Dampers with Forward Traffic Isolation

Johannes Specht, University of Duisburg-Essen

Introduction

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

ATS (P802.1Qcr)

- Bounded delay, robust, integrated policing

Related work

- Concept known: DJ-Regulators/Dampers
- Bounded delay **and** bounded jitter without global synchronization/[g]PTP
- Challenge: Integrity, Traffic Isolation

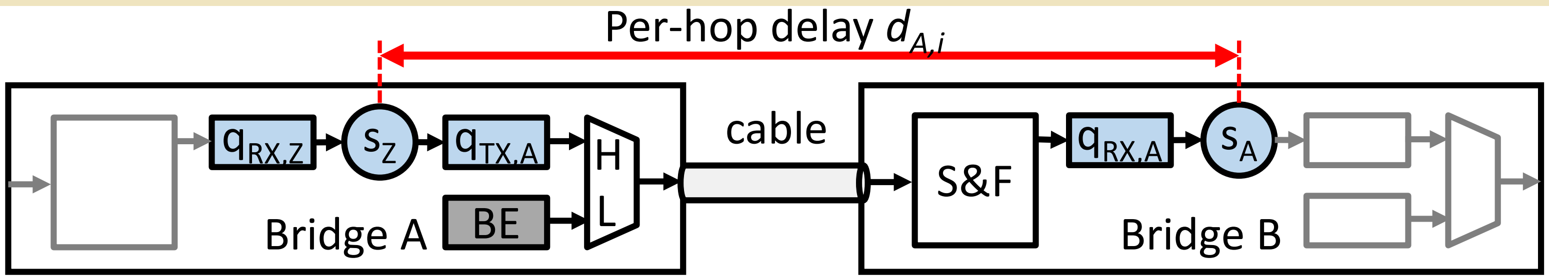
This Slidedeck

- How it works: Rate-based Shaping (ATS) vs. Damping
- Pros and Cons
- Forward Taffic Isolation (new)

No Goal: Let’s do this in P802.1Qcr

Dampers

Initial Assumptions and Simplifications



Symbols

s_k : Shaper with associated with Bridge k
$q_{TX/RX,k}$: FIFO queues associated with Bridge k
$d_{A,i}$: Delay of the ith frame from A (s_Z to s_A)

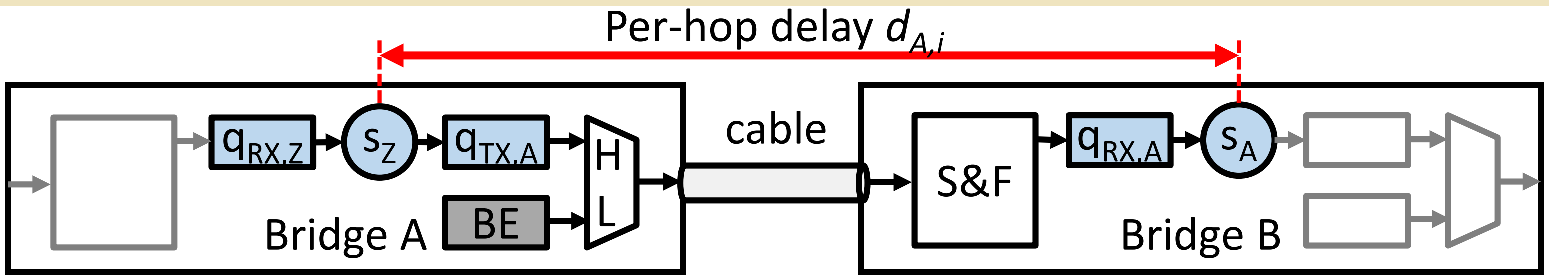
1. **Perfect cables:** No propagation delays
2. **Simple Bridges:** No delays in relays & MACs and cables, no oscillator variations, no numeric imprecision, no gates, no preemption, etc.
3. **Two-level queuing model:** FIFO→shaper→FIFO
4. **Single hop:** Bridge A → Bridge B
5. **Two traffic classes:** Shaped class (High), Best Effort (Low)
6. **Simple traffic:** Periodic small frames, sporadic large best effort frames

Trust me ☺

- Most of these are just to keep subsequent slides simple. E.g., dealing with oscillator variations, numeric imprecision, etc. would just expand math and this slide set.
- Some aspects need further investigation.

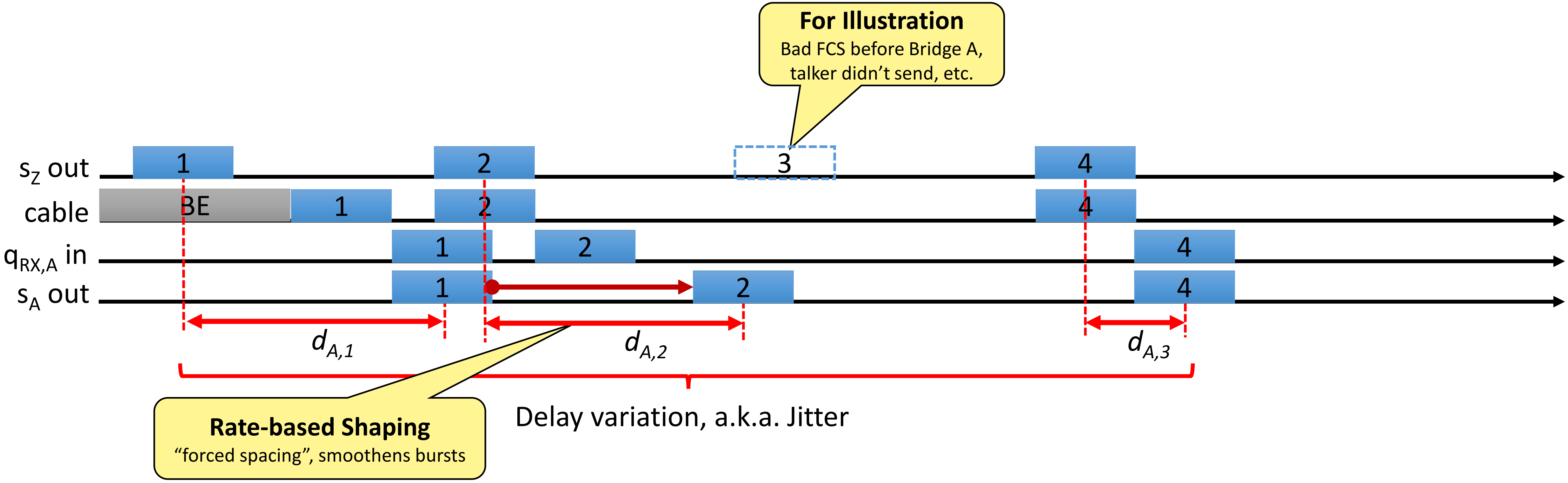
Rate-based Shaping (e.g., P802.1Qcr)

UNIVERSITÄT DUISBURG ESSEN
Open-Minded



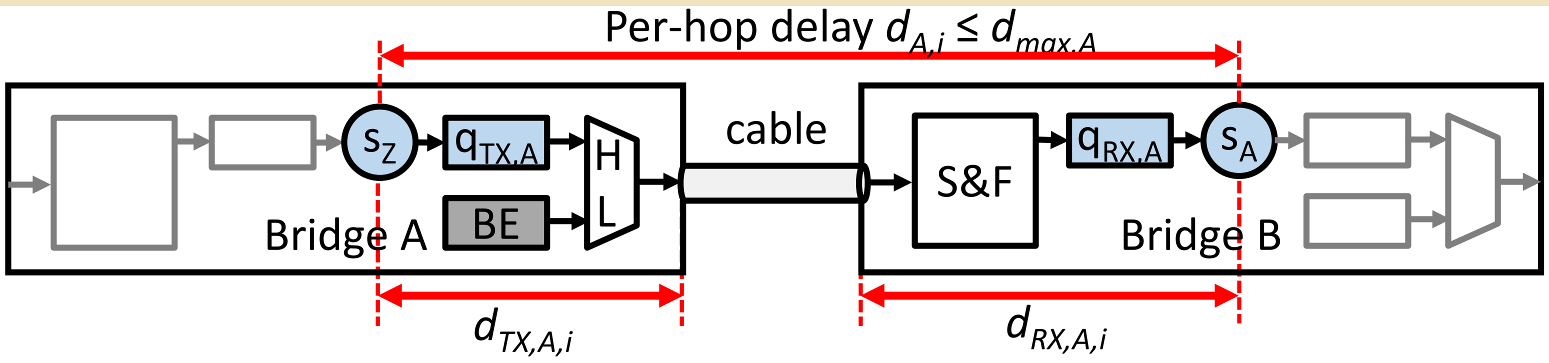
Symbols

s_k : Shaper with associated with Bridge k
$q_{TX/RX,k}$: FIFO queues associated with Bridge k
$d_{A,i}$: Delay of the ith frame from A (s_Z to s_A)



