Bundle In Bundle Encapsulation (BIBE)

Scott Burleigh

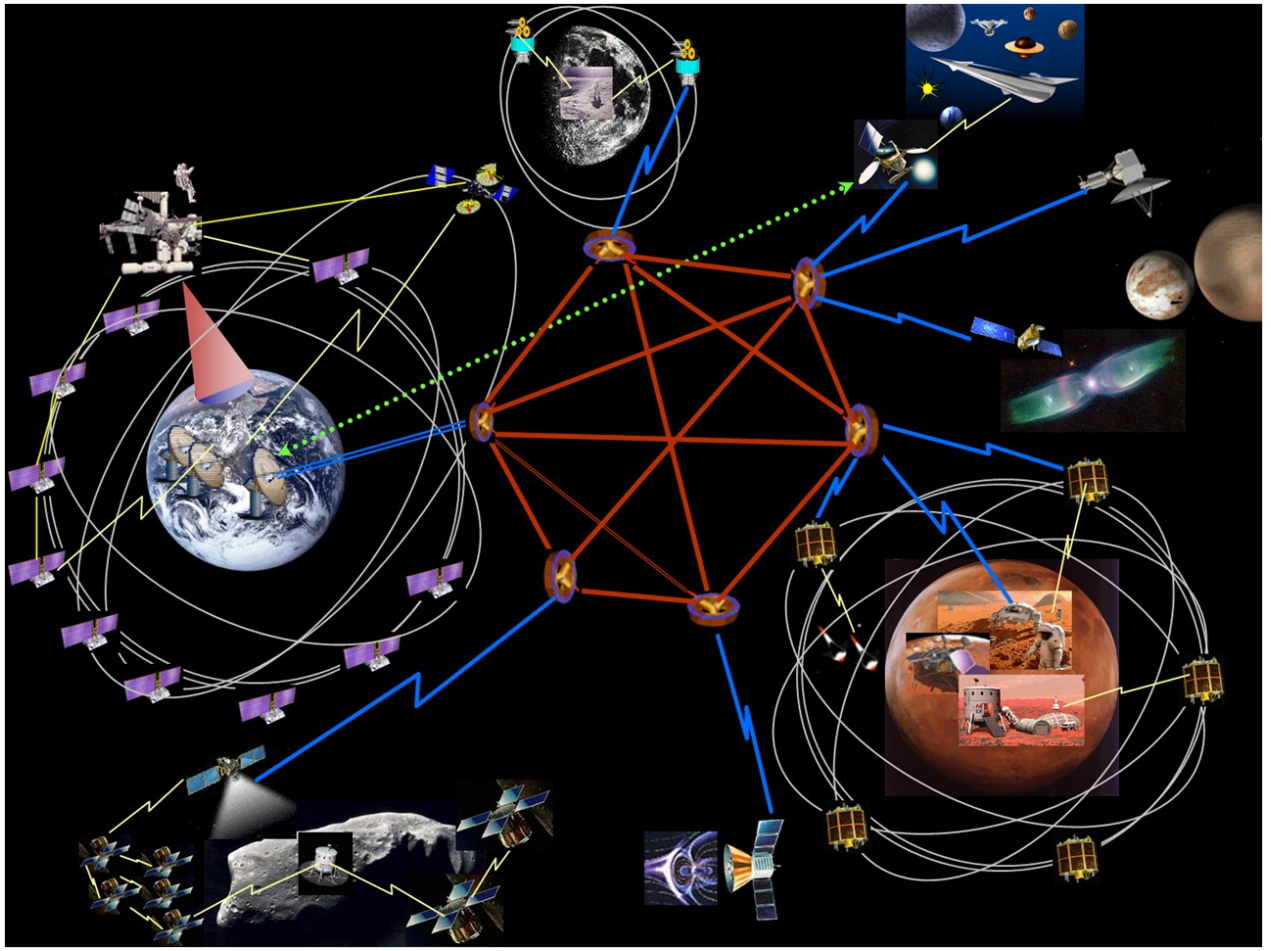
IPN Group

22 March 2022

(originally presented to IETF in Spring of 2021)

Overview

- A short history of BIBE
- Aggregate custody signaling
- Current BIBE design
- Applications
- Future



A short history of BIBE

The original BIBE specification

- Posted as draft-irtf-dtnrg-bundle-encapsulation-06 in August of 2009; authors were Susan Symington, Bob Durst, and Keith Scott of the MITRE Corporation.
- Conceived as a capability of the BP node's application agent – that is, as a BP *application*.
- Motivations:
  - Support for content-centric networking (forwarding of cached bundles).
  - Efficient (targeted) custodial retransmission of multicast bundles.
  - Security tunneling, particularly defense against traffic analysis.

BIBE at the convergence layer

- BIBE idea resurrected in 2013, posted as draft-irtf-burleigh-bibe-00 in March of 2013; author was Scott Burleigh.
- Conceived as a convergence-layer protocol under BP.
- Motivation: a way of helping to disentangle routing from security. The BIBE tunnel takes the place of the “security source” and “security destination” features of the original Bundle Security Protocol specification.

