

LOOPS Generic Information Set

draft-welzl-loops-gen-info-00

LOOPS BoF

IETF 105 - Montreal

Michael Welzl, U. Oslo

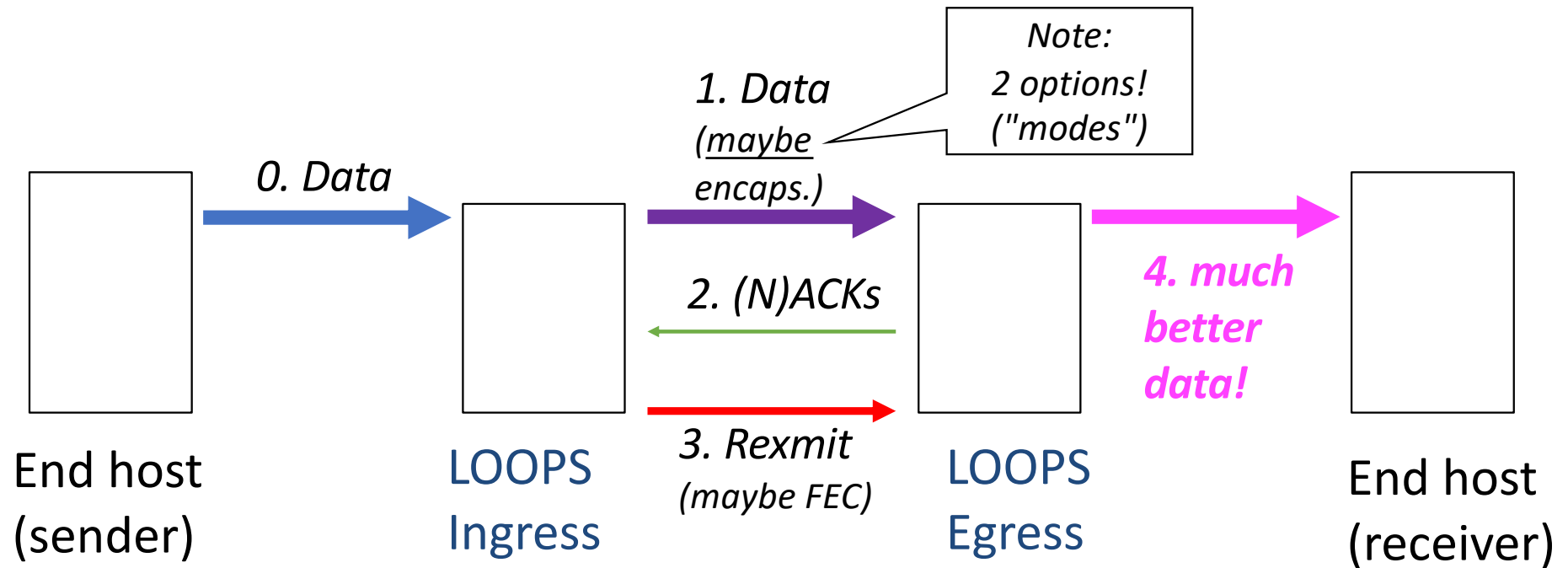
Carsten Bormann, ed., U. Bremen

What is this draft?

- "The present document is a **strawman** for the set of information that would be interchanged in a LOOPS protocol, without already defining a specific data packet format."

→ an overview of how a LOOPS protocol could work.

Context



Problems to address

- From previous slide:
 - Loss detection/retransmission
 - FEC control
- Also: handle congestion on ingress-egress segment
 - Measurement / congestion detection (limit extra traffic from ingress)
 - Congestion signaling to end hosts
- Next: some concrete ideas on how to deal with these problems
 - Just a strawman !

Tunnel mode

1. Ingress forwards

- **Encapsulate**: not tied to any specific overlay protocol
 - Document contains sketches of bindings to GUE and Geneve
- We don't try to understand data after the IP header
 - Hence, need to add a **Packet Sequence Number (PSN)**
- Some more information added
 - **Tunnel type**
 - **"ACK desirable" flag** (asks for feedback block 1, next slide)
 - **Anything needed by FEC**

2. Egress answers

- **Block 1** (optional, only upon "ACK desirable")
 - PSN being acknowledged
 - Absolute time of reception for the packet acked (PSN)
- **Block 2** (optional)
 - an ACK bitmap (based on PSN)
 - a delta indicating the end PSN of the bitmap
- Can be interspersed and repeated
- Can be piggybacked on a reverse direction data packet or sent separately
- Usually aggregated in some useful form

3. Ingress retransmits

- Detects need for retransmit via **NACK** or **RTO**
 - But should we really? What about congestion?
 - Use ECN if possible
 - Calculate latency from timestamps in feedback blocks 1
 - Assume ingress-egress time-sync
- ... Or, rather than "just retransmit", send **FEC** repair packets

4. Egress forwards

- De-FEC
- Inform end hosts about congestion if needed
 - Might be able to distinguish "real" congestion from, e.g., corruption loss
 - ECN much preferred as a signal!

Summary: information exchanged

- **Forward:** encapsulation, containing...
 - Packet Sequence Number (PSN)
 - Tunnel type
 - "ACK desirable" flag (asks for feedback block 1, next slide)
 - Anything needed by FEC
- **Backward:** optional blocks type 1 and 2...
(can be piggybacked, aggregated, interspersed, repeated, ...)
 - **Block 1** (optional, only upon "ACK desirable")
 - PSN being acknowledged
 - Absolute time of reception for the packet acked (PSN)
 - **Block 2** (optional)
 - an ACK bitmap (based on PSN)
 - a delta indicating the end PSN of the bitmap

Transparent mode

A bit more radical... describing least intrusive method here:

"Never delay and don't even tunnel"

Just discussing retransmit; FEC could also be done

1. Ingress forwards

- Just forward
- Also, keep a copy of packets, with a hash for identification
 - From immutable header fields
 - May need to include data beyond IP

2. Egress answers

- ACK everything; no NACK possible
 - Same hash calculation as ingress
- ACK format similar to tunnel mode

3. Ingress retransmits

- Detects need for retransmit **via RTO only**
 - Should we really? Congestion estimation as before
- Hash collisions: miss retransmit opportunity, and possibly latency measurement errors
 - Could exclude collisions from latency calculation

4. Egress forwards

- That's all it does. There will be re-ordering.

Summary: information exchanged

- **Forward:** nothing
- **Backward:** roughly as before - optional blocks type 1 and 2...
(can be piggybacked, aggregated, interspersed, repeated, ...)
 - **Block 1**
(limited in some way: was optional, only upon "ACK desirable" for tunnel mode, but egress doesn't get this information in transparent mode)
 - PSN being acknowledged
 - Absolute time of reception for the packet acked (**hash**)
 - **Block 2 (optional)**
 - an ACK bitmap (based on **hash values**)
 - a delta indicating the end **hash** of the bitmap

Conclusion

- Spectrum of possibilities, from "full-fledged" to min-intrusive
 - Various trade-offs between these options
- In all cases: LOOPS can be very beneficial when the LOOPS segment RTT is much shorter than the e2e RTT
 - Wireless NICs use this fact
- Some packet drops really cause pain
 - LOOPS can help



**Tail
loss!**

Clarifying questions?

(Don't forget to think "strawman".)