



InfluxDB+Telegraf+Grafanaによる ネットワークモニタリング

Tetsuhiro Sato

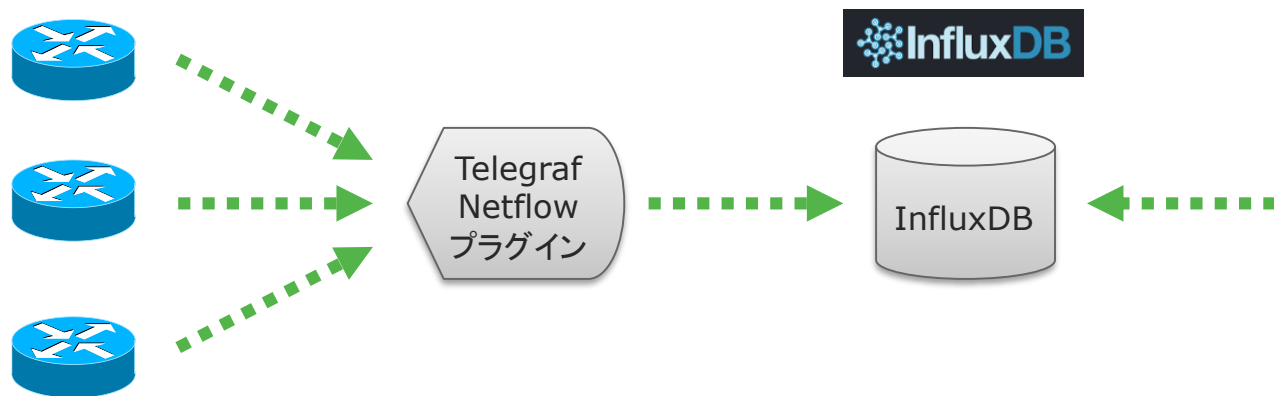
Cisco Systems G.K.

自己紹介

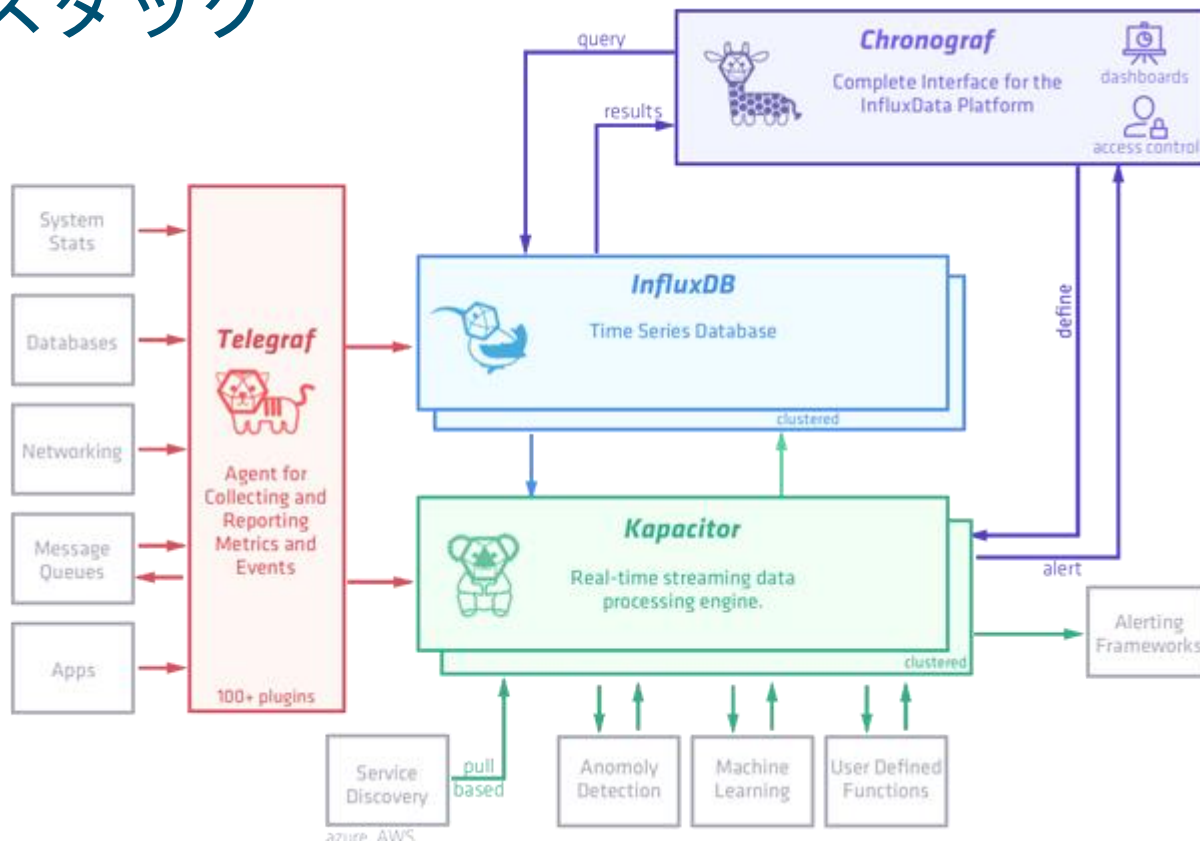
- ・ 仙台生まれ、仙台育ち
 - ・ シスコシステムズ合同会社っていうネットワーク機器ベンダでSDNとかプログラマビリティっぽい仕事してます
 - ・ ネットワークプログラマビリティ勉強会っていうIT勉強会を企画、運営してます
 - ・ ネットワークの運用（Ops）の課題をソフトウェアエンジニアリングによって解決すべき事柄と捉える人々で集まって意見交換
- <https://network-programmability.connpass.com/>
- ・ 勉強会で何か発表してくれる人探してます！

