# 4 Subscription Drafts

## IETF #96 Berlin

## 20-July-2016

NETCONF Charter Item 6:

*“Enhance RFC 5277 with the ability to delete subscriptions without closing the client session, to modify existing subscriptions, and to have multiple subscriptions on a established client session. These changes should not affect older clients that do not support these particular subscription requirements. The RPCs and the data models in RFC 5277 should be converted to YANG*

Authors

Sharon Chisholm
**Alexander Clemm**
Einar Nilsen-Nygaard  Alberto
Gonzalez Prieto
Hector Trevino
Ambika Prasad Tripathy
**Eric Voit**

+ Contributors

Andy Bierman
Yan Gang
Peipei Guo
Susan Hares
Tim Jenkins
Balazs Lengyel
Kent Watsen
Guangying Zheng (Walker)

weekly call

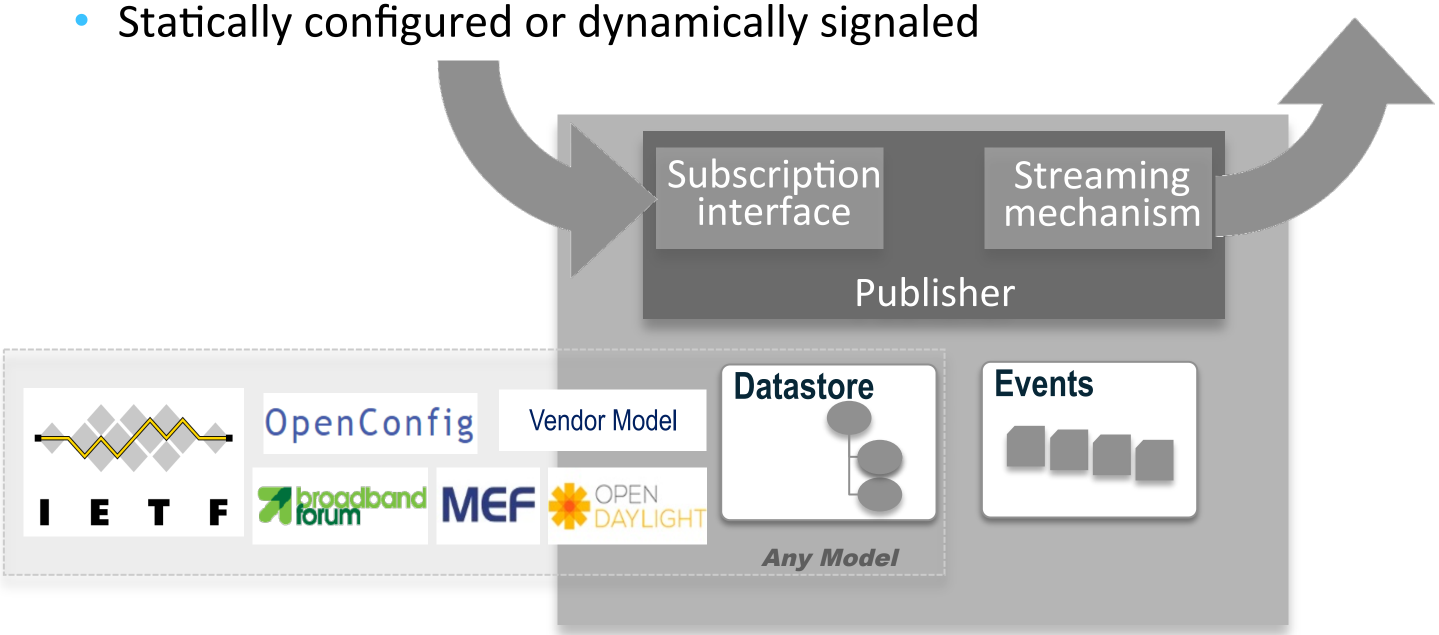
# Event & YANG Subscriptions
## Context

## Subscribing to updates

- Event Stream or YANG Datastore Subtree(s)
- Statically configured or dynamically signaled

