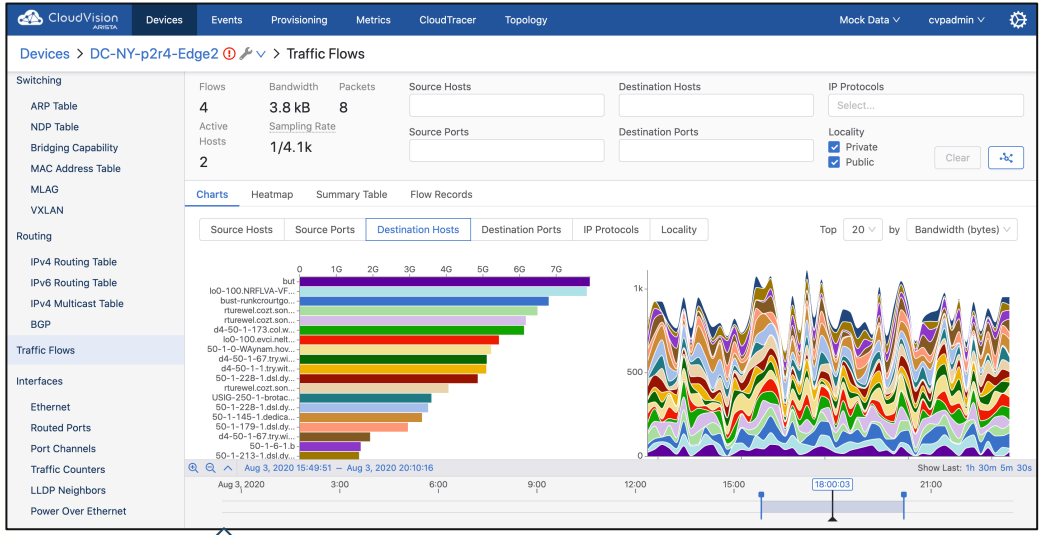
ARISTA

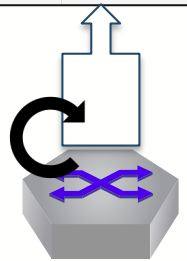
Inband Flow Analyzer–Inband Telemetry



- Inband Flow Analyzer
  - https://tools.ietf.org/html/draft-kumar-ippm-ifa
- Hop-by-Hopもしくは宛先でメタデータを埋め込む
- その情報を元にフロー毎(パケット毎)のネットワークにおける遅延時間などをコントローラーで収集する

Flow データ to ストリーミング



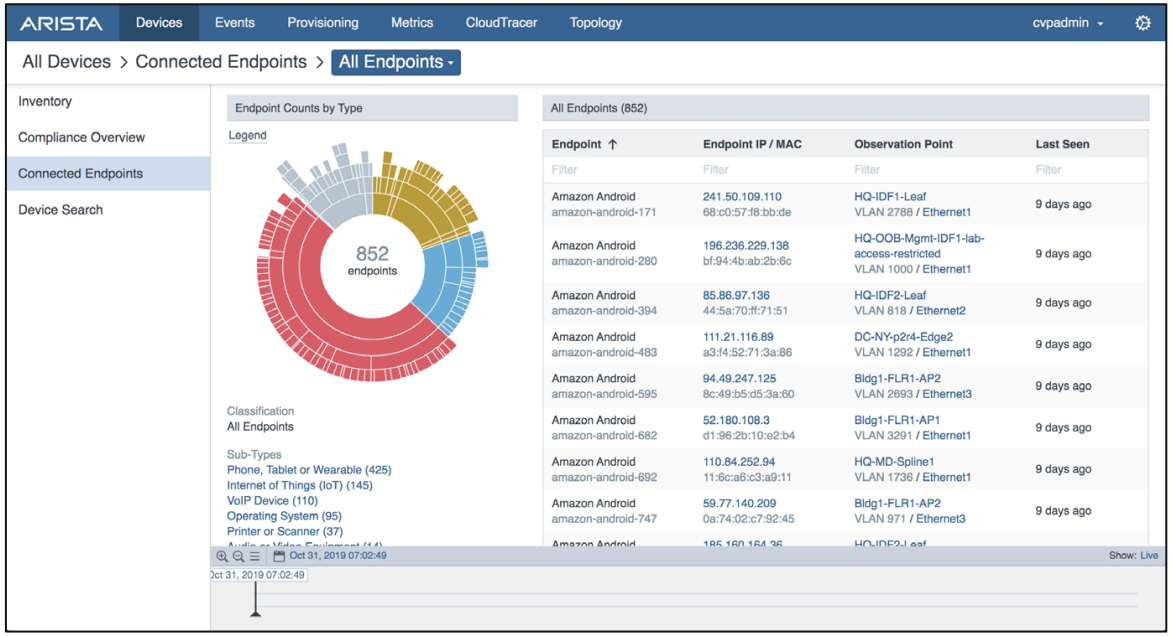


```
sflow destination 127.0.0.1
sflow source-interface Management1
sflow run
```

- フローデータはトラフィックトレンドを抑えるに非常に便利
- ストリーミングデータは加工がしやすい
- Flow->Telemetryへ

ARISTA

端末情報認識(DHCPフィンガープリント)



https://github.com/karottc/fingerbank

- OUIやDHCPリクエストのベンダー識別子(60)およびパラメーター要求リスト(55)を元にベンダーを特定する

まとめ

- トラフィックをスキトオラセル。手法に関してまとめた
- トラブルシューティングやトラフィック解析でやっている手法など共有議論したい