# THE CASE FOR MOVING CONGESTION CONTROL OUT OF THE DATAPATH

Akshay Narayan, Frank Cangialosi, Prateesh Goyal, Srinivas Narayana, Mohammad Alizadeh, Hari Balakrishnan

MIT CSAIL

## CONGESTION CONTROL

DCQCN TIMELY

DCTCP pFabric

Hybla NV RC3

Algorithms

Westwood FAST RCP Sprout ABC

EBCC Binomial XCP Illinois Remy

Vegas Eifel Veno LEDBAT PRR PCC

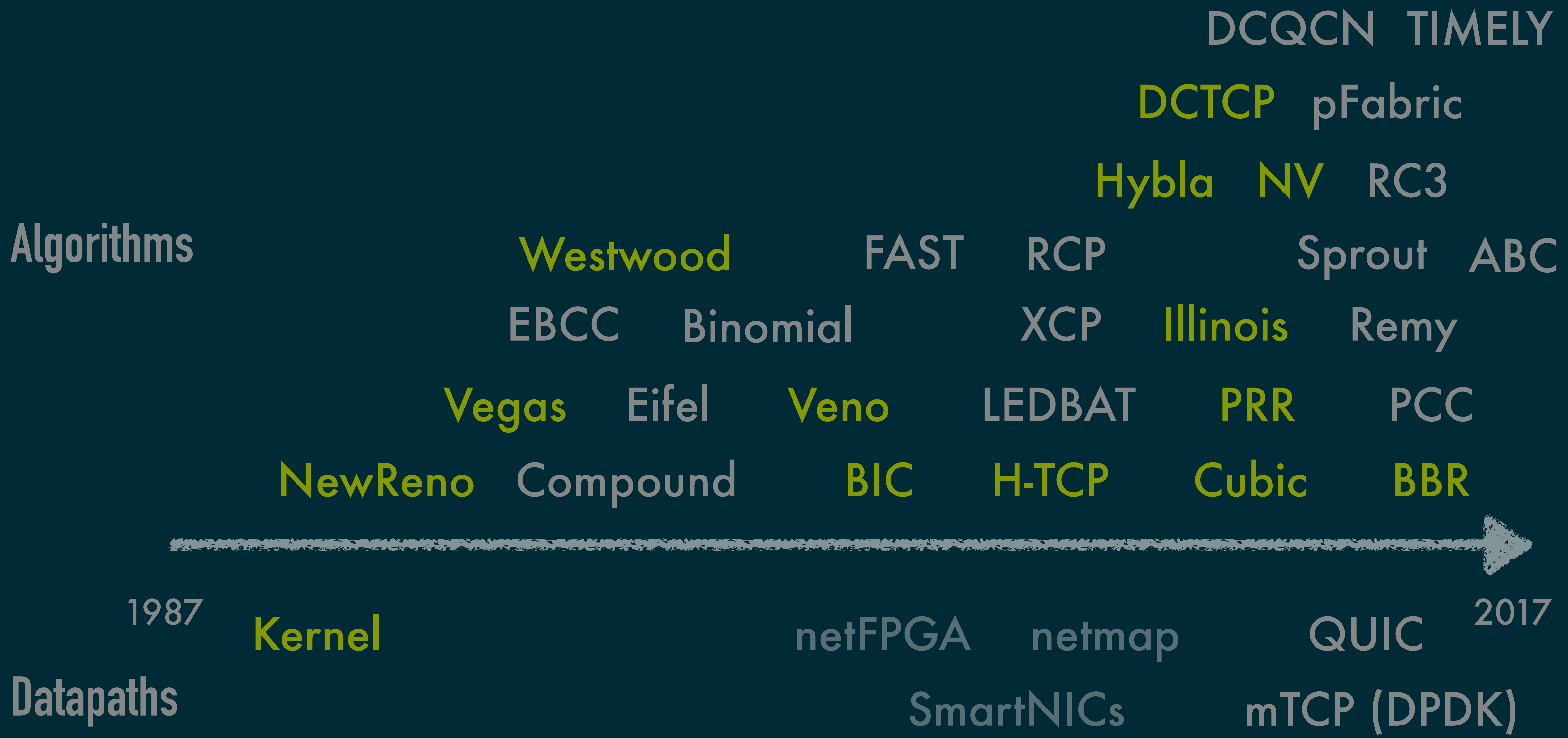
NewReno Compound BIC H-TCP Cubic BBR

1987 Kernel netFPGA netmap QUIC 2017

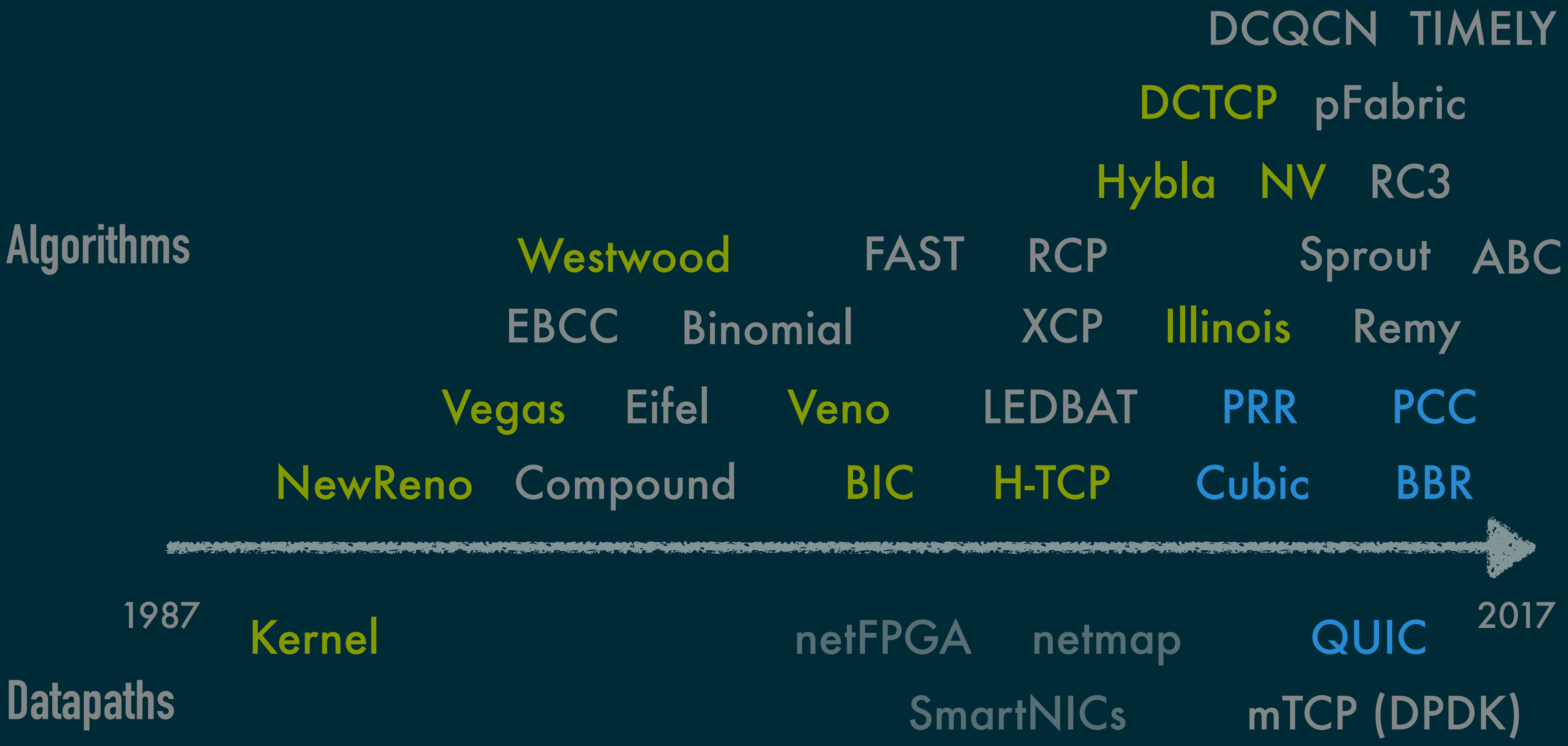
Datapaths

SmartNICs mTCP (DPDK)

# CONGESTION CONTROL


DCQCN TIMELY
DCTCP pFabric
Hybla NV RC3
Algorithms
Westwood FAST RCP Sprout ABC
EBCC Binomial XCP Illinois Remy
Vegas Eifel Veno LEDBAT PRR PCC
NewReno Compound BIC H-TCP Cubic BBR
1987
2017
Kernel netFPGA netmap QUIC
Datapaths
SmartNICs mTCP (DPDK)

# CONGESTION CONTROL



DCQCN TIMELY
DCTCP pFabric
Hybla NV RC3
Algorithms
Westwood FAST RCP Sprout ABC
EBCC Binomial XCP Illinois Remy
Vegas Eifel Veno LEDBAT PRR PCC
NewReno Compound BIC H-TCP Cubic BBR
1987
2017
Kernel netFPGA netmap QUIC
Datapaths
SmartNICs mTCP (DPDK)