本日のお題

- ・ ネットワークのメトリック情報をInfluxDBにぶち込むTelegrafのNetflowプラグインを作ってみたよ
- ・ さらに、Grafanaで可視化してみたよ

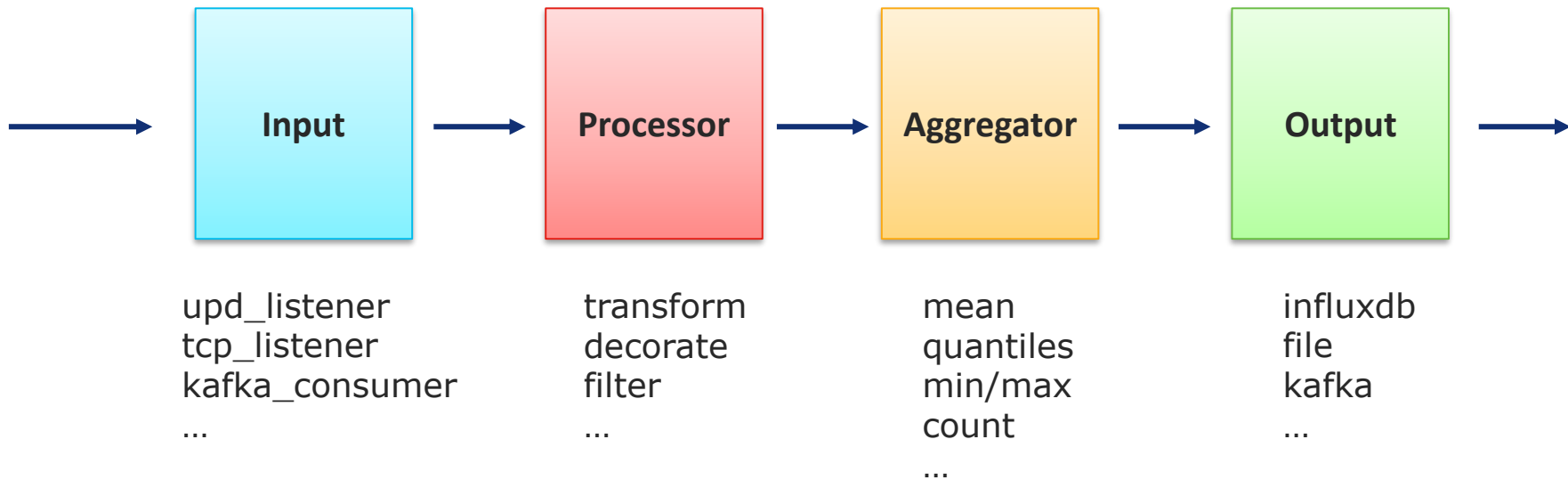


TICK スタック



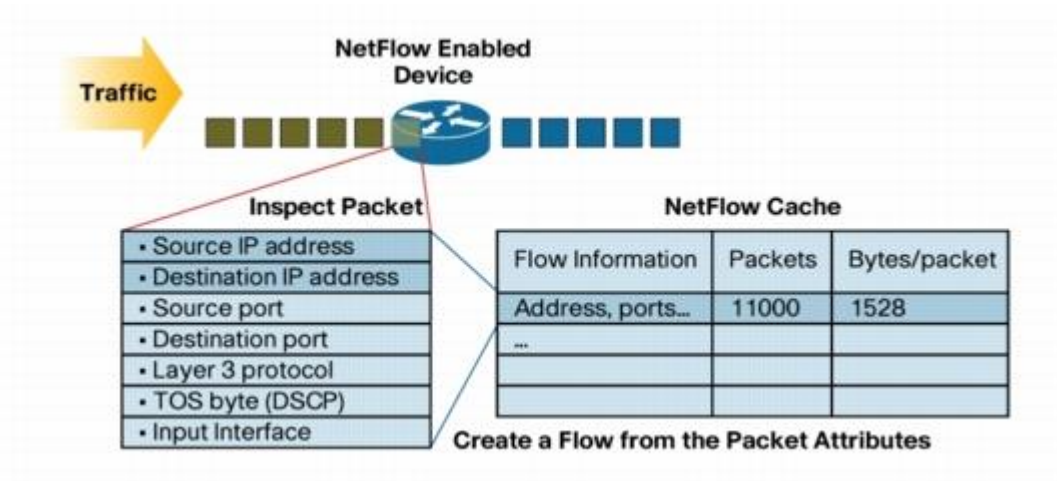
<https://www.influxdata.com/time-series-platform/>

Telegraf Pluggable アーキテクチャ



