Verification of GossipSub in ACL2s

ACL2 Workshop 2023

Ankit, Max, Pete and Cristina
Northeastern University

Motivation

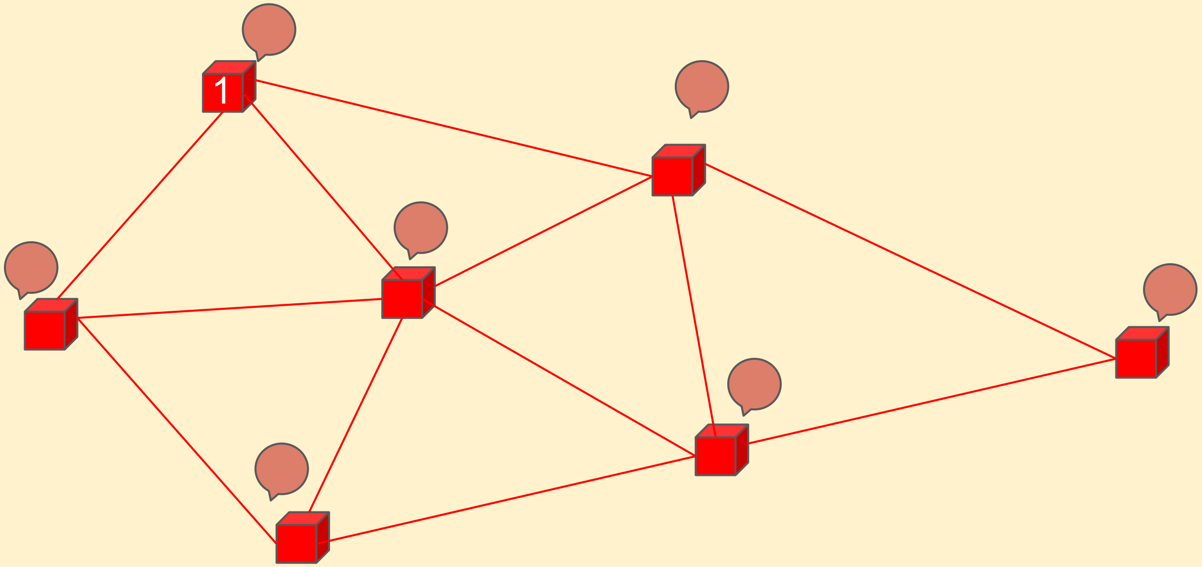
- Popular Web3.0 P2P protocol
- Used by Ethereum and Filecoin, market cap > $145B
- Interesting design, peers decide locally who to talk to
- Claimed resilient against sybil attacks
- We proved otherwise. MITRE CVE-2022-47547
- This work is a companion for our Oakland-24 paper “Formal Model-Driven Analysis of Resilience of GossipSub to Attacks from Misbehaving Peers”

Talk Outline

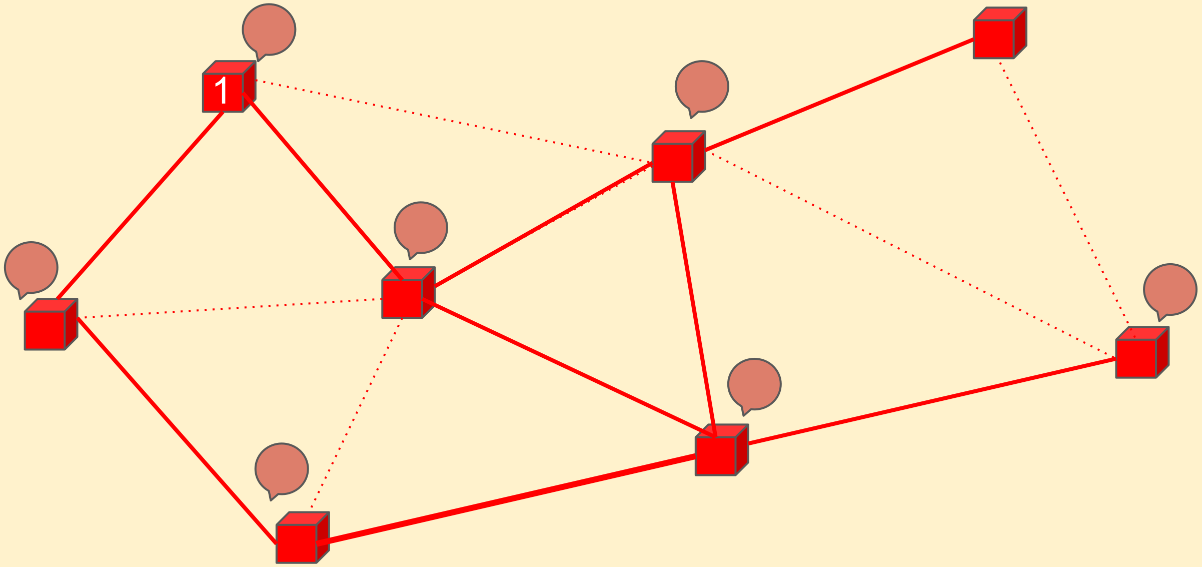
- GossipSub
- Our formal model in ACL2s
- Peer Scoring
- Security Properties
- Attack Generation
- Limitations
- Future Work
- End

