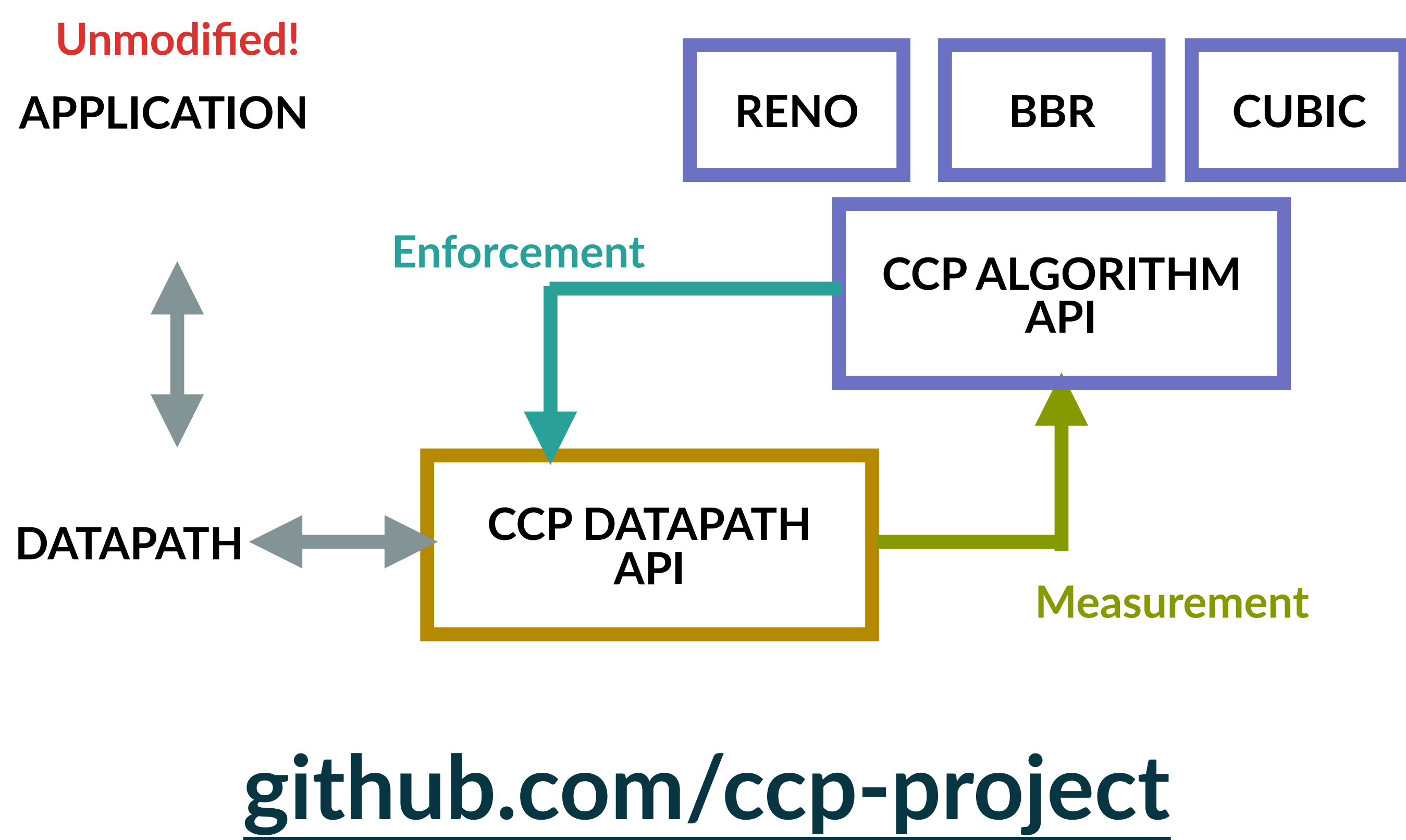


Restructuring Endpoint Congestion Control

Akshay Narayan, Frank Cangialosi, Deepti Raghavan, Prateesh Goyal
Srinivas Narayana, Radhika Mittal*, Mohammad Alizadeh, Hari Balakrishnan