Damping in a Nutshell

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

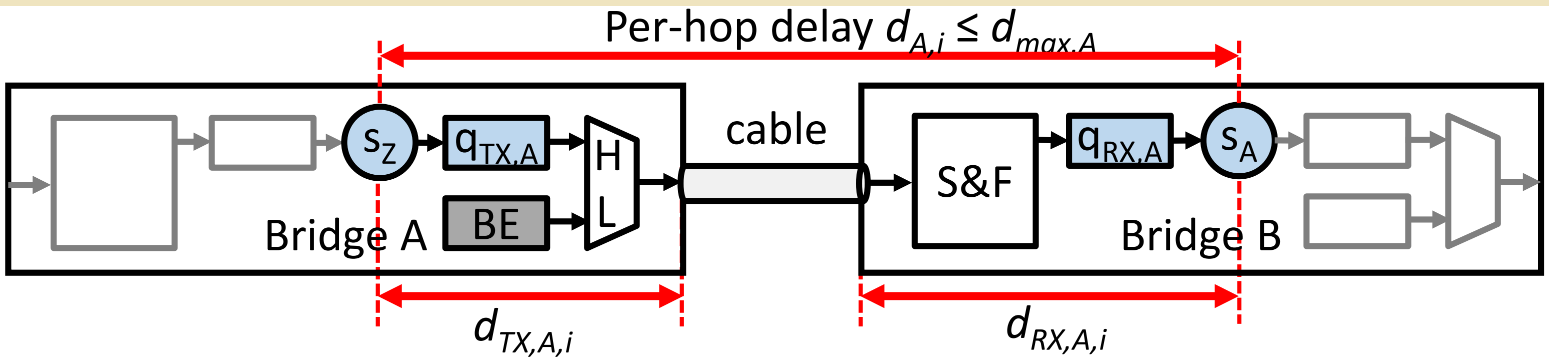


Symbols

- s_k : Shaper with associated with Bridge k
- $q_{TX/RX,k}$: FIFO queues associated with Bridge k
- $d_{A,i}$: Delay of the ith frame from A (s_Z to s_A)
- $d_{max,A}$: Per-hop delay bound for A
- $d_{TX,A,i}$: Residence time in $q_{TX,A}$
- $d_{RX,A,i}$: Residence time in $q_{RX,A}$ and S&F

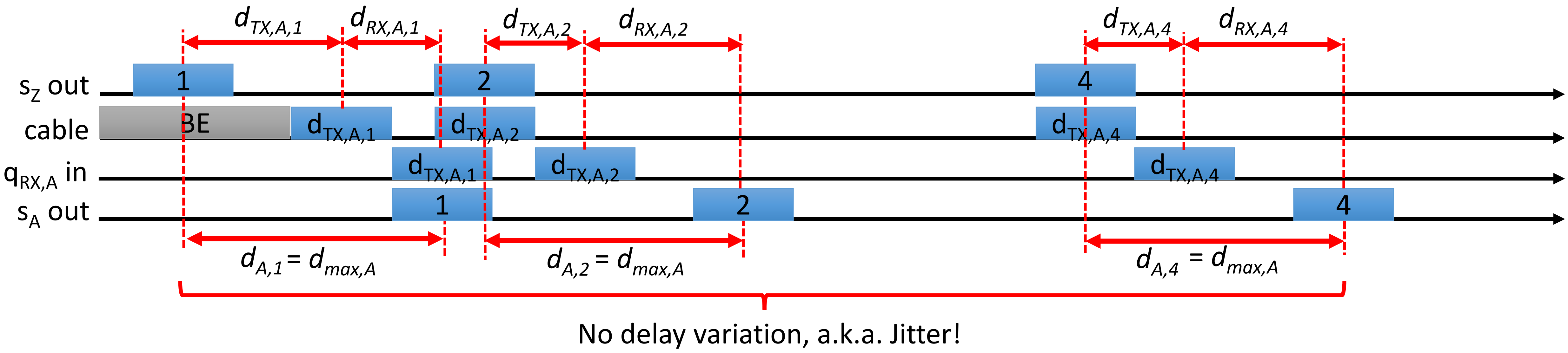
1. **A pre-configured per-hop delay bound $d_{max,k}$**
   - Trust me … again – not too complicated, cmp. ATS
   - Similar to CQF cycle duration – though it can differ per hop
2. **Define $d_{TX,A,i}$ and $d_{RX,A,i}$**
   - $d_{TX,k,i}$: post-shaper residence time in the upstream Bridge/Station
   - $d_{RX,k,i}$: pre-shaper residence time in the downstream Bridge
3. **Transfer $d_{TX,k,i}$ per frame → Dynamic Packet State**
   - Encoding is not the main point here (this is not a Standard!)
   - Data integrity addressed later
4. **Shape differently → Force $d_{RX,k,i} = d_{max,k} - d_{TX,k,i}$**
   - I know, S&F, …, would just add more symbols to my slides (this is not a Standard!)

Damping Illustrated



Symbols

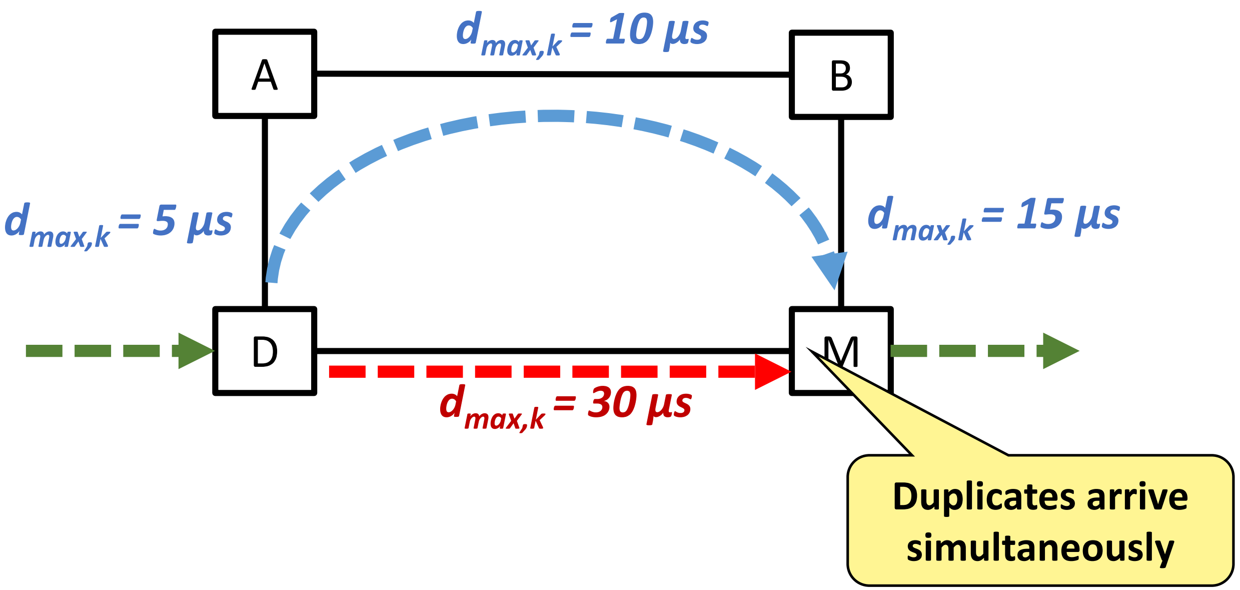
- s_k : Shaper with associated with Bridge k
- $q_{TX/RX,k}$: FIFO queues associated with Bridge k
- $d_{A,i}$: Delay of the i[th] frame from A (s_Z to s_A)
- $d_{max,A}$: Per-hop delay bound for A
- $d_{TX,A,i}$: Residence time in $q_{TX,A}$
- $d_{RX,A,i}$: Residence time in $q_{RX,A}$ and S&F



Example Use-Case: FRER Path-Delay Balancing

Symbols

$d_{max,k}$: Per-hop delay bound for k



Description

- Post-merging burstiness nearly identical to the pre-duplication burstiness
- MatchRecoveryAlgorithm sufficient
- More discussion: https://opus.bibliothek.uni-wuerzburg.de/frontdoor/index/index/docId/20582

Pros and Cons

Pros and Cons

UNIVERSITÄT DUISBURG ESSEN

Open-Minded

Pros

- **Low/no Jitter**
- **No state** (Shaper FSMs): All information in Dynamic Packet State
- Should work with simplified ATS queuing (**“interleaved shaping”**), i.e. no FIFO queue per flow needed.
- [g]PTP **Hardware re-use**

