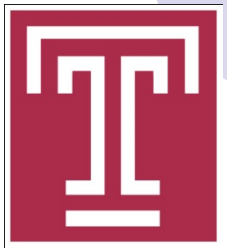
Optimal Filter Assignment Policy Against Transit-link Distributed Denial-of-Service Attack

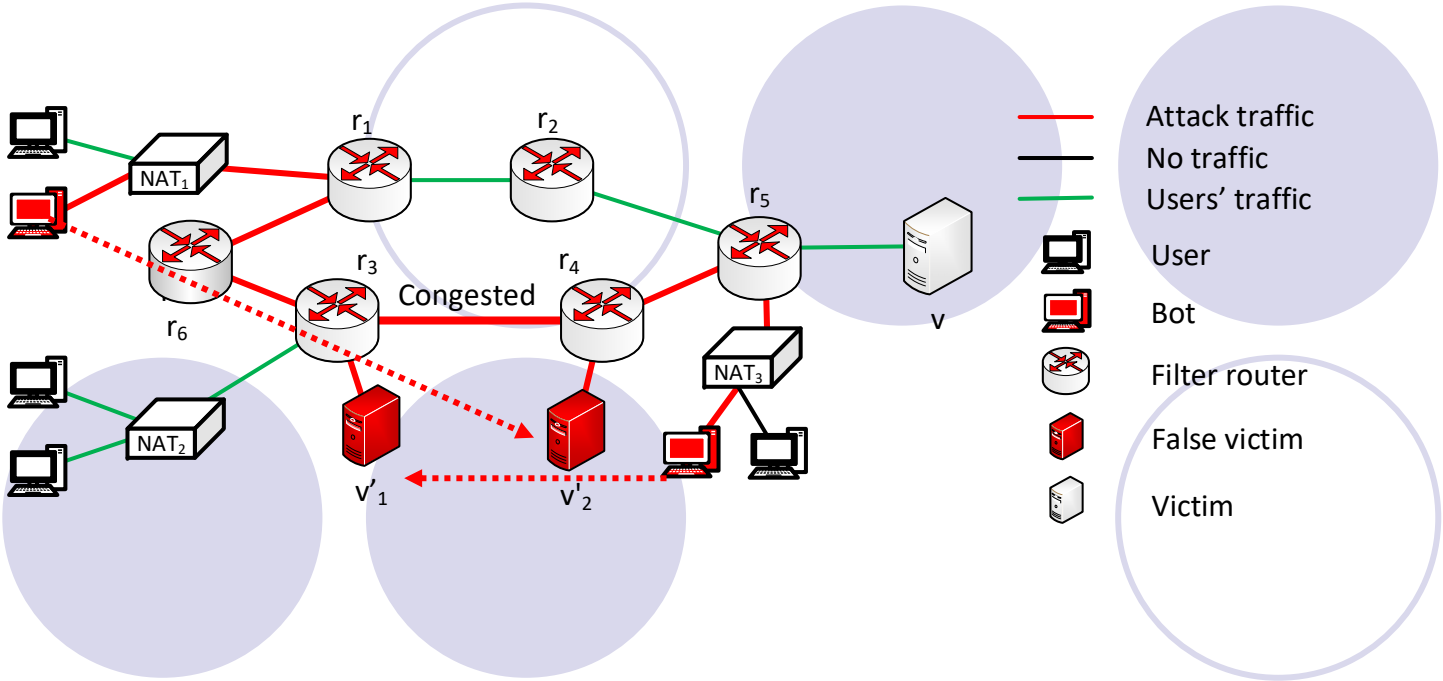
Rajorshi Biswas, Jie Wu
Dept. of Computer and Info. Sciences, Temple University

Wei Chang
Department of Computer Science, Saint Joseph's University

Pouya Ostovari
Charles W. Davidson College of Engineering, San Jose State University

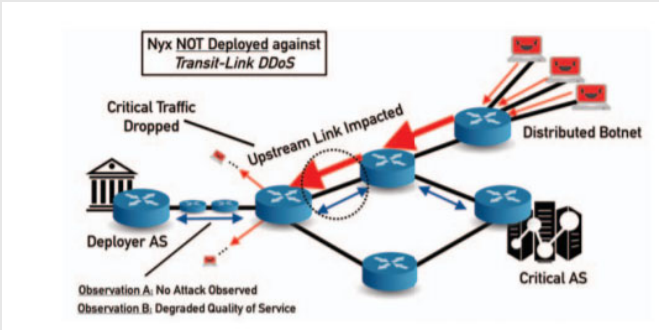


Transit-Link DDoS Attack



- Attackers generated traffic to the false victim.
- Attack traffic congest some links on the way to victim from users.
- Also known as Link-flooding attack.

Previous work

Systems	Limitations
SPIFFY (router level attack packet)  SPIFFY: Inducing Cost Detectability Tradeoffs for Persistent Link-Flooding Attacks (M. Suk Kang et al. NDSS, 2016)	• Increased router overhead.
Routing Around Congestion (BGP-based)  Routing Around Congestion: Defeating DDoS Attacks and Adverse Network Conditions via Reactive BGP Routing (J. M. Smith et al. S&P 2018)	• Victim far from the congested link is tough without additional access. • Hard to detect whether the congestion is created by attacker or user.