GossipSub

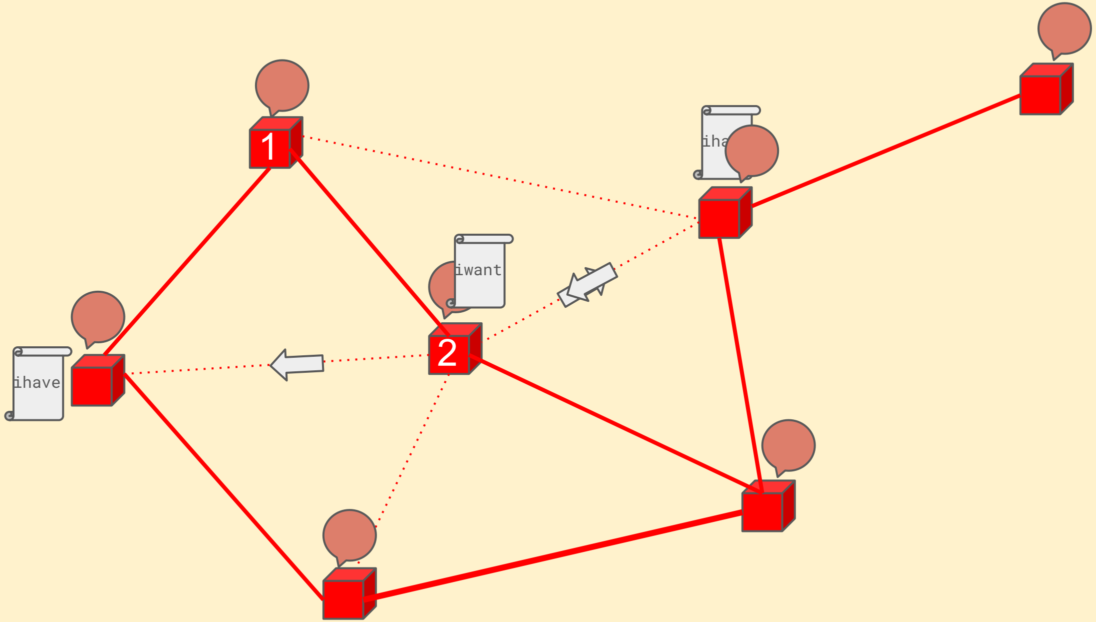
In the beginning was FloodSub



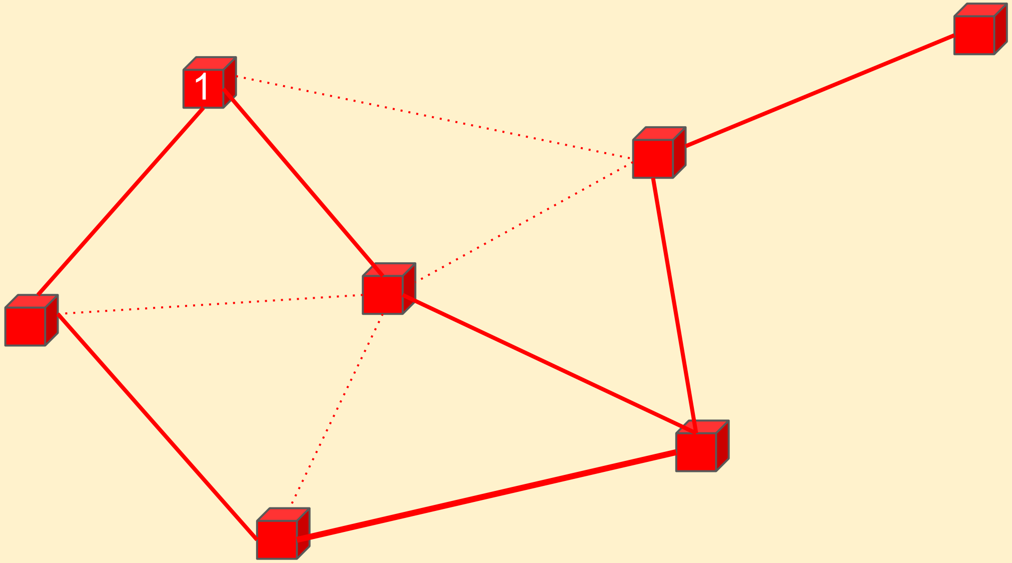
MeshSub



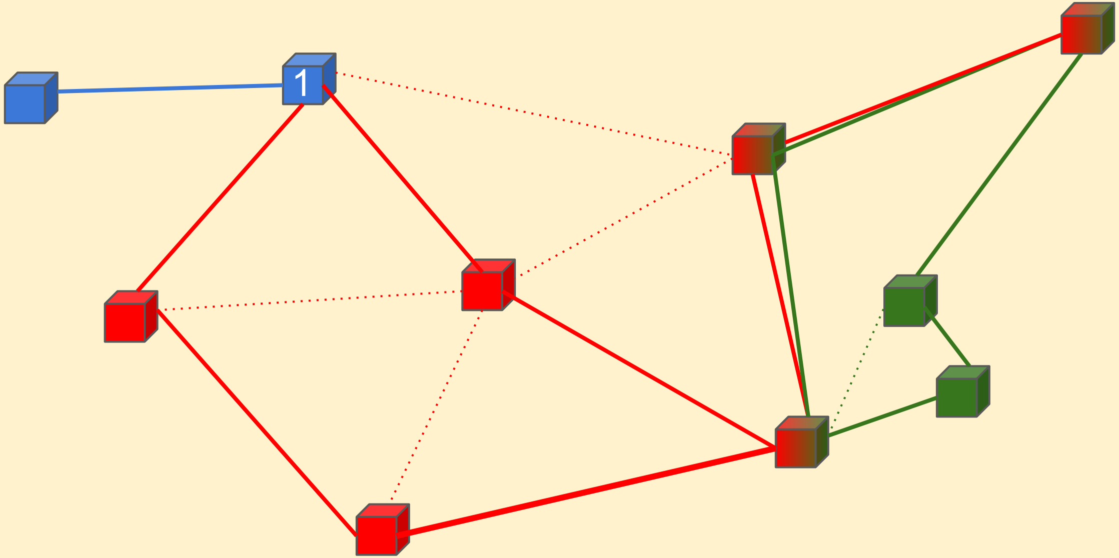
GossipSub



GossipSub



GossipSub Topics

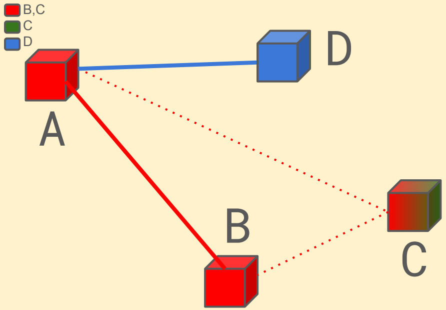


ACL2s Formal Model of GossipSub

State

```
(defdata peer-state     ;; local to a peer
  (record (nts          . nbr-topic-state)
          (mst          . msgs-state)
          (nbr-tctrs    . pt-tctrs-map)
          (nbr-gctrs    . p-gctrs-map)
          (nbr-scores . peer-rational-map)))
```

```
;; state of the entire network
(defdata group (map peer peer-state))
```

