

Streaming Telemetry Under the Hood: Something to Think About

Viktor Osipchuk (vosipchu@cisco.com)

05-10-2018

Do They Select Cars By Color?



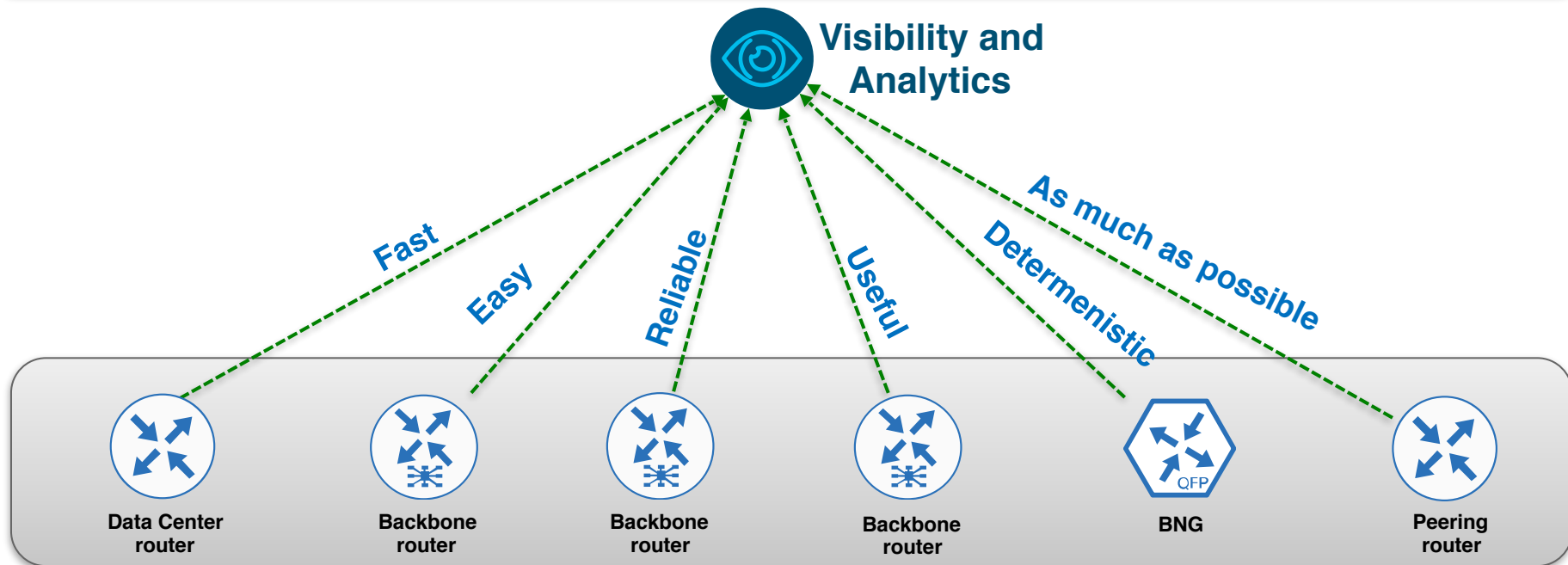
Agenda

- 1 Brief Telemetry Overview
- 2 Closer Look at Router
- 3 Closer Look at the Link Between
- 4 Closer Look at Collector
- 5 Final Thoughts

“Scream Stream If You Wanna Go Faster”

Telemetry: an automated communications process by which measurements and other data are collected at remote or inaccessible points and transmitted to receiving equipment for monitoring.

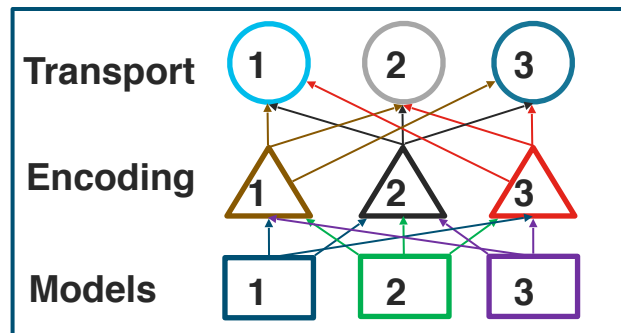
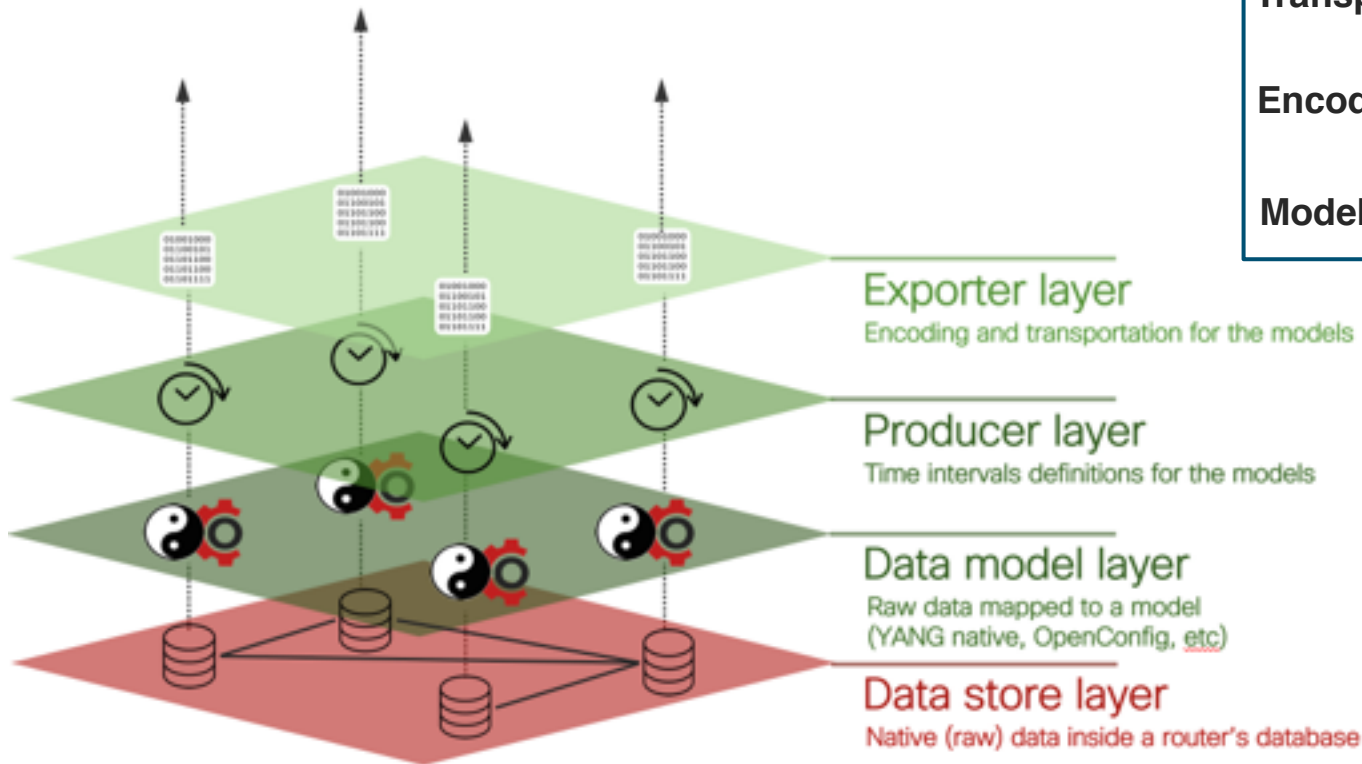
<https://en.wikipedia.org/wiki/Telemetry>



Agenda

- 1 Brief Telemetry Overview
- 2 Closer Look at Router**
- 3 Closer Look at the Link Between
- 4 Closer Look at Collector
- 5 Final Thoughts

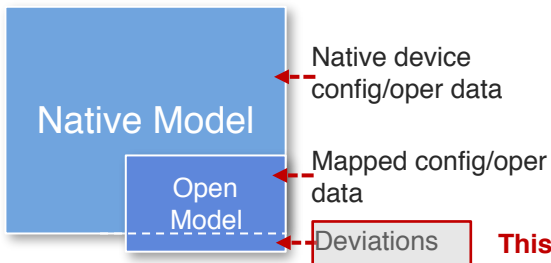
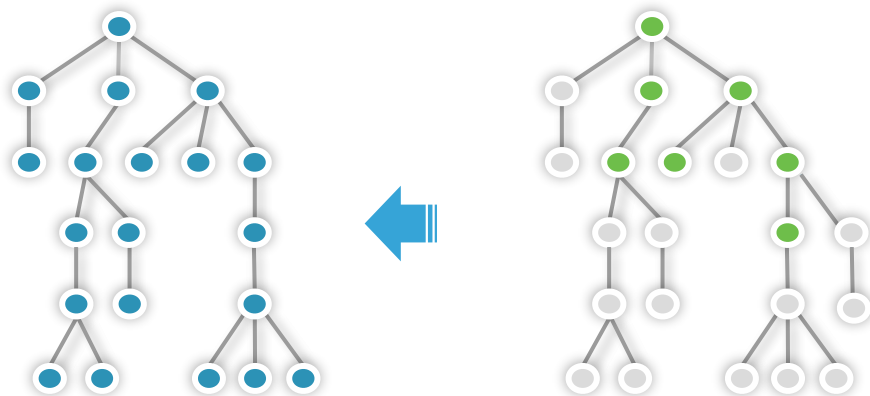
How Do You See Telemetry In a Router?