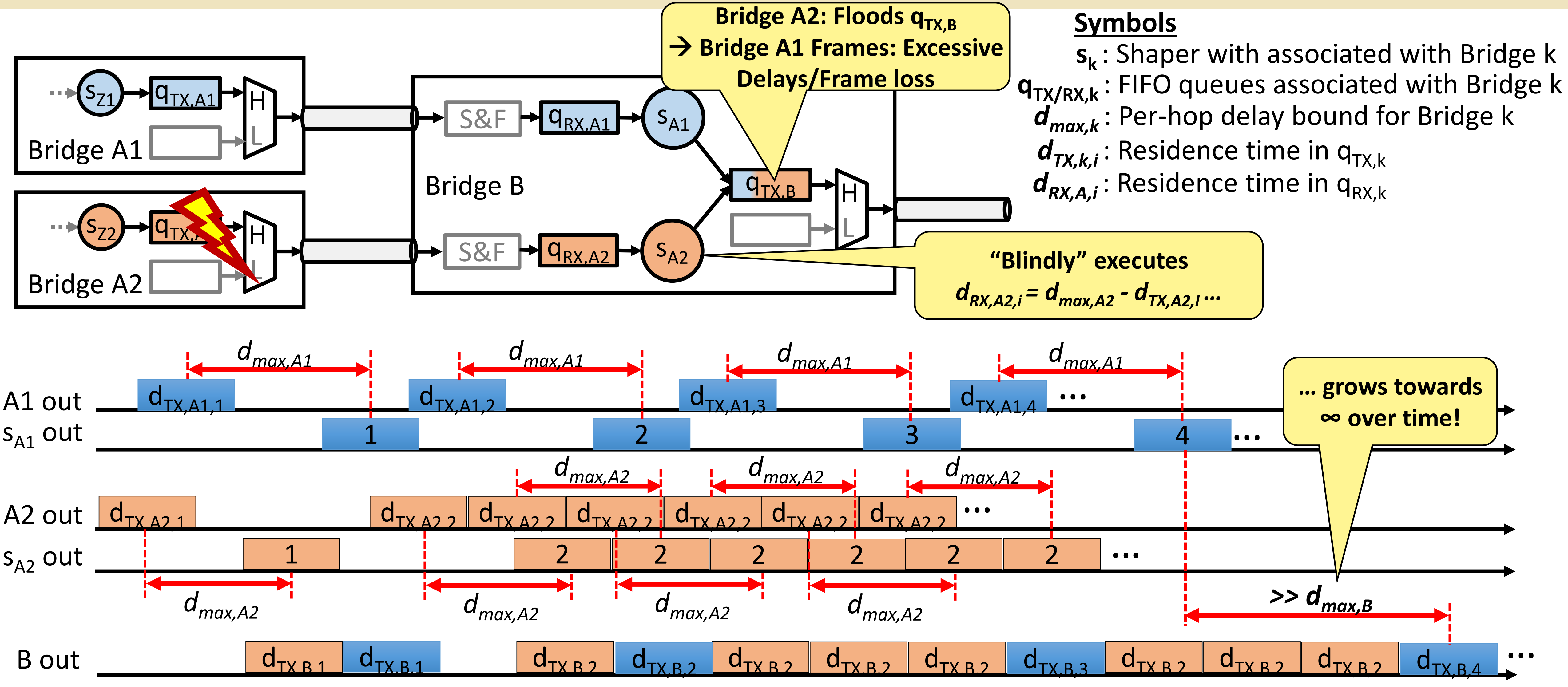
Cons

- **Increased Overhead** for Dynamic Packet State
- FCS re-calculation per Hop required → **Decreased data integrity**
- No state (Shaper FSMs): → **No protection and isolation** against malicious traffic/**babbling idiots**!

Protection & Isolation

Babbling Idiot Impact (e.g., Frame Repetition)

UNIVERSITÄT DUISBURG ESSEN
Open-Minded



Note: No BE frames and S&F delays shown (unnecessary for illustration).

Is this an Issue? – Depends on the Network

Case 1: Conventional Networks

The edge (=Station) is considered problematic, the core (=Bridges) is considered to never fail (or if it does, only fail silent is considered).

→Protection: Edge Bridge Ports only
(i.e., Bridge ports connected to Stations)

→State: Edge Bridge Ports only
(# of Streams from a single Station is limited)

Case 2: Dependable Networks

It doesn't matter whether Station or Bridge. Devices can fail arbitrarily according to their failure rate (MTBF, etc.). And we don't know how (i.e., babbling idiot behavior) …

→ Protection: Every Bridge Port
(no matter whether it's a Station or a Bridge upstream)

→ State: Every Bridge Port in every Bridge
("Per-stream Filtering and Policing" in every Port)

Is this an Issue? – Depends on the Network

Issue Summarized

Faulty cross-traffic disrupts traffic on fault free paths

Goals

1. Protect traffic on fault free paths against faulty cross-traffic
2. 100% protection requires (up to) per-flow state → get close to this level, but with less state

No Goals

Distinguish between faulty and fault free traffic across the same (faulty) bridge

Case 1: C

The ec … does, only fail sile

→Prot

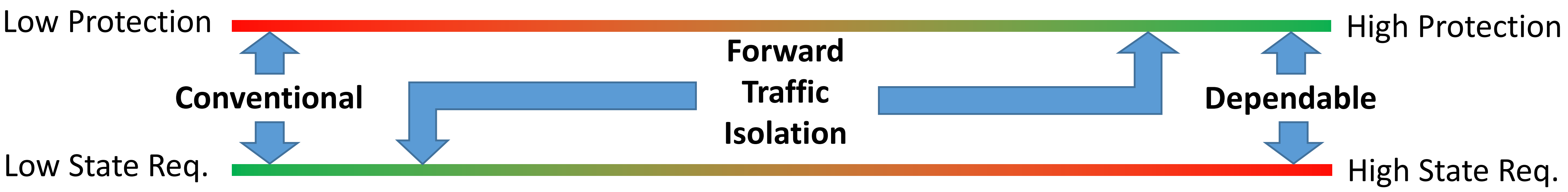
→Stat

Case 2: D

It does … (MTBF, etc.). A

→ Pro

→ Sta



Forward Traffic Isolation (FTI)

Forward Traffic Isolation (FTI) – Key Concepts

UNIVERSITÄT DUISBURG ESSEN

Open-Minded

1. Enhanced PSFP on edges only
   - Enhanced Flow meters ("PSFP+")
   - Max. SDU size filtering
2. Additional Validation Data in Frames
   - Part of Dynamic Packet State (DPS)
3. Exploit Redundant HW on Paths
   - Example: One bridge with 10^{-6} failure/h → two nodes with ~10^{-12} failure/h
   - FTI interleaves along the path – validation data tunneled through the next (potentially faulty) Bridge downstream
4. Validation Data is Signed
   - Asymmetric: Read/verify with public key, modification requires private key
   - Important notes:
     - Signature algorithms against HW faults, not necessarily against intelligent/human attacks
       → less computation, several literature on this topic
       (e.g., *K. Echtle and T. Kimmeskamp, Fault-Tolerant and Fail-Safe Control Systems Using Remote Redundancy, 22th International Conference on Architecture of Computing Systems 2009)*
     - Symmetric signatures (e.g., CRCs) are possible, but with more DPS and "clever" key distribution
       → subsequent slides stick to asymmetric concepts