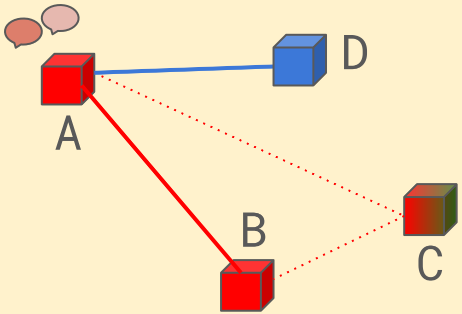


```
nts
 nbr-topicsubs
  (( 🟥 . (B C))
   ( 🟩 . (C))
   ( 🟦 . (D)))

 topic-mesh
  (( 🟥 . (B)))

 topic-fanout
  (( 🟦 . (D)))
 …
```

State

```
(defdata peer-state     ;; local to a peer
  (record (nts          . nbr-topic-state)
          (mst          . msgs-state)
          (nbr-tctrs    . pt-tctrs-map)
          (nbr-gctrs    . p-gctrs-map)
          (nbr-scores . peer-rational-map)))
```

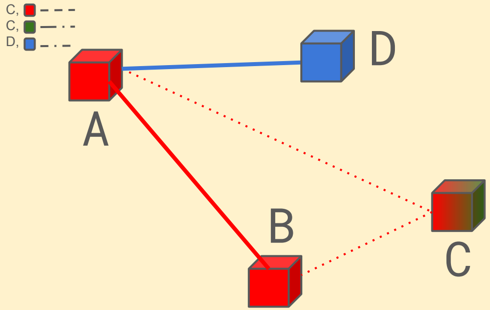
```
;; state of the entire network
(defdata group (map peer peer-state))
```



```
mst
 recently-seen
  (((💬. C) . 5)
   ((💬. B) . 10))

 pld-cache
  ((💬. C)
   (💬. B))

 hwindows
  (1 1)
 …
```

State

```
(defdata peer-state     ;; local to a peer
  (record (nts          . nbr-topic-state)
          (mst          . msgs-state)
          (nbr-tctrs    . pt-tctrs-map)
          (nbr-gctrs    . p-gctrs-map)
          (nbr-scores . peer-rational-map)))


;; state of the entire network
(defdata group (map peer peer-state))
```



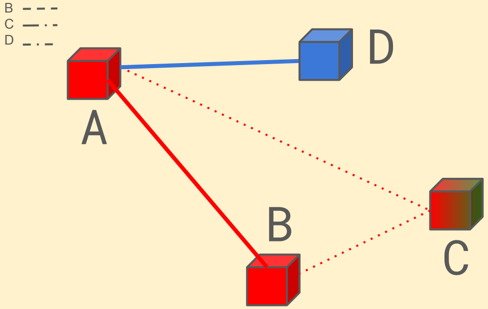
nbr-tctrs

```
                      tctrs
(((C . 🟥) . ((:firstmessagedeliveries . 0) 👍
               (:invalidmessagedeliveries . 0) 👎
               (:meshfailurepenalty . 0) 👎
               (:meshmessagedeliveries . 1) 👍
               (:meshtime . 42)) 👍
                      tctrs
 ((C . 🟩) . ((:firstmessagedeliveries . 324)
               (:invalidmessagedeliveries . 0)
               (:meshfailurepenalty . 0)
               (:meshmessagedeliveries . 330)
               (:meshtime . 377)))

  (D . 🟦) .    ...)
```

State

```
(defdata peer-state      ;; local to a peer
  (record (nts          . nbr-topic-state)
          (mst          . msgs-state)
          (nbr-tctrs    . pt-tctrs-map)
          (nbr-gctrs    . p-gctrs-map)
          (nbr-scores . peer-rational-map)))
```

```
;; state of the entire network
(defdata group (map peer peer-state))
```