## CONGESTION CONTROL

DCQCN TIMELY

DCTCP pFabric

Hybla NV RC3

Algorithms

Westwood FAST RCP Sprout ABC

EBCC Binomial XCP Illinois Remy

Vegas Eifel Veno LEDBAT PRR PCC

NewReno Compound BIC H-TCP Cubic BBR

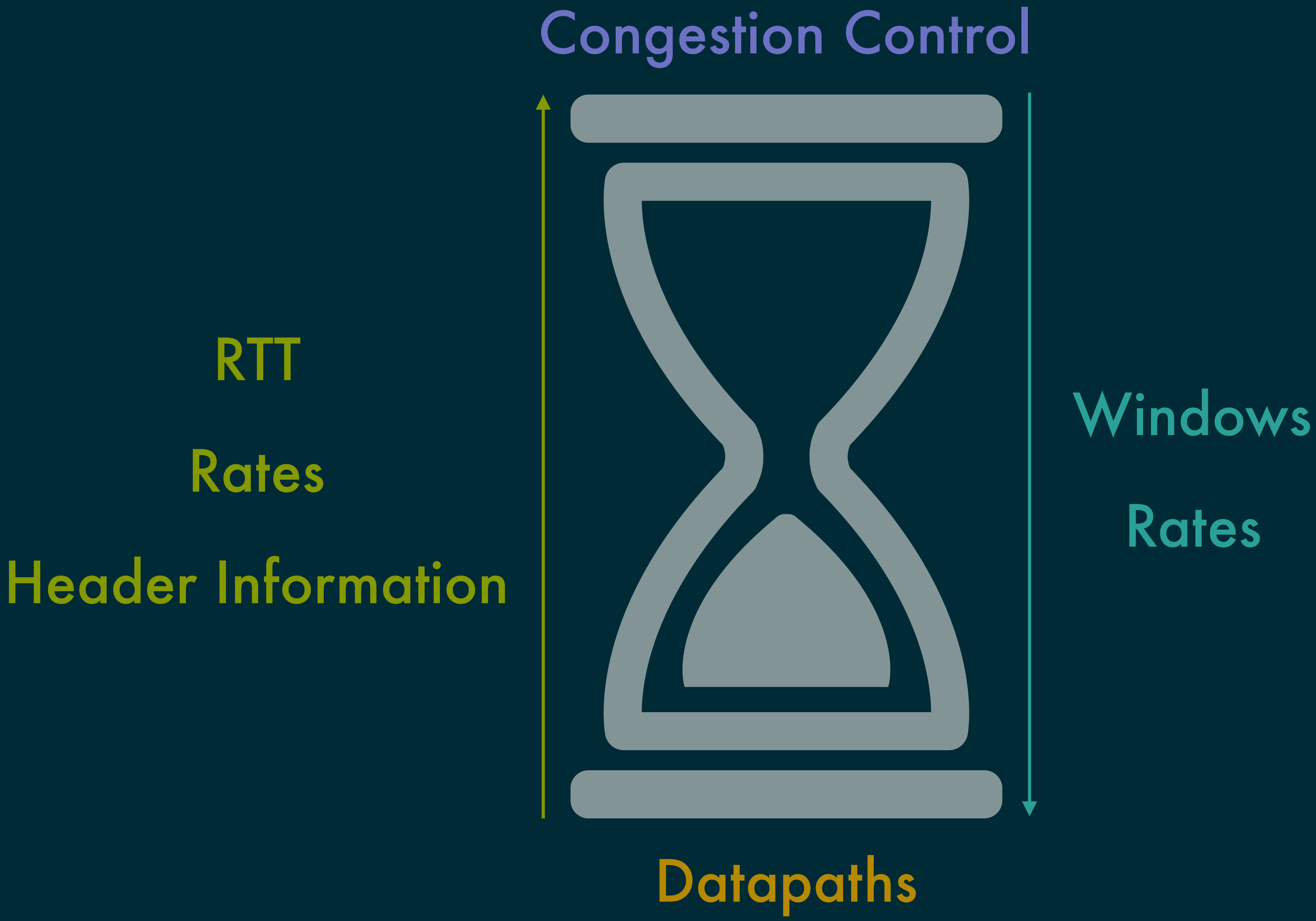
1987 2017

Kernel netFPGA netmap QUIC

Datapaths

SmartNICs mTCP (DPDK)