Failure Assumptions

UNIVERSITÄT DUISBURG ESSEN

Open-Minded

1. **One "Box" fails at a time**

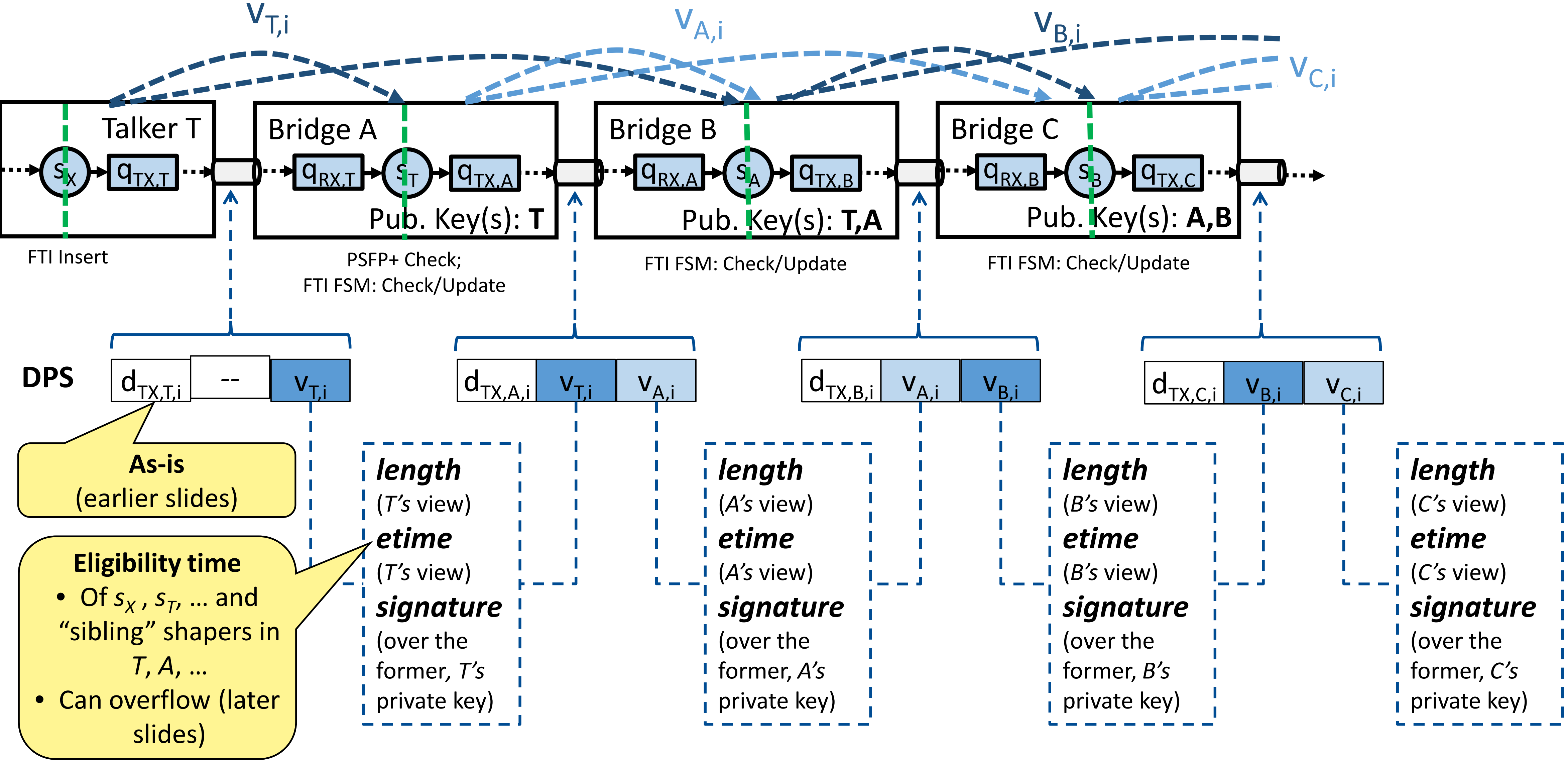
   We can support more, but this one is simple and enough for illustration, plus system failure probability already goes notably lower.

2. **A faulty box cannot find out the private key another fault free box**

   A faulty box has a private key, but this is different then the private keys of its upstream neighbors 1 and 2 hops upwards. It cannot "find out" the other boxes' private key by e.g. random hardware faults.

FTI - Keys, Roles, Dynamic Packet State (DPS)

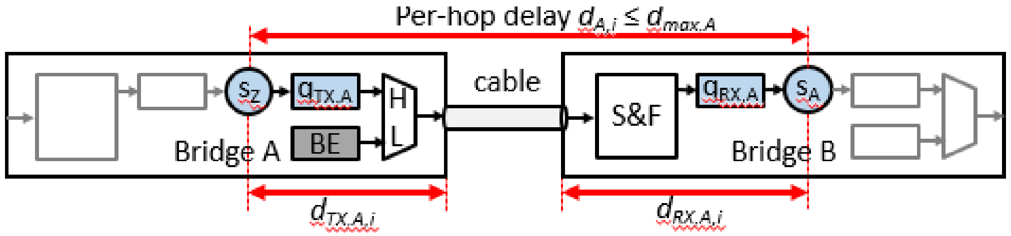
UNIVERSITÄT DUISBURG ESSEN
Open-Minded



First Bridge: Enhanced Flow Meters for Dampers (“PSFP+”)

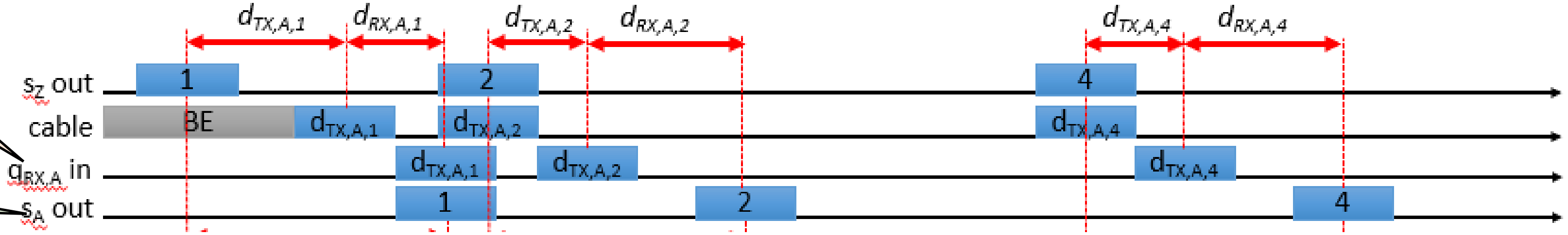
Arrival times of conventional flow meters

(Virtual) Arrival times of enhanced flow meters for dampers



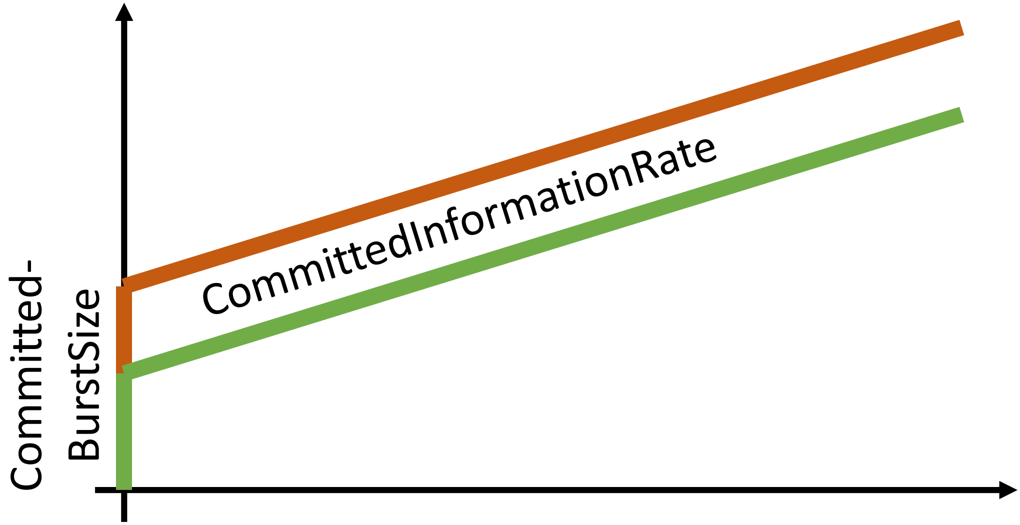
Symbols

s_k : Shaper with associated with Bridge k
$q_{TX/RX,k}$: FIFO queues associated with Bridge k
$d_{A,i}$: Delay of the i[th] frame from A (s_Z to s_A)
$d_{max,A}$: Per-hop delay bound for A
$d_{TX,A,i}$: Residence time in $q_{TX,A}$
$d_{RX,A,i}$: Residence time in $q_{RX,A}$ and S&F



Summary

- Flow meters algorithm executed after computation $\boldsymbol{d_{RX,k,i} = d_{max,k} - d_{TX,k,i}}$
- Eliminates jitter after shaper output
- Flattens the arrival curve (i.e., lower CommittedBurstSize values possible)



FTI: Simplified FSM in Bridges

UNIVERSITÄT DUISBURG ESSEN

Open-Minded

UNALIGNED

```
rxFrameEna = FALSE;
aligned = FALSE;
```

rxFrameEna →

ALIGNED

```
rxFrameEna = FALSE;

/* Check the signatures */
if (!isSigValid(frame.v[0]) break;
If (frame.v[1] != null && !isSigValid(frame.v[1])) break;

/* Check the frame lengths */
if (!isLengthValid(frame.length,frame.v[0].length) break;
if (frame.v[1] != null && !isLengthValid(frame.length,frame.v[1].length) break;

etime = d_max – frame.d;

/* Align, if unaligned */
If (!aligned) {
  tLastFrame = now;
  delta0 = frame.v[0].etime – etime;
  if (frame.v[1] != null) delta1 = frame.v[1].etime – etime;
  aligned = TRUE;
  proceed(frame,etime);
  break;
}

/* Check eligibility time consistency */
if (delta0 != frame.v[0].etime – etime) break;
if (frame.v[1] != null) && (delta1 != frame.v[1].etime – etime) break;

/* Update timer, forward the frame (i.e., not discarded) */
tLastFrame = now;
proceed(frame, etime);
```

rxFrameEna

aligned &&
now > (tLastFrame + etimeOverrunTimeout)