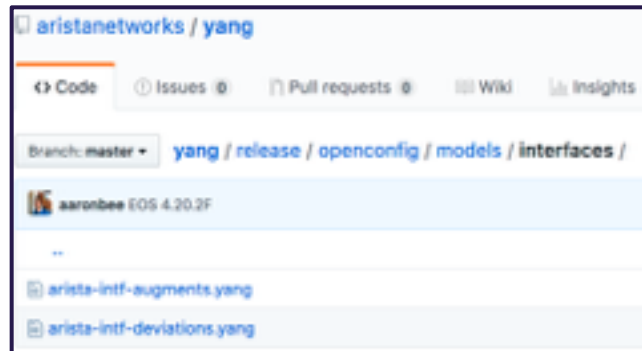
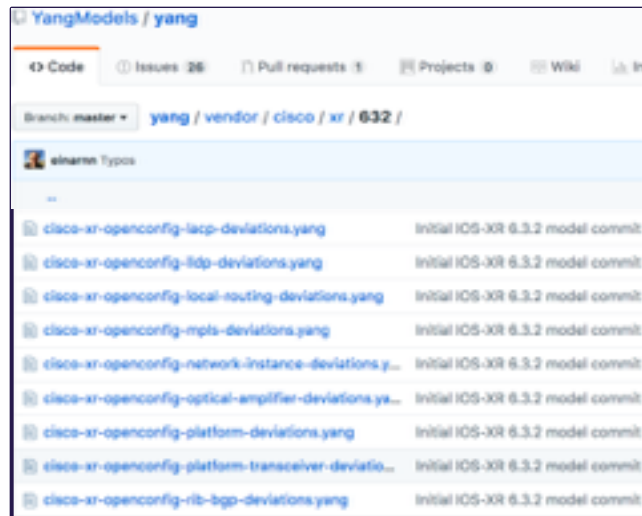
Check Your Models

Native

OpenConfig

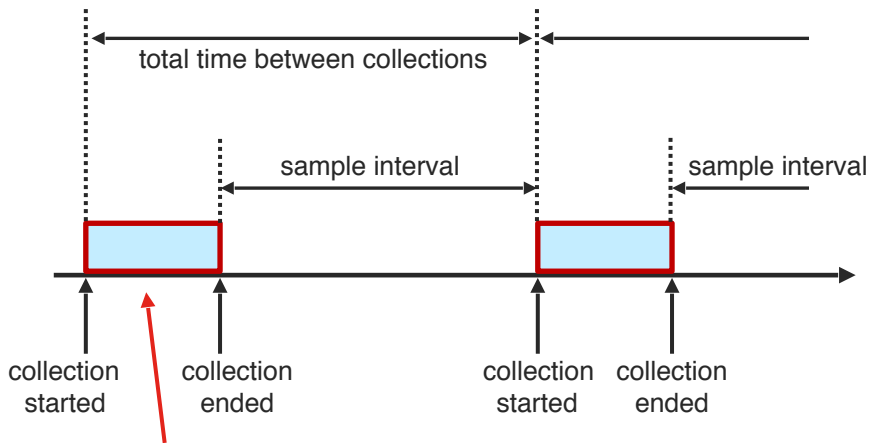


**This might not always
work for Telemetry**



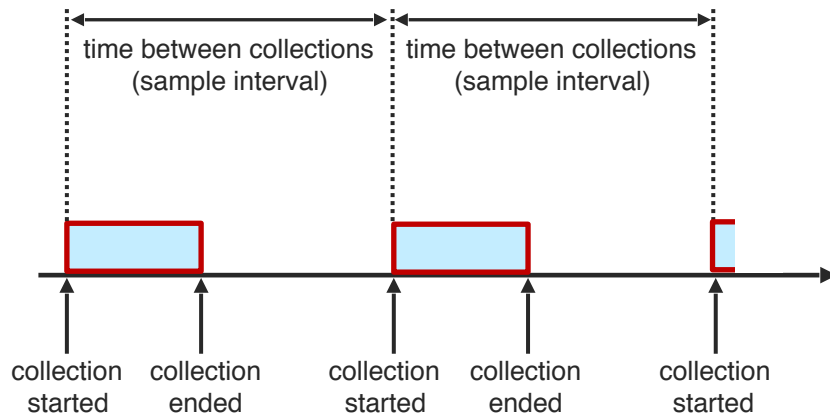
What Does Sample Interval Really Mean?

- ✓ Works fine for small collections
- ✓ You should never see missed collections
- 🤔 Hard to automate, no consistent behavior



**This might vary
in time**

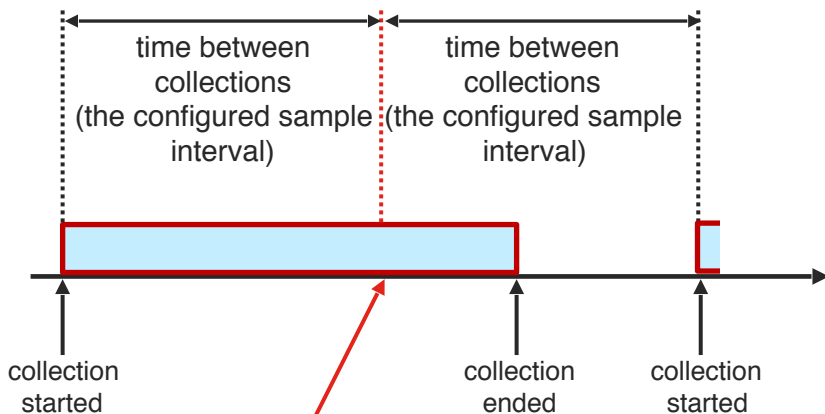
- ✓ Consistent behavior, easy to automate
- ✓ Sample interval must be more than max collection time
- 🤔 You might see missed collections



Missed Collections. What To Do?

What do you expect from your router?

- Send all 1s / all 0s?
- Send previous collection values?
- Leave things as is and don't send anything?

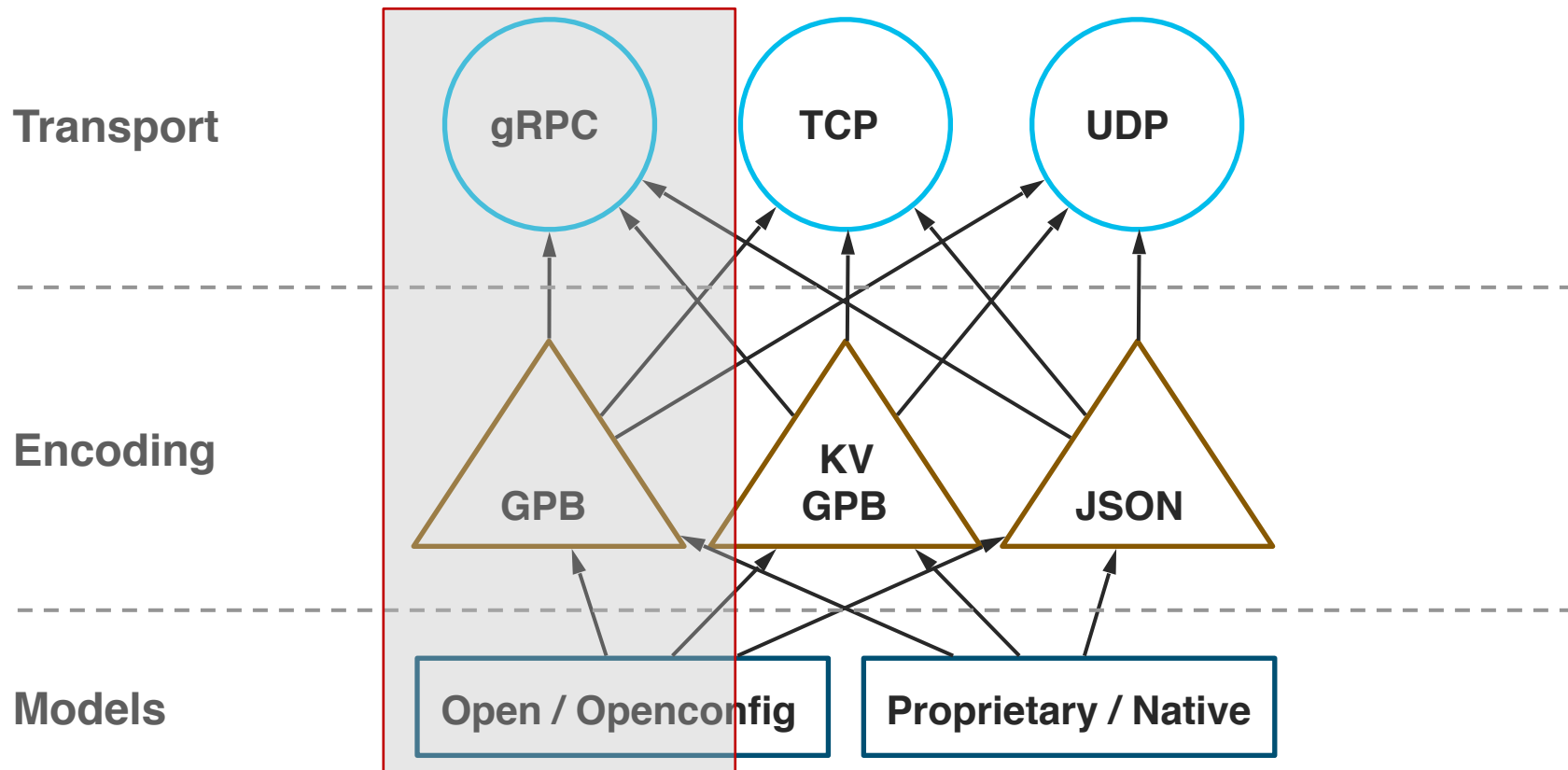


What will your router do here?

Timestamp	CPU	(Epoch converted)
1522936669977000000	9	(April 5, 2018 1:57:49.977 PM)
1522936674977000000	9	(April 5, 2018 1:57:54.977 PM)
1522936679979000000	9	(April 5, 2018 1:57:59.979 PM)
1522936684976000000	9	(April 5, 2018 1:58:04.976 PM)
1522936689978000000	9	(April 5, 2018 1:58:09.978 PM)
1522936694973000000	9	(April 5, 2018 1:58:14.973 PM)
1522936699983000000	9	(April 5, 2018 1:58:19.983 PM)
1522936704976000000	9	(April 5, 2018 1:58:24.976 PM)
1522936709976000000	9	(April 5, 2018 1:58:29.976 PM)
1522936714976000000	9	(April 5, 2018 1:58:34.976 PM)
1522936719975000000	9	(April 5, 2018 1:58:39.975 PM)
1522936724975000000	9	(April 5, 2018 1:58:44.975 PM)
1522936729982000000	9	(April 5, 2018 1:58:49.982 PM)
1522936734976000000	8	(April 5, 2018 1:58:54.976 PM)
1522936739978000000	8	(April 5, 2018 1:58:59.978 PM)
1522936749977000000	8	(April 5, 2018 1:59:09.977 PM)
1522936754975000000	8	(April 5, 2018 1:59:14.975 PM)

Do you want to fill the gap or not?