What is Netflow

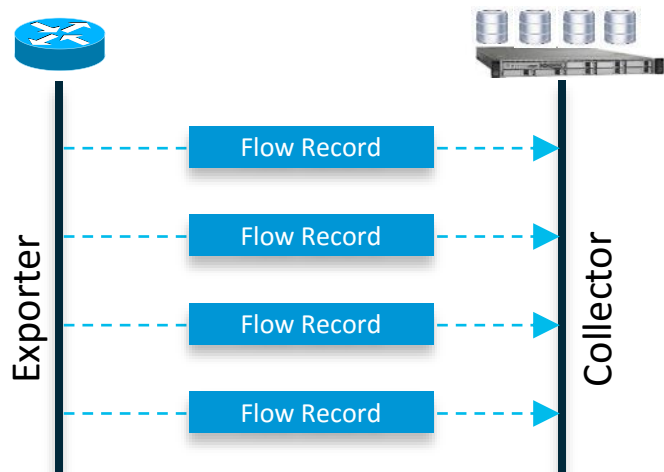
- NetFlowはネットワーク利用の可視化を提供する仕組み
- ネットワーク統計情報を収集してエクスポート
- Version 5, Version 9, IPFIX (Version 10) が主に利用される
 - **Version 5** は、固定フィールドをサポート
 - **Version 9 & IPFIX** は、テンプレートにより、柔軟かつ拡張性のあるフィールド定義が可能