Background: Filter Router and Filter

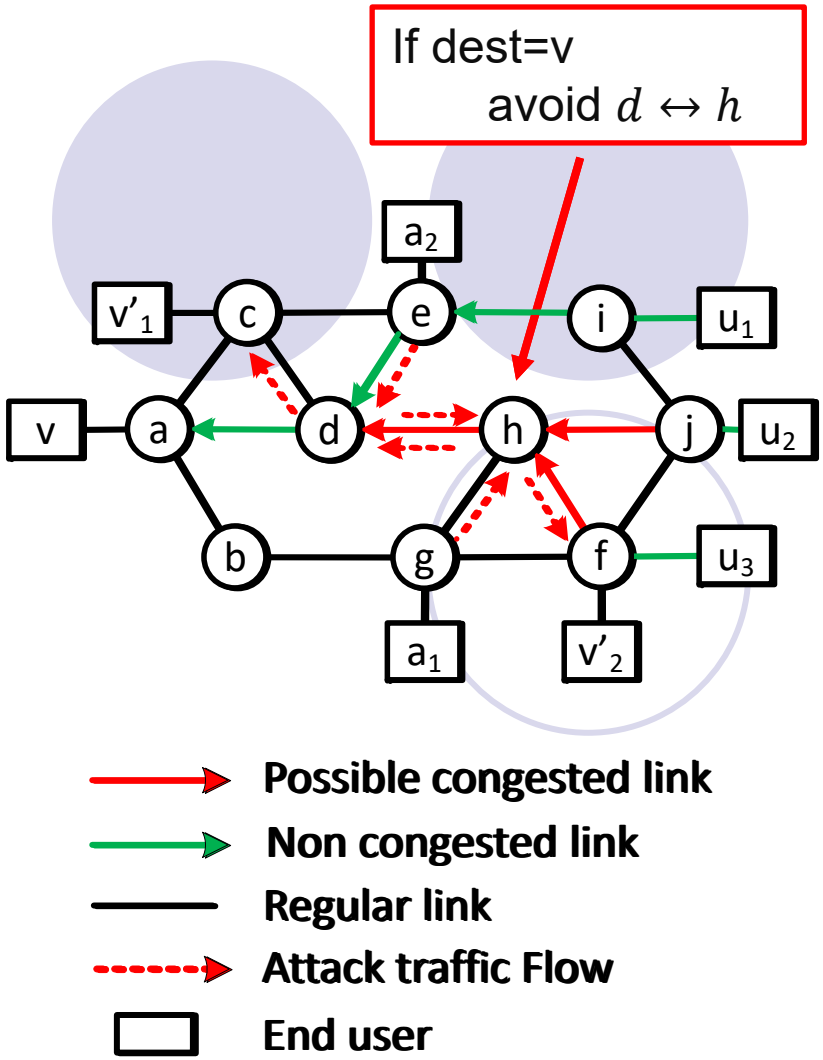
- Assumption: Victim knows the topology.
  - (ISP, packet marking)
- Finds possible congested links.
  - Based on User traffic rate.
- Send filter to Filter router to change route.

Filter Router:

Accepts filter and apply that to block links based on destination.

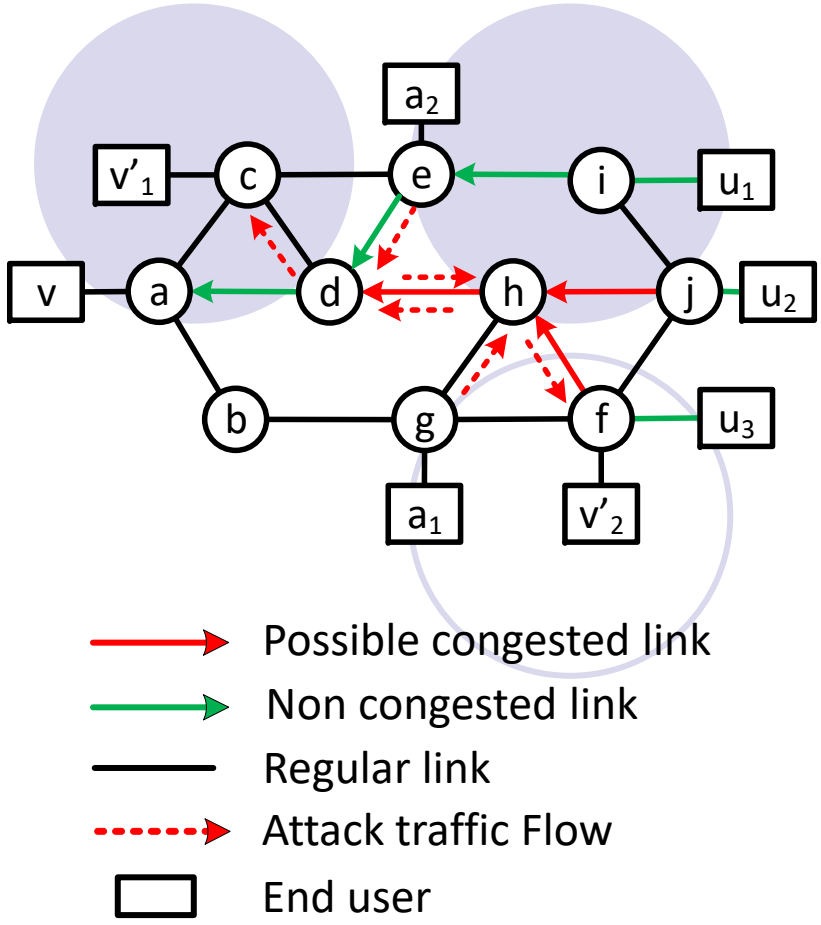
Filter:

Link blocking rule.