## Streaming of updates

- Customized to recipient
- On-change, Periodic, Event



General Introduction Session

Thursday 10-11:30AM, Tegel Conf Room

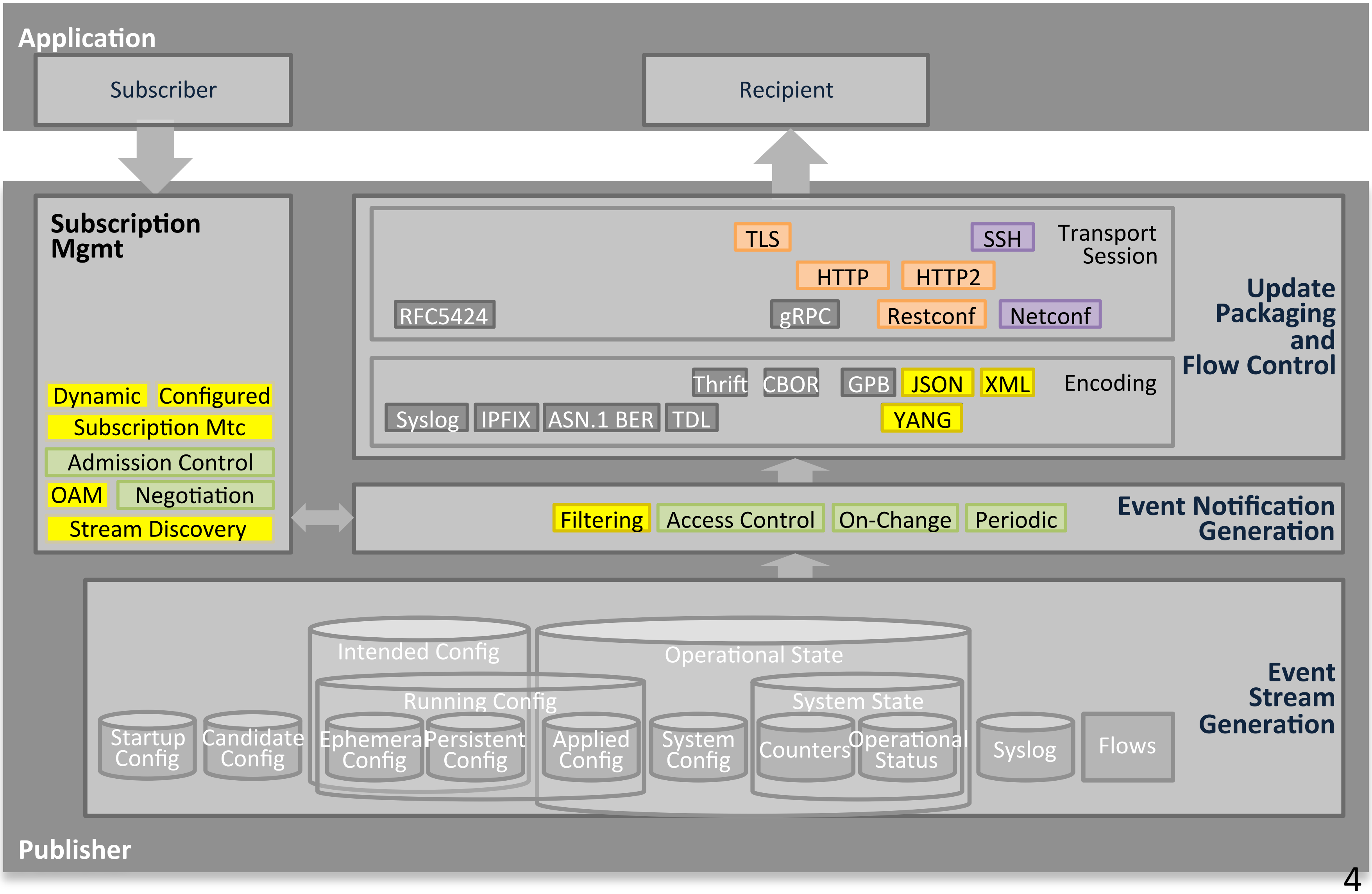
# Event & YANG Subscriptions
# 4 Drafts

| | Current draft | Git name | |
|---|---|---|---|
| Subscription Mechanism: | **YANG Datastore Push**<br>draft-ietf-netconf-yang-push | yang-push | adopted |
| | **Subscriptions for Event Notifications**<br>draft-gonzalez-netconf-5277bis | rfc5277bis | Request for Adoption |
| Choice of Transports: | **NETCONF Transport for Event Notifications**<br>draft-gonzalez-netconf-event-notifications | notif-netconf | |
| | **RESTCONF & HTTP Transport for Event Notifications**<br>draft-voit-netconf-restconf-notif | notif-restconf | |
| | Future Transport Notification drafts | | |

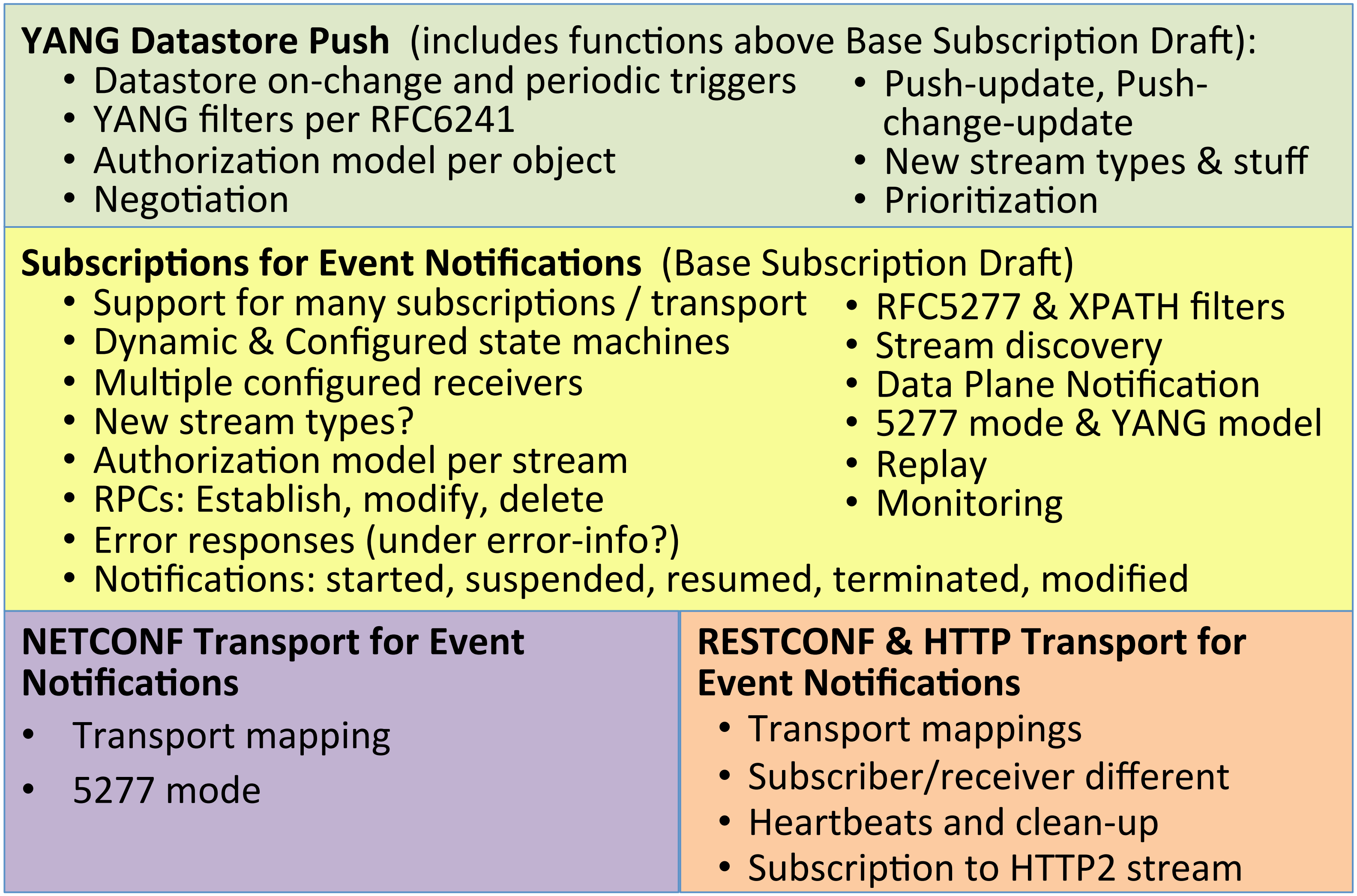
Github repository https://github.com/netconf-wg