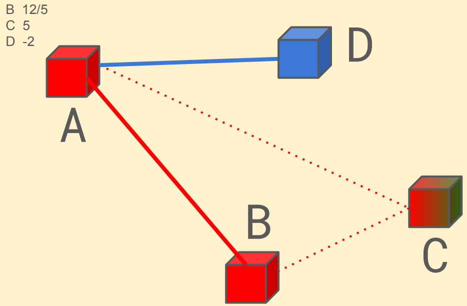
nbr-gctrs

```
        gctrs
((B . ((:apco . 100) 👍
       (:bhvo . 0)   👎
       (:ipco . 0))) 👎
        gctrs
 (C . ((:apco . 500)
       (:bhvo . 20)
       (:ipco . 0)))
        gctrs
 (D . ((:apco . 0)
       (:bhvo . 0)
       (:ipco . 0))))
```

State

```
(defdata peer-state     ;; local to a peer
  (record (nts         . nbr-topic-state)
          (mst         . msgs-state)
          (nbr-tctrs   . pt-tctrs-map)
          (nbr-gctrs   . p-gctrs-map)
          (nbr-scores . peer-rational-map)))
```

```
;; state of the entire network
(defdata group (map peer peer-state))
```



nbr-scores

```
((B . 12/5
 (C . 5)
 (D . -2))
```

Fundamental Security Property

Peers who behave poorly 👎 will be demoted ⬇️ by their neighbors.

Peers who behave better-than-average 👍 will be promoted ⬆️ by their neighbors.

Promotion⬆️/demotion⬇️ is entirely based on local peer behavior👍/👎.

Peer Scoring

Score Calculation

overall-score =

topic-score-cap($\Sigma_{t \in topics}$ topic-score(t))

$\sum_{t \in topics}$ topic-weight(t).(👍(t).weight(t)+… - 👎(t).weight(t)-…))

$+$ 👍.weight + … - 👎.weight - …

Score Calculation

$$
\begin{aligned}
\mathrm{Score(peer)} = \mathrm{TC}\Big(\sum_{t \in \mathrm{topics}} \mathrm{tw}(t) \big(& \; w1(t) * P1(t) \\
& + w2(t) * P2(t) \\
& + w3(t) * P3(t) \\
& + w3b(t) * P3b(t) \\
& + w4(t) * P4(t) \big) \Big) + w5 * P5 \\
& \qquad\qquad\qquad\qquad\qquad + w6 * P6 \\
& \qquad\qquad\qquad\qquad\qquad + w7 * P7
\end{aligned}
$$

P1(t)	time in mesh
P2(t)	first mesh message deliveries
P3(t)	mesh message delivery rate
P3b(t)	mesh message delivery failures
P4(t)	invalid messages
P5	application specific score
P6	IP co-location factor
P7	behavioral penalty

Security Properties

Score function properties for security

	ETH2.0
1) ⬜(topic-score < 0) ⇒ ◇(overall-score < 0)	✖
2) ⬆️ bad performance counters ⇒ ⬇️ overall score	✖
3) ⬆️ good performance counters ⇒ ⬆️ overall score	✓
4) Identical performance counters achieve identical score	✓

Property 1 in ACL2s (without the temporal operators)

```
(property (ptc :pt-tctrs-map pcm :p-gctrs-map p :peer top :topic)
   :hyps (^ (member-equal ‘(,p . ,top)
                           (acl2::alist-keys ptc))
            (> (lookup-score p (calc-nbr-scores-map ptc pcm *eth-twp*))
               0)) ;; overall-score > 0
    :body (> (calcScoreTopic (lookup-tctrs p top ptc) ;; topic-score > 0
             (mget top *eth-twp*)) 0))
```

Stay tuned, for the counter-example of the temporal version appearing shortly!