Protecting User by Sending Filter

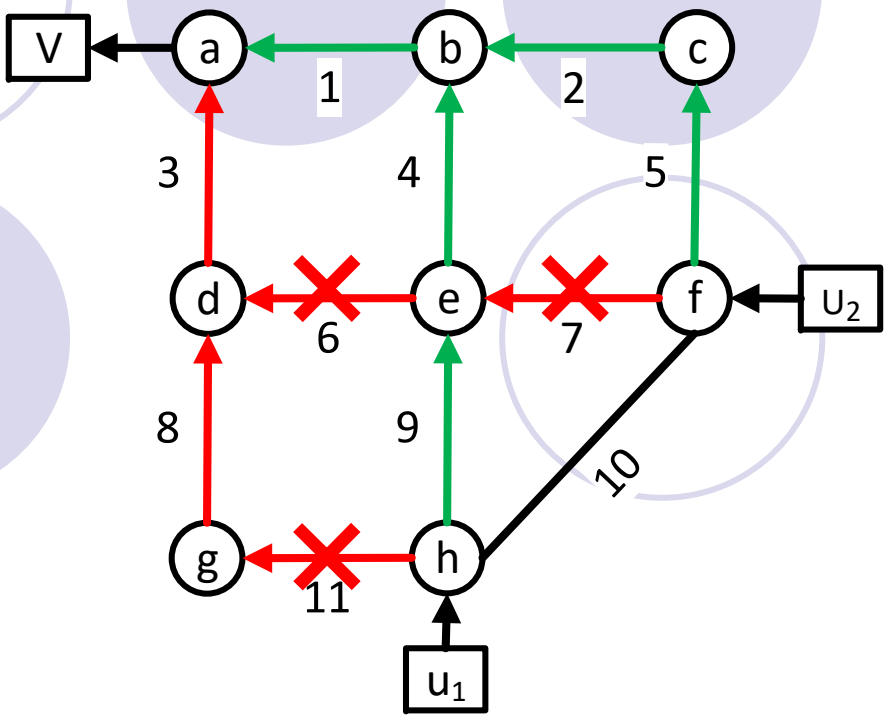
- Assumption:
  - There exist a non congested path from each user.
  - Shortest path routing is used.
- Send filter to block all possible congested links
  - Owner of the FR changes money for applying filter.
- Need to block links wisely.



Problem: Minimizing Blocked Links

- Given topology and possible congested links.
- Block minimum number of links so that all user traffic follow non congested paths.

- Option 1: block 11, 6, 7
  - Three blocks (not minimum)
- Option 2: Block 3,7
  - Two blocks (not minimum)

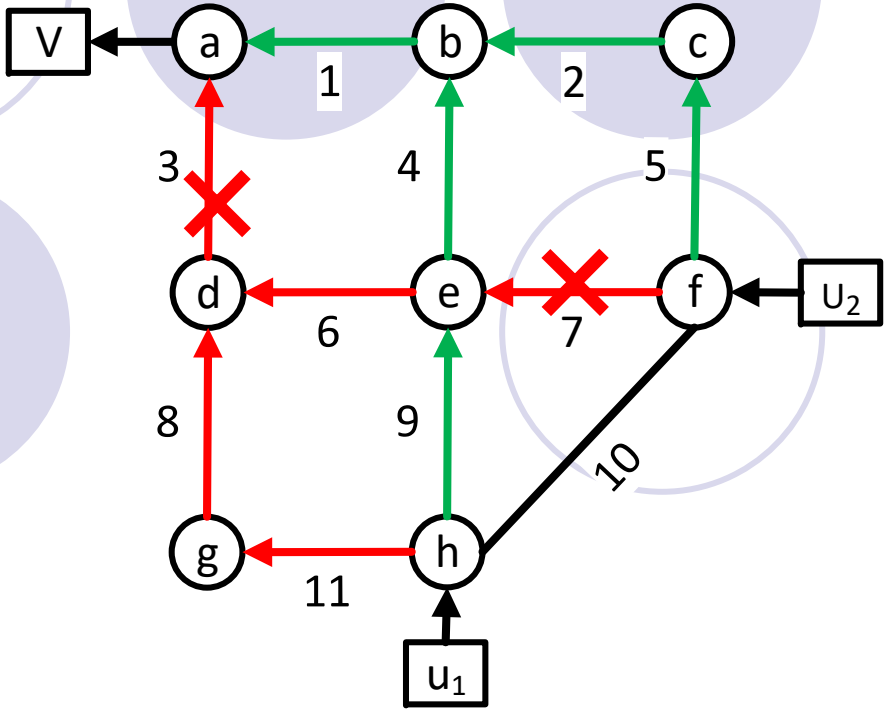


No traffic travels through possible congested links

Problem: Minimizing Blocked Links

- Given topology and possible congested links.
- Block minimum number of links so that all user traffic follow non congested paths.