Netflow Version9 / IPFIX

固定のフローフォーマット

Netflow Version 5

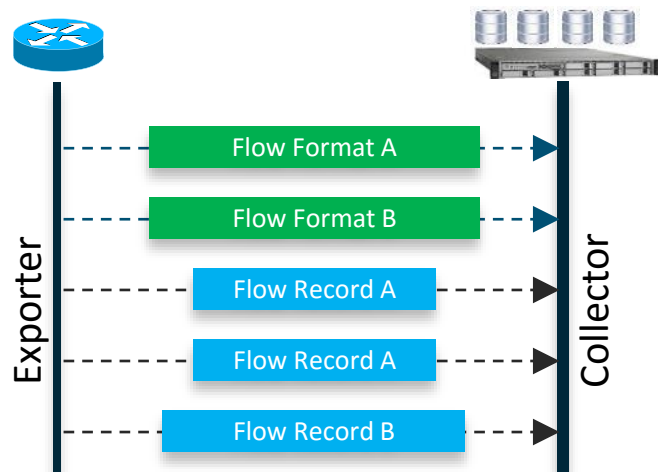


- 固定フィールド（18 フィールド）

送信元/宛先 IP & ポート番号、入力/出カインターフェイス、パケット数、バイト数、ToSなど

柔軟なフローフォーマット

Netflow v9 / IPFIX



- ユーザー定義のフローレコードフォーマット
- エクスポートは、フローレコードとは別に、テンプレート（フローレコードフォーマット）をコレクタに通知

Netflow Plugin

- TelegrafのInput Service Pluginとして実装
- 主要なNetflowバージョンをサポート
 - Version 5
 - Version 9
 - IPFIX (Version 10)

<https://github.com/tetsusat/telegraf/tree/netflow/plugins/inputs/netflow>

使い方

- 設定ファイルの生成

```
$ telegraf -sample-config -input-filter netflow -output-filter influxdb > netflow.conf
```

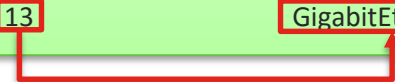
- Telegrafの起動

```
$ telegraf -config netflow.conf
2017-06-28T23:14:35Z I! Starting Telegraf v1.2.0~rc1
2017-06-28T23:14:35Z I! Loaded outputs: influxdb
2017-06-28T23:14:35Z I! Loaded inputs: inputs.netflow
2017-06-28T23:14:35Z I! Tags enabled: host=ubuntu
2017-06-28T23:14:35Z I! Agent Config: Interval:10s, Quiet:false, Hostname:"ubuntu", Flush Interval:10s
2017-06-28T23:14:35Z I! Started Netflow listener service on :2055
2017-06-28T23:14:35Z I! Netflow listening on: [::]:2055
```

ちょっと便利な機能①

- IfIndexからIfNameを動的に解決して、IfNameタグを追加
- IfIndexとIfNameの対応はOptionテンプレートから動的に学習

```
> select interface_input_snmp, interface_input_name from netflow limit 5
name: netflow
time                interface_input_snmp  interface_input_name
-----
1498692453423955383  12                   FastEthernet8
149869245342401170  13                   GigabitEthernet0
1498692453424343326  12                   FastEthernet8
1498692453424486283  12                   FastEthernet8
1498692453424631017  13                   GigabitEthernet0
```



telegrafの設定

```
[[inputs.netflow]]
  service_address = ":2055"

  resolve_ifname_by_ifindex = true
  ...
```

ルータの設定

```
flow exporter EXPORTER-V9
  destination X.X.X.X
  transport udp 2055
  template data timeout 10
  option interface-table timeout 10
```

ちょっと便利な機能②

- application_idからapplication_nameを動的に解決、application_nameタグを追加
- application_idとapplication_nameの対応は、プラグイン内部で保持

```
> select application_id, application_name from netflow limit 5
name: netflow
time                application_id  application_name
-----
1498692550621951178 218104261    ssl
1498692551617468331 16777217     icmp
1498692568533525934 218103809    unknown
1498692572501574018 50331771     ntp
1498692572501692164 50331771     ntp
```



