EasyTrans: Enable Fast Iteration of Transport Protocol

Jie Zhang* , Chuan Ma* , Wei Wang†, Kai Zheng†, Yong Cui*

Tsinghua University* , Huawei Technologies†

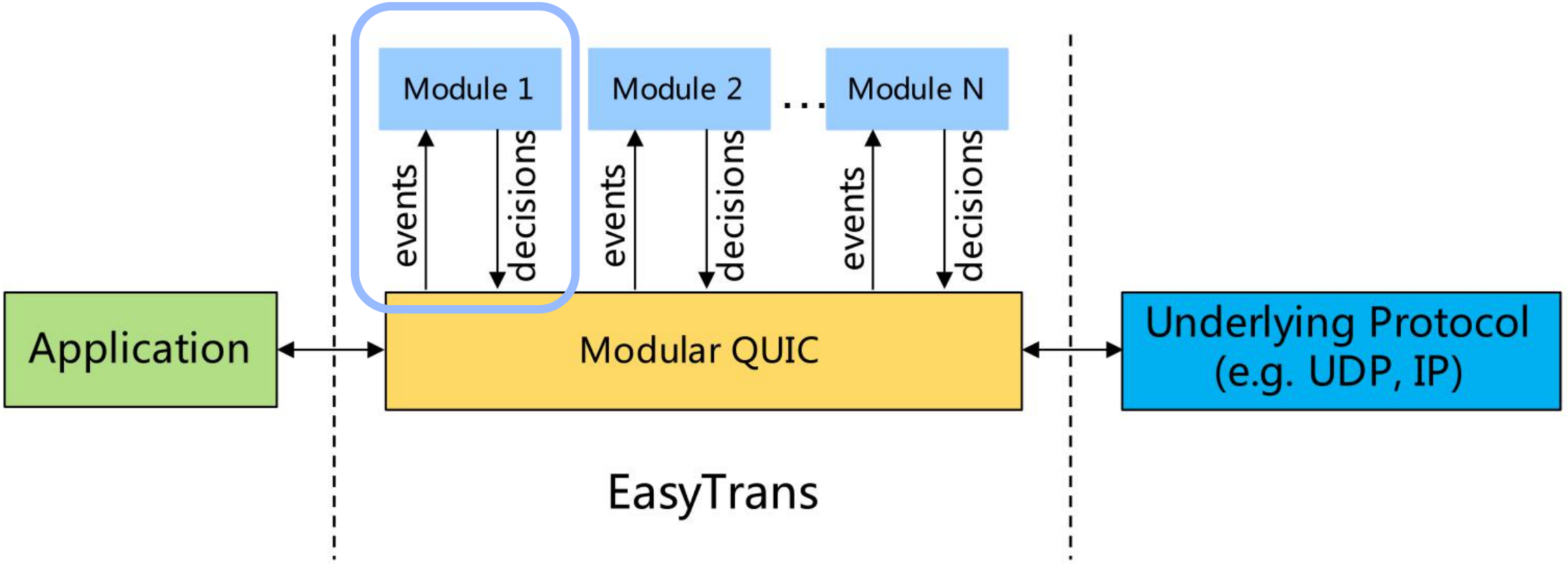
Fast development **High performance**

Background and Motivation

- **Transport protocol iterations accelerate**
  - Emerging along with evolving applications
- **Trouble in developing protocols**
  - “Too coupled to use” or “Too extensible to perform well”
- **Goal: EasyTrans**
  - Enable easy development and retains good performance

EasyTrans: Architecture

- **Focus on** specific module and **ignore** unnecessary parts
- Various **calling modes** that **improve performance**



EasyTrans: Calling modes

- Four different calling modes
- **High performance even with sophisticated computation**
  - E.g. Machine learning

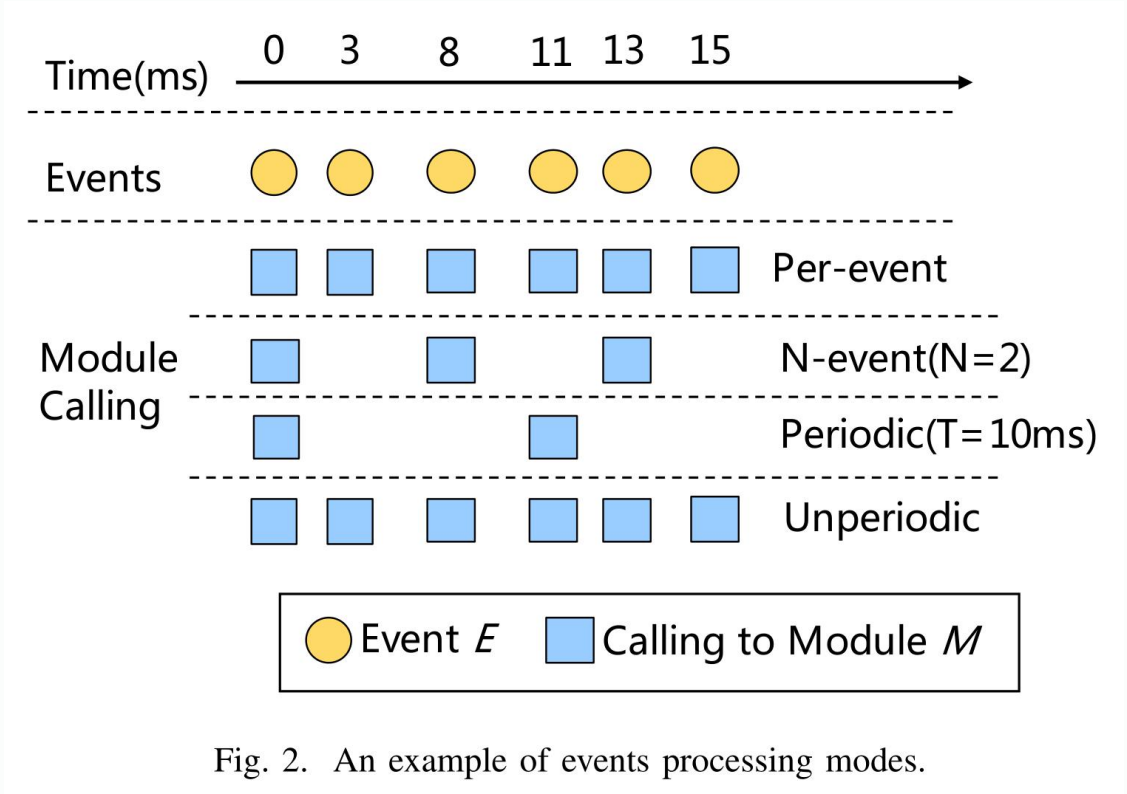


Fig. 2. An example of events processing modes.

EasyTrans: Evaluation

- QUIC-based implementation
- Modules and Calling modes introduce **slight** overhead (<5%)

TABLE I
ACHIEVED LOOPBACK GOODPUT OF DIFFERENT CALLING MODES

Modules	Calling Modes	Goodput(Gbps)
No module	-	1.895
Congestion control (a) and stream scheduling (b)	a: Unperiodic b: Per-event	1.799
a and b	a: Periodic (T=10ms) b: Per-event	1.790
a and b	a: Per-event b: Per-event	1.786

EasyTrans: Future work

- Develop **more modules**:
  - **ACK**
    - Offers diverse strategies to packet **acknowledgment**
    - Challenge: Hard to decouple
  - **Multipath**
    - Make full use of the **bandwidth**
    - Challenge: Change the structure of EasyTrans
- **Extensive evaluation**
  - Performance
  - Fidelity