Why Property 1 failed for ETH2.0

FIRSTMESSAGEDELIVERIES	0
INVALIDMESSAGEDELIVERIES	0
MESHFAILUREPENALTY	0
MESHMESSAGEDELIVERIES	1
MESHTIME	42

FIRSTMESSAGEDELIVERIES	194
INVALIDMESSAGEDELIVERIES	0
MESHFAILUREPENALTY	0
MESHMESSAGEDELIVERIES	200
MESHTIME	147

FIRSTMESSAGEDELIVERIES	182
INVALIDMESSAGEDELIVERIES	0
MESHFAILUREPENALTY	0
MESHMESSAGEDELIVERIES	188
MESHTIME	135

scoring-function

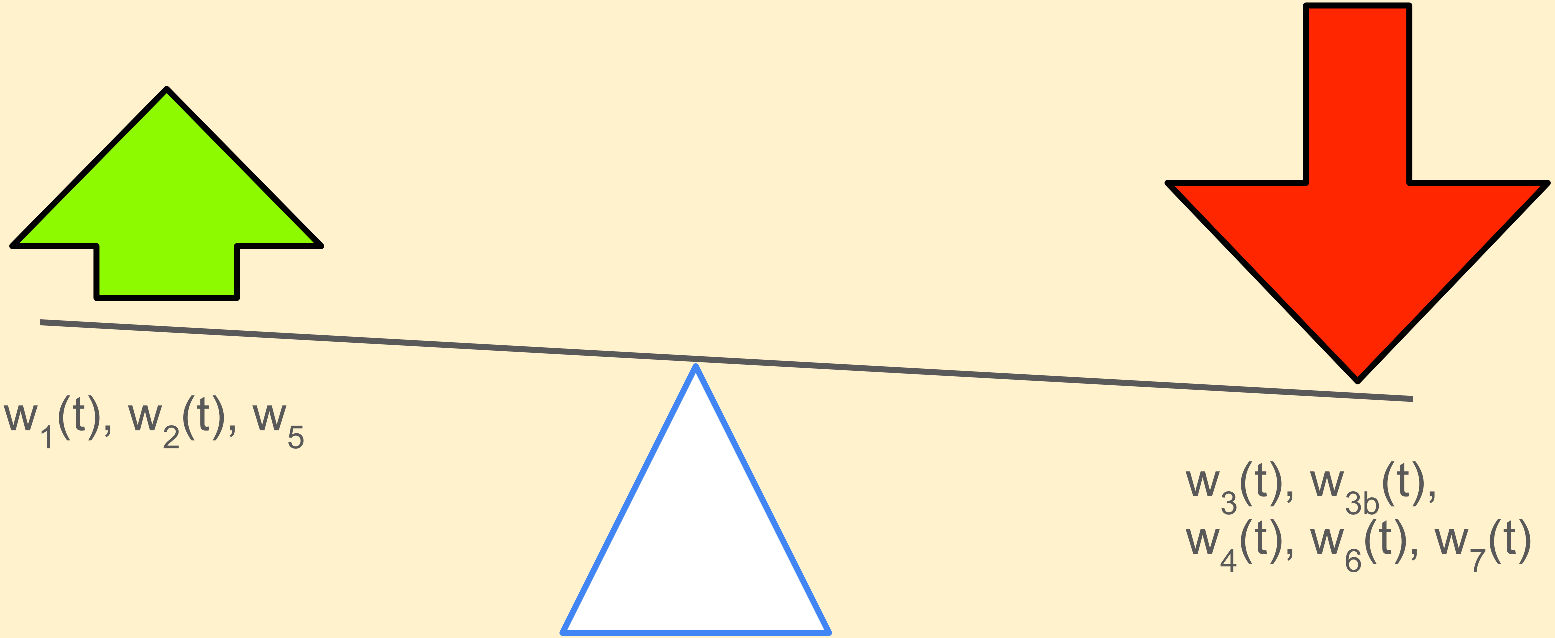
-25

5

22.21

7.78

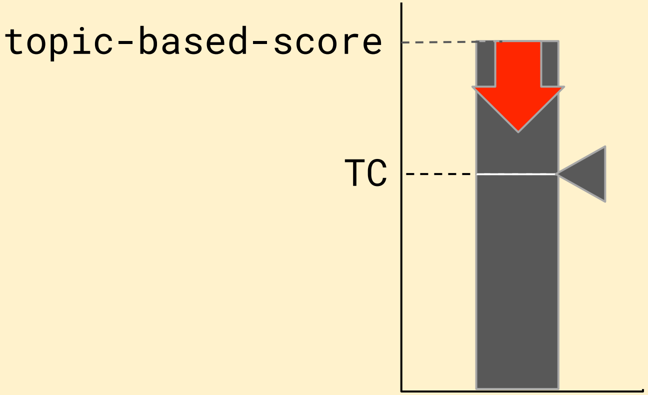
Custom Enumerators for generating Counter-examples