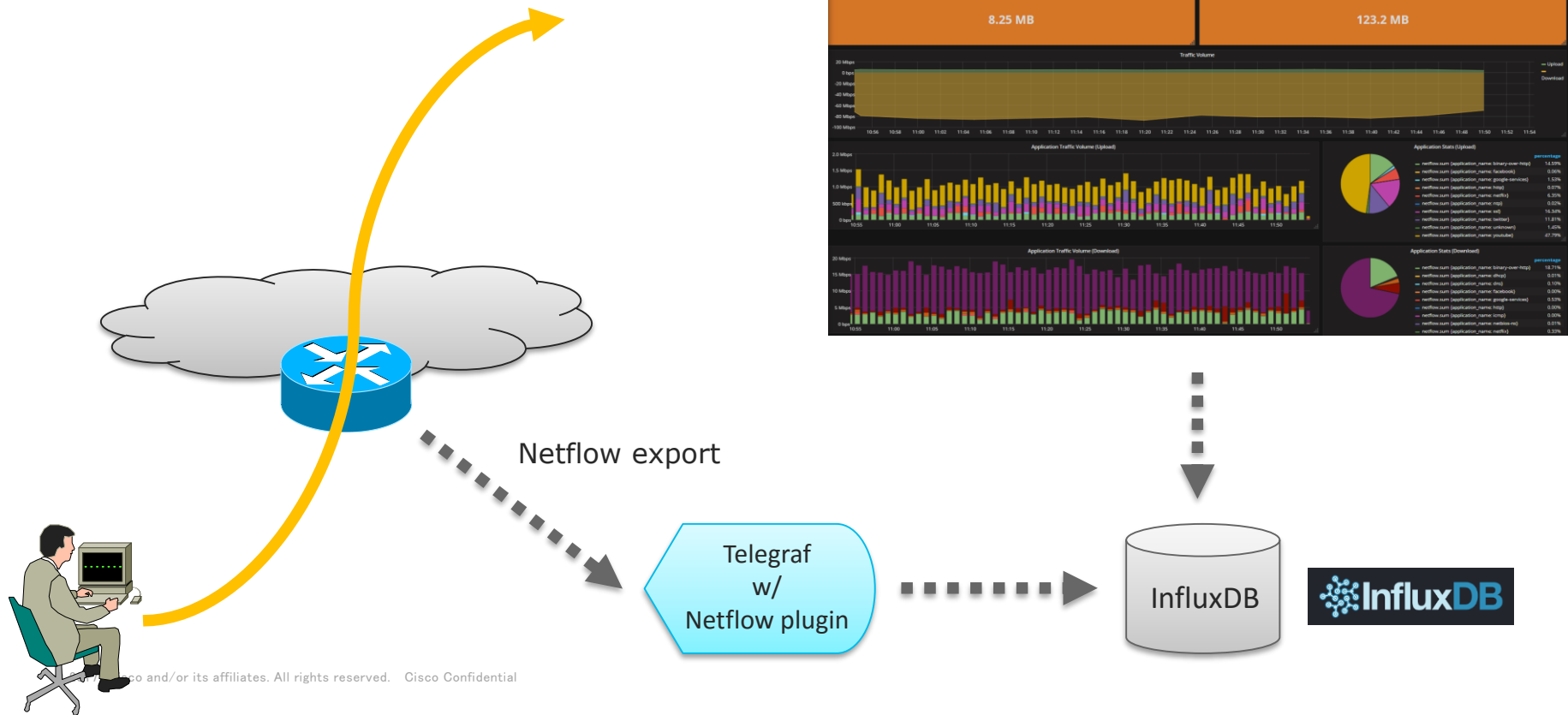
telegrafの設定

```
[[inputs.netflow]]
  service_address = ":2055"

  resolve_application_name_by_id = true

  ...
```