How Do You Want Your Data Out?



Is It Enough To State gRPC/GPB Support?

Define your Telemetry message

```
syntax = "proto3";
option go_package = "telemetry_bis";

/* Common Telemetry message */ // this is common for both
message Telemetry {
  oneof node_id {
    string node_id_str = 1;
  }
  oneof subscription {
    string subscription_id_str = 3;
  }
  string encoding_path = 6;
  uint64 collection_id = 8;
  uint64 collection_start_time = 9;
  uint64 msg_timestamp = 10;
  repeated TelemetryField data_gpbkv = 11;
  TelemetryGPBTable data_gpb = 12;
  uint64 collection_end_time = 13;
}
```

Define your gRPC calls

```
service gRPCConfigOper {
  // Configuration related commands
  rpc GetConfig(ConfigGetArgs) returns(stream ConfigGetReply) {};
  rpc MergeConfig(ConfigArgs) returns(ConfigReply) {};
  rpc DeleteConfig(ConfigArgs) returns(ConfigReply) {};
  rpc ReplaceConfig(ConfigArgs) returns(ConfigReply) {};
  rpc CliConfig(CliConfigArgs) returns(CliConfigReply) {};
  rpc CommitReplace(CommitReplaceArgs) returns (CommitReplaceReply) {};
  rpc CommitConfig(CommitArgs) returns(CommitReply) {};
  rpc ConfigDiscardChanges(DiscardChangesArgs) returns(DiscardChangesReply)
  {};
  // Get only returns oper data
  rpc GetOper(GetOperArgs) returns(stream GetOperReply) {};

  // Get Telemetry Data
  rpc CreateSubs(CreateSubsArgs) returns(stream CreateSubsReply) {};
}
```

Is It Enough To State gRPC/GPB Support?

Juniper's OC .proto

```
package telemetry;

// Interface exported by Agent
service OpenConfigTelemetry {
    // Request an inline subscription for data at the specified path
    // The device should send telemetry data back on the same
    // connection as the subscription request.
    rpc telemetrySubscribe(SubscriptionRequest)

    // Terminates and removes an existing telemetry subscription
    rpc cancelTelemetrySubscription(CancelSubscriptionRequest)

    // Get the list of current telemetry subscriptions from the
    // target. This command returns a list of existing subscriptions
    // not including those that are established via configuration.
    rpc getTelemetrySubscriptions(GetSubscriptionsRequest)

    // Get Telemetry Agent Operational States
    rpc getTelemetryOperationalState(GetOperationalStateRequest)

    // Return the set of data encodings supported by the device for
    // telemetry data
    rpc getDataEncodings(DataEncodingRequest)
}
```

<https://github.com/nileshsimaria/jimon/blob/master/telemetry/telemetry.proto>

Juniper's UDP .proto

```
import "pbj.proto";
import "google/protobuf/descriptor.proto";

extend google.protobuf.FieldOptions {
    optional TelemetryFieldOptions telemetry_options = 1024;
}

message TelemetryFieldOptions {
    optional bool is_key = 1;
    optional bool is_timestamp = 2;
    optional bool is_counter = 3;
    optional bool is_gauge = 4;
}

message TelemetryStream {
    // router name or export IP address
    required string system_id = 1 [([telemetry_options].is_key = true,
    {pbj_field_option}.type = FT_POINTER)];

    // line card / RE (slot number)
    optional uint32 component_id = 2 [([telemetry_options].is_key = true)];

    // PFE (if applicable)
    optional uint32 sub_component_id = 3 [([telemetry_options].is_key = true)];

    // configured sensor name
    optional string sensor_name = 4 [([telemetry_options].is_key = true,
    {pbj_field_option}.type = FT_POINTER)];
}
```

https://github.com/nileshsimaria/jmonudp/blob/master/protos/telemetry_top/telemetry_top.proto

GNMI Should Be The Answer. Right?


Defines the gRPC call

```
service gNMI {  
    // Capabilities allows the client to retrieve the set of capabilities that  
    // is supported by the target. This allows the target to validate the  
    // service version that is implemented and retrieve the set of models that  
    // the target supports. The models can then be specified in subsequent RPCs  
    // to restrict the set of data that is utilized.  
    // Reference: gNMI Specification Section 3.2  
    rpc Capabilities(CapabilityRequest) returns (CapabilityResponse);  
    // Retrieve a snapshot of data from the target. A Get RPC requests that the  
    // target snapshots a subset of the data tree as specified by the paths  
    // included in the message and serializes this to be returned to the  
    // client using the specified encoding.  
    // Reference: gNMI Specification Section 3.3  
    rpc Get(GetRequest) returns (GetResponse);  
    // Set allows the client to modify the state of data on the target. The  
    // paths to modified along with the new values that the client wishes  
    // to set the value to.  
    // Reference: gNMI Specification Section 3.4  
    rpc Set(SetRequest) returns (SetResponse);  
    // Subscribe allows a client to request the target to send it values  
    // of particular paths within the data tree. These values may be streamed  
    // at a particular cadence (STREAM), sent one off on a long-lived channel  
    // (POLL), or sent as a one-off retrieval (ONCE).  
    // Reference: gNMI Specification Section 3.5  
    rpc Subscribe(stream SubscribeRequest) returns (stream SubscribeResponse);  
}
```

Defines the message

```
message Update {  
    Path path = 1;           // The path (key) for the update.  
    Value value = 2 [deprecated=true]; // The value (value) for the update.  
    TypedValue val = 3;      // The explicitly typed update value.  
    uint32 duplicates = 4;   // Number of coalesced duplicates.  
}  
  
// TypedValue is used to encode a value being sent between the client and  
// target (originated by either entity).  
message TypedValue {  
    // One of the fields within the val oneof is populated with the value  
    // of the update. The type of the value being included in the Update  
    // determines which field should be populated. In the case that the  
    // encoding is a particular form of the base protobuf type, a specific  
    // field is used to store the value (e.g., json_val).  
    oneof value {  
        string string_val = 1;           // String value.  
        int64 int_val = 2;               // Integer value.  
        uint64 uint_val = 3;            // Unsigned integer value.  
        bool bool_val = 4;              // Bool value.  
        bytes bytes_val = 5;            // Arbitrary byte sequence value.  
        float float_val = 6;            // Floating point value.  
        Decimal64 decimal_val = 7;      // Decimal64 encoded value.  
        ScalarArray leaflist_val = 8;   // Mixed type scalar array value.  
        google.protobuf.Any any_val = 9; // protobuf.Any encoded bytes.  
        bytes json_val = 10;            // JSON-encoded text.  
        bytes json_ietf_val = 11;       // JSON-encoded text per RFC7951.  
        string ascii_val = 12;          // Arbitrary ASCII text.
```

What to select
here?



Agenda

- 1 Brief Telemetry Overview
- 2 Closer Look at Router
- 3 Closer Look at the Link Between**
- 4 Closer Look at Collector
- 5 Final Thoughts

How Can a Router Send Its Data Out?



gRPC



TCP



UDP

gRPC Comes With an Overhead...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.30.110.41	10.30.110.40	TCP	74	64652 → 57500 [SYN] Seq=0 Win=...
2	0.000093450	10.30.110.40	10.30.110.41	TCP	74	57500 → 64652 [SYN, ACK] Seq=0...
3	0.000514062	10.30.110.41	10.30.110.40	TCP	66	64652 → 57500 [ACK] Seq=1 Ack=...
4	0.000603066	10.30.110.41	10.30.110.40	HTTP2	90	Magic
5	0.000628645	10.30.110.40	10.30.110.41	TCP	66	57500 → 64652 [ACK] Seq=1 Ack=...
6	0.000639815	10.30.110.41	10.30.110.40	HTTP2	75	SETTINGS
7	0.000652105	10.30.110.40	10.30.110.41	TCP	66	57500 → 64652 [ACK] Seq=1 Ack=...
8	0.000660746	10.30.110.41	10.30.110.40	HTTP2	79	WINDOW_UPDATE
9	0.000670352	10.30.110.40	10.30.110.41	TCP	66	57500 → 64652 [ACK] Seq=1 Ack=...
10	0.000686445	10.30.110.41	10.30.110.40	HTTP2	152	HEADERS
11	0.000694705	10.30.110.40	10.30.110.41	TCP	66	57500 → 64652 [ACK] Seq=1 Ack=...
12	0.000810032	10.30.110.40	10.30.110.41	HTTP2	75	SETTINGS
13	0.000839324	10.30.110.40	10.30.110.41	HTTP2	79	WINDOW_UPDATE
14	0.001087631	10.30.110.41	10.30.110.40	TCP	66	64652 → 57500 [ACK] Seq=133 Ac...
15	0.001118306	10.30.110.40	10.30.110.41	HTTP2	75	SETTINGS
16	0.001134535	10.30.110.41	10.30.110.40	TCP	66	64652 → 57500 [ACK] Seq=133 Ac...
17	0.001409008	10.30.110.41	10.30.110.40	HTTP2	75	SETTINGS
18	0.011716945	10.30.110.41	10.30.110.40	HTTP2	250	DATA
19	0.011798380	10.30.110.40	10.30.110.41	TCP	66	57500 → 64652 [ACK] Seq=32 Ack=...
20	24.000822908	10.30.110.41	10.30.110.40	TCP	66	64652 → 57500 [FIN, ACK] Seq=3...
21	24.001077163	10.30.110.40	10.30.110.41	TCP	66	57500 → 64652 [FIN, ACK] Seq=3...
22	24.001617400	10.30.110.41	10.30.110.40	TCP	66	64652 → 57500 [ACK] Seq=336 Ac...