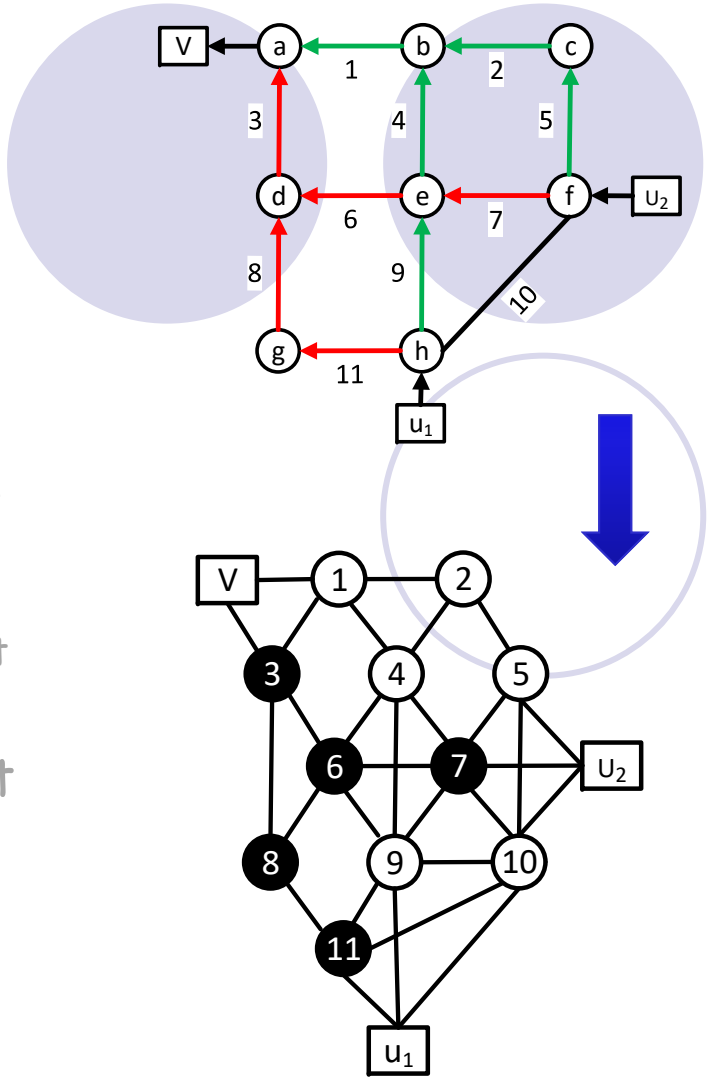
- Option 1: block 11, 6, 7
  - Three blocks (not minimum)
- Option 2: Block 3,7
  - Two blocks (not minimum)



No traffic travels through possible congested links

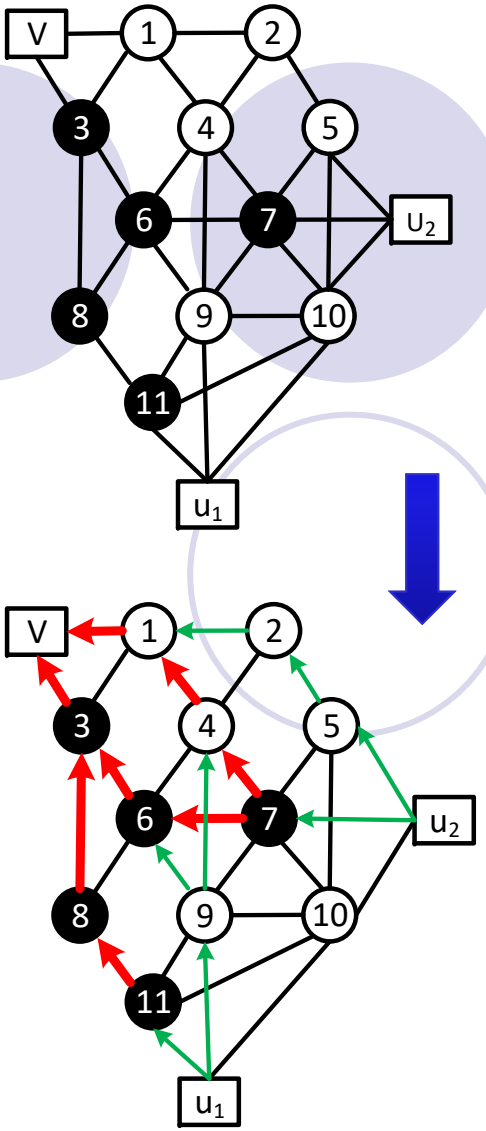
Solution Steps

- Step 1: Transformation to line graph
  - Edges-> Nodes, red links-> black nodes
- Step 2: Create traffic flows
  - From all sources, follow all shortest paths.
- Step 3: Remove white nodes
  - Remove white nodes and concatenate red links
- Step 4: Add super user
  - Add links to the black nodes who have incident green links
- Step 5: Find minimum d-separating set
  - Using Acid and Campos's method