## NARROW WAIST



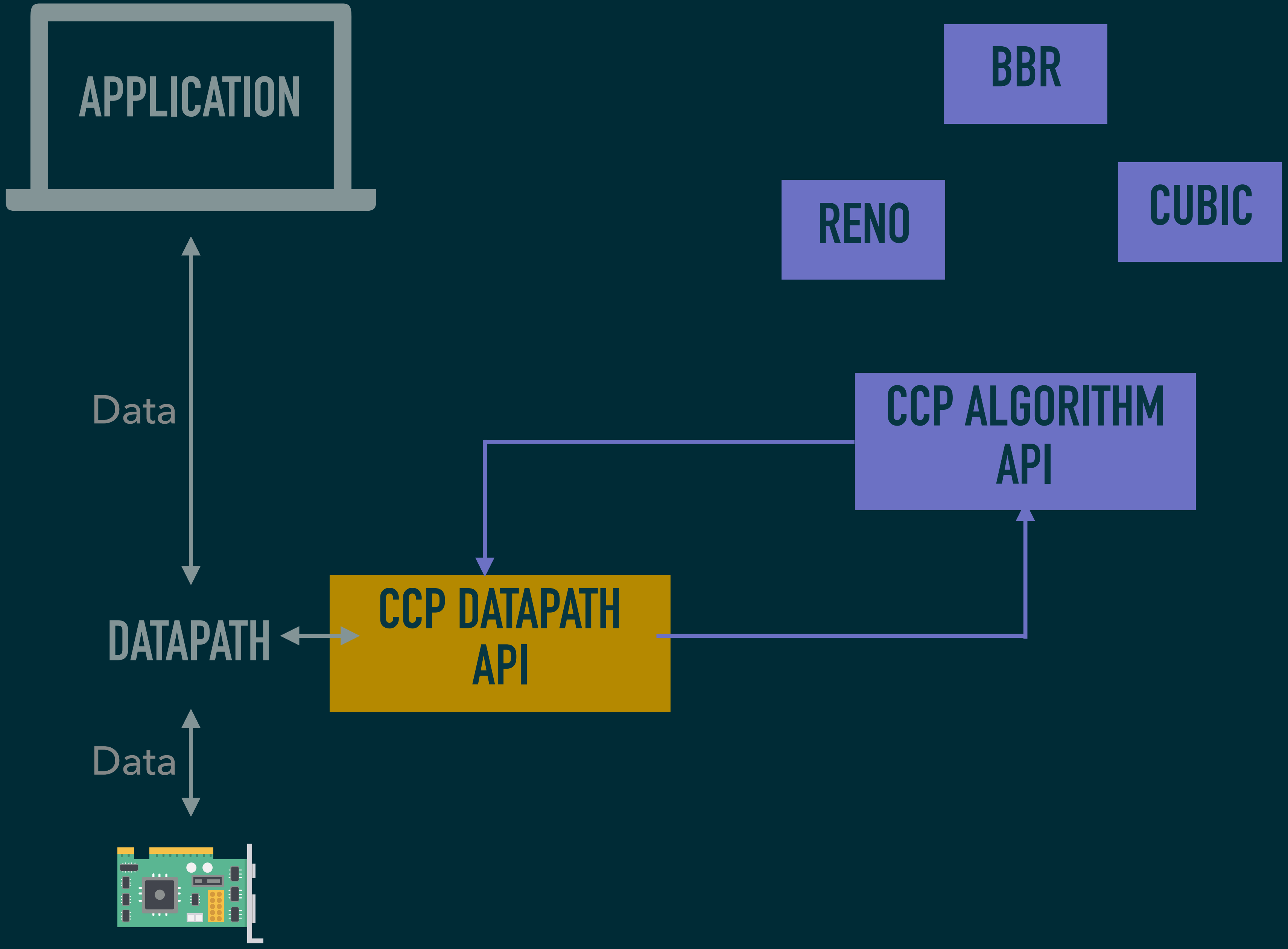
## PRIMITIVES

| Signal | Protocols |
|---|---|
| ACKs | Cubic, DCTCP, NewReno |
| Loss | Cubic, DCTCP, NewReno, PCC |
| RTT | BBR, Remy, Sprout, TIMELY, Vegas |
| Rates | BBR, PCC, Remy, Sprout |
| ECN | ABC, DCTCP |

## CONGESTION CONTROL PLANE DESIGN

Off Datapath

Asynchronous



Without compromising performance

## ALGORITHM API

Event Handler

```
fn OnMeasurement(m: Info) {