Note: Preliminary/untested

d	v[1]	v[0]

```
/* Transient */
Frame frame;
Time etime;
Time now;

/* Scope: Per-Rx-Port */
Boolean rxFrameEna;

/* Scope: Per-RX-Port-per-Class */
Boolean aligned;
Time tLastFrame;
Time delta0;

/* Scope: Per-Upstream-RX-Port-per-Class */
Time delta1;

/* Helpers */
void forward(Frame frame); // Forwards the frame
Signature sign(Frame frame); //Generates a local signature
Boolean isLengthValid(int l1,l2); // Length consistency check
Boolean isSigValid(Validation v); // Validates a signature

void proceed(Time etime, Frame frame) {
  frame.v[1] = frame.v[0];
  frame.v[0].etime;
  frame.v[0].length = frame.length;
  frame.v[0].signature =
    sign(frame.v[0].etime, frame.v[0].length);
  forward(frame);
}
```

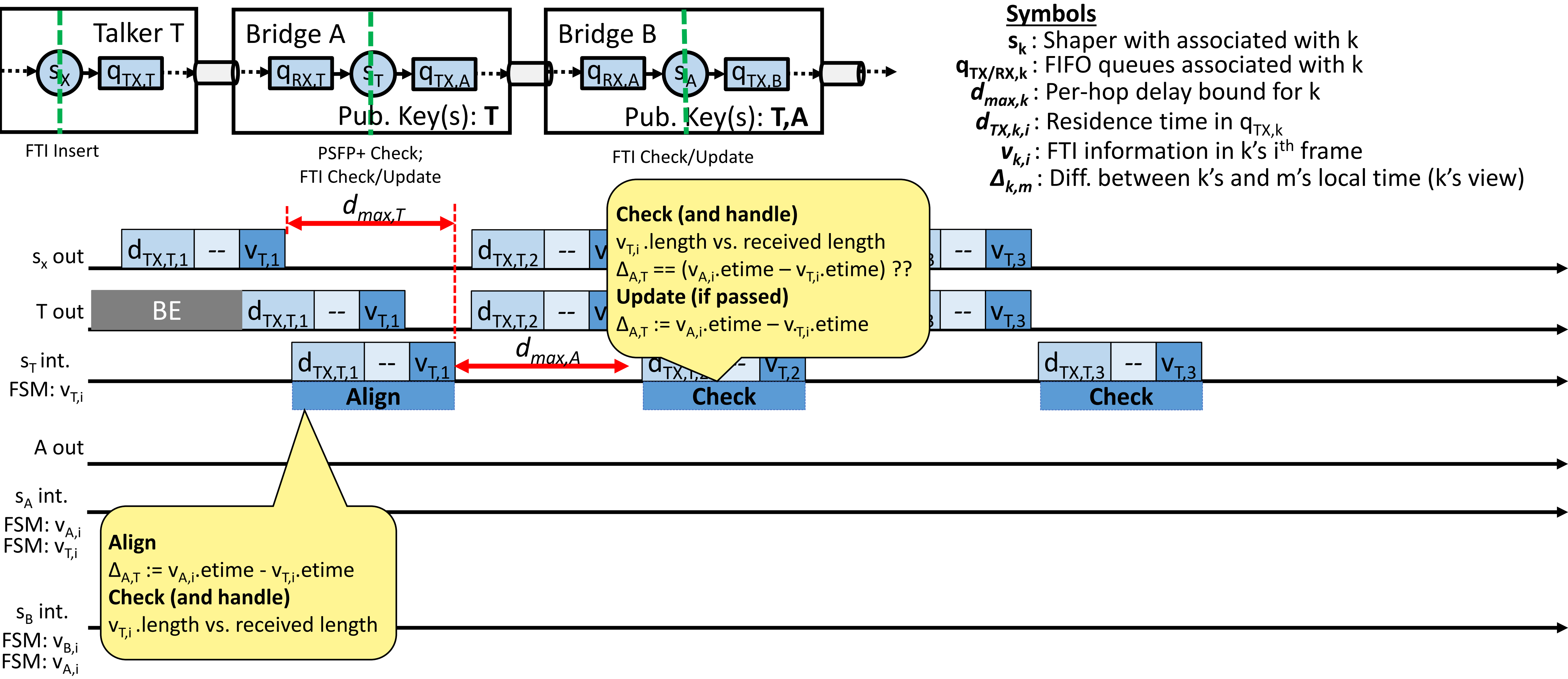
Simplified
Lookups/tables not explicitly shown

Local Device Size
Table size depends on the local device's dimensioning

Limited Topology Dependency
Like for public keys, the table size here depends on the number of dual-hop upstream devices (see later slides)

Simplified
Tolerances for imprecisions (oscillators, limited numeric resolutions, etc.) omitted

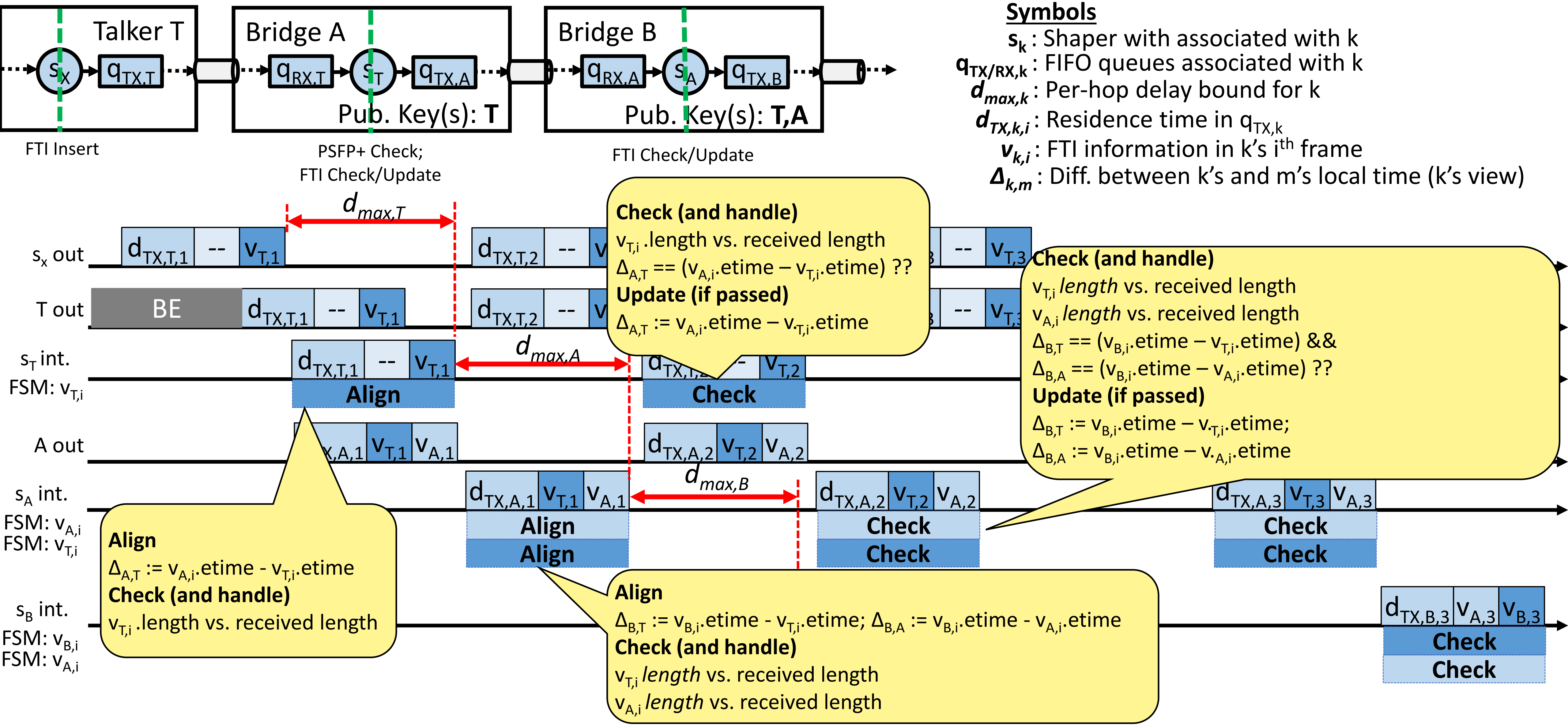
FTI – Illustration and FSMs



FTI – Illustration and FSMs

Symbols

- s_k : Shaper with associated with k
- $q_{TX/RX,k}$: FIFO queues associated with k
- $d_{max,k}$: Per-hop delay bound for k
- $d_{TX,k,i}$: Residence time in $q_{TX,k}$
- $v_{k,i}$: FTI information in k's i[th] frame
- $\Delta_{k,m}$: Diff. between k's and m's local time (k's view)