0040	62 41 00 00 b8 00 00 00 00 00 01 00 00 00 00 b3	bA.....
0050	08 ff 05 12 a0 01 0a 0b 4e 43 53 35 35 30 31 5f NC55001
0060	74 6f 70 1a 04 74 65 73 74 32 2e 43 69 73 63 6f	top..tes t2.Cisco
0070	2d 49 4f 53 2d 58 52 2d 73 68 65 6c 6c 75 74 69	-IOS-XR- shelluti
0080	6c 2d 6f 70 65 72 3a 73 79 73 74 65 6d 2d 74 69	l-oper:s system-tl
0090	6d 65 2f 75 70 74 69 6d 65 3a 0a 32 30 31 35 2d	me/uptime: 2015-

Magic number to start
HTTP2 phase

Settings from
the router

Window size from
the router

HTTP2 details

Window size/settings
from the collector

Settings
confirmations

**Negotiation
happens
before
the data
is streamed**

...But Brings Some Good Benefits

No.	Time	Source	Destination	Protocol	Length	Info
4.	16.020449012	10.30.110.40	10.30.110.41	HTTP2	79	WINDOW_UPDATE
4.	16.020453129	10.30.110.41	10.30.110.40	HTTP2	16459	DATA
4.	16.020465328	10.30.110.40	10.30.110.41	HTTP2	79	WINDOW_UPDATE
4.	16.020469649	10.30.110.41	10.30.110.40	HTTP2	16459	DATA
4.	16.020476567	10.30.110.41	10.30.110.40	TCP	1514	64969 → 57500 [ACK] Seq=768790 Ack=534 W..
4.	16.020620532	10.30.110.40	10.30.110.41	TCP	66	57500 → 64969 [ACK] Seq=534 Ack=770238 W..
4.	16.021387959	10.30.110.41	10.30.110.40	TCP	66	64945 → 57500 [ACK] Seq=1179628 Ack=612 ..
4.	16.021406127	10.30.110.41	10.30.110.40	TCP	4410	64969 → 57500 [ACK] Seq=770238 Ack=534 W..
4.	16.021410692	10.30.110.41	10.30.110.40	HTTP2	16459	DATA
4.	16.021412266	10.30.110.40	10.30.110.41	TCP	66	57500 → 64969 [ACK] Seq=534 Ack=774582 W..
4.	16.021419367	10.30.110.41	10.30.110.40	HTTP2	10293	DATA
4.	16.021421080	10.30.110.41	10.30.110.40	HTTP2	527	DATA
4.	16.021424808	10.30.110.41	10.30.110.40	HTTP2	237	DATA
4.	16.021430133	10.30.110.41	10.30.110.40	HTTP2	679	DATA
4.	16.021472771	10.30.110.40	10.30.110.41	HTTP2	79	WINDOW_UPDATE
4.	16.021480055	10.30.110.41	10.30.110.40	HTTP2	1048	DATA
4.	16.021509967	10.30.110.41	10.30.110.40	HTTP2	676	DATA
4.	16.021513831	10.30.110.40	10.30.110.41	TCP	66	57500 → 64945 [ACK] Seq=612 Ack=1198248 ..
4.	16.021583417	10.30.110.41	10.30.110.40	HTTP2	670	DATA
4.	16.021604650	10.30.110.40	10.30.110.41	TCP	66	57500 → 64945 [ACK] Seq=612 Ack=1199462 ..
4.	16.021708748	10.30.110.40	10.30.110.41	HTTP2	79	WINDOW_UPDATE
4.	16.021836970	10.30.110.41	10.30.110.40	TCP	1514	64969 → 57500 [ACK] Seq=784809 Ack=547 W..
4.	16.021871368	10.30.110.41	10.30.110.40	HTTP2	1223	DATA

Flags: 0x00	
0...	= Reserved: 0x0
.000 0000 0000 0000 0000 0000 0001	= Stream Identifier: 1
0...	= Reserved: 0x0
.000 0000 0000 0000 0110 1001 1110 0101	= Window Size Increment: 27109

Speed Control
(from the
collector side)

TCP And UDP Are Simple

No.	Time	Source	Destination	Protocol	Leng	Info
1	0.000000000	10.30.110.41	10.30.110.40	TCP	62	18577 → 5432 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 ..
2	0.000000063	10.30.110.40	10.30.110.41	TCP	62	5432 → 18577 [SYN, ACK] Seq=0 Ack=1 Win=20200 Len=...
3	0.000696242	10.30.110.41	10.30.110.40	TCP	60	18577 → 5432 [ACK] Seq=1 Ack=1 Win=16384 Len=0
4	0.010749238	10.30.110.41	10.30.110.40	TCP	239	18577 → 5432 [PSH, ACK] Seq=1 Ack=1 Win=16384 Len=...
5	0.010788711	10.30.110.40	10.30.110.41	TCP	54	5432 → 18577 [ACK] Seq=1 Ack=186 Win=30336 Len=0
6	9.072228347	10.30.110.41	10.30.110.40	TCP	60	18577 → 5432 [FIN, ACK] Seq=186 Ack=1 Win=16384 Le...
7	9.072517389	10.30.110.40	10.30.110.41	TCP	54	5432 → 18577 [FIN, ACK] Seq=1 Ack=187 Win=30336 Le...
8	9.073190068	10.30.110.41	10.30.110.40	TCP	60	18577 → 5432 [ACK] Seq=187 Ack=2 Win=16384 Len=0

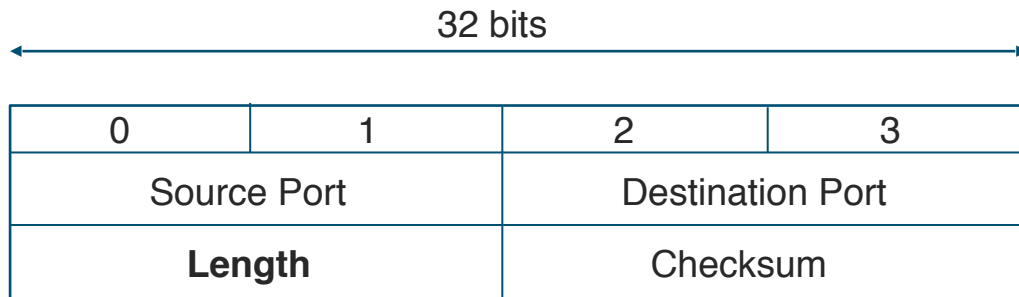
0030	40 00 b7 e4 00 00 00 01 00 01 00 01 00 00 00 00	0.....
0040	00 ad 0a 0b 4e 43 53 35 35 30 31 5f 74 6f 70 1aNCS5501_top.
0050	04 74 65 73 74 32 2e 43 69 73 63 6f 2d 49 4f 53	..test2.Cisco IOS
0060	2d 58 52 2d 73 68 65 6c 6c 75 74 69 6c 2d 6f 70	..XR-shell util-op
0070	65 72 3a 73 79 73 74 65 6d 2d 74 69 6d 65 2f 75	er:system-time/ut
0080	70 74 69 6d 65 3a 0a 32 30 31 35 2d 30 31 2d 30	ptime: 2015-01-0
0090	37 40 b5 a8 cd 09 48 ff 83 80 cb a9 2c 50 ff 83	70....H.....P..

Good to know if
there is any
additional header
inside

No.	Time	Source	Destination	Protocol	Leng	Info
1	0.000000000	10.30.110.41	10.30.110.40	UDP	227	26539 → 18765 Len=103

0020	0e 28 67 ab 49 4d 00 c1 1a 66 00 01 00 01 00 01	m(g.1M...f.....
0030	00 00 00 00 00 ad 0a 0b 4e 43 53 35 35 30 31 5fNCS5501_
0040	74 6f 70 1a 04 74 65 73 74 32 2e 43 69 73 63 6f	top..tes t2.Cisco
0050	2d 49 4f 53 2d 58 52 2d 73 68 65 6c 6c 75 74 69	..IOS-XR- shelluti
0060	6c 2d 6f 70 65 72 3a 73 79 73 74 65 6d 2d 74 69	l-oper:system-ti
0070	6d 65 2f 75 70 74 69 6d 65 3a 0a 32 30 31 35 2d	me/uptime: 2015-
0080	30 31 2d 30 37 40 f8 ea cd 09 48 d8 f3 ac cb a9	01-070....H.....
0090	2c 50 d8 f3 ac cb a9 2c 5a 42 08 df f3 ac cb a9	..P.....,ZB.....

UDP Is Fast, But...



Max datagram length is:

$2^{16} - 20 \text{ (IPH)} - 8 \text{ (UDPH)} = \mathbf{65,507}$ bytes

Sep 19 01:35:01.452 m2m/mdt/backend-timer 0/RP0/CPU0 t15234 45924 [mdtbk_bte_encode_cb]: sub_id 5, /oper/optics/
if/*/optics_info, **len 77580: mdt_send_encoded_data returned error**

To Encrypt or Not To Encrypt?

gRPC Dial-in (NO-TLS)

Password exchange

```

No.    Time    Source                Destination            Protocol Length Info
  10 0.0014569542 30.30.130.43         10.30.150.40          79 WINDOW_UPDATE
  11 0.0014570061 30.30.130.43         10.30.150.41          TCP        60 43738 - 57500 [ACK] Seq=47 Ack=29 Win=29...
  12 0.0014705400 30.30.130.43         10.30.150.40          HTTP/2     75 SETTING
  13 0.0014854803 30.30.130.43         10.30.150.41          TCP        60 43738 - 57500 [ACK] Seq=47 Ack=32 Win=29...
  14 0.0015106032 30.30.130.43         10.30.150.41          HTTP/2     75 SETTING
  15 0.0015200009 30.30.130.43         10.30.150.41          HTTP/2    220 HEADERS, DATA
  16 0.0017124511 30.30.130.43         10.30.150.41          HTTP/2     75 DATA
  17 0.0017339921 30.30.130.43         10.30.150.40          TCP        60 57500 - 43738
  18 0.0019060780 30.30.130.43         10.30.150.41          HTTP/2    220 HEADERS, DATA
  19 0.0019394693 30.30.130.43         10.30.150.41          TCP        60 43738 - 57500 [ACK] Seq=227 Ack=247 Win=...

Header Block Fragment: 830644ee6326ad079670794064e1a308a075b4c5d66abc9...
[Header Length: 267]
[Header Count: 9]
# Header: :method: POST
# Header: :scheme: http
# Header: :path: /v300xxExtensibleManagementService.gRPCConf30oper/CreateSubs
# Header: :authority: 30.30.130.43
# Header: content-type: application/grpc
# Header: user-agent: grpc-go/1.0
# Header: x-protocol: grpc
# Header: password: cisco
# Header: username: cisco
# Header: x-protocol: grpc
# Stream: DATA, Stream ID: 1, Length 23
Length: 23