}
```

## ALGORITHM API

Event Handler

State Update

```
fn OnMeasurement(m: Info) {
  newlyAcked = m.Ack - lastAck;
  cwnd += newlyAcked / cwnd;




}
```

# ALGORITHM API

Event Handler

State Update

Decision

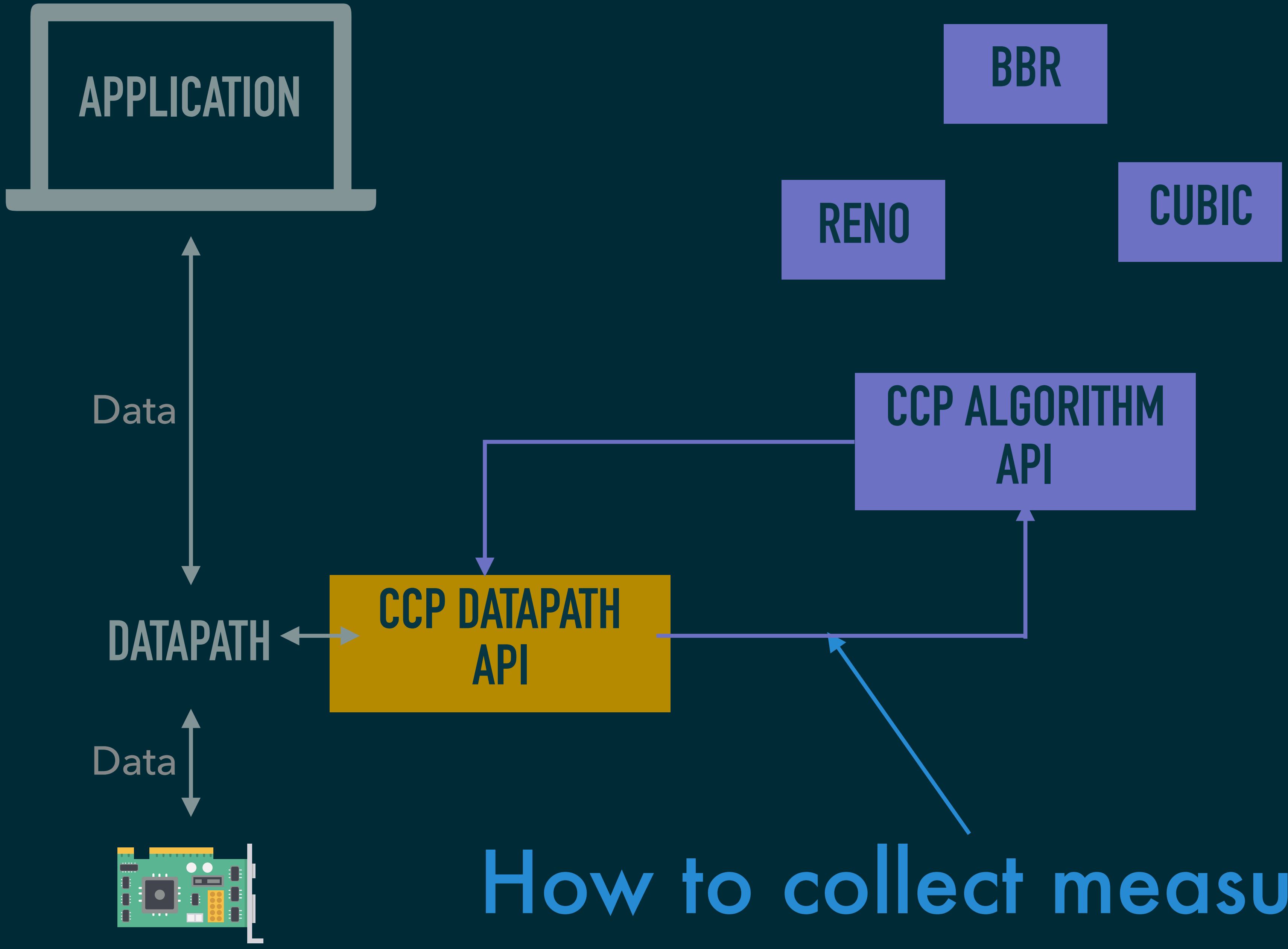
```
fn OnMeasurement(m: Info) {
  newlyAcked = m.Ack - lastAck;
  cwnd += newlyAcked / cwnd;

  run_on_datapath(
   SetCwnd(cwnd) => WaitRtts(1.0) => Report()
  );

}
```