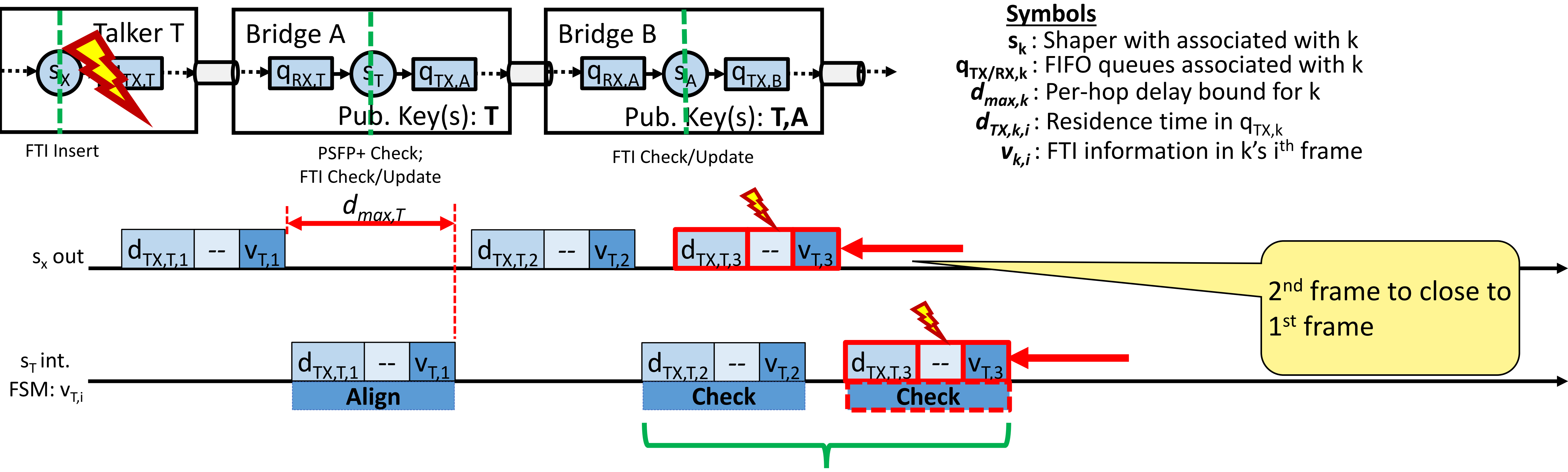


Some Failure Scenarios for Illustration

Goal: Capture malicious traffic immediately after the faulty device. Merge point not shown subsequently, though capturing immediately after the faulty is enough.

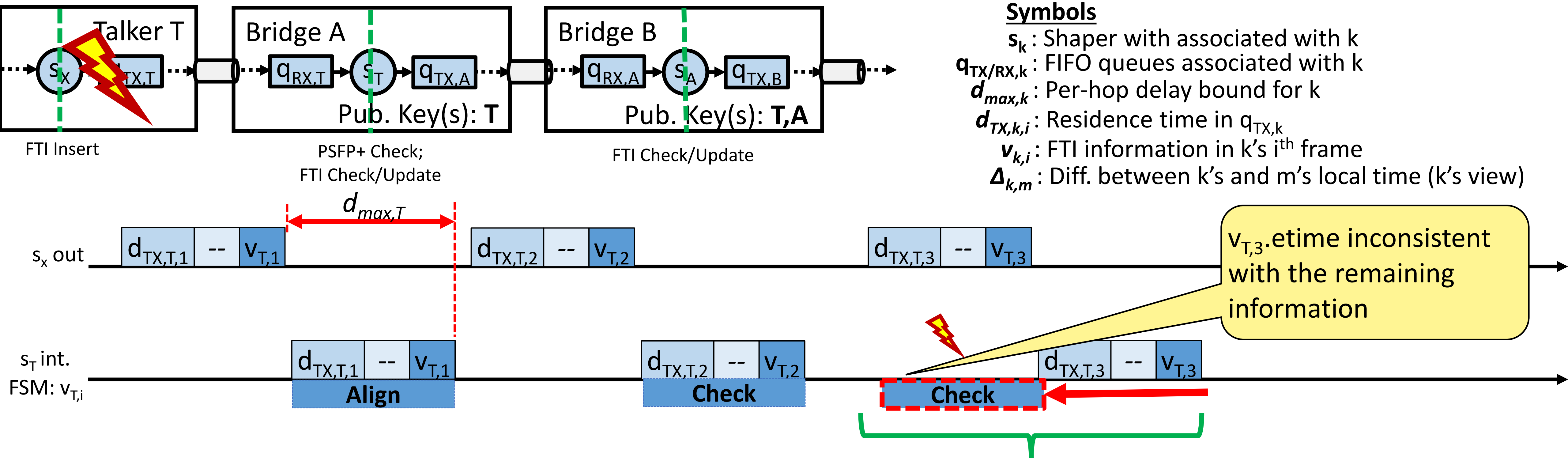
Note: Compared to earlier slides, the blue path contains the faulty node.

FTI – Faulty T, excessive burst



→ **Caught by PSFP+, Committed Burst Size exceeded!**

FTI – Faulty T, bad etime in $v_{T,i}$

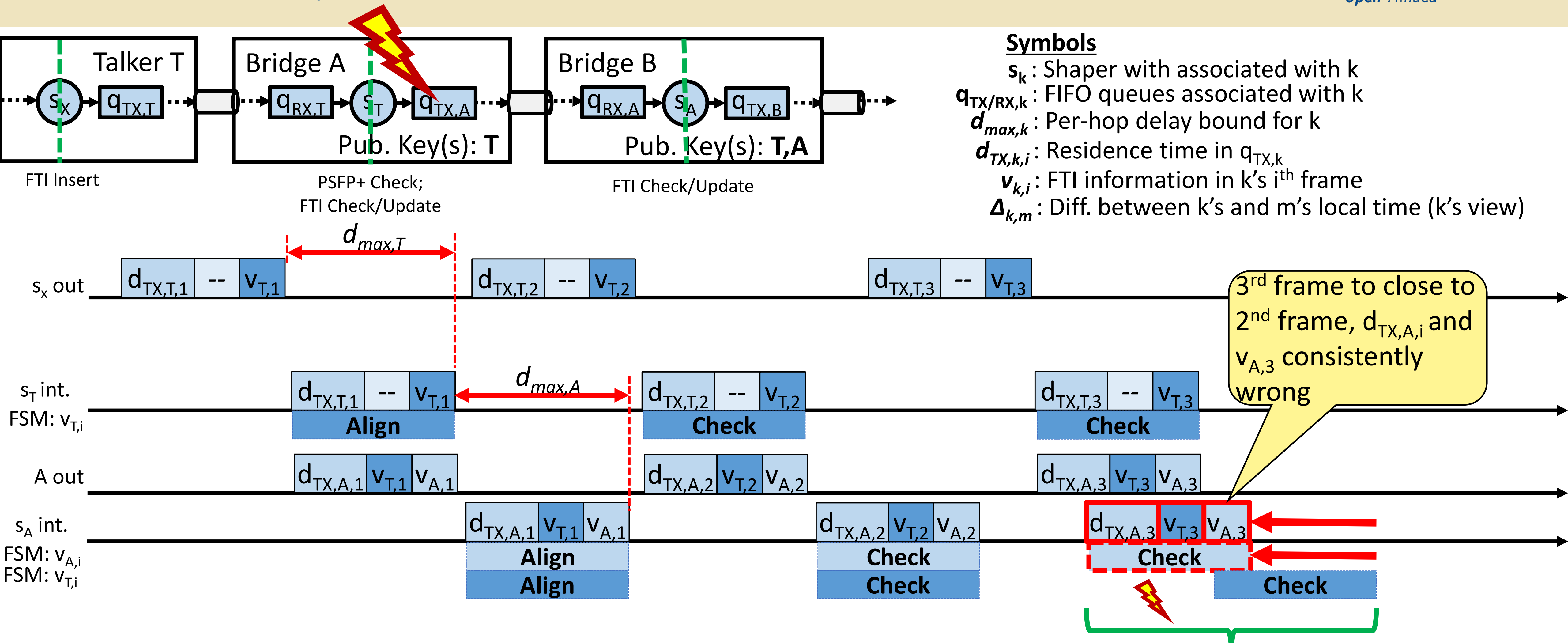


Symbols

- s_k : Shaper with associated with k
- $q_{TX/RX,k}$: FIFO queues associated with k
- $d_{max,k}$: Per-hop delay bound for k
- $d_{TX,k,i}$: Residence time in $q_{TX,k}$
- $v_{k,i}$: FTI information in k's i^{th} frame
- $\Delta_{k,m}$: Diff. between k's and m's local time (k's view)

→ Caught by FTI Check, T's offset $\Delta_{A,T}$ known by A!