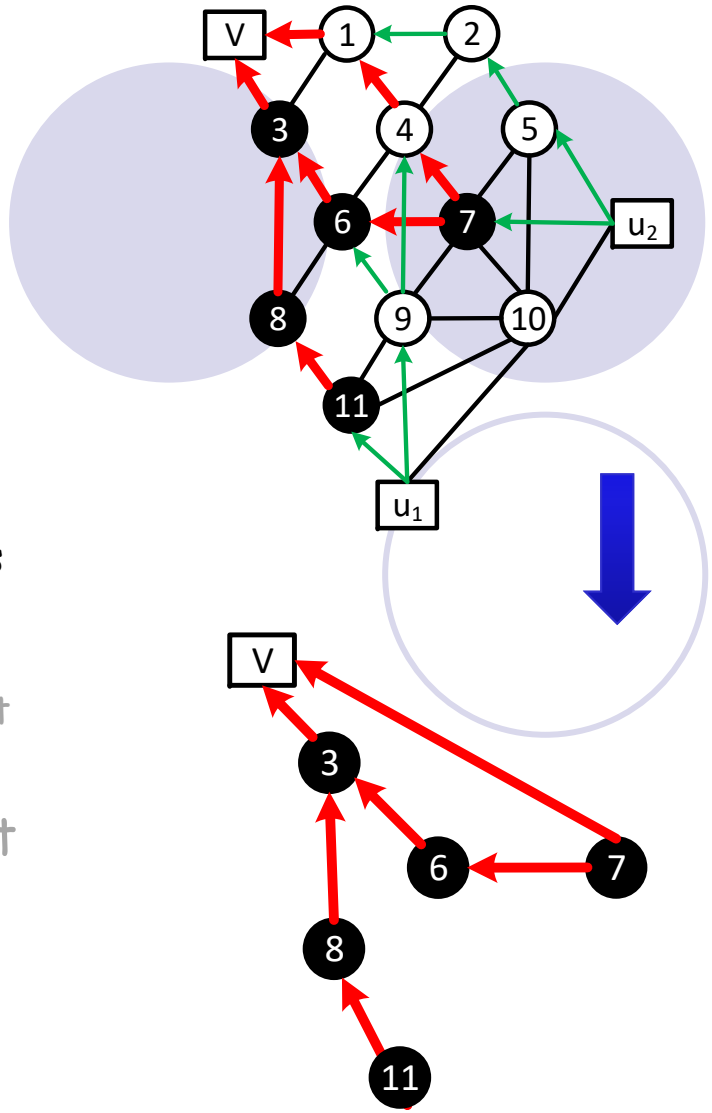
Solution Steps

- Step 1: Transformation to line graph
  - Edges-> Nodes, red links-> black nodes
- Step 2: Create traffic flows
  - From all sources, follow all shortest paths.
- Step 3: Remove white nodes
  - Remove white nodes and concatenate red links
- Step 4: Add super user
  - Add links to the black nodes who have incident green links
- Step 5: Find minimum d-separating set
  - Using Acid and Campos's method



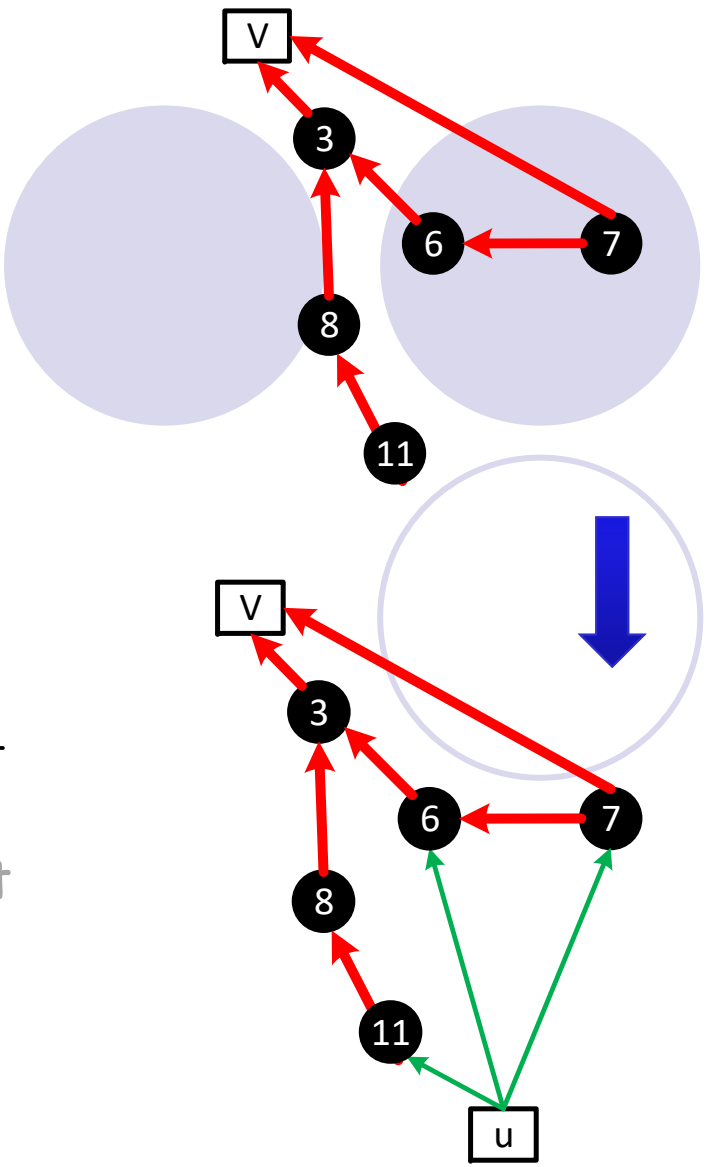
Solution Steps

- Step 1: Transformation to line graph
  - Edges-> Nodes, red links-> black nodes
- Step 2: Create traffic flows
  - From all sources, follow all shortest paths.
- Step 3: Remove white nodes
  - Remove white nodes and concatenate red links
- Step 4: Add super user
  - Add links to the black nodes who have incident green links
- Step 5: Find minimum d-separating set
  - Using Acid and Campos's method



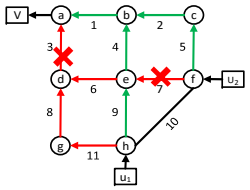
Solution Steps

- Step 1: Transformation to line graph
  - Edges-> Nodes, red links-> black nodes
- Step 2: Create traffic flows
  - From all sources, follow all shortest paths.
- Step 3: Remove white nodes
  - Remove white nodes and concatenate red links
- Step 4: Add super user
  - Add links to the black nodes who have incident green links
- Step 5: Find minimum d-separating set
  - Using Acid and Campos's method