Telegraf Netflow Plugin デモ



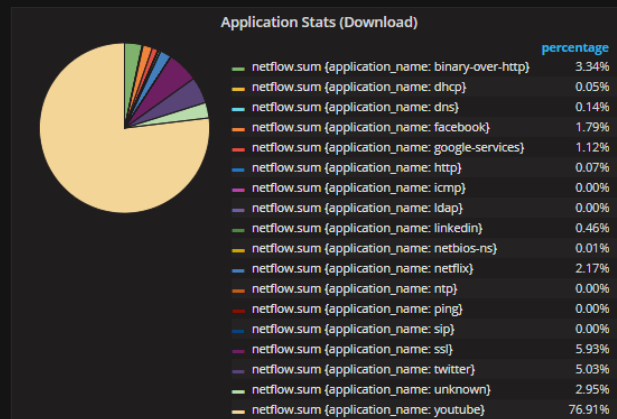
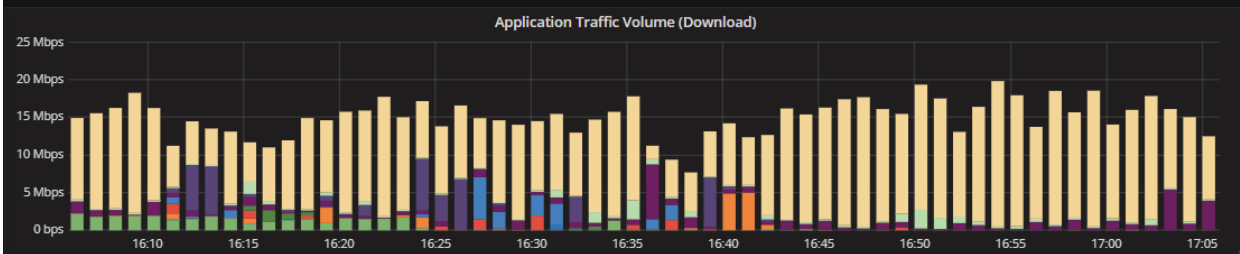
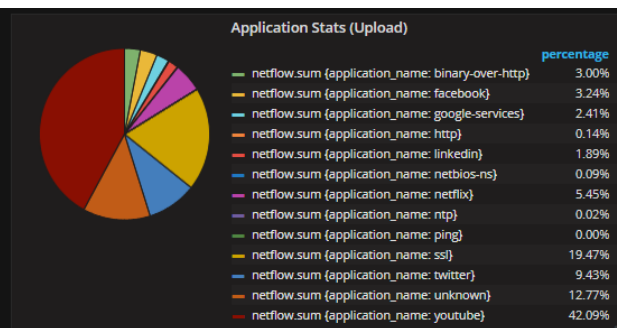
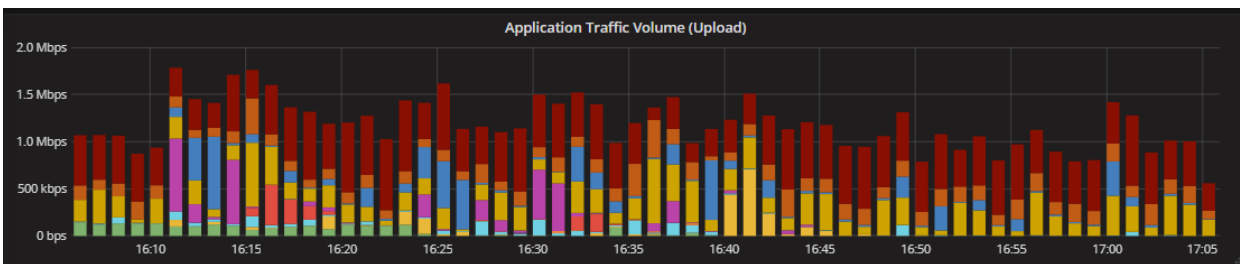
デモ① Traffic Volume