```

Message content

```
%=LMed.....
.NCS5501_top..test2.Cisco-IOS-XR-shellutil-oper:system-time/uptime:
2015-01-07@...
H.....P.....ZB.....,z...keysz1..contentz...hostname*.NCS5501_topz
..uptime8.....h.....
```

gRPC Dial-in (TLS)

Password exchange

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.30.130.40	10.30.130.41 TCP	74	44522 → 57500	SYN Seq=0 Win=0 Len=0
2	0.000034247	10.30.130.41	10.30.130.40 TCP	74	57500 → 44522	ACK Seq=0 Ack=1 Win=0
3	0.000044292	10.30.130.40	10.30.130.41 TCP	66	44522 → 57500	ACK Seq=1 Ack=1 Win=2021
4	0.000170814	10.30.130.40	10.30.130.41 TCP	239	44522 → 57500	PSH, ACK Seq=0 Ack=1 Win=0
5	0.000170942	10.30.130.40	10.30.130.41 TCP	66	57500 → 44522	ACK Seq=1 Seq=174 Win=0
6	0.0001591306	10.30.130.41	10.30.130.40 TCP	129	57500 → 44522	ACK Seq=0 Ack=174 W=0
7	0.000008875	10.30.130.40	10.30.130.41 TCP	66	44522 → 57500	ACK Seq=174 Ack=0 Win=0
8	0.000046298	10.30.130.41	10.30.130.40 TCP	1358	57500 → 44522	PSH, ACK Seq=0 Ack=174 W=0
9	0.0000462963	10.30.130.40	10.30.130.41 TCP	66	44522 → 57500	ACK Seq=174 Ack=1358 Win=0
10	0.0001705258	10.30.130.40	10.30.130.41 TCP	404	57500 → 44522	PSH, ACK Seq=1503 Ack=17
11	0.0001705756	10.30.130.41	10.30.130.40 TCP	66	44522 → 57500	ACK Seq=1503 Ack=17
12	0.0007042303	10.30.130.41	10.30.130.40 TCP	75	57500 → 44522	PSH, ACK Seq=1503 Ack=17
13	0.0007054496	10.30.130.40	10.30.130.41 TCP	66	44522 → 57500	ACK Seq=174 Ack=1503 Win=0
14	0.0349205133	10.30.130.40	10.30.130.41 TCP	382	44522 → 57500	PSH, ACK Seq=174 Ack=150
15	0.0349505380	10.30.130.41	10.30.130.40 TCP	72	57500 → 44522	PSH, ACK Seq=1503 Ack=17
16	0.0350045641	10.30.130.40	10.30.130.41 TCP	113	57500 → 44522	ACK Seq=1503 Ack=17
17	0.0350871290	10.30.130.41	10.30.130.40 TCP	394	57500 → 44522	PSH, ACK Seq=1504 Ack=30
18	0.0350908005	10.30.130.41	10.30.130.40 TCP	388	57500 → 44522	PSH, ACK Seq=1503 Ack=30
19	0.0350959593	10.30.130.40	10.30.130.41 TCP	66	44522 → 57500	ACK Seq=300 Ack=1034 Win=0

Frame 25: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits) on interface 0

Ethernet II, Src: VMware_A7:20:2A (08:00:56:A7:20:2A), Dest: Cisco_Ba:8a:ad:00 (00:0c:2b:ba:8a:ad:00)

Internet Protocol Version 4, Src: 10.30.130.40, Dest: 10.30.130.41

Message content

[illegible]

Which Encoding To Use?

GPB

Message length: 330 bytes

74	66	12	84	62	85	78	46	43	53	35	33	31
74	67	12	84	74	85	78	74	32	52	43	35	33
72	49	47	53	28	58	52	69	66	66	62	61	73
61	74	61	74	69	73	74	69	34	69	66	62	61
74	74	61	74	69	73	74	69	34	69	66	62	61
66	61	63	65	73	21	69	65	62	72	66	61	63
66	61	74	65	73	21	69	65	62	72	66	61	63
66	61	75	66	74	75	72	73	3a	00	32	38	31
28	38	39	48	68	68	68	68	9a	9b	9a	9b	28
9a	9b	9a	9b	2c	62	68	9a	9b	9a	2c	62	68
8a	8b	81	88	84	9a	9b	8a	8b	81	88	84	9a
64	72	64	72	64	67	69	67	52	52	14	21	28
5a	8b	81	98	83	71	81	87	19	98	83	85	8b
8b	8b	83	72	8b	08	18	8b	07	8b	83	81	8b
81	39	58	83	87	94	8b	83	8b	83	84	8b	86
83	83	83	87	87	8b	8b	83	8b	83	84	8b	86
83	83	83	87	87	8b	8b	83	8b	83	84	8b	86
84	83	83	87	87	8b	8b	84	8b	84	8b	84	8b
84	83	84	8b	8b	84	8b	84	8b	84	8b	84	8b
78	84	78	84	85	8b	8b	8b	85	8b	7c	9a	85
78	85	8b	85	72	8b	8b	8b	85	8b	85	85	85

KV-GPB

Message length: 1142 bytes

```

.....NCS00#
top...tes12/Cisco
-105--resinfra-st
atad-ope r:infra
statisti cs/inter
faces/in terface/
latest/g emeric-
ounters:-2015-11
-@kg....H.....P
.....n.....Bu
ndredsig...R...Mu
Z.....80/6/1/0
9.....
.....
.....
.....
.....

```

JSON

Message length: 1325 bytes

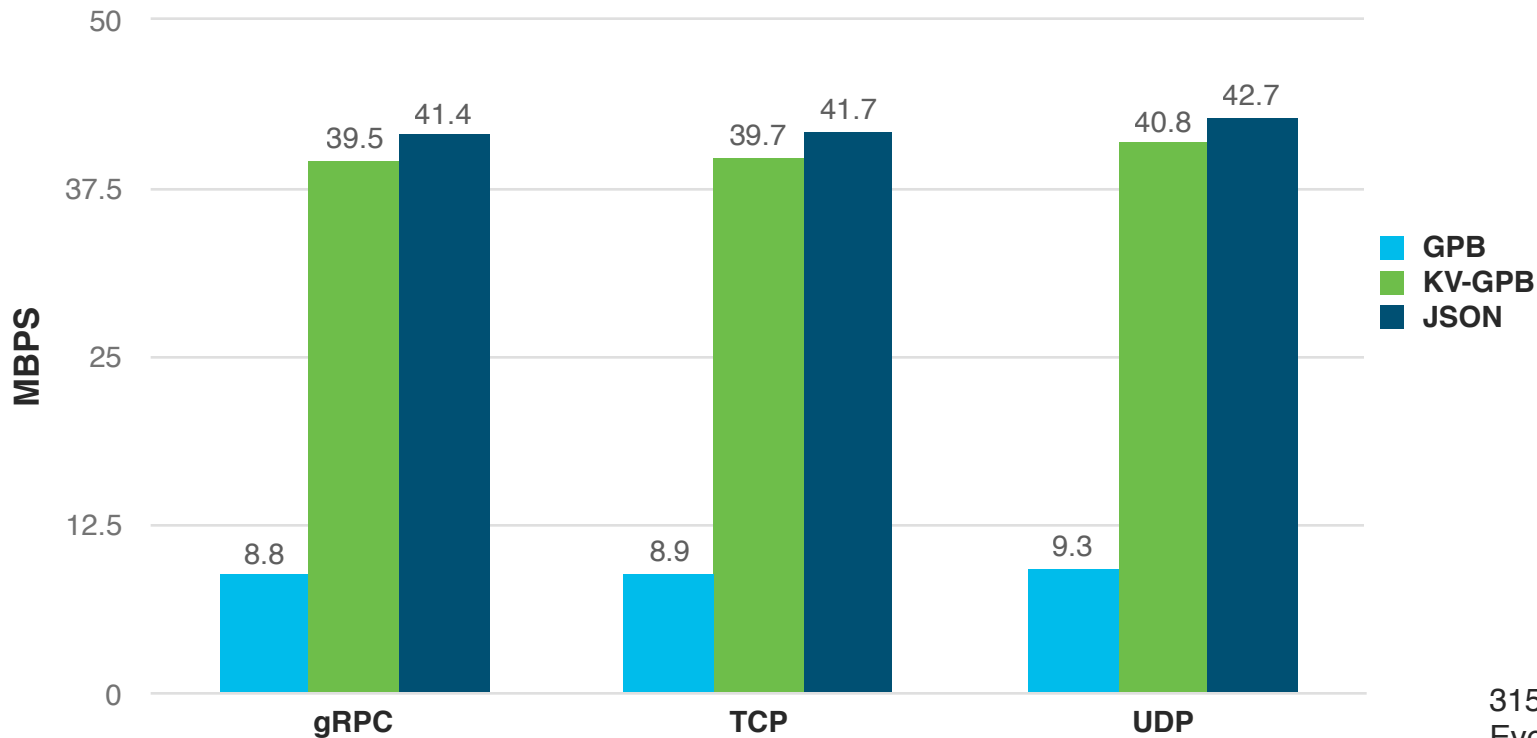
74	61	73	86	12	78	88	84	86	74	43	32	35	35	38	31
74	61	73	78	18	74	74	75	73	72	42	35	43	37	33	63
28	49	47	53	28	38	52	29	69	66	66	61	20	73	63	63
63	74	61	74	60	73	67	78	65	72	38	66	66	72	61	26
74	73	61	74	60	73	74	69	63	73	21	68	67	74	65	73
66	61	63	65	73	21	69	66	65	62	72	66	61	63	65	25
6c	61	74	65	73	74	21	67	65	65	65	72	69	63	26	63
67	75	76	74	62	72	73	38	82	32	38	31	35	26	31	31
28	38	39	48	63	88	88	89	87	86	69	67	67	67	2c	2c
76	59	67	61	62	25	3a	62	87	88	82	82	82	82	82	82
7a	22	12	84	60	65	79	73	7a	24	12	86	69	73	73	73
72	66	61	61	65	29	66	61	72	66	61	61	21	75	66	66
63	63	63	63	63	63	63	63	63	63	63	63	63	63	63	63
67	12	87	63	73	66	74	65	64	7a	17	12	78	78	78	78
63	60	65	63	73	28	72	65	63	65	69	63	65	64	68	65
93	67	17	75	17	12	82	62	79	74	65	63	73	26	72	65
65	69	76	63	64	64	64	64	c2	6b	64	65	74	15	12	66
78	61	63	66	75	73	73	26	73	65	66	74	48	97	ad	6c
67	68	87	74	14	12	8a	62	74	74	65	73	28	73	65	65
74	68	63	c2	6e	58	65	64	61	39	7a	21	12	1a	66	75
74	69	63	61	74	74	24	78	63	69	63	63	63	73	26	73
65	63	65	66	76	65	64	64	fd	97	66	63	76	12	1a	1a
62	72	67	61	64	63	61	73	74	27	78	63	63	66	65	65
63	73	65	65	74	63	61	73	74	27	78	63	63	66	65	65
63	73	65	65	74	63	61	73	74	27	78	63	63	66	65	65
72	67	63	64	63	61	73	74	2d	78	63	63	66	65	74	73
28	73	65	66	74	48	85	78	18	12	8c	67	75	74	78	73
74	24	64	72	67	78	73	38	78	7a	16	12	12	67	73	73
78	75	74	76	75	75	75	75	65	28	64	62	67	78	73	73
88	7a	78	12	86	69	78	65	75	74	28	64	72	67	78	73
38	88	7a	15	12	11	69	66	78	75	74	28	71	75	65	75
65	24	64	72	67	78	73	38	78	7a	19	12	15	72	73	73
74	28	78	61	63	66	65	73	73	28	72	65	63	65	69	73
64	64	64	64	7a	12	16	64	67	61	61	65	65	65	28	78
74	12	12	12	12	12	12	12	74	64	66	65	65	64	28	78