## DIFFICULTY OF DATAPATH PROGRAMMING

```
fn OnMeasurement(m: Info) {
  let K = pow(max(0, WlastMax - cwnd) / 0.4), 1/3)
  cwnd = WlastMax + 0.4 * pow(t - K, 3)
}
```

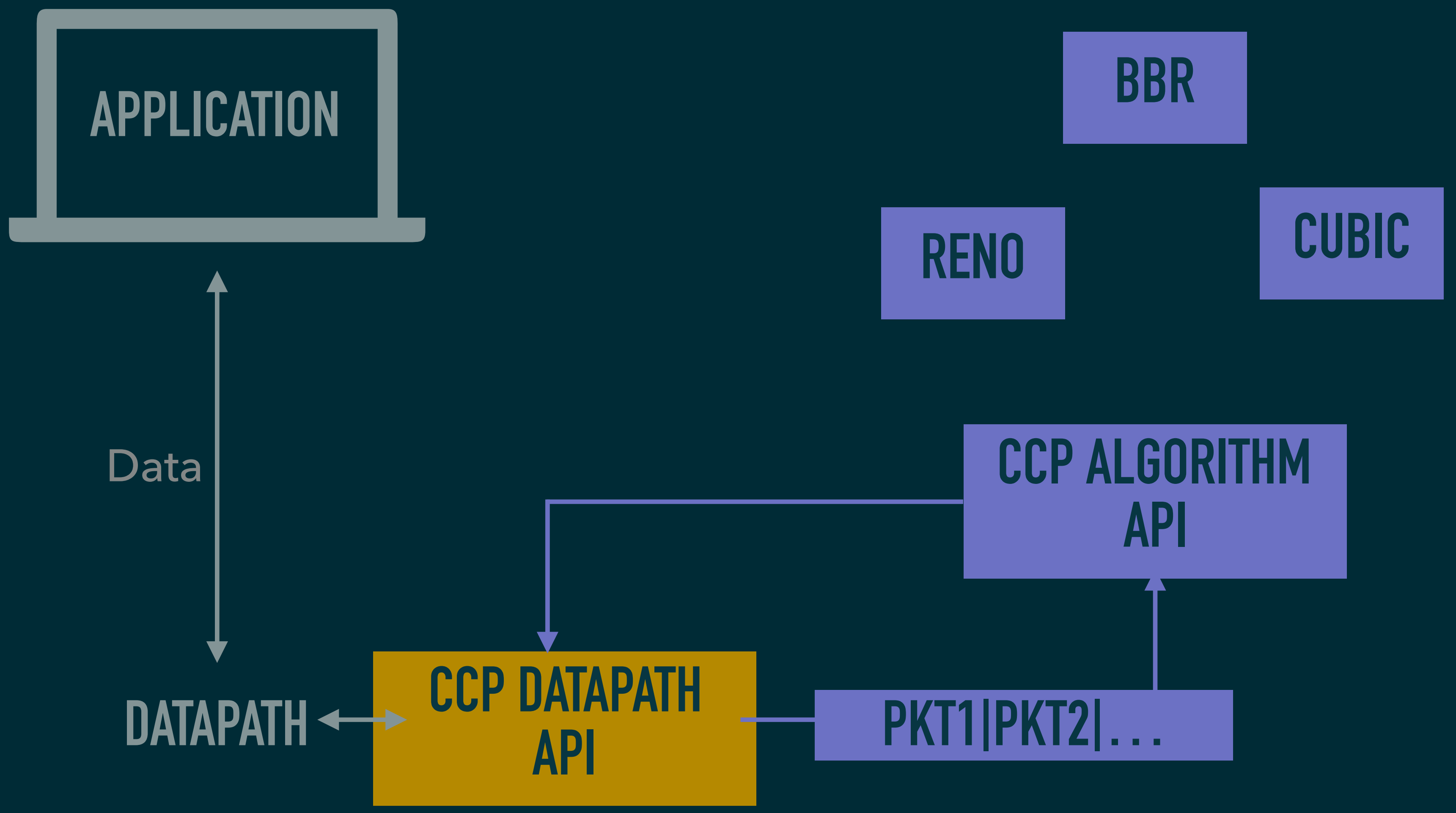
```
net/ipv4/tcp_cubic.c
/* calculate the cubic root of x using a table lookup followed by one
 * Newton-Raphson iteration.
 * Avg err ~= 0.195%
 */
static u32 cubic_root(u64 a) // 40 lines of code
```