$w_1(t)$, $w_2(t)$, w_5

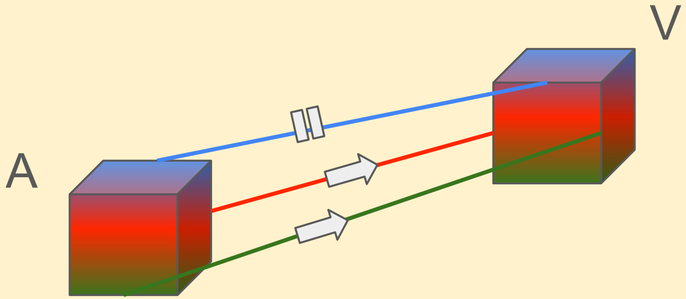
$w_3(t)$, $w_{3b}(t)$,
$w_4(t)$, $w_6(t)$, $w_7(t)$

Why Property 2 failed for ETH2.0



Attack Generation

Attack Gadgets



An AG_1 Attack Gadget

Constructing Attacks

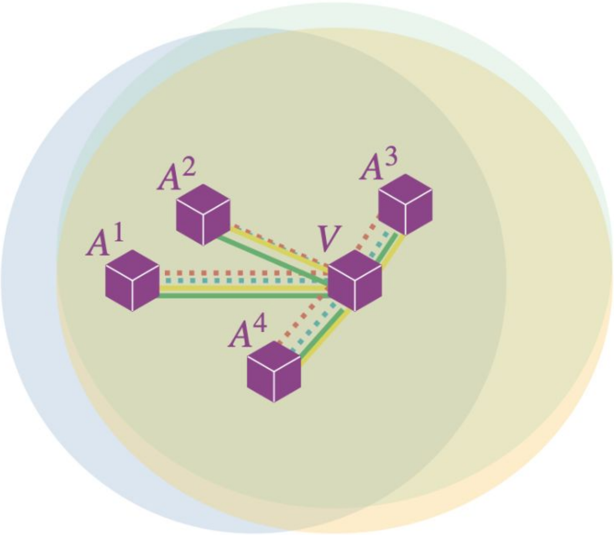


Figure 5: An eclipse attack using AG_2 gadgets

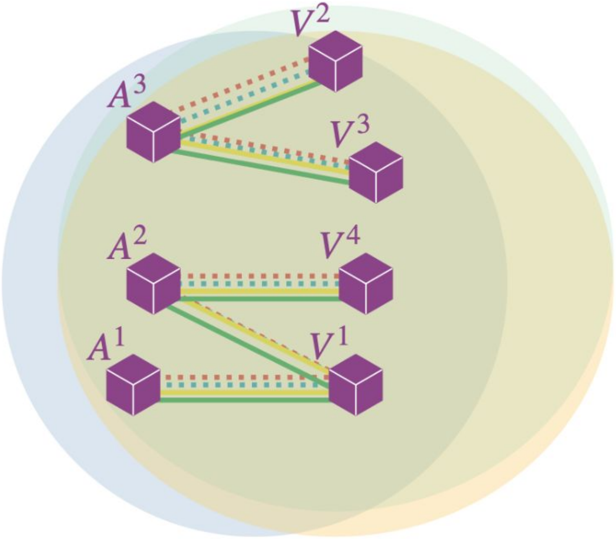
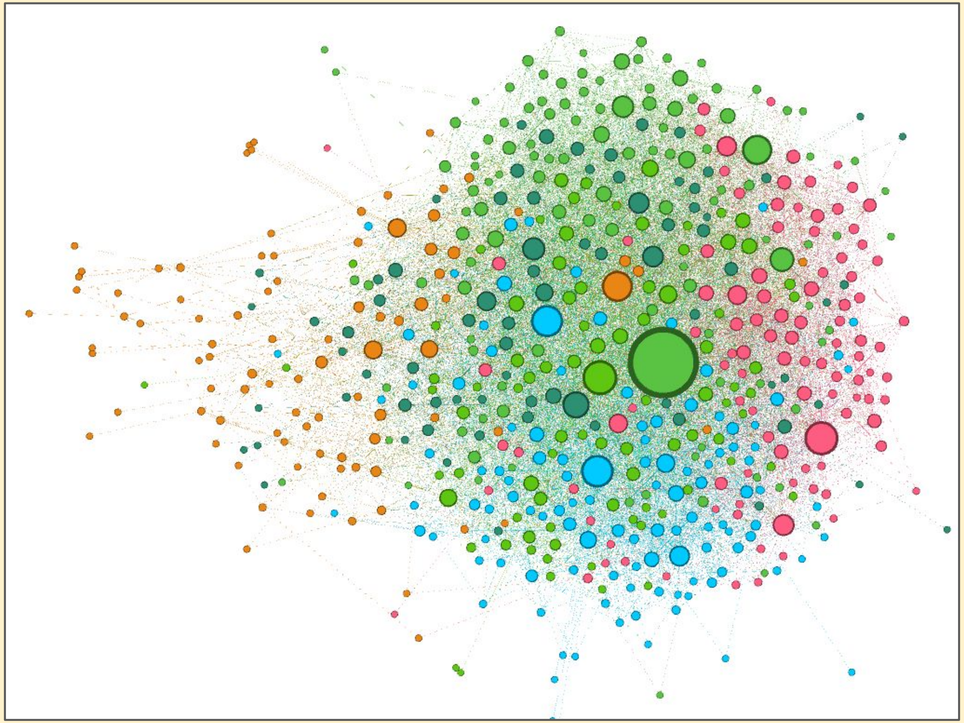