[illegible]

87	00	85	88	89	90	00	00	01	00	00	05	33	node_id
88	16	00	12	00	79	22	65	67	64	65	57	69	64	57
89	74	72	22	34	22	46	43	53	35	30	31	51	74	67
90	22	22	22	73	65	62	73	62	69	78	74	69	67	6e
91	09	64	57	73	74	72	36	22	74	65	73	74	22	2c
92	05	65	63	61	64	66	66	67	57	70	61	74	68	22
93	43	69	73	63	67	20	49	47	53	20	58	52	20	69
94	72	61	20	73	74	61	74	64	64	20	67	70	65	72
95	60	66	72	61	20	73	74	64	74	60	73	74	60	63
96	72	69	65	74	65	72	86	61	63	65	73	21	69	65
97	66	61	63	65	27	6c	61	74	65	73	74	27	67	65
98	72	69	63	20	63	67	75	66	74	65	72	73	22	2c
99	63	67	6c	6c	65	74	69	67	57	69	64	22	34	73
00	31	38	36	31	39	32	2c	22	63	67	6c	6c	65	63
01	69	67	57	73	74	61	72	72	57	74	69	66	65	22
02	69	67	63	65	65	65	65	65	65	65	65	65	65	22
03	73	67	74	69	66	65	73	74	61	66	78	22	3c	2d
04	32	33	38	32	34	33	32	32	38	39	2c	22	64	61
05	74	69	63	73	61	6c	22	34	5b	70	22	74	69	65
06	74	61	66	70	22	31	31	35	32	33	38	32	34	33
07	32	38	38	2c	22	6b	65	79	73	22	34	72	69	6e
08	72	66	61	63	65	20	6c	65	65	65	65	22	34	73
09	65	64	72	65	64	67	69	67	45	30	27	30	27	30
10	22	76	22	22	63	67	62	74	65	65	74	22	36	72
11	63	65	65	74	73	22	67	65	63	65	65	76	65	64
12	34	39	37	39	34	31	38	36	22	62	69	74	65	73
13	24	72	65	63	65	65	76	65	64	22	34	38	35	34
14	32	32	38	30	32	2c	22	78	63	63	65	74	73	2d
15	73	65	64	74	22	34	32	35	37	35	34	38	33	35
16	38	32	22	62	79	74	65	73	23	73	65	66	74	22
17	35	34	32	38	36	37	31	32	33	39	36	39	32	2c
18	74	73	24	72	63	63	65	69	78	65	64	22	34	37
19	36	33	34	33	2c	62	72	67	61	64	63	61	73	24
20	61	63	65	65	74	73	2d	73	65	63	65	65	76	65
21	22	34	34	2c	22	6d	75	6c	74	69	63	63	74	26
22	01	63	65	74	73	2d	73	65	64	74	22	34	35	2c
23	32	37	36	2c	62	72	67	61	64	63	61	73	24	30
24	70	61</												

Design Your Transport Network Properly

Peak bandwidth consumption



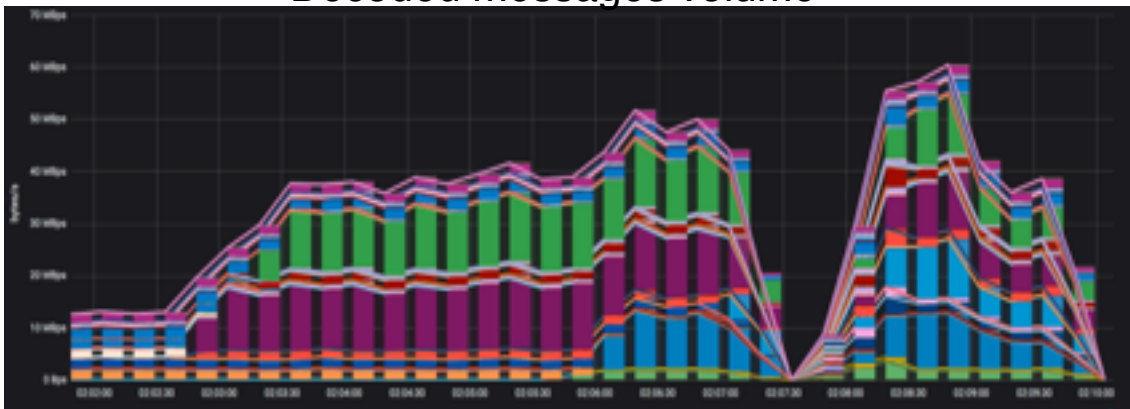
315k counters
Every 5 seconds

Agenda

- 1 Brief Telemetry Overview
- 2 Closer Look at Router
- 3 Closer Look at the Link Between
- 4 Closer Look at Collector**
- 5 Final Thoughts

Is Your Collector Fast Enough?

Decoded messages volume

[illegible]

Make sure the collector has enough power to process your telemetry data.

Is Your Hard Drive Write Speed Fast Enough?

HDD-based server (SAS)



SSD-based server (SAS)



What To Think About Selecting a TSDB

Metric	InfluxDB	Prometheus	Elasticsearch	OpenTSDB
Website	https://influxdata.com/	https://prometheus.io/	https://www.elastic.co/products/elasticsearch	http://opentsdb.net
Category	Real-time Analytics	Monitoring System	Real-time Search	Real-time Analytics
Supported Measurements	metrics, events	metrics	metrics, events	metrics
High Availability (HA)	Double writing 2 servers	Double writing 2 servers	Clustering	Clustering
Underlying Technology	Golang	Golang	Java	Java, Hadoop
Storage Backend	Custom	Custom	Document	Hadoop (Columnar)
Supported Data Types	int64, float64, bool, and string	float64	string, int32, int64, float32, float64, bool, null	int64, float32, float64
Bytes per point after compression	2.2	1.3	22	12
Metric Precision	nano second	milli second	milli second	milli second
Write Performance - Single Node	470k metrics / sec (custom HW)	800k metrics / sec	30k metrics / sec	32k metrics /sec (calculated)
Query Performance (1 host, 12hr by 1m)	3.78 ms (min), 8.17 (avg)	tbd	13.23 ms (min), 28.6 (avg)	tbd
Query Language	InfluxQL (SQL like)	PromQL	Query DSL	lookup only
Community Size	large	large	large	medium
Maturity	Stable	Stable	Stable	stable

Full table: <https://tinyurl.com/jsd4esy>
 Good to read: <https://tinyurl.com/ybaw4ww6>

InfluxDB vs OpenTSDB: <https://tinyurl.com/y8ofbjyy>
 InfluxDB vs Cassandra: <https://tinyurl.com/y83vv9ys>
 DB ranking: <https://tinyurl.com/ya8rrrpj>
 InfluxDB vs Elasticsearch: <https://tinyurl.com/y7yxjf6v>

Everything is working, where are my stats?