- Minutes, Meeting Recordings, Terminology, Issues

# 4 Drafts in Layered Framework



# 4 Drafts Functional Partitioning



**YANG Datastore Push** (includes functions above Base Subscription Draft):

- Datastore on-change and periodic triggers
- YANG filters per RFC6241
- Authorization model per object
- Negotiation
- Push-update, Push-change-update
- New stream types & stuff
- Prioritization

**Subscriptions for Event Notifications** (Base Subscription Draft)

- Support for many subscriptions / transport
- Dynamic & Configured state machines
- Multiple configured receivers
- New stream types?
- Authorization model per stream
- RPCs: Establish, modify, delete
- Error responses (under error-info?)
- Notifications: started, suspended, resumed, terminated, modified
- RFC5277 & XPATH filters
- Stream discovery
- Data Plane Notification
- 5277 mode & YANG model
- Replay
- Monitoring

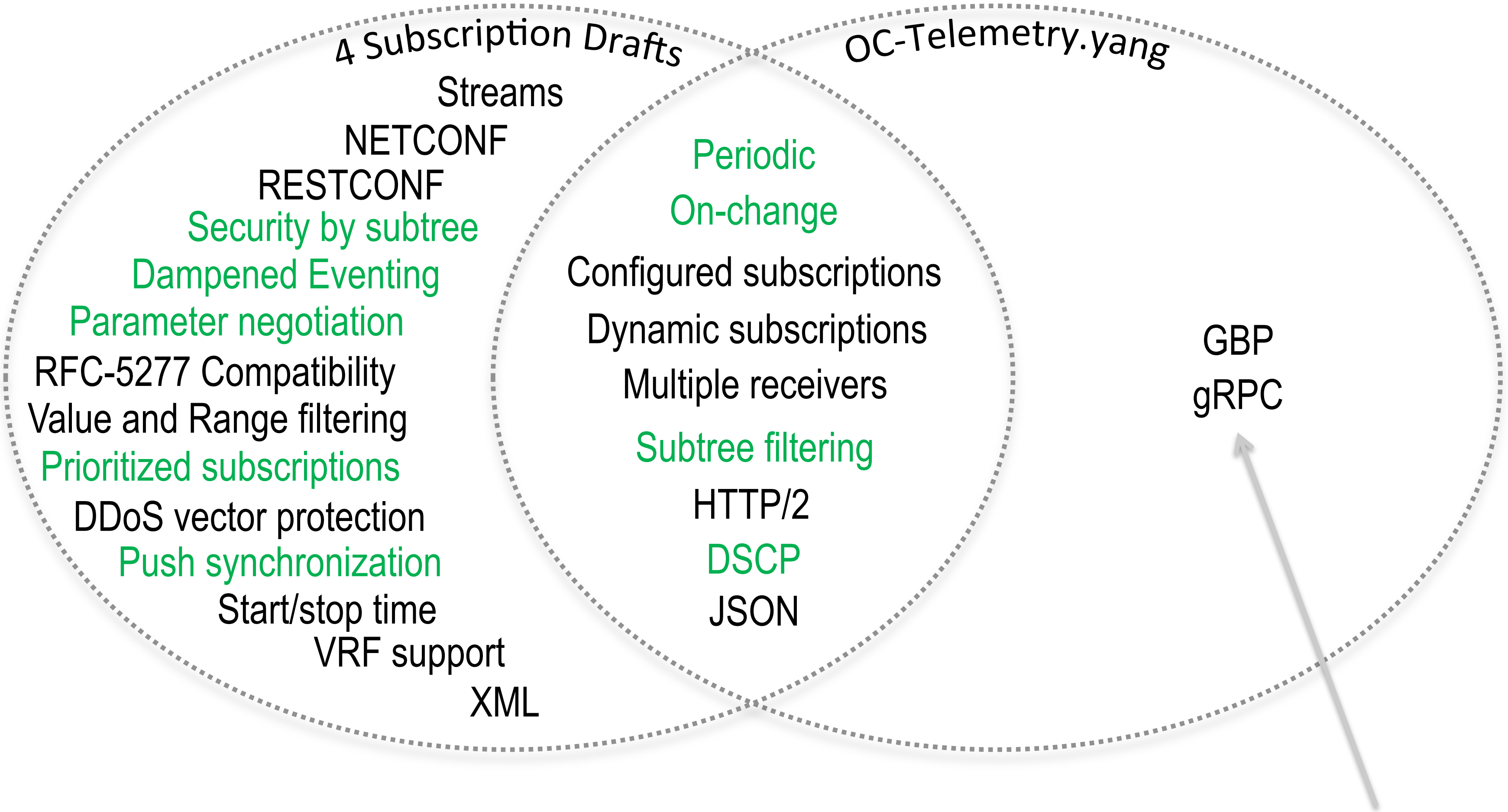
**NETCONF Transport for Event Notifications**