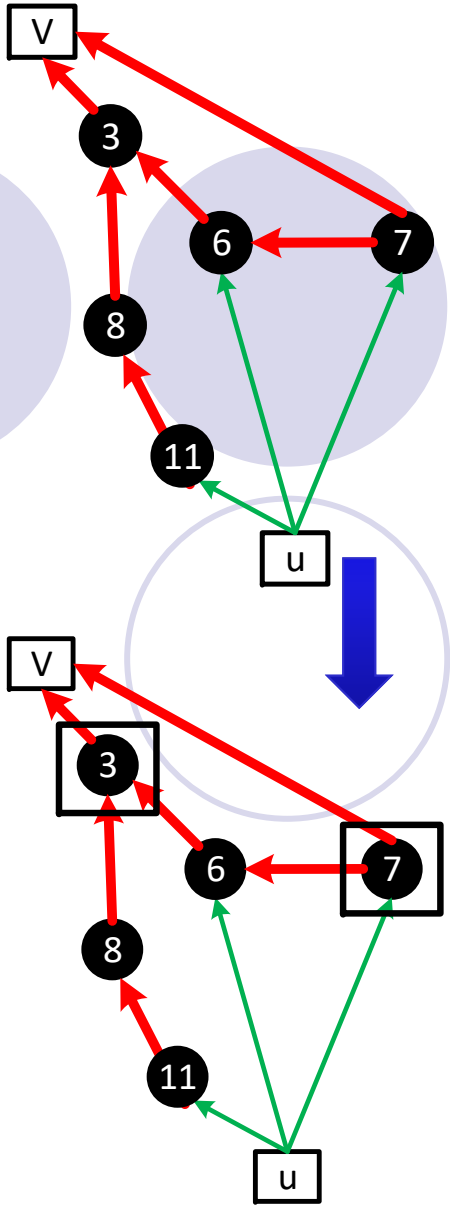


Solution Steps

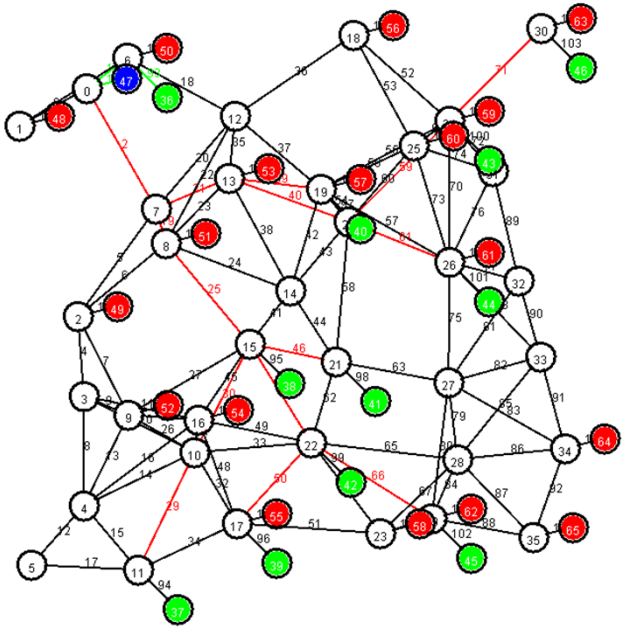
- Step 1: Transformation to line graph
  - Edges-> Nodes, red links-> black nodes
- Step 2: Create traffic flows
  - From all sources, follow all shortest paths.
- Step 3: Remove white nodes
  - Remove white nodes and concatenate red links
- Step 4: Add super user
  - Add links to the black nodes who have incident green links
- Step 5: Find minimum d-separating set
  - Using Acid and Campos's method



Solution: Block links 3,7



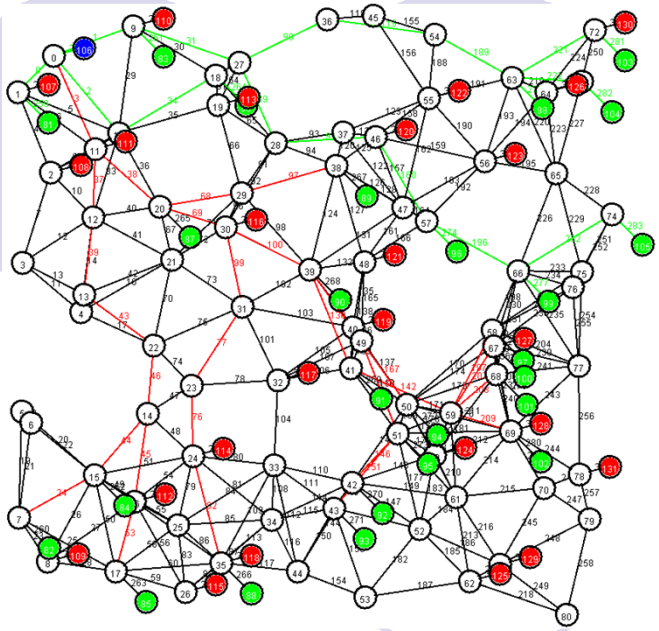
Simulation: Random Topology Generation



Attacker

User

Victim



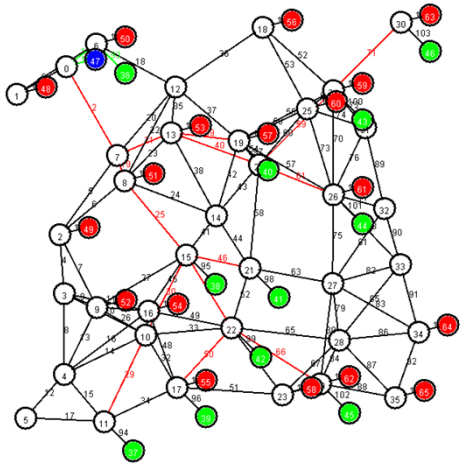
Topology I
Nodes: 65, Links: 123

Topology I
Nodes: 131, Links: 310

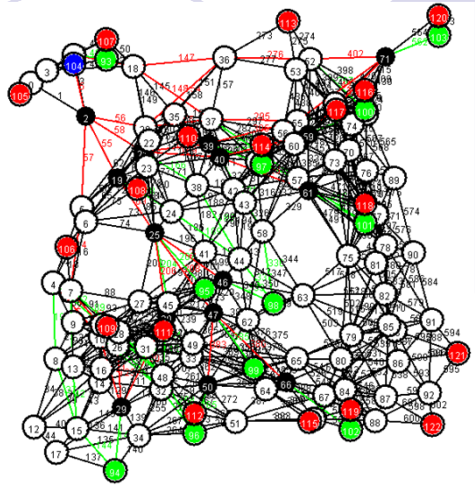
Unite disk graph, Randomly placed nodes (uniform),
Area: 500x500, Neighborhood radius: 70, 50% attackers (chosen randomly)