RP/0/RP0/CPU0:ios-xr# sh clock

Sun Apr 1 20:56:15.074 PDT
20:56:15.167 PDT Sun Apr 1 2018

cisco@ubuntu51-1:~\$ date

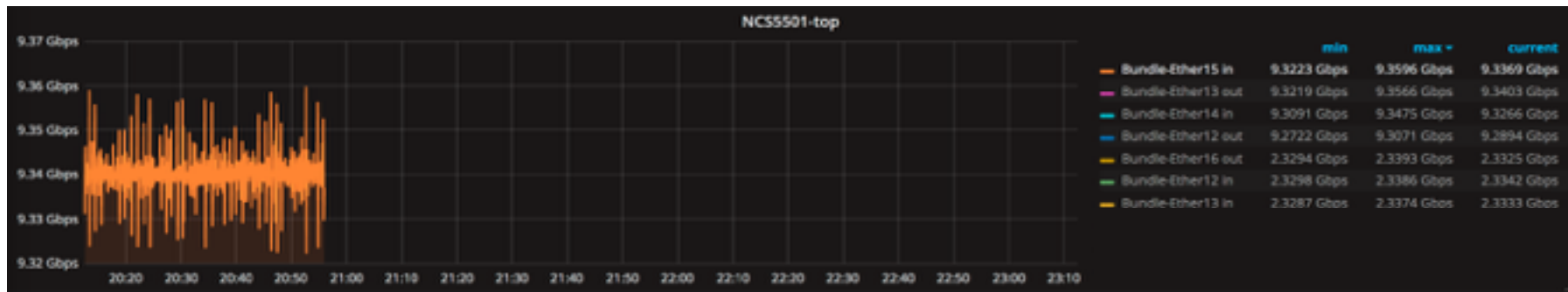
Sun Apr 1 23:13:11 PDT 2018

RP/0/RP0/CPU0:ios-xr#sh tele m subscription if-stats

Sun Apr 1 20:50:17.883 PDT

Subscription: if-stats

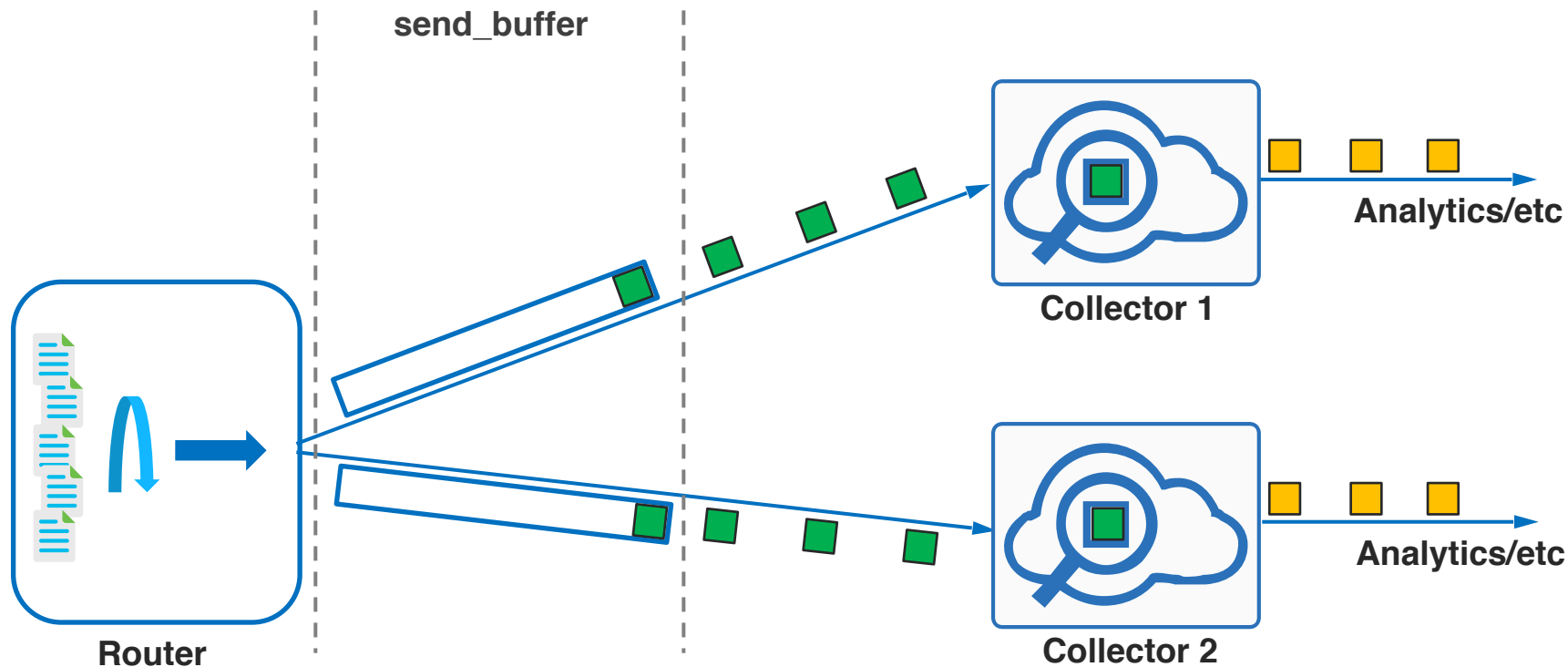
```
-----
State: ACTIVE
BSCP/0os marked value: Default
Sensor groups:
Id: if-stats
  Sample Interval: 5000 ms
  Sensor Path: Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/
interface[interface-name='Bundle-Ether*']/latest/generic-counters
  Sensor Path State: Resolved
Destination Groups:
Group Id: DGroup1
  Destination IP: 10.30.110.40
  Destination Port: 57500
  Encoding: self-describing-gpb
  Transport: grpc
  State: Active
-----
```



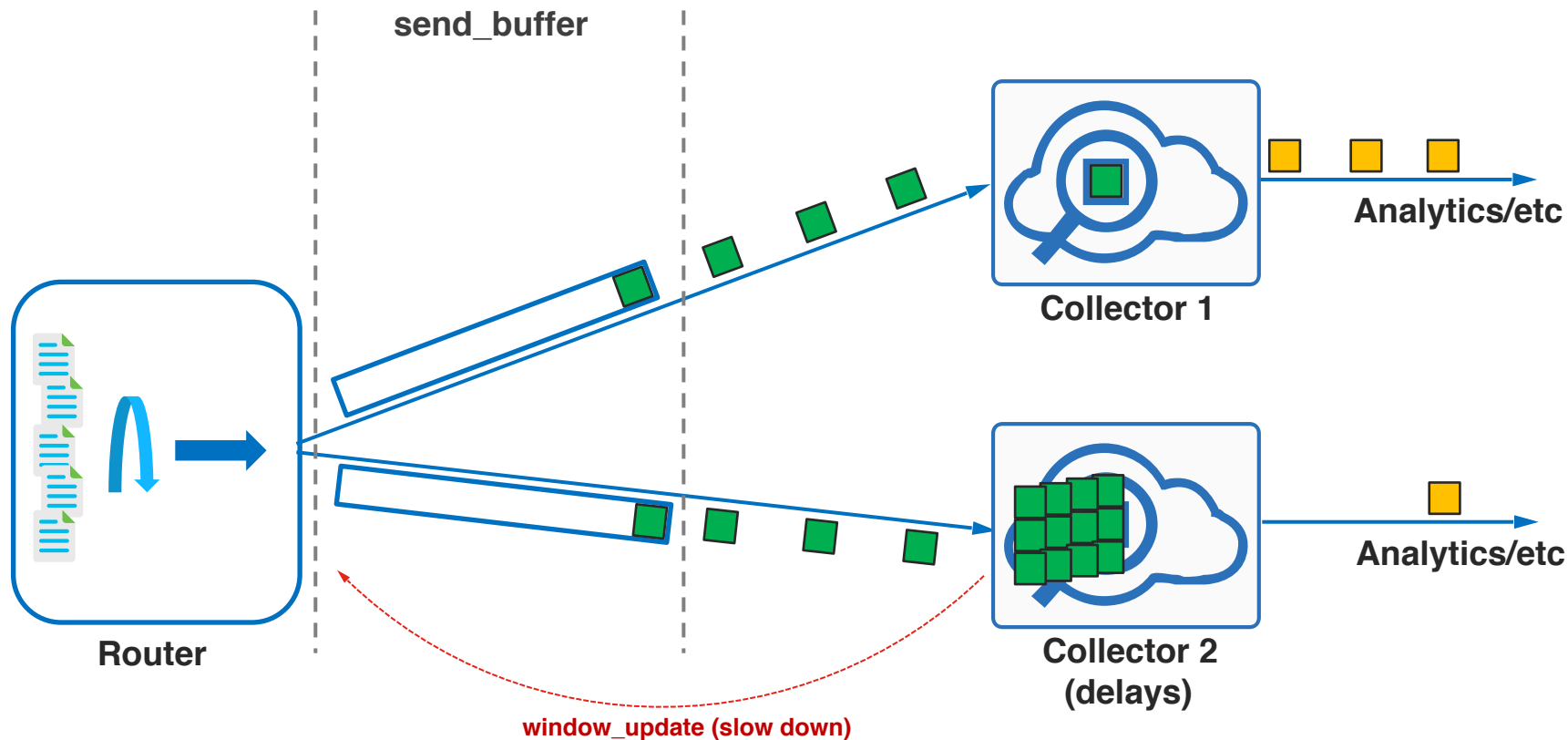
Agenda

- 1 Brief Telemetry Overview
- 2 Closer Look at Router
- 3 Closer Look at the Link Between
- 4 Closer Look at Collector
- 5 **Final Thoughts**