FTI – Faulty A, excessive burst



Symbols

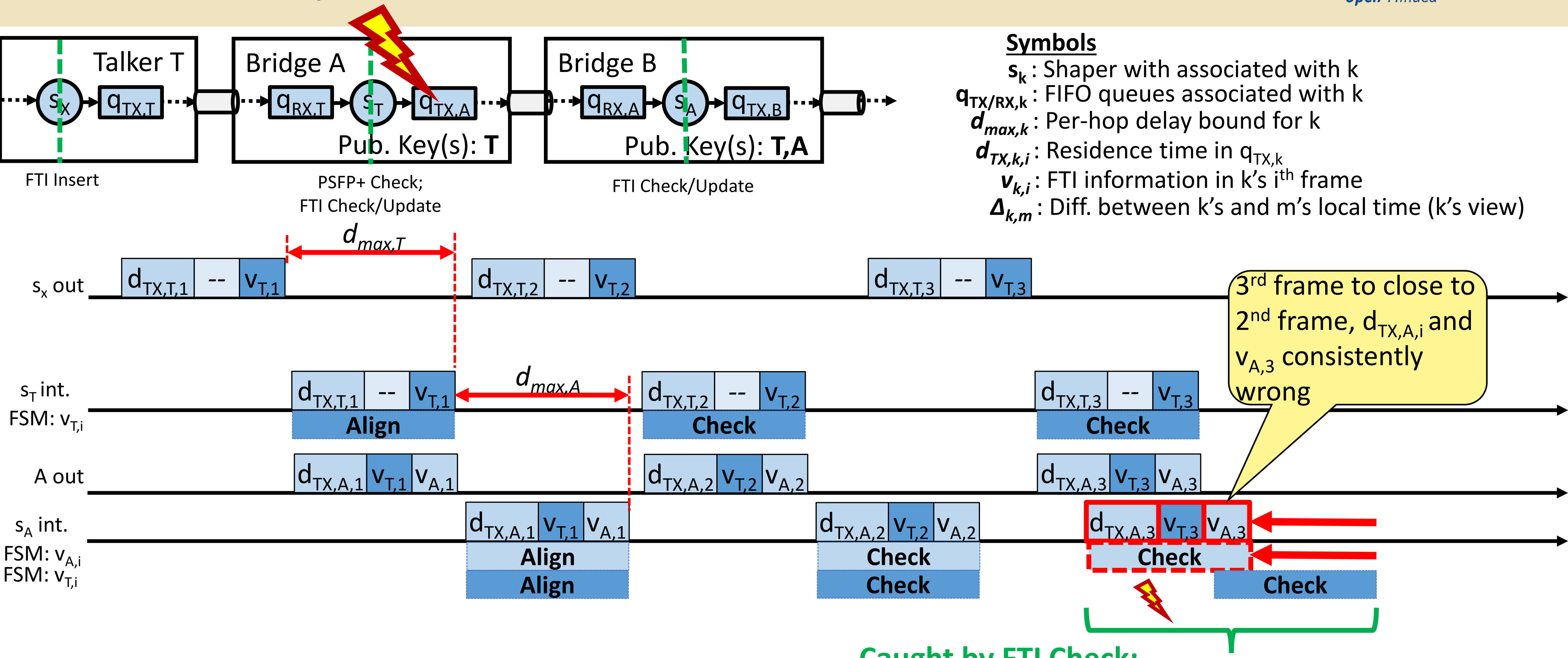
- s_k : Shaper with associated with k
- $q_{TX/RX,k}$: FIFO queues associated with k
- $d_{max,k}$: Per-hop delay bound for k
- $d_{TX,k,i}$: Residence time in $q_{TX,k}$
- $v_{k,i}$: FTI information in k's i^{th} frame
- $\Delta_{k,m}$: Diff. between k's and m's local time (k's view)

Caught by FTI Check:

→ **T's offset $\Delta_{B,T}$ known by B!**

→ **T's $v_{T,i}$ values cannot be "faked" by A!**

FTI – Faulty A, excessive burst



Symbols

- s_k : Shaper with associated with k
- $q_{TX/RX,k}$: FIFO queues associated with k
- $d_{max,k}$: Per-hop delay bound for k
- $d_{TX,k,i}$: Residence time in $q_{TX,k}$
- $v_{k,i}$: FTI information in k's i^{th} frame
- $\Delta_{k,m}$: Diff. between k's and m's local time (k's view)

In fact, it doesn't matter whether A's frame is too early or too late. A cannot "fake" T's $v_{T,I}$ information, $v_{T,i}$.etime in particular. A does not know T's private key. Same for $v_{T,i}$.length (not illustrated).

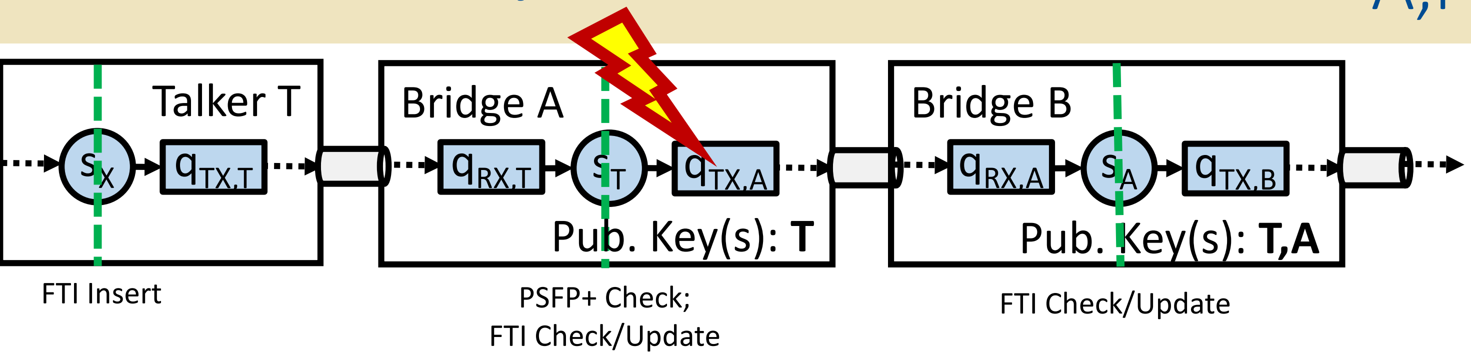
Caught by FTI Check:

→ T's offset $\Delta_{B,T}$ known by B!

→ T's $v_{T,i}$ values cannot be "faked" by A!