# DATAPATH API



How to collect measurements?

## VECTOR BATCHING

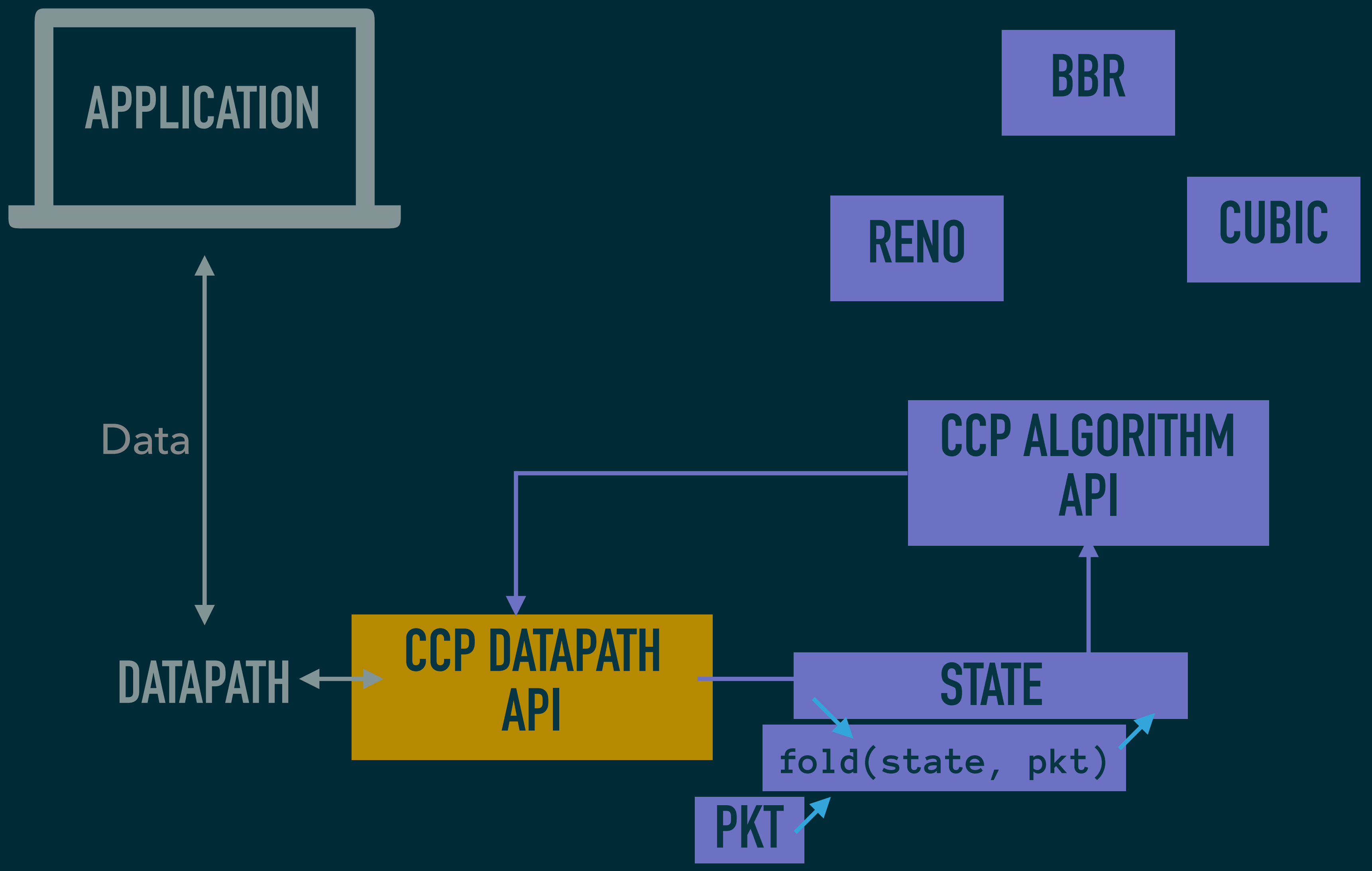


Store per-packet information

Send vector of measurements to CCP

Compute RTT, Rates, etc in CCP

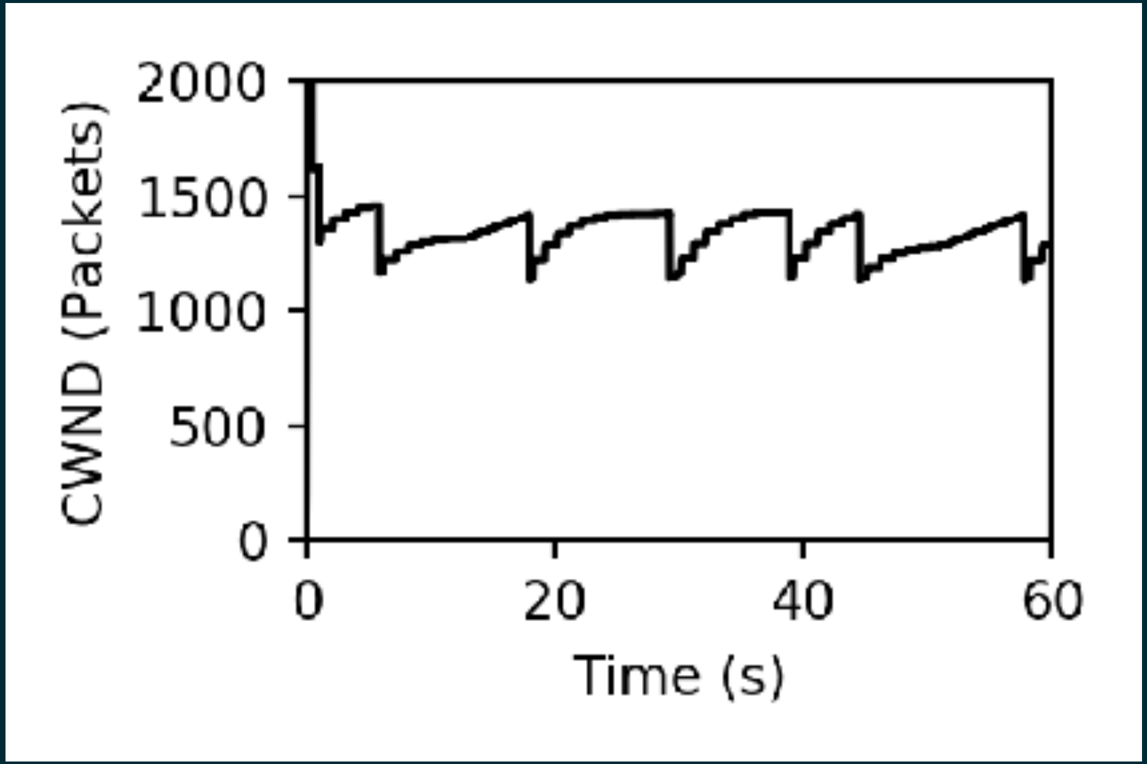
## IN-DATAPATH AGGREGATION



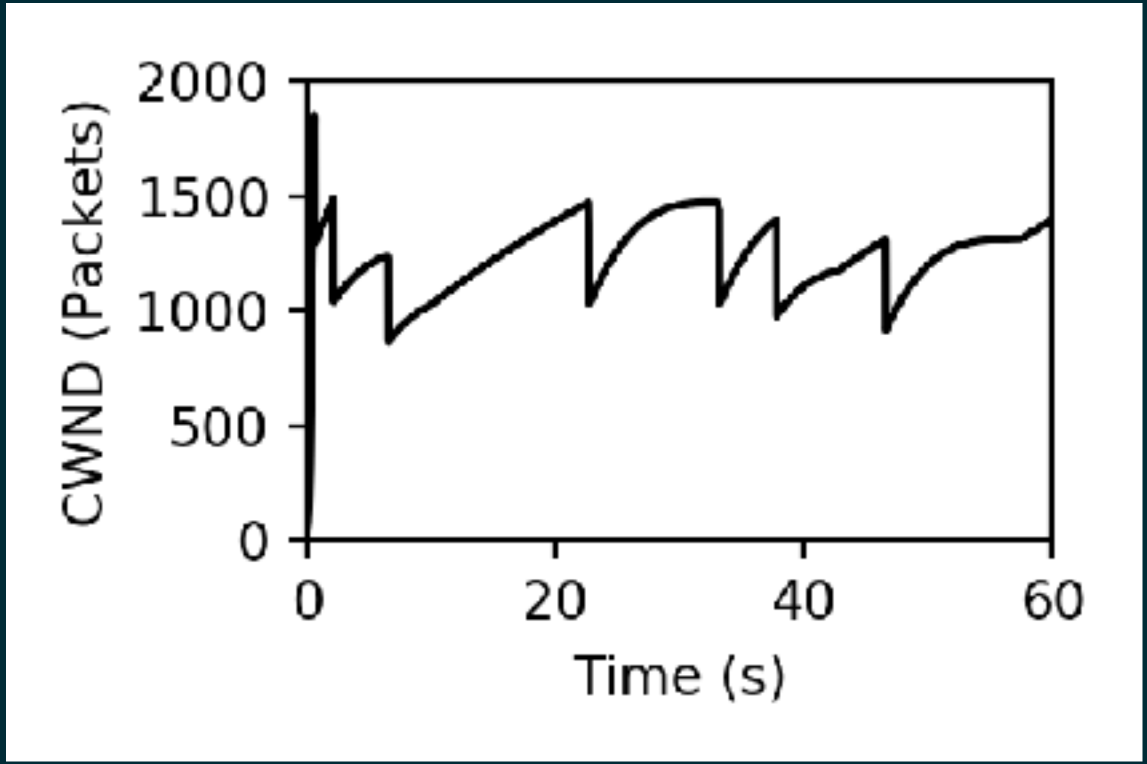
Expose primitives to user-defined fold

Compute state aggregate in datapath