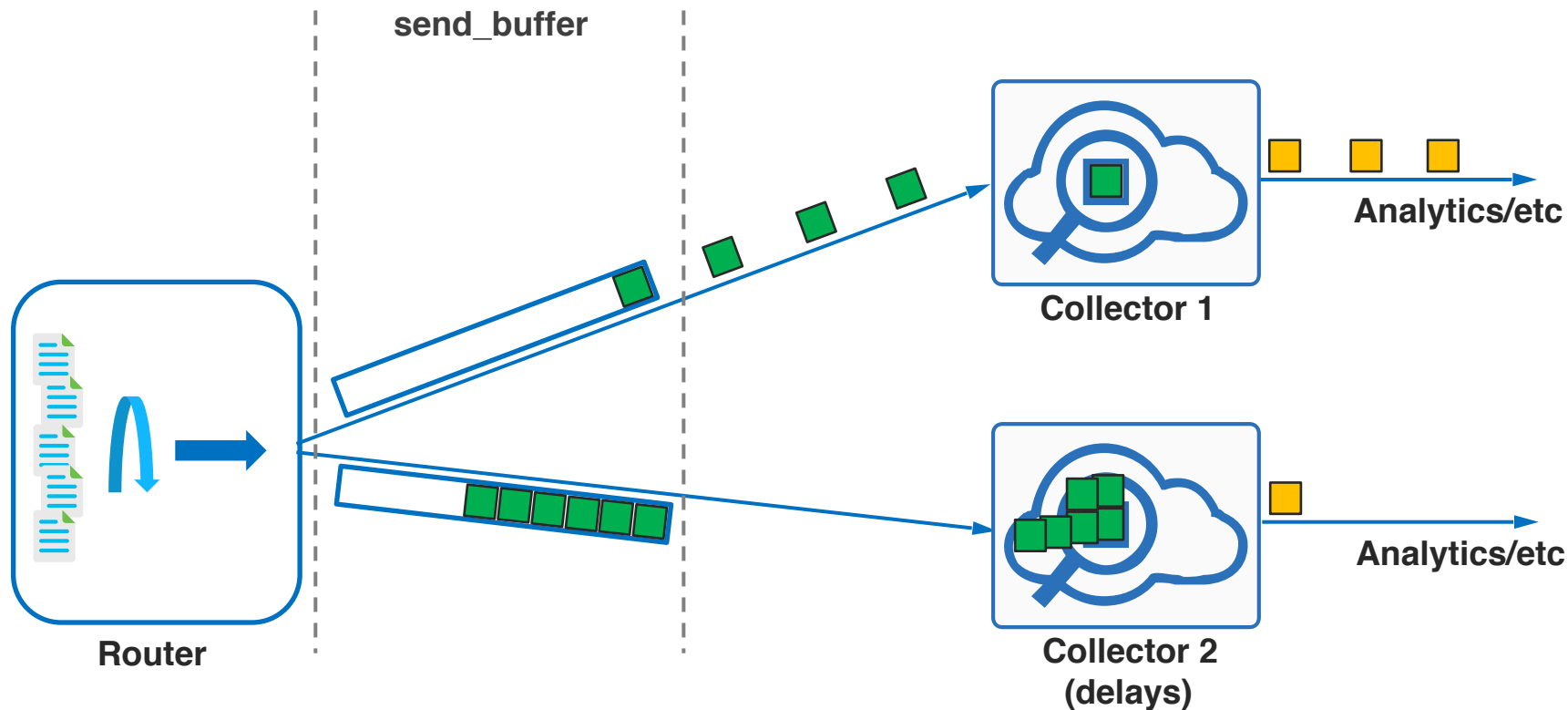
Think About Your Overall Design



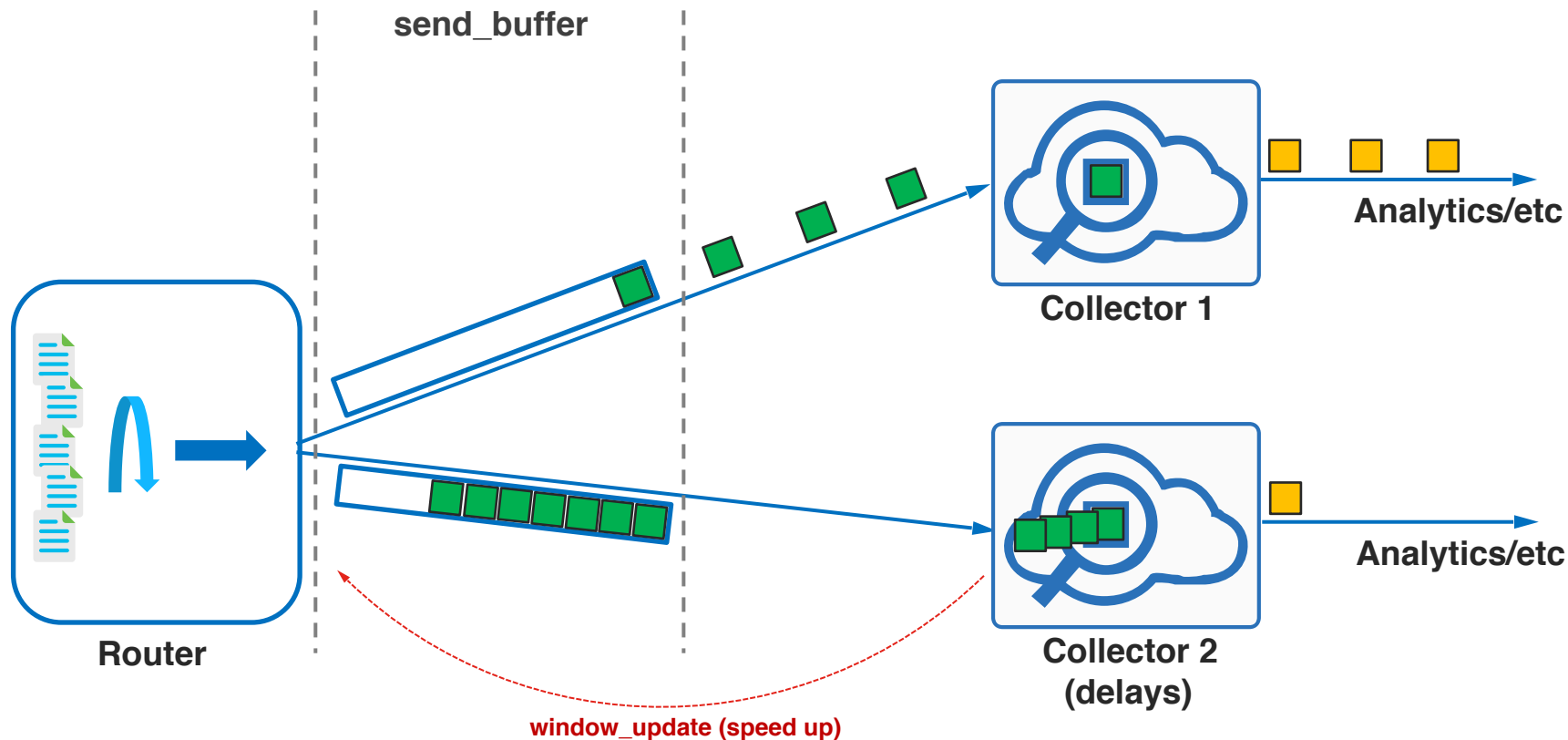
Think About Your Overall Design



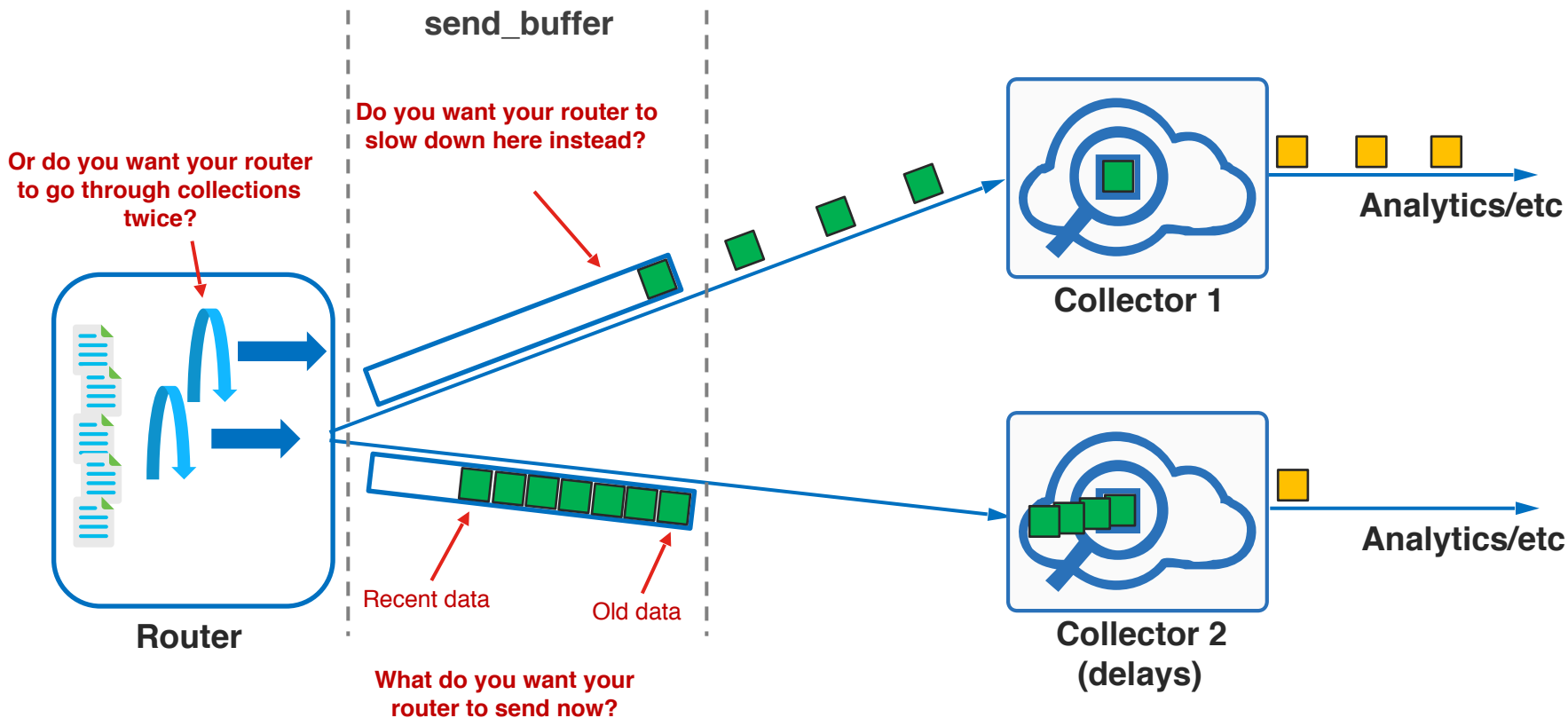
Think About Your Overall Design



Think About Your Overall Design



Think About Your Overall Design



It Is Good to Know More



A snapshot from <https://www.youtube.com/watch?v=ZMZJ3ZaEclQ>