- Transport mapping
- 5277 mode

**RESTCONF & HTTP Transport for Event Notifications**

- Transport mappings
- Subscriber/receiver different
- Heartbeats and clean-up
- Subscription to HTTP2 stream

# Context with OC-Telemetry.yang



4 Subscription Drafts

Streams
NETCONF
RESTCONF
Security by subtree
Dampened Eventing
Parameter negotiation
RFC-5277 Compatibility
Value and Range filtering
Prioritized subscriptions
DDoS vector protection
Push synchronization
Start/stop time
VRF support
XML

Periodic
On-change
Configured subscriptions
Dynamic subscriptions
Multiple receivers
Subtree filtering
HTTP/2
DSCP
JSON

OC-Telemetry.yang

GBP
gRPC

Recurring requirement: specification of market requested, non-IETF technologies

For both Events & YANG Datastores
For only YANG Datastore Push

# rfc5277bis
# Highlights

- Support for dynamic subscriptions (via RPC) and configured subscriptions
  - Dynamic:
    - Subscription lifetime tied to subscriber session
    - Supports negotiation of subscription parameters
  - Configured:
    - Established via configuration
    - Allows multiple receivers
  - No mix-and-match – cannot terminate configured subscription via RPC and vice-versa
- Subscriptions are directed at an Event Stream
  - NETCONF (all notifications, per RFC 5277)
  - Potentially others: OAM, Push, more
- Subscriptions can specify filters
- Support for multiple transport and encoding mappings

# rfc5277bis
# Model overview: Streams, filters

```
module: ietf-event-notifications
   +--ro streams
   |  +--ro stream*   notif:stream
   +--rw filters
   |  +--rw filter* [filter-id]
   |     +--rw filter-id    filter-id
   |     +--rw (filter-type)?
   |        +--:(rfc5277)
   |           +--rw filter
   +--…
```

# rfc5277bis
# Model overview: subscription configuration

```
module: ietf-event-notifications
   +--…
   +--rw subscription-config {configured-subscriptions}?
   |  +--rw subscription* [subscription-id]
   |     +--rw subscription-id     subscription-id
   |     +--rw stream?             stream
   |     +--rw (filter-type)?
   |     |  +--:(rfc5277)
   |     |  |  +--rw filter
   |     |  +--:(by-reference)
   |     |     +--rw filter-ref?         filter-ref
   |     +--rw startTime?          yang:date-and-time
   |     +--rw stopTime?           yang:date-and-time
   |     +--rw encoding?           encoding
   |     +--rw receivers
   |     |  +--rw receiver* [address]
   |     |     +--rw address     inet:host
   |     |     +--rw port        inet:port-number
   |     |     +--rw protocol?   transport-protocol
   |     +--rw (push-source)?
   |        +--:(interface-originated)
   |        |  +--rw source-interface?   if:interface-ref
   |        +--:(address-originated)
   |           +--rw source-vrf?         uint32
   |           +--rw source-address      inet:ip-address-no-zone
   +--…
```

# rfc5277bis
# Model overview: subscription state

```
module: ietf-event-notifications
   +--…
   +--ro subscriptions
      +--ro subscription* [subscription-id]
         +--ro subscription-id           subscription-id
         +--ro configured-subscription?   empty {configured-subscriptions}?
         +--ro subscription-status?       subscription-status
         +--ro stream?                    stream
         +--ro (filter-type)?
         |  +--:(rfc5277)
         |  |  +--ro filter
         |  +--:(by-reference)
         |     +--ro filter-ref?                filter-ref
         +--ro startTime?                 yang:date-and-time
         +--ro stopTime?                  yang:date-and-time
         +--ro encoding?                  encoding
         +--ro receivers
         |  +--ro receiver* [address]
         |     +--ro address     inet:host
         |     +--ro port        inet:port-number
         |     +--ro protocol?   transport-protocol
         +--ro (push-source)?
            +--:(interface-originated)
            |  +--ro source-interface?          if:interface-ref
            +--:(address-originated)
               +--ro source-vrf?                uint32
               +--ro source-address             inet:ip-address-no-zone
```

# rfc5277bis
# Other model aspects

- RPCs (for dynamic subscriptions)
  - Establish-subscription
  - Modify-subscription
  - Delete-subscription
- Notifications (OAM)
  - OAM notifications are used by server to signal receivers certain events concerning the subscription itself
  - Basic lifecycle
    - Subscription-started, -modified, -terminated
    - Added-to-subscription, removed-from-subscription (configured subscriptions only)
  - Temporary suspension by server
    - Subscription-suspended, subscription-resumed
    - Server has the option to suspend the subscription when needed
    - YANG-Push: server might not be able to keep up with update events in some circumstances – e.g. large number of instances, high velocity of changes, etc.
    - Receivers need to be able to "count on" subscription (unless told otherwise) to not have to revert to polling
    - Defined in RFC5277bis, as might be applicable beyond YANG-Push
  - Replay complete

# rfc5277bis
# Control Plane Notifications aka OAM Messages

- Servers need to indicate to receivers relevant events about the subscription itself
- Events to be signaled by the server modeled as YANG Notifications
  - Notifications can be initiated by the server
- Issues
  - Standard YANG Notifications are “general purpose” – anybody can subscribe
  - How to allow receiver to only get notifications regarding “its” subscriptions, not everyone else’s
- Proposed solution
  - Add YANG extension “control-plane-notif”
    - Use to tag OAM notifications
    - Tagged notifications are not part of regular event stream but signaling stream
    - Notification receiver is automatically subscribed to signaling stream of “its” subscription
  - Consideration: more general tag to indicate event stream as part of notification definition (default: NETCONF)