Simulation: Graph Transformations

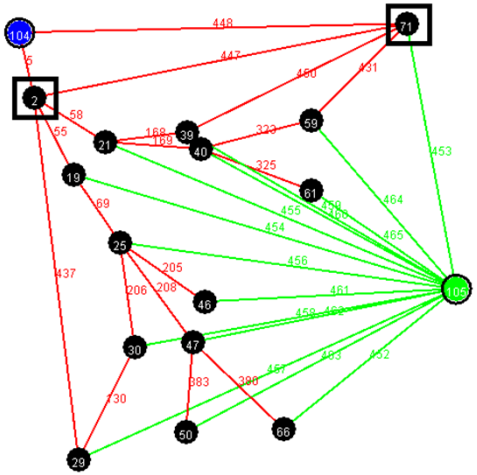
Topology I



Original Topology

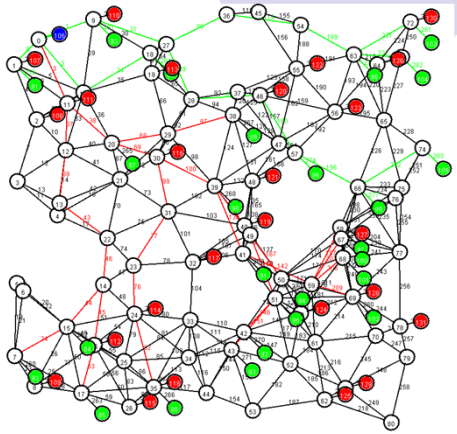


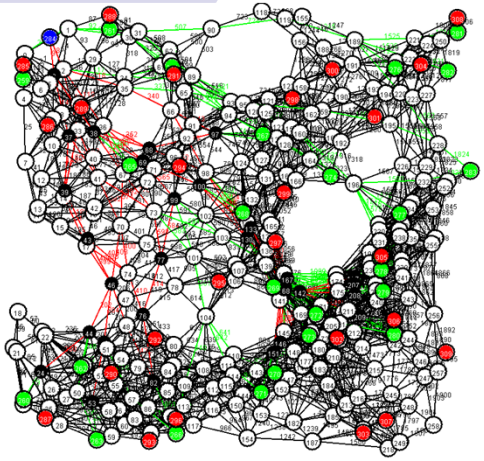
Line graph

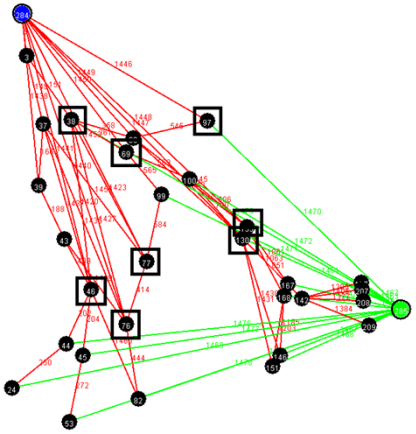


Final Graph

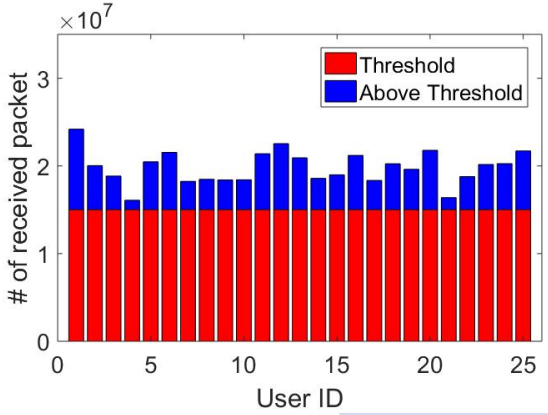
Topology II



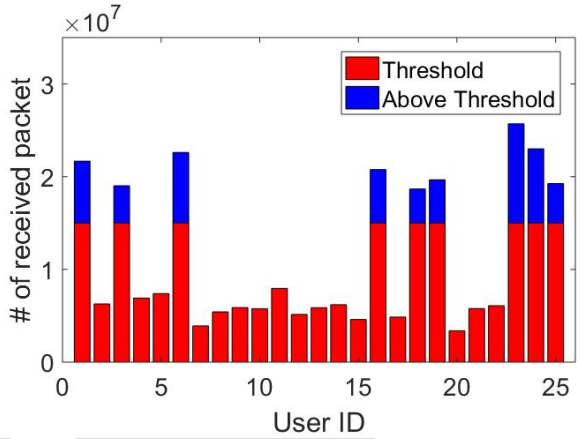