Custody transfer and unidirectional links

- Much discussion of custody transfer in 2015 and 2016:
  - In the general case, custody transfer with custodial retransmission could not be made efficient:
    - Accurate estimation of round-trip time was – in the general case – not possible.
    - Bundle fragmentation by non-custodians could not be prevented and would always defeat custody transfer.
  - BUT in some deployment scenarios, especially those including unidirectional links, a delay-tolerant asymmetric acknowledgment mechanism is needed. Bundle protocol is the obvious choice.

Extracting custody transfer from BP

- Proposition: all BP transmission reliability should be accomplished between neighboring BP nodes, i.e., at the convergence layer.
- So do custody transfer at the convergence layer; that is, use BP as a convergence-layer protocol.
- Wait a minute….BIBE already does exactly that.
- So let’s just build custody transfer into BIBE and use BIBE for both purposes, independently or together:
  - Cross-domain security. (Security sources and destinations, and defense against traffic analysis.)
  - Reliable convergence-layer transmission over asymmetric paths.

Aggregate custody signaling

- Separately, in 2012 researchers at University of Colorado, Boulder, had designed an alternative, more bandwidth-efficient definition of custody transfer. Documented in draft-kuzminsky-aggregate-custody-signals-04 (not posted). Authors were Sebastian Kuzminsky and Andrew Jenkins.
- Conceived as an alternative administrative record plus an additional extension block in BP. Implemented as an option in ION.
- Motivation: enable custodial retransmission to be used for reliable BP communications with the International Space Station (ISS) over extremely asymmetrical link data rates.

The new BIBE specification

- ACS has been highly successful in ISS operational use of DTN, strongly endorsed by that user community.
- The result of merging the ACS concept into BIBE, replacing the original BP custody transfer design, is draft-burleigh-dtn-bibect-01, posted 20 May 2018, author Scott Burleigh.
  - Operates as an optionally reliable convergence-layer protocol under BP.
  - Encapsulated bundle (the payload of the encapsulating [convergence-layer] bundle) may be encrypted and/or signed.

A reliable convergence-layer protocol

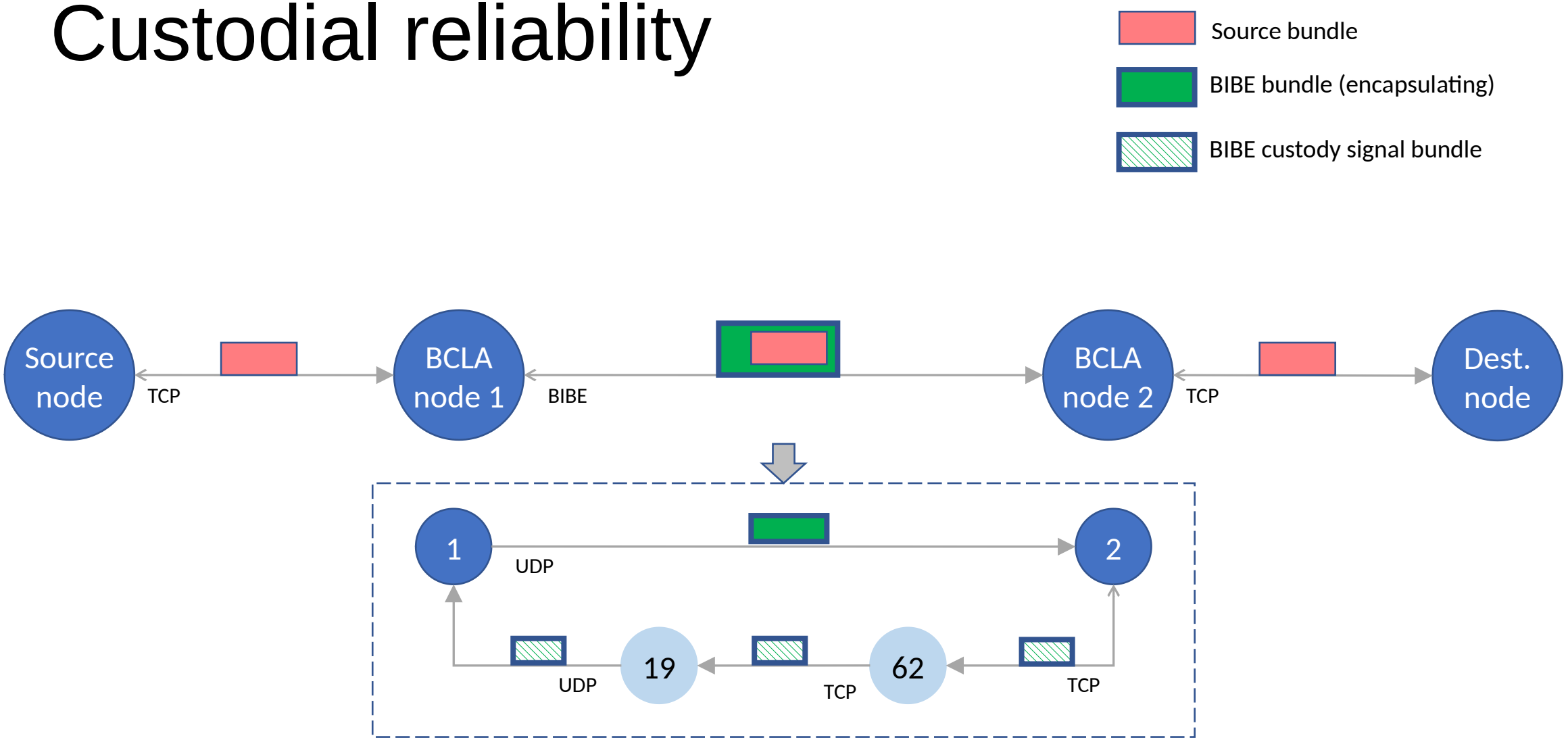
- Payload of the encapsulating bundle comprises:
  - Transmission ID (zero if custody transfer is not requested).
  - Expected time of acknowledgment (zero if custody transfer is not requested).
  - Encapsulated bundle.
- Acknowledgment of the encapsulating bundle is aggregated into a new administrative record sent in a responding bundle.
  - Custody transfer disposition code (“custody accepted” or reason for refusal).
  - Sequences of consecutive transmission IDs of received bundles.
- If acknowledgement is not received by the expected time, transmission of the encapsulated bundle is assumed to have failed; the encapsulated bundle is queued to be re-forwarded.