- ・ 上りと下りのトラフィック総流量を可視化



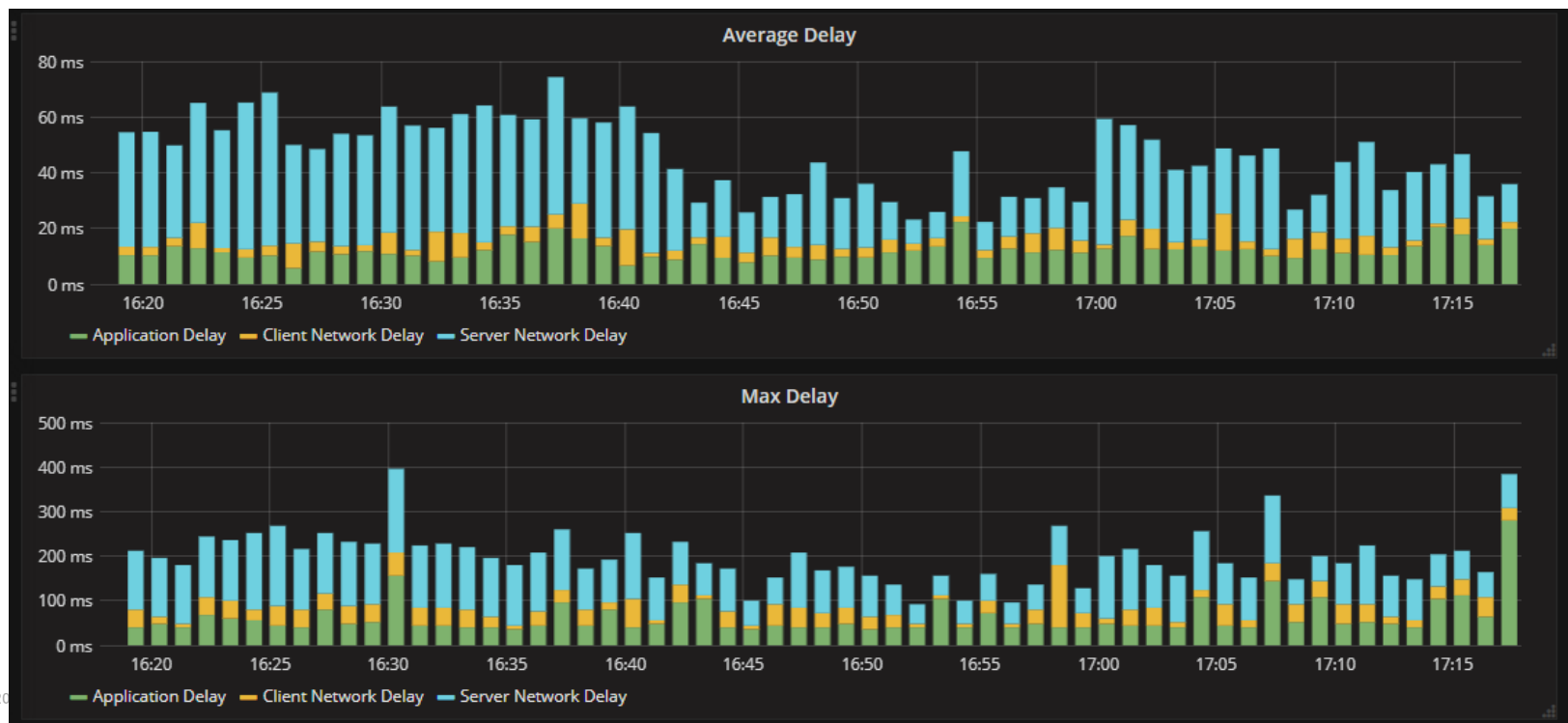
デモ② Application Metrics

- アプリケーション毎の割合を可視化



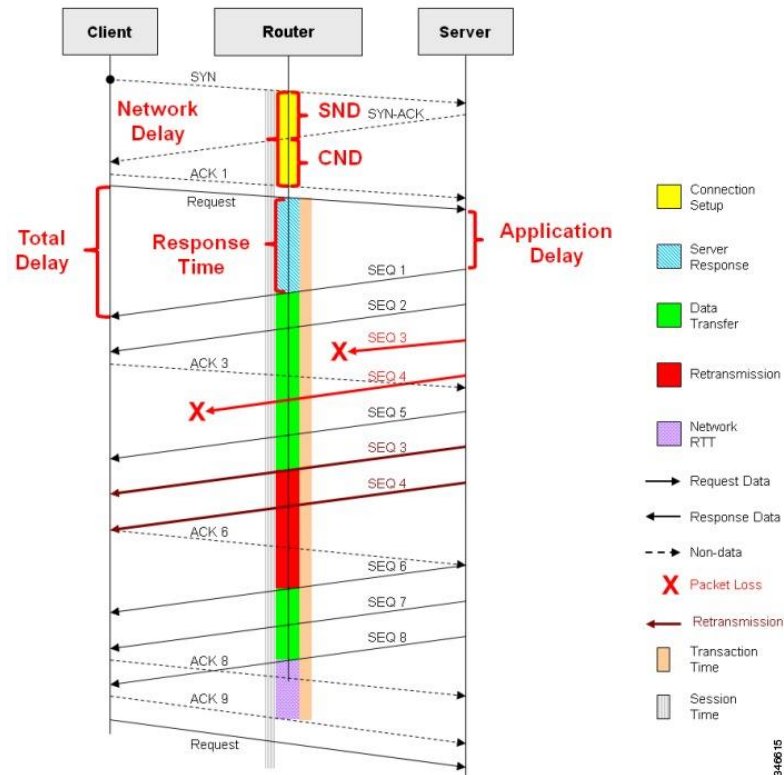
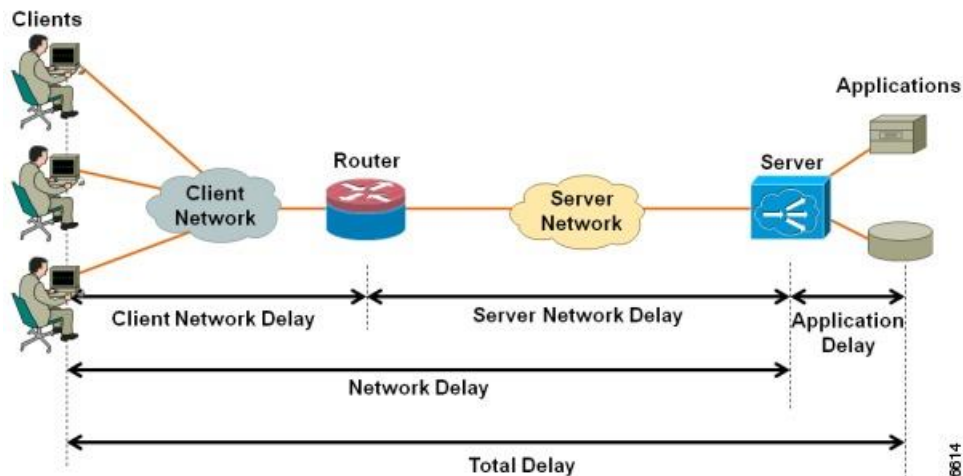
デモ③ TCP Performance Metrics

- Server Network Delayとクライアントネットワーク遅延とアプリケーション遅延を可視化
- ユーザーが体感する応答時間が悪いときに原因を絞り込む



(参考) TCP Performance Metrics

- 3-wayハンドシェークと直後のRequest-ResponseからServer Network DelayとClient Network DelayとApplication Delayを計測



(おまけ) Calculator Plugin

- TelegrafのProcessor Pluginとして実装
- 2つのフィールドを四則演算して新しいフィールドとして追加
 - "add" 足す
 - "sub" 引く
 - "mul" 掛ける
 - "div" 割る

TCP Performance Metrics関連のフィールドには遅延の平均値を直接表すものがなかったため、遅延の合計と遅延の数から算出するために作成

Application Delayの平均値を算出するTelegrafの設定

```
[[processors.calculator]]

[[processors.calculator.operations]]
  field_name = "connection_delay_application_ave"
  operator = "div"
  left = "connection_delay_application_sum"
  right = "connection_server_counter_responses"

...
```

<https://github.com/tetsusat/telegraf/tree/calculator/plugins/processors/calculator>