Figure 6: A partition attack using AG_2 gadgets

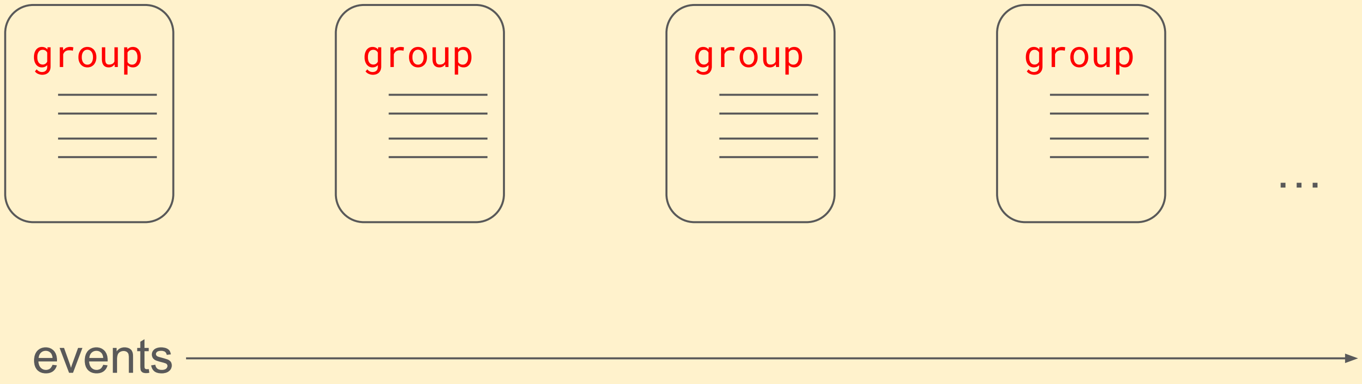
Constructing Attacks on Actual Topologies



Raw data from Kai Li, Yuzhe Tang, Jiaqi Chen, Yibo Wang & Xianghong Liu (2021): TopoShot. In: Internet Measurement Conference

Actual Reaction of Eth Devs to our findings presented in IPFS Camp 2022

Temporal Property 1 when executing an Attack



Limitations

- Properties depend on complex types. Writing helpful enumerators required insight.
- Testing properties for new applications will likewise require writing new custom enumerators.
- And possibly new ways of generating attacks, based on the application being attacked.

Future Work

- Refinement based characterization of libP2P protocols
- Reasoning about application layer on top of the network layer

Questions