A recently identified issue

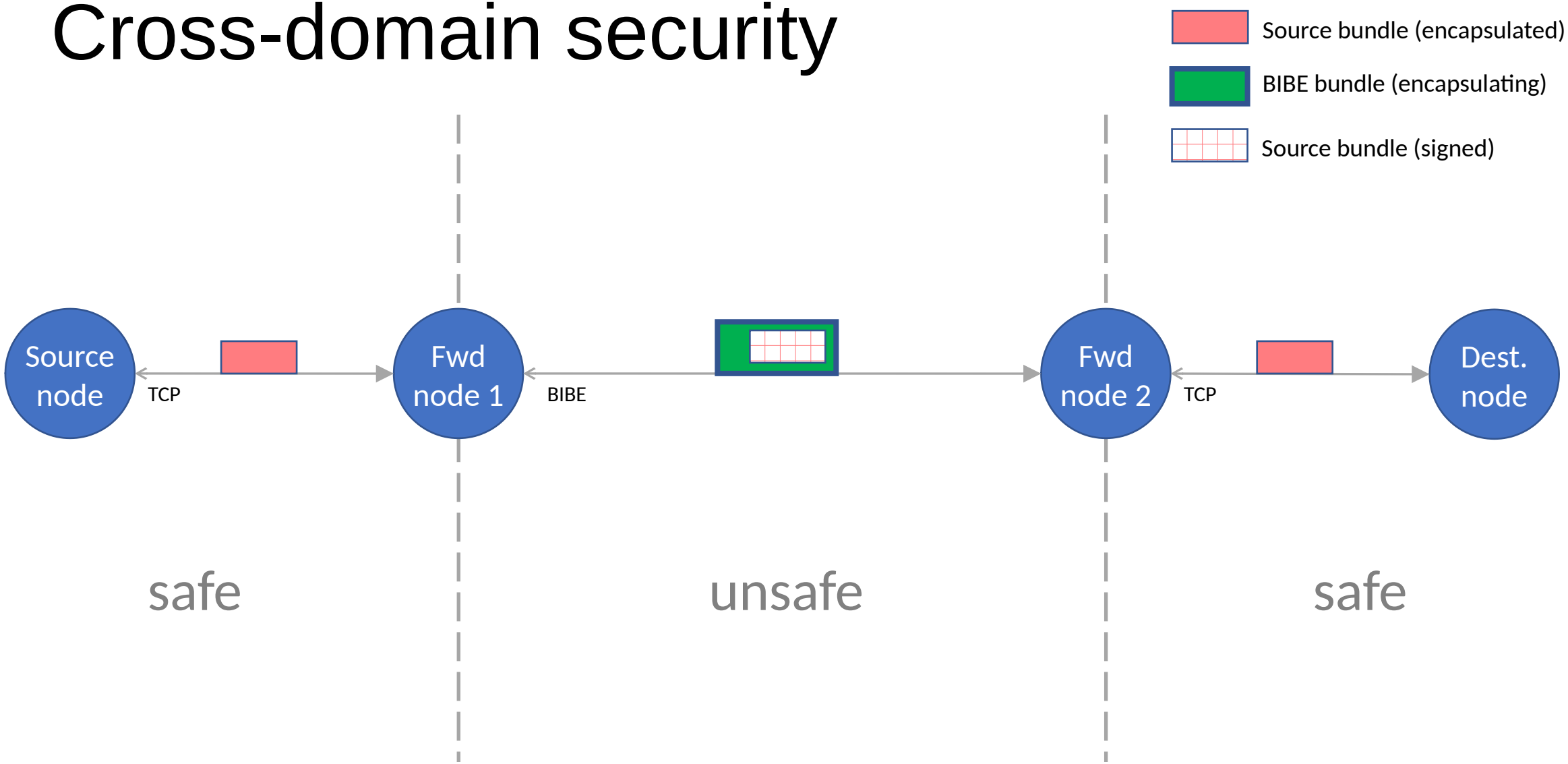
- What if one or both of the participating nodes lack accurate clocks?
  - How does the sender compute or otherwise express the expected time of acknowledgment?
  - How does the receiver parse and utilize the expected time of acknowledgment?