*UC Berkeley
ccp@csail.mit.edu



Split Programming Model

- CCP: new API for congestion control, separates congestion control algorithms from datapath
- Asynchronous event handlers process measurements, enforcements (rate, CWND)
- Flexibility and performance

```
Event Handler : def OnReport(info):
State Update  : cwnd += info.acked / cwnd;
Decision      : datapath.update(["Cwnd", cwnd]);
```

Sophistication

- Isolate developers from datapath programming
- Use powerful user-space libraries

```
let K = pow(
    max(0, WlastMax - cwnd) / 0.4,
    1/3
)
cwnd = WlastMax + 0.4 * pow(t - K, 3)
```

Cubic update function, in CCP Agent

Datapath Programs

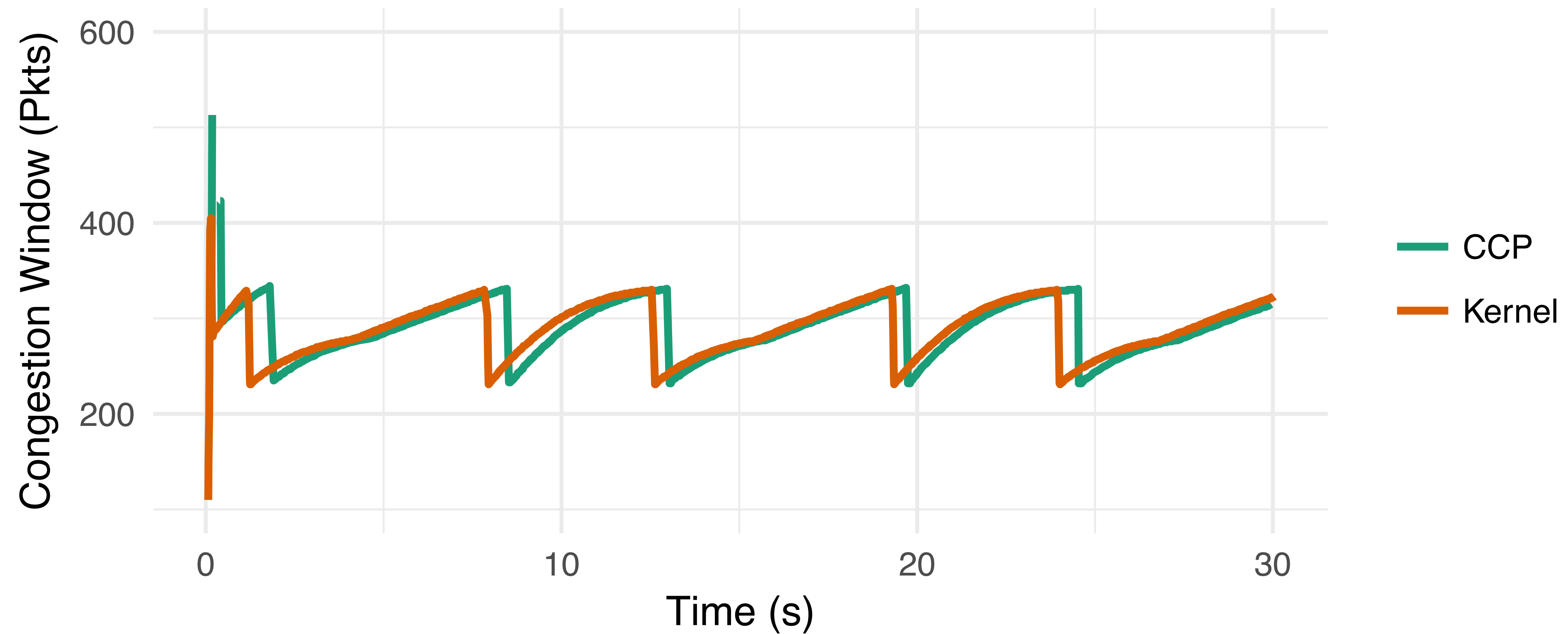
- Collect measurements, specify enforcement

```
(def (Report (acked 0)))
(when true
  (: Report.acked
    (+ Report.acked Ack.bytes_acked))
  (: Cwnd (+ Cwnd Report.acked))
  (fallthrough))
(when (> Flow.lost_pkts_sample 0)
  (report)))
```

Slow Start, in datapath

Fidelity

- CCP algorithms (at right, TCP Cubic) match the behavior of their in-datapath counterparts



Portability

- Run unmodified algorithm implementations across datapaths
- Write-once, run-anywhere