# rfc5277bis Event Streams

- Which Event Streams can, which Event Stream must a server provide?
- NETCONF – RFC 5277bis
  - NETCONF stream per RFC 5277
  - Any event that is raised per YANG Notification definition in YANG Modules implemented by the server
  - Superset of all notifications that can be raised
- Subscriber can apply filtering, but single stream appears too broad, unwieldy to handle
  - Not every event is of interest to everyone – e.g. OAM messages
  - Many use cases require only well-defined notification subsets – eg YANG Push
  - Separate streams avoid need for complex filters – greater efficiency, simpler to use
- Which additional streams to provide is up to server implementations
  - Only NETCONF "MUST" be available
  - System-provided streams are "discoverable":  
    List of system streams as oper data that is part of the YANG model
  - Other streams may make sense to provide, some of which may makes sense to define in a standard

# rfc5277bis
# Next steps

- Adopt as WG draft?
- Address issues:

| | |
|---|---|
| EN1 | Definition and domain of basic set of Stream types. What streams are provided and what do they contain (includes default 5277 stream). |
| EN2 | Clarify interplay between filter definitions and different streams. Includes information in subtrees of event payloads. |
| EN3 | Mechanisms for diagnostics, e.g. deal with dropped updates, monitoring when they occur, etc. |
| EN4 | How to allow for seamless integration with non-standard encodings and transports (like GPB/GRPC). Specify requirements encoding and transport must meet, provide examples. |
| EN5 | Along with Netconf-notif, should this draft obsolete 5277 or be in parallel with it? |
| EN6 | Stream discovery. Are adjustments needed for maximal transport independence? |
| EN7 | Detecting loss of a sequential update notification, and mechanisms to resend. Sequence numbers: facilitate detection of event messages that have been dropped within a subscription (on a stream, after filtering was applied) |
| EN8 | Should we have a mandatory transport? |
| EN9 | Notification ID: facilitate deduplication of events seen on multiple subscriptions and overlapping streams |

# yang-push
# Updates since IETF #95

- One revision update
- Pulled basic subscription model out
  - draft-gonzalez-5277-bis
  - YANG-Push now builds on top of this
  - YANG-Model now an augmentation
- Augmentations to RPC definitions to include YANG-Push subscription parameters, as applicable
- Associated editorial updates throughout
  - Including discussion of issues being worked through

# yang-push
# Model overview: subscription configuration

```
module: ietf-event-notifications
   +--…
   +--rw subscription-config {configured-subscriptions}?
   |  +--…
      +--rw (update-trigger)?
      |  +--:(periodic)
      |  |  +--rw period                      yang:timeticks
      |  +--:(on-change)   {on-change}?
      |     +--rw no-synch-on-start?         empty
      |     +--rw dampening-period           yang:timeticks
      |     +--rw excluded-change*           change-type
      +--rw dscp?                      inet:dscp
      |    {notif-bis:configured-subscriptions}?
      +--rw subscription-priority?     uint8
      +--rw subscription-dependency?   string
|     +--rw (filter-type)?
   |  |  +--:…
         +--:(update-filter)
            +--rw (update-filter)?
               +--:(subtree)
               |  +--rw subtree-filter
               +--:(xpath)
                  +--rw xpath-filter?     yang:xpath1.0
```

*YANG-Push augmentations to RFC5277bis*

*Update triggers*

*Push QoS*

*Addl. filter options*

# yang-push
# Model overview: subscription state

```
module: ietf-event-notifications
   +--…
   +--ro subscriptions
      +--ro subscription* [subscription-id]
         +--…
         +--ro (update-trigger)?
         |  +--:(periodic)
         |  |  +--ro period                     yang:timeticks
         |  +--:(on-change) {on-change}?
         |     +--ro no-synch-on-start?         empty
         |     +--ro dampening-period           yang:timeticks
         |     +--ro excluded-change*           change-type
         +--ro dscp?                      inet:dscp
         |                          {notif-bis:configured-subscriptions}?
         +--ro subscription-priority?     uint8
         +--ro subscription-dependency?   string
         +--…
         +--ro (filter-type)?
         |  +--:…
            +--:(update-filter)
               +--ro (update-filter)?
                  +--:(subtree)
                  |  +--ro subtree-filter
                  +--:(xpath)
                     +--ro xpath-filter?     yang:xpath1.0
```

# yang-push
# Next Steps

- Update the spec as we close on discussion items

| | |
|---|---|
| YP1 | Which stream types to introduce. Current list includes streams for all operational and for all config data. Consider adding stream for operational data minus counters. Also: assess implications of opstate implications on required data streams. |
| YP2 | In addition to identifying which items go to which streams, identifying and calling out which items (such as counters) should not be "on-change subscribable" may be useful. Consider introducing a Yang extension to define if an object: is-a-counter and/or not-notifiable. |
| YP3 | What QoS parameters should be supported for subscriptions? |
| YP4 | Implications of ephemeral requirements from I2RS |
| YP5 | Filters: YANG 1.1 allows filters to be defined in multiple places. How do they intersect each other in a deterministic way. |
| YP6 | On-change subscription: consider providing publisher with capability to initiate a refresh of contents rather than send deltas. Current proposal allows for a "synch-on-start" option; such an option might be useful also e.g. on resumption of a subscription that had been suspended. |
| YP7 | Do we need an extension for NACM to support filter out datastore nodes for which the receiver has no read access? |