Applications

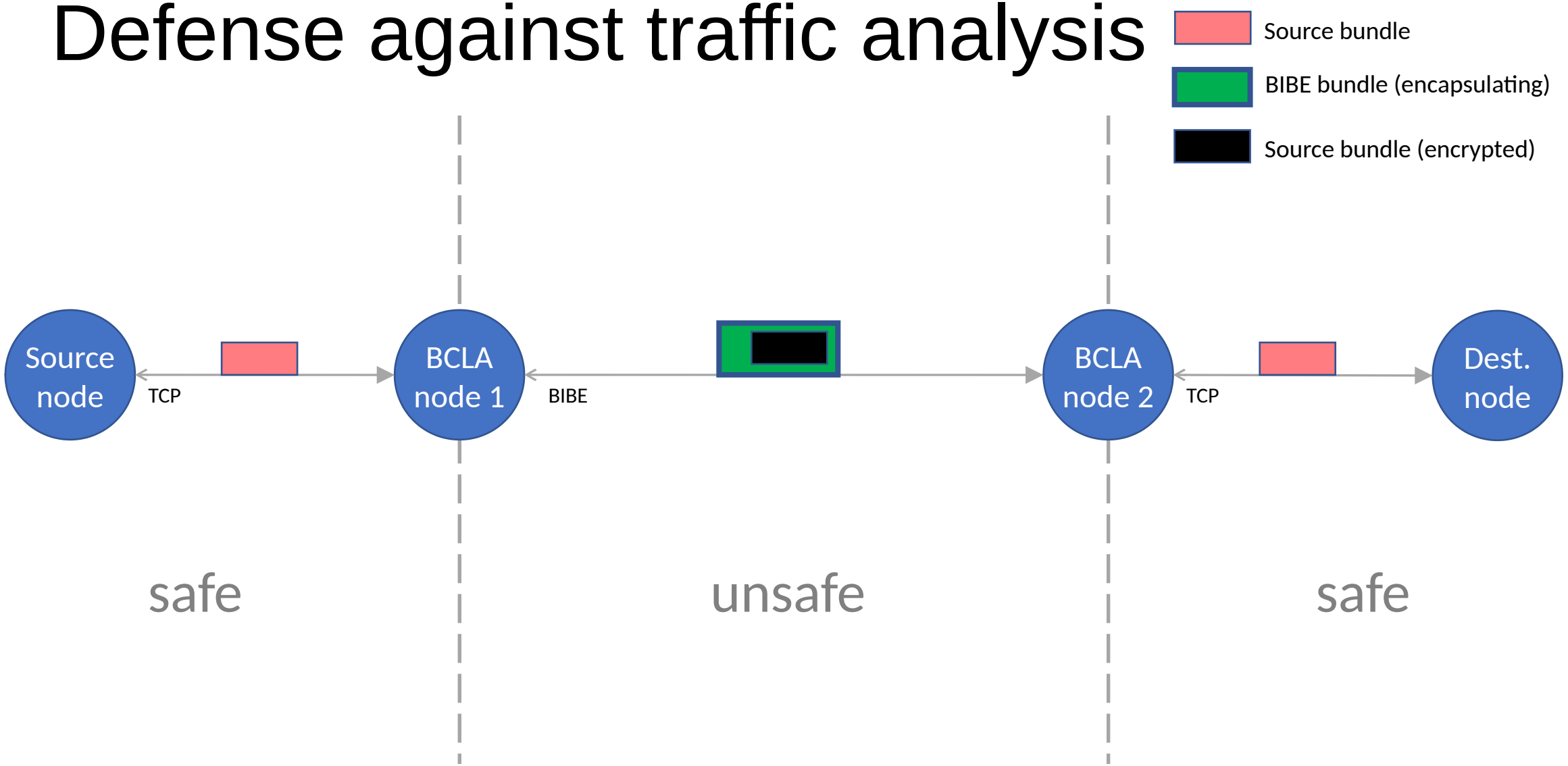
Custodial reliability



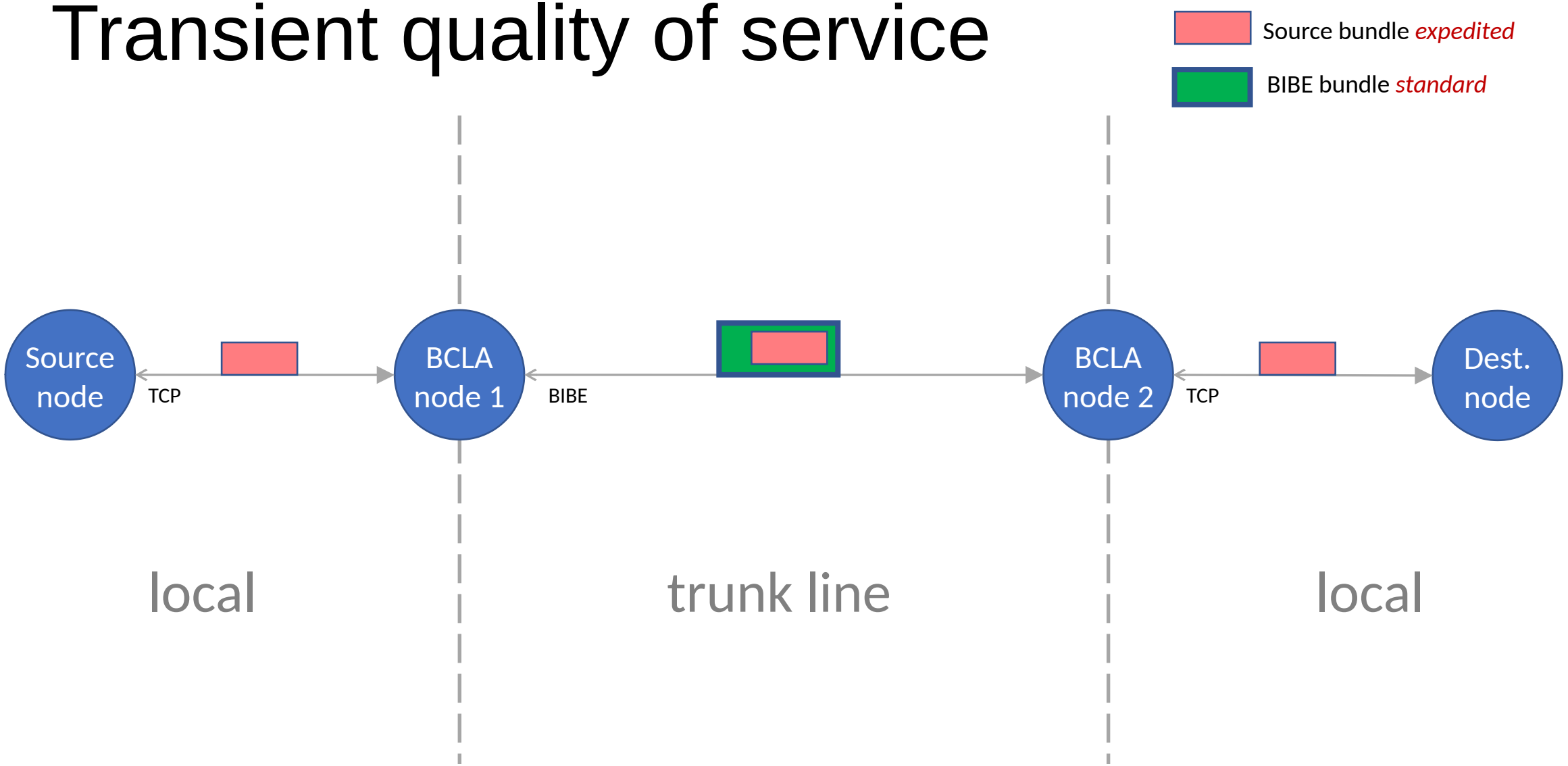