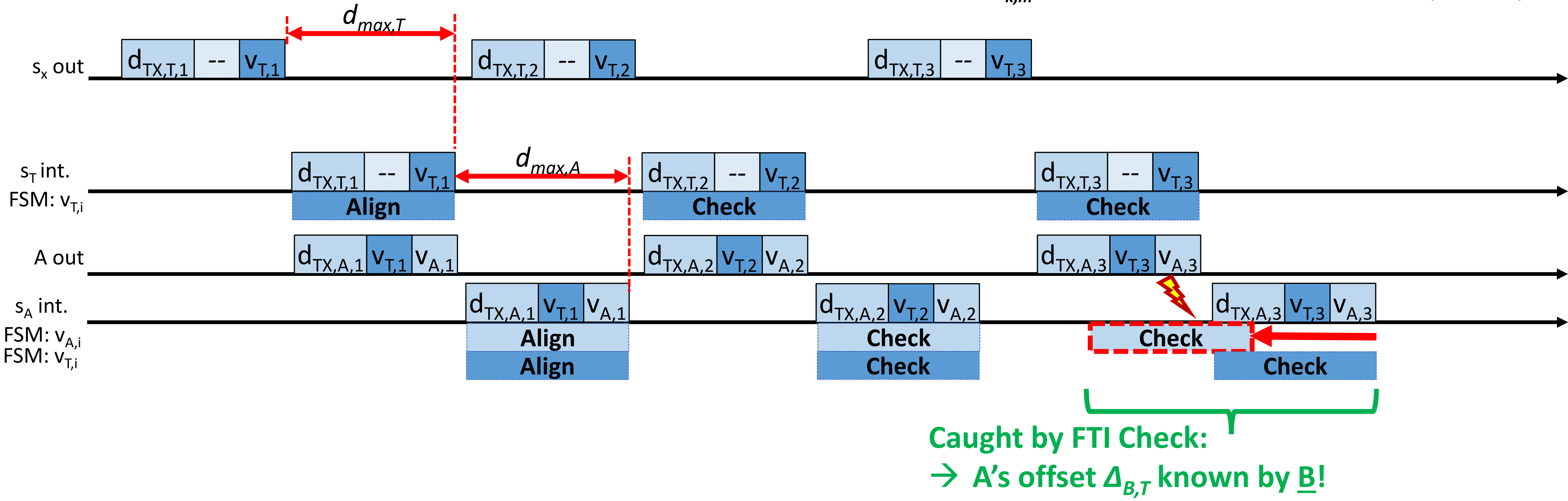
FTI – Faulty A, bad etime in $v_{A,i}$

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

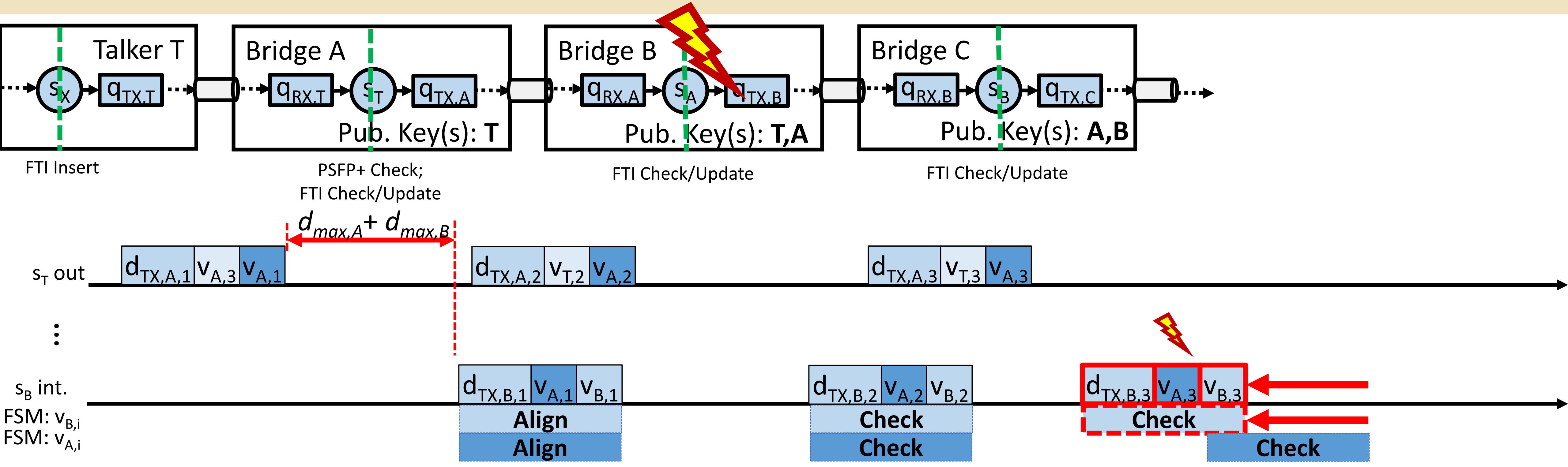


Symbols

- s_k : Shaper with associated with k
- $q_{TX/RX,k}$: FIFO queues associated with k
- $d_{max,k}$: Per-hop delay bound for k
- $d_{TX,k,i}$: Residence time in $q_{TX,k}$
- $v_{k,i}$: FTI information in k's ith frame
- $\Delta_{k,m}$: Diff. between k's and m's local time (k's view)



FTI – Faulty B, excessive burst



Caught by FTI Check:
→ **A's offset $\Delta_{C,A}$ known by C!**

Symbols

- s_k : Shaper with associated with k
- $q_{TX/RX,k}$: FIFO queues associated with k
- $d_{max,k}$: Per-hop delay bound for k
- $d_{TX,k,i}$: Residence time in $q_{TX,k}$
- $v_{k,i}$: FTI information in k's i^{th} frame
- $\Delta_{k,m}$: Diff. between k's and m's local time (k's view)

Note: Case just to simplify illustration how FTI operates along the path

Further Aspects

Not shown in earlier slides

Keys

UNIVERSITÄT DUISBURG ESSEN

Open-Minded

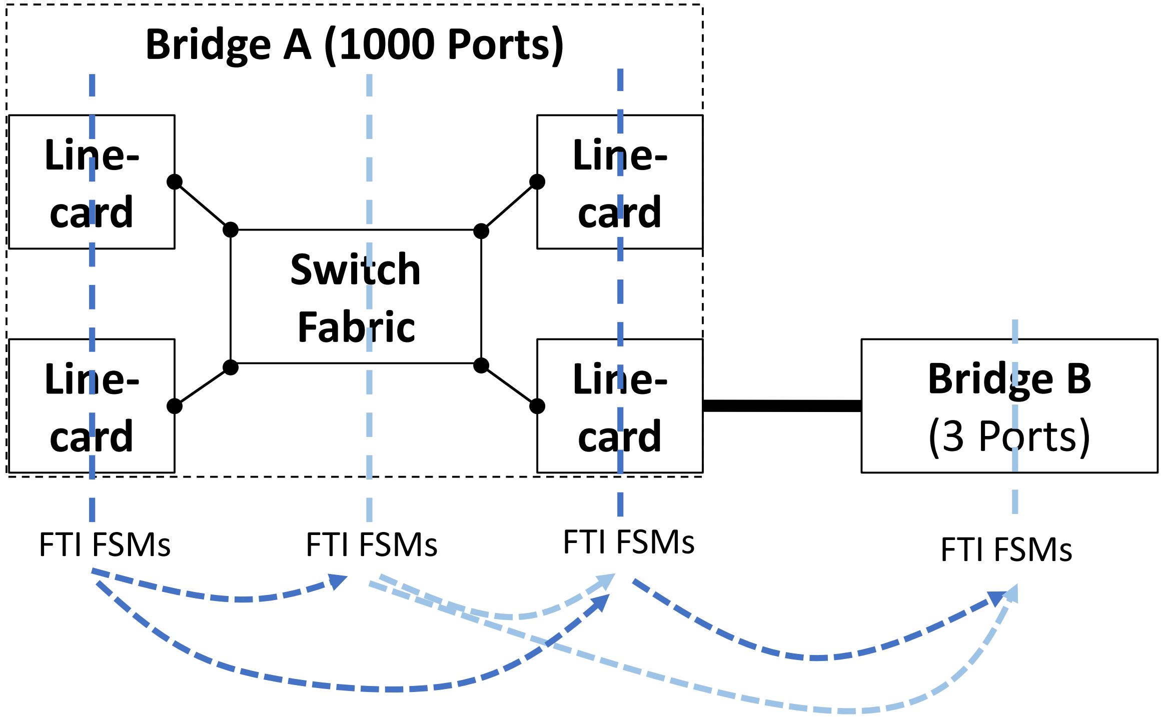
Public Key Distribution

Either static, or via a protocol. A protocol has not been presented, though this is not so critical, given it is the slow, not so critical, path (control plane).

Public Key Identification/Lookup

On frame reception, the associated public key for $v_{k,i}$ values must be identified. This aspect wasn't covered, though it can be an extra field of $v_{k,i}$ not covered by the signature (think of the following: If a faulty node in the middle "fakes" this field, a wrong public key is selected and signature check fails).

Reducing State Requirements



Dual-hop Upstream State

Consider Bridge A has 1000 ports, connected 999 Talkers, and to Bridge B downstream, which is a small 3 Port Bridge. B would require 1000 $\Delta_{k,m}$ state variables just to serve these 1000 talkers. However, Bridge A will comprise multiple Chips, ASICs, etc. which can reasonably independent from each other in terms of reliability. There can be multiple FTI check and update points with associated FSMs in Bridge A (e.g., one per ASIC), thus massively reducing the required $\Delta_{k,m}$ state variables in Bridge B (i.e., think of every ASIC in Bridge A is a Bridge itself).

Various

UNIVERSITÄT DUISBURG ESSEN

Open-Minded

$v_{k,i}$.etime Overflows and Timeouts

Each FSMs times out if the time range of $v_{k,i}$.etime is exceeded. The FSMs then fall back to unaligned state. A faulty node can exploit this, however, it can at most send one bad frame per time range. The resulting maximum “noise” caused by such a node consumes considerable low bandwidth - appears ok for worst-case consideration.

Missing Frames

Due to FCS errors, different routing, etc. a frame sequence upstream can be incomplete at the next two hops downstream. This is no issue, the exact sequence can contain gaps. It’s just $\Delta_{k,m}$ state variables that are updated less frequently.

FTI in other Areas

Though dampers provide higher delay-performance, there is e.g. a DPS-based asynchronous Cyclic Queueing and Forwarding derivate (https://datatracker.ietf.org/doc/draft-qiang-detnet-large-scale-detnet/). FTI can be applied here, too, just think of eligibility times with “low resolution” (i.e., cycle numbers).

Summary

Dampers

- Low jitter asynchronous traffic shaping
- Stateless in Bridges
- Dynamic Packet State is used → Integrity is an Issue

Forward Traffic Isolation

- New concept for traffic isolating against babbling idiots
- No 100% solution - residual errors hard to quantify – but qualitatively high degree of protection from an engineers point of view
- Moderate state requirements (i.e., topology dependent, limited to two hops) – typically significantly lower than per flow state
- Scheme applicable in other Areas

Thank you for your Attention!

Open-Minded

Questions, Opinions, Ideas?

Johannes Specht
Dipl.-Inform. (FH)

Dependability of Computing Systems
Institute for Computer Science and
Business Information Systems (ICB)
Faculty of Economics and
Business Administration
University of Duisburg-Essen

Schuetzenbahn 70
Room SH 502
45127 Essen
GERMANY
T +49 (0)201 183-3914
F +49 (0)201 183-4573

Johannes.Specht@uni-due.de
http://dc.uni-due.de