# CONGESTION WINDOW DYNAMICS



CCP



Linux

Overall, the window evolution is similar

## IMPLICATIONS

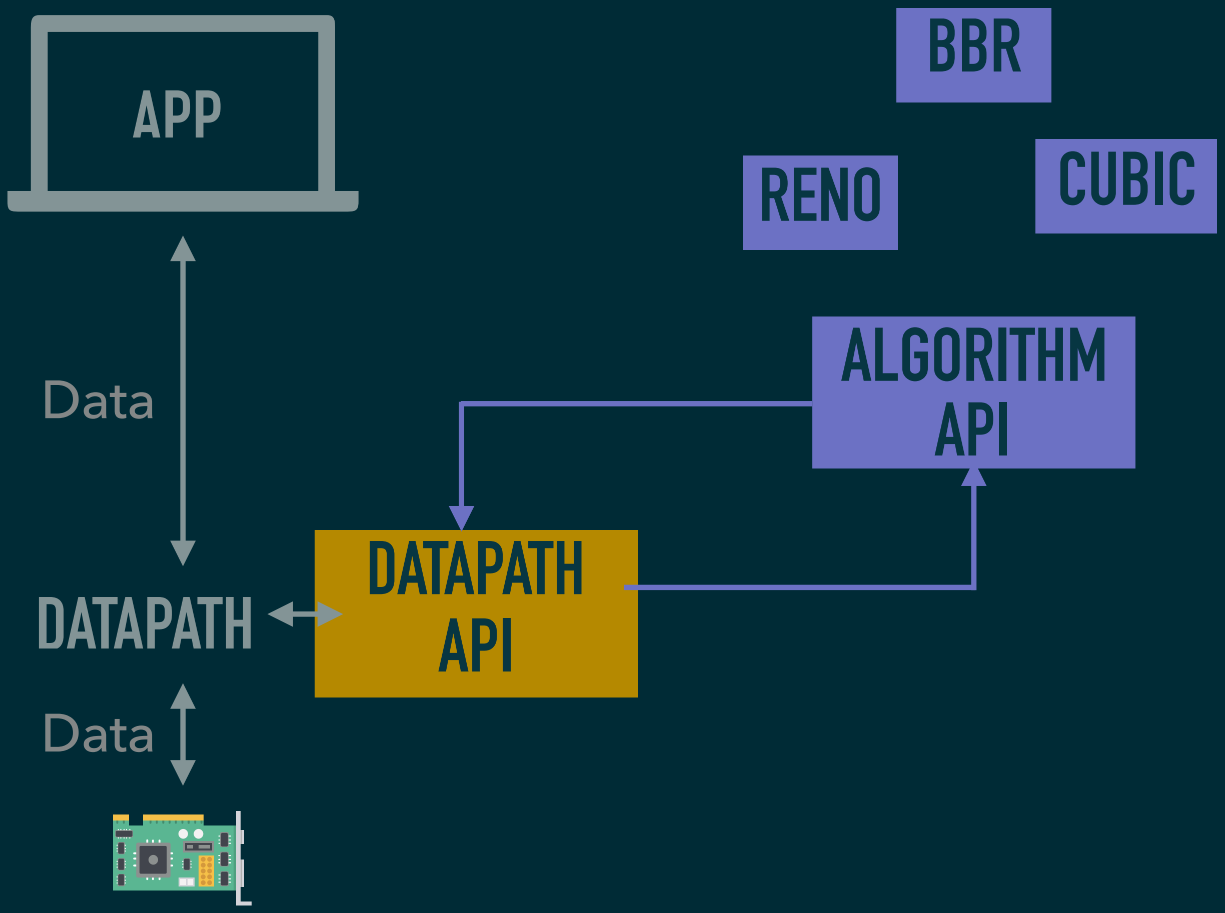
Velocity of congestion control research

New algorithms

Flexibility need not preclude performance

## OPEN QUESTIONS

- New algorithms?
- Hardware support for CCP primitives?
- Low-RTT paths
    - Make decisions less frequently?



github.com/mit-nms/ccp