Cross-domain security



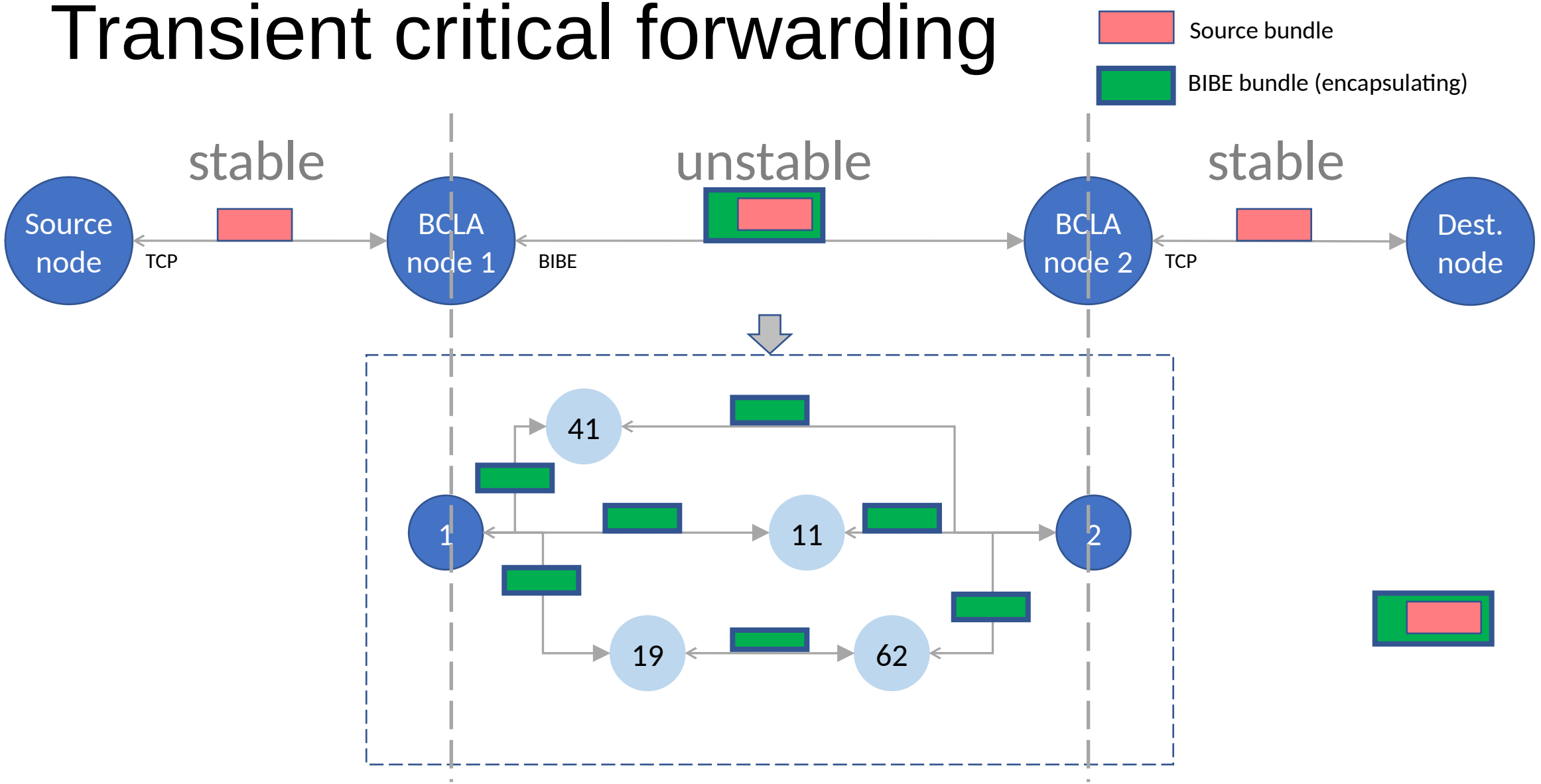
Defense against traffic analysis



