APRIL 24-25

P4

WORKSHOP 2023

P4.org Architecture Work Group

Mario Baldi &
Andy Fingerhut

If you have ever experienced something that feels strangely familiar, as if the exact same thing has happened to you before, then you are experiencing what the French call "déjà vu.”

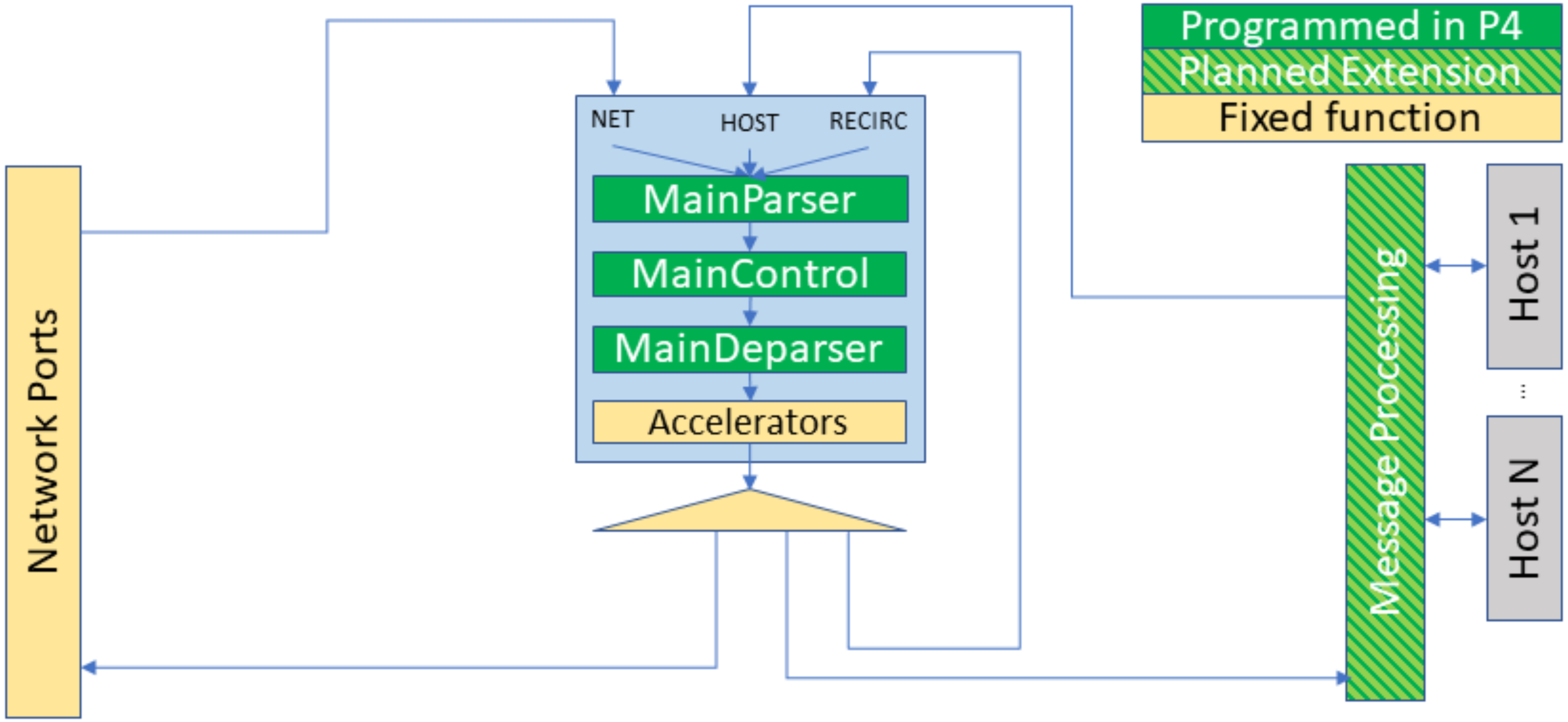
“The Carnivorous Carnival”, Lemony Snicket

[on the next page of the book]

If you have ever experienced something that feels strangely familiar, as if the exact same thing has happened to you before, then you are experiencing what the French call "déjà vu.”

“The Carnivorous Carnival”, Lemony Snicket

PNA structure



Significant features defined in past two years

- Add-on-miss
  - Add new entries to tables at high rate *in data plane*
- Auto-delete
  - Delete old entries *in the data plane* when they have been unmatched for configurable duration.
  - The timeout duration of an entry is *modifiable* at packet processing time.
- Packet encryption
  - Data plane APIs and P4 architecture flow for encryption & decryption

In progress now

- Data-plane-writable action data
  - e.g. maintain expected TCP sequence numbers independently for each table entry, in TCP connection tracking.
  - This could be specified before using externs – the main new thing here is a more concise syntax.
- Nailing down details of recirculation

Many things still to do

- Multicast
- Review packet mirroring
- Common definition across targets for relative priorities of drop, unicast, multicast, etc.
- Customizable Tx/Rx host descriptor formats
- Define message processing features
    - Interaction with host memory, e.g. reading/writing descriptors and packet data
    - Potentially enabling segmentation and reassembly/coalescing features
    - Programmable RDMA?

Active participants in architecture WG today include

- AMD
- Google
- Intel
- Marvell
- Nvidia

Getting involved in P4.org

How to get involved in a working group?

- From P4.org home page top right, choose Community -> Working Groups
- You will find:
  - Calendar of P4.org meetings, all open to the public, all currently on-line (Zoom, Teams, etc.)
  - Who the co-chairs are
  - Mailing list for each working group
- Good ways to jump in:
  - Join the email list, send questions there
  - Attend a meeting
  - Contact the co-chairs before a meeting if you have a topic you would like to discuss

What is it like?

- Many regular work group members are experts in their fields
  - Compiler design and implementation
  - programming language features
  - one or more P4 target devices
  - runtime implementers, or users of runtime APIs in production
- Implementers of compilers and runtimes worry about corner cases
  - Does this new feature interact in odd ways with existing features?
  - Would it be excessively difficult to implement, test, or use?
  - Are there performance implications?
- Expect proposals to go through multiple rounds of feedback and revision
  - One-on-one or “side” meetings among the most interested people of a feature are often good ways to speed things up.

Thank You!

Andy Fingerhut