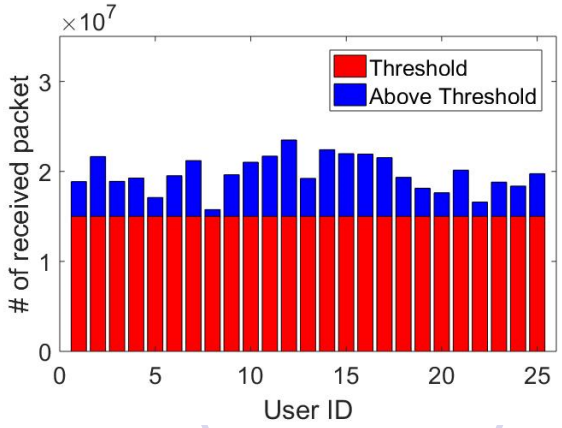
Effect of Filter Deployment



Before attack



After attack



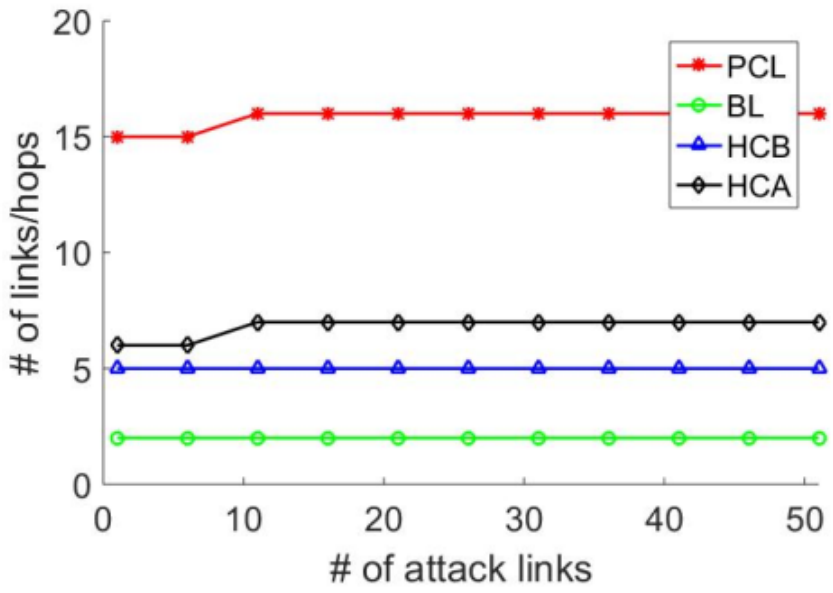
After defense

25 Users, Topology I used

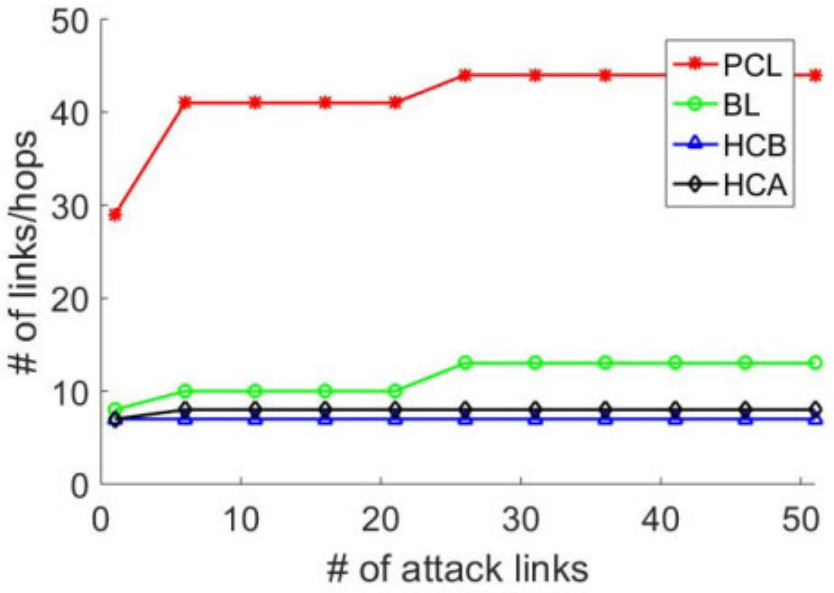
After defense all the users' data rate is above threshold

Effect of Number of Attacked Links

Topology I



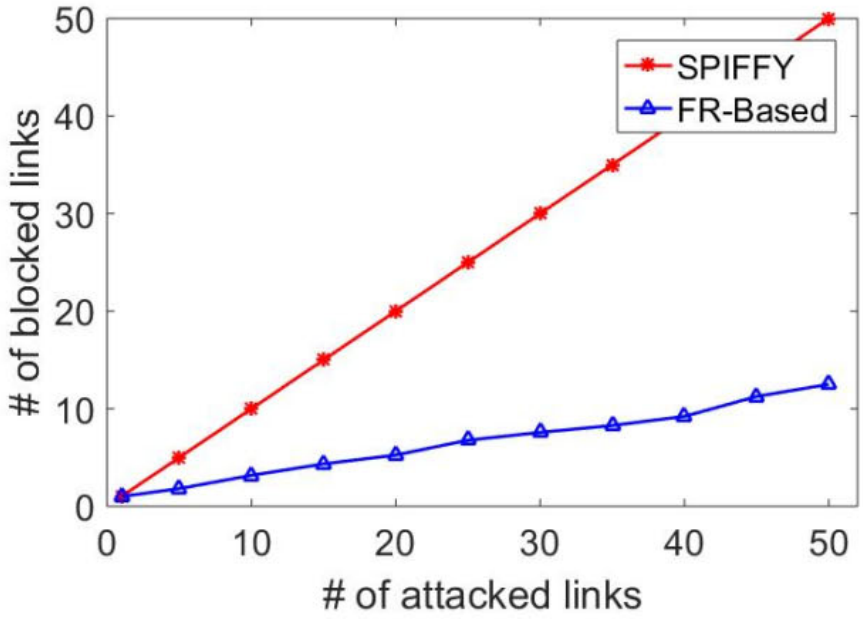
Topology II