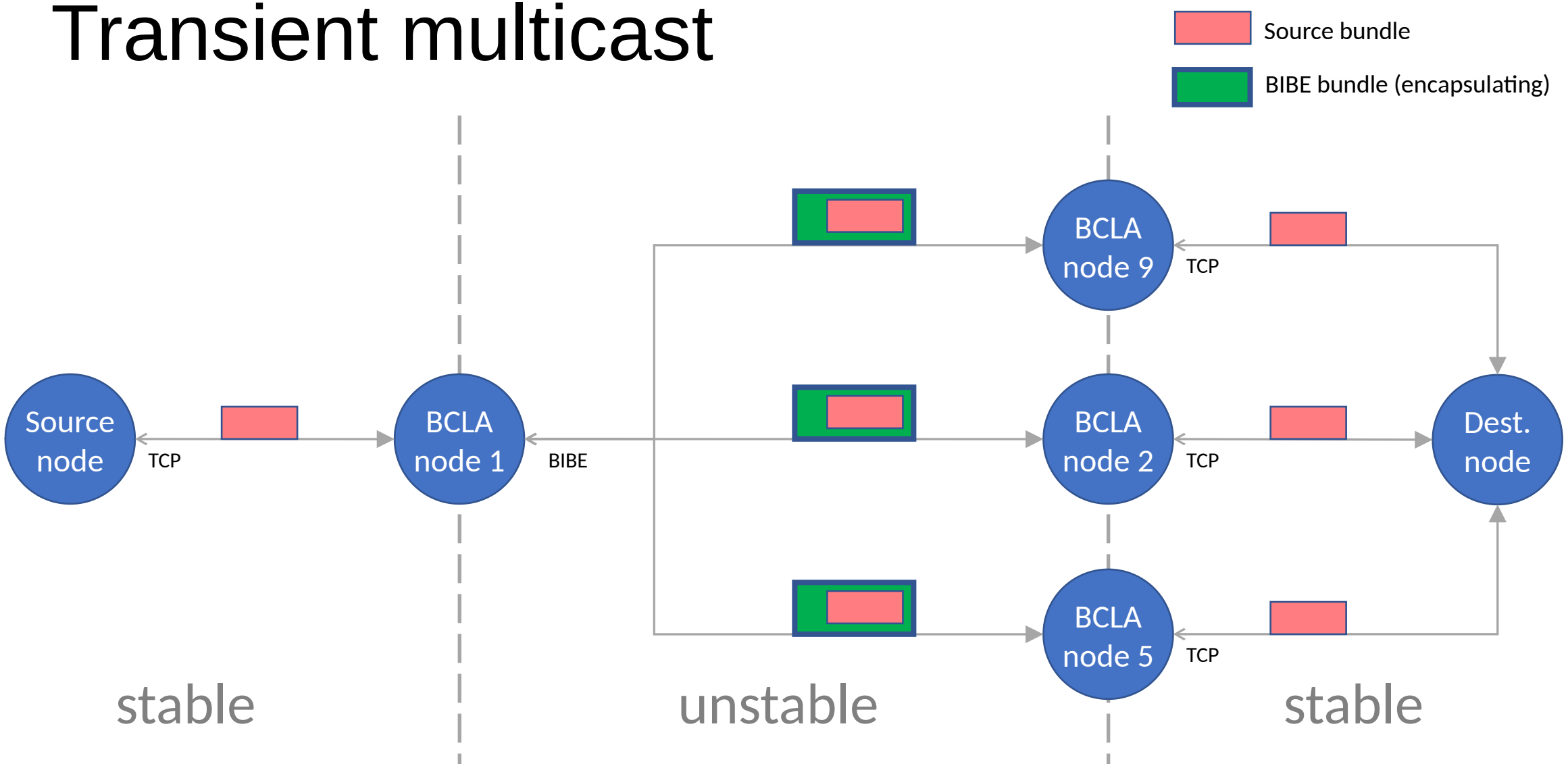
Transient quality of service



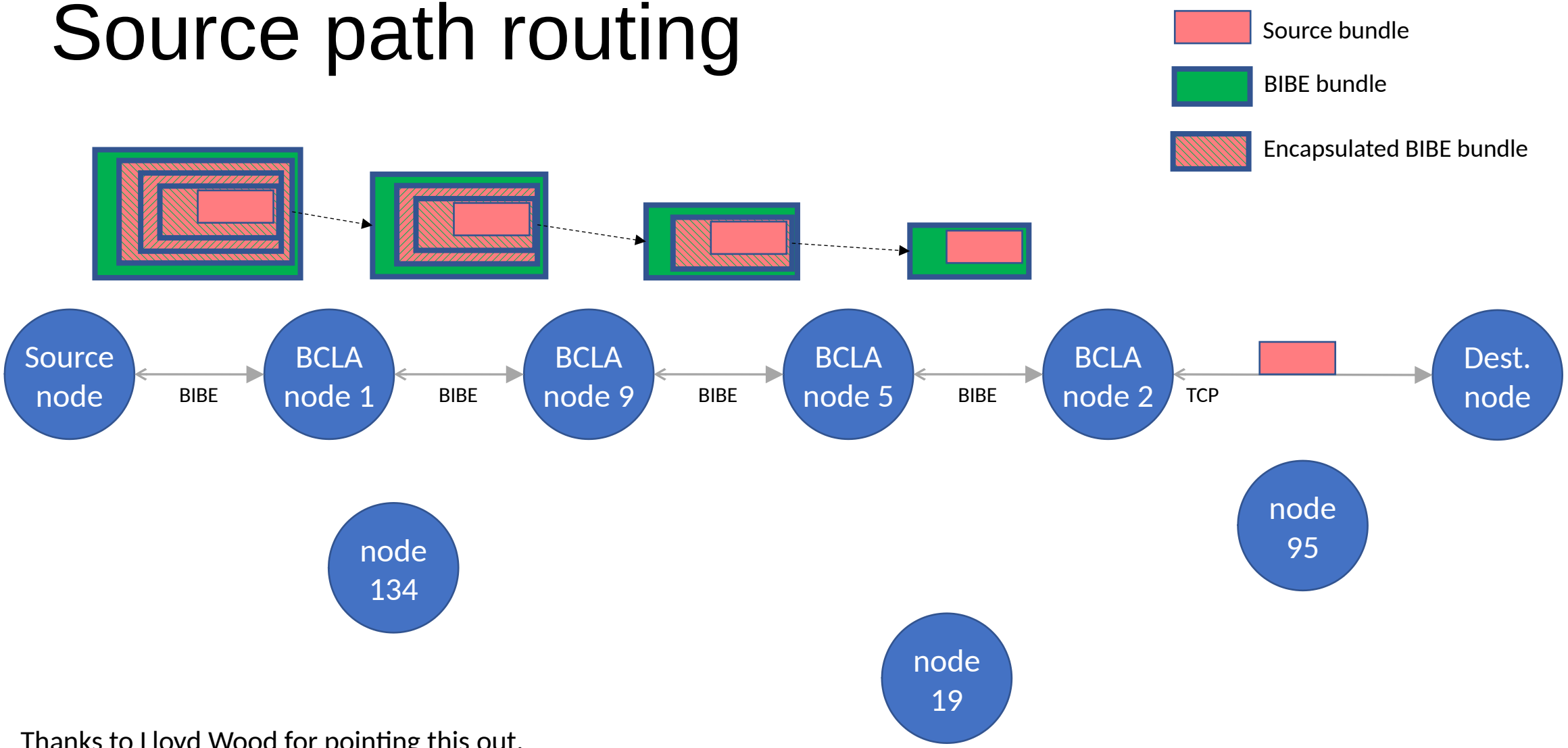
Transient critical forwarding



Transient multicast

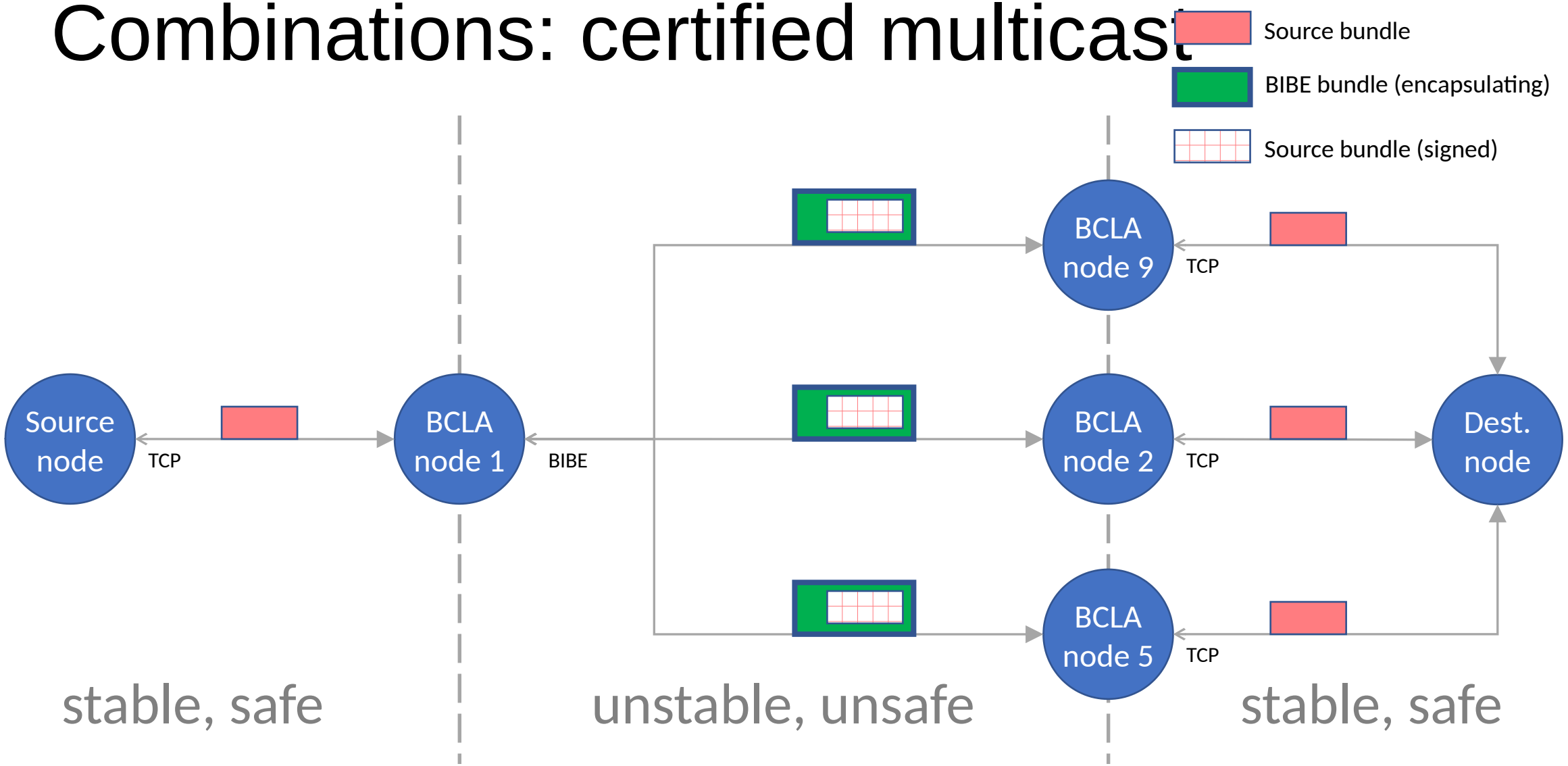


Source path routing



Thanks to Lloyd Wood for pointing this out.

Combinations: certified multicast



Questions?