# notif-netconf Highlights

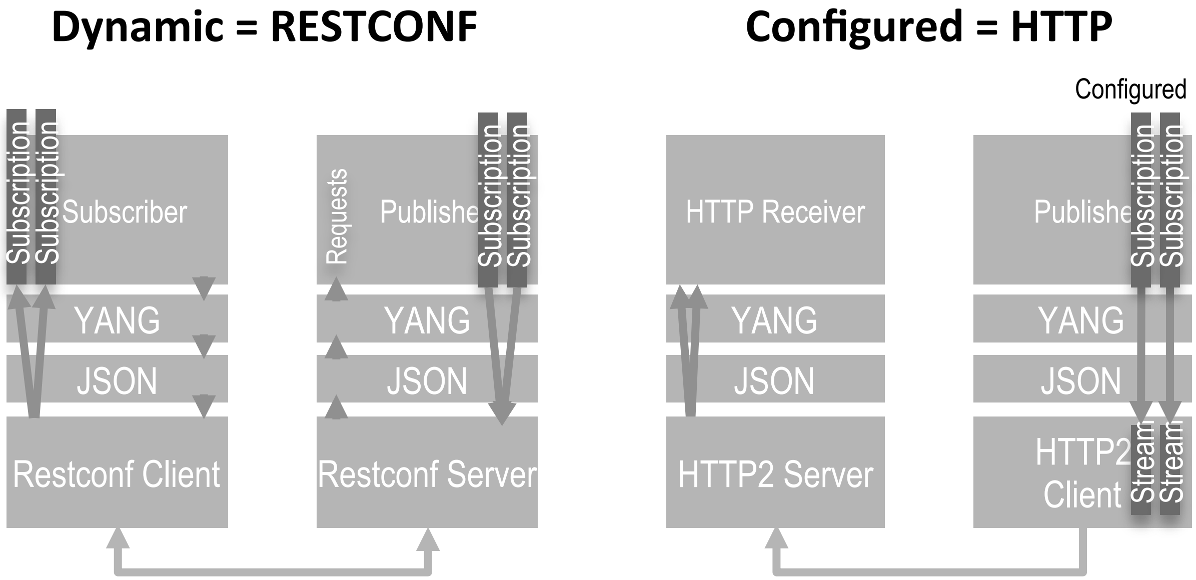
- Support multiple subscriptions over a single NETCONF session.
- Support a revised definition of the default NETCONF stream (capabilities exchange)
- Backwards compatibility with RFC 5277

# notif-netconf
# Next Steps

- Adopt as WG draft?
- Address issues:

| | |
|---|---|
| NT1 | Support multiple create-subscriptions over a single NETCONF session? or only multiple establish-subscription? |
| NT2 | Configured subscription need to be refined in [event-notifications] and then adjust this document based on it. |
| NT3 | Express filter in JSON should be documented. |
| NT4 | Call Home support |

# notif-restconf Highlights



- Leverage HTTP/2 QoS capabilities where viable
  - Subscription multiplexing over independent HTTP/2 streams
  - Stream prioritization and stream dependencies
  - Flow control on independent streams
- Considering proxy subscription transport issues

# notif-restconf
# Next Steps

- Adopt as WG draft?
- Address issues:

| | |
|---|---|
| RT1 | Integration specifics for Restconf capability discovery on different types of Streams |
| RT2 | In what way to we position "Event notifications" model in this document vs. current solution in Restconf. |
| RT3 | Do we include 3rd party signaled subscriptions within models that need to be supported generically, or for a particular type of transport. |
| RT6 | We need to define encodings of rfc5277bis notifications for both Restconf and HTTP. |
| RT7 | HTTP native option doesn't currently use SSE.  But we should evaluate moving to that as possible.  It will make development integration easier and more consistent. |

Thank you!

# Some Terms

| |
|---|
| Configured Subscription |
| Data Node |
| Data Node Filter |
| Data Node Security Filter |
| Data Node Update |
| Dynamic Subscription |
| Datastore |
| Event |
| Event Notification |
| Event Stream |
| Filter |

| |
|---|
| Notification |
| Publisher |
| Receiver |
| Subscriber |
| Subscription |
| Subscription Policy |
| Update Notification |
| Update Record |
| Update Record Filter |
| Update Stream |
| Update Trigger |

Working definitions at:
https://github.com/netconf-wg/yang-push/wiki/Terminology
(Expect tweaks/changes)

# yang-push
# Selected discussion items

- YANG-Push stream types
  - YANG-PUSH: Covers all YANG data, both configuration and operational
  - OPER-PUSH: operdata only – superset of “statsonly” and “nostats”
  - CONFIG-PUSH: config data only
  - Other candidates
    - Operdata-nostats
      Exclude stats – use cases include monitoring for state changes; on-change and periodic subscriptions
    - Operdata-statsonly:
      Use cases include performance management, time series analysis; periodic subscriptions only
    - Custom: user definable
  - Other potential streams for push of applied config, derived state, other operational data
- On-change subscribable data
  - Consider introducing tags or metadata to distinguish stats
- NACM implications
  - Filter out data to which subscriber has no read-access vs. accept only subscriptions when subscriber has access to all subscribed data

# rfc5277bis
# Candidate Event Streams

- NETCONF
  - General purpose stream per RFC 5277, every system supports
  - Any event that is raised per YANG notification definition in a YANG module
    (caveat as per below)
- CONTROL
  - Designate Control Plane Notifications as such – YANG tag
  - Special purpose stream - excluded from other event streams incl NETCONF
  - Clients are automatically subscribed as applicable – e.g. for subscription
- YANG-Push-related streams

# Differentiating Event Notifications & YANG Datastore Push

| | Events | Datastore |
|---|---|---|
| What you need | Consume a stream of Publisher generated **messages at the cadence determined by the Publisher** | Consume a stream of Publisher generated **YANG data updates at a cadence negotiated with the Subscriber** |
| What to use | 5277bis Event Notifications | YANG Push |
| Requirements | RFC 5277 + NETCONF WG Discussions | RFC 7923 |

complementary

# Functional Partitioning

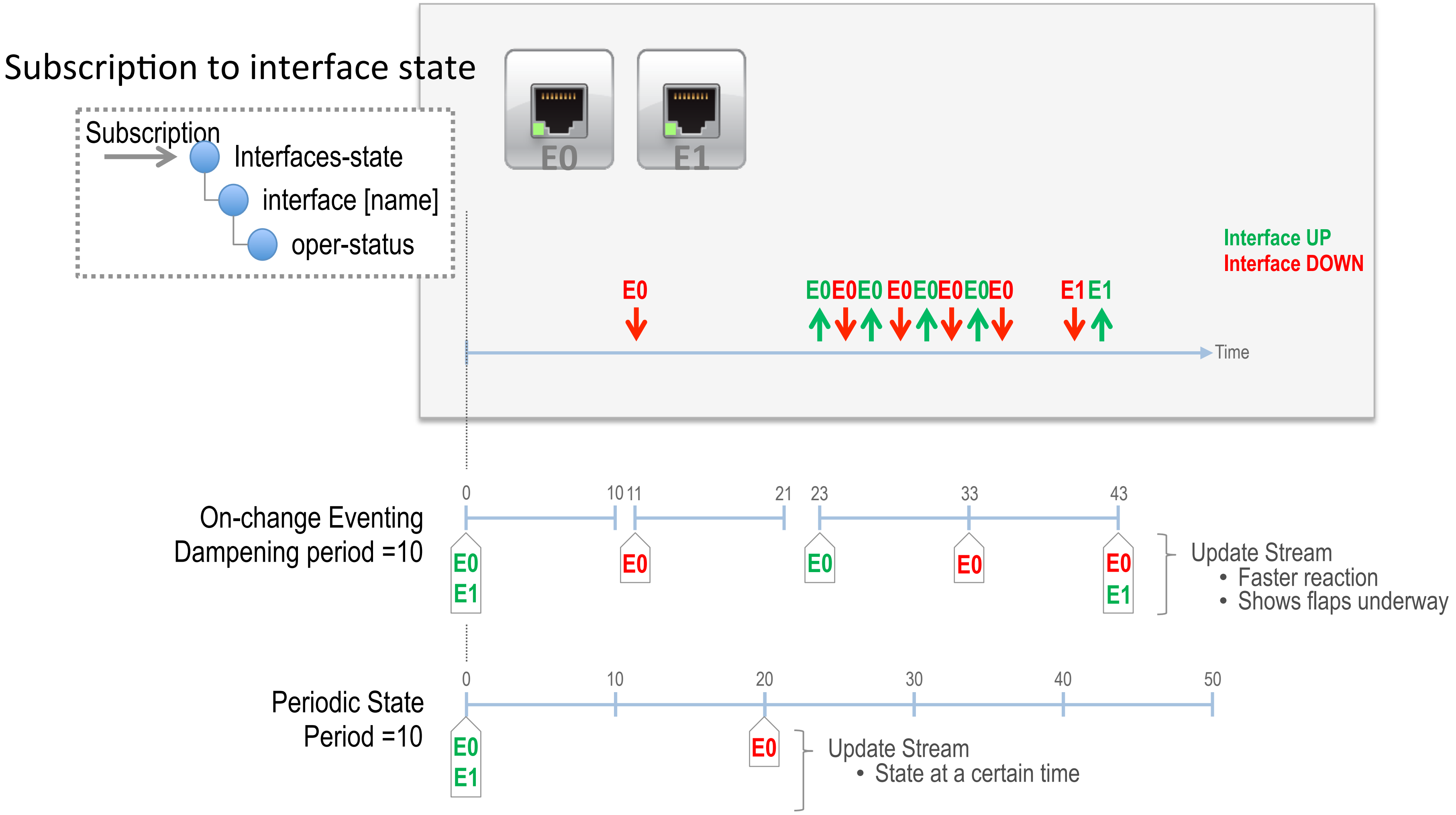
| | | Event Notifications | | YANG Datastore Push |
|---|---|---|---|---|
| | | 5277 Mode | Enhanced | |
| Subscription | Types of Subscription | Dynamic | Dynamic and Configured | |
| | Subscriptions per Session | one | many | |
| | Negotiation | No | Yes | |
| | RPCs | create | establish, modify, delete | |
| | Control Plane Notifications | None | started, suspended, resumed, terminated, modified | |
| Data Plane Notifications | | notification | +subscription-id | push-update,<br>push-change-update |
| Transport | NETCONF | Yes | | |
| | RESTConf, HTTP, HTTP2 | No | Yes | |

Legend

| |
|---|
| 🟩 YANG Datastore Push |
| 🟨 Subscriptions for Event Notifications |
| 🟪 NETCONF Transport for Event Notifications |
| 🟧 RESTCONF Transport for Event Notifications |

Compatibility with RFC-5277 ⬚

# Dampening Eventing vs. Periodic Behavior (1)



Subscription to interface state

Subscription

Interfaces-state

interface [name]

oper-status

On-change Eventing
Dampening period =10

Update Stream

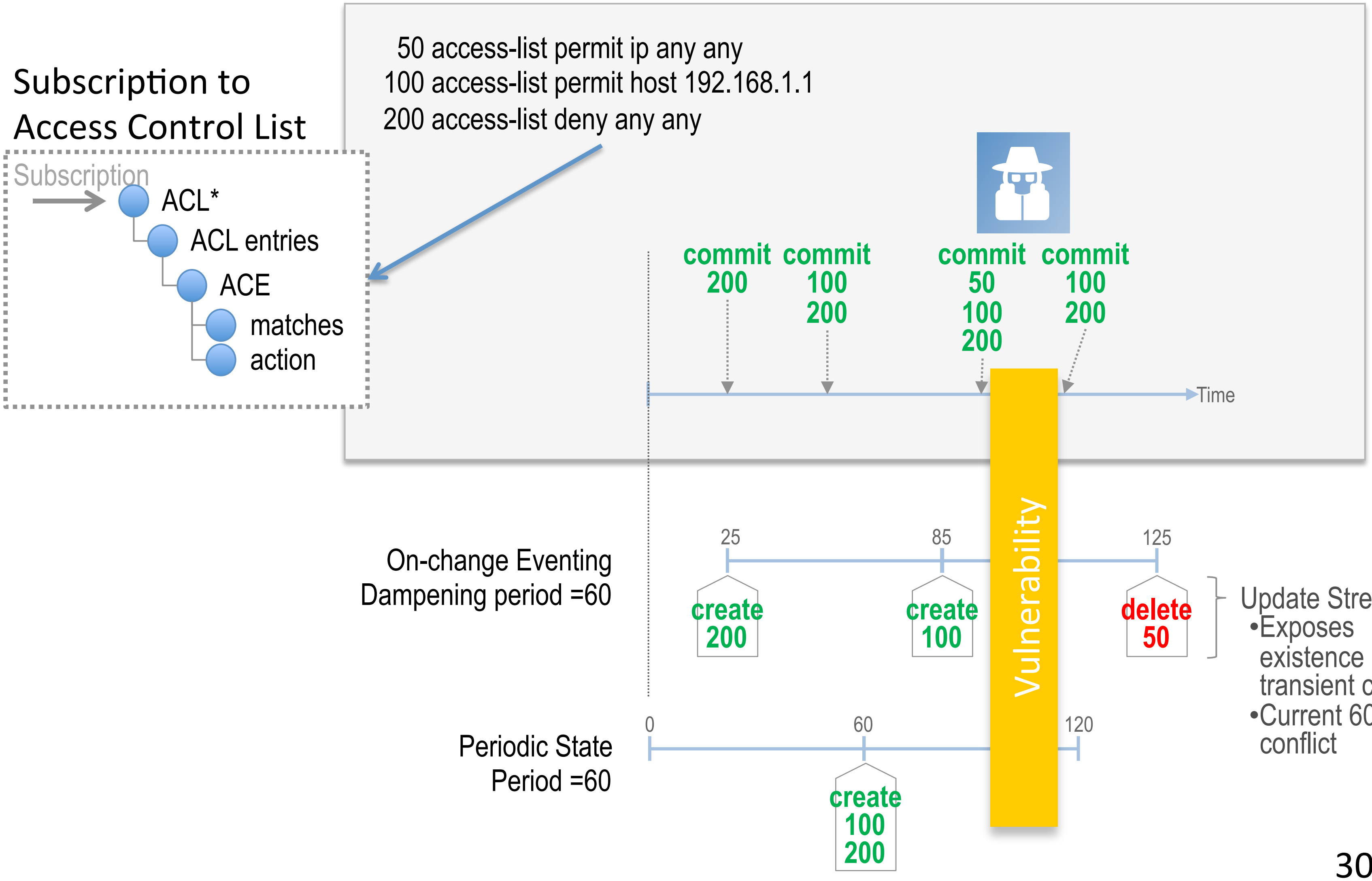
- Faster reaction
- Shows flaps underway

Periodic State
Period =10

Update Stream

- State at a certain time

# Dampening Eventing vs. Periodic Behavior (2)



Subscription to Access Control List

50 access-list permit ip any any
100 access-list permit host 192.168.1.1
200 access-list deny any any

Update Stream
- Exposes existence of transient config
- Current 6020 conflict

# Prioritization of Subscriptions

subscription-priority (8bit integer, optional)
priority of a subscription

Weight (8bit integer)
enables proportional bandwidth when there
are multiple streams to same TCP Peer

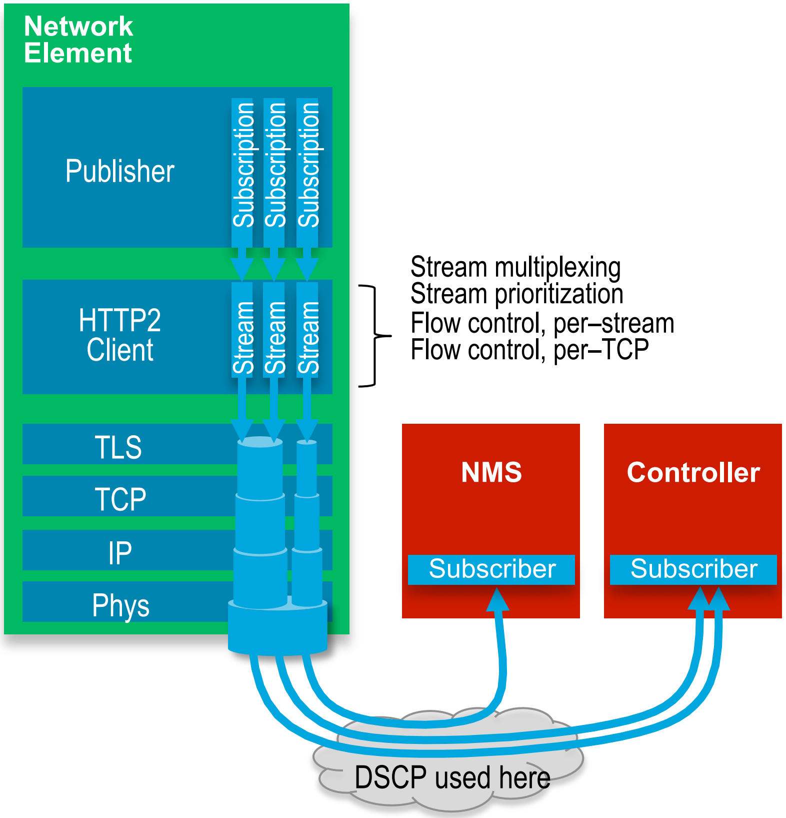
subscription-dependency (string, optional)
points to single parent subscription

Stream Dependency (31bit integer)
preempts the marshalling of updates
for any dependent streams

(re)Transmit frames at rate consumable into destination

Prioritize and rate shape

Dequeue



Stream multiplexing
Stream prioritization
Flow control, per–stream
Flow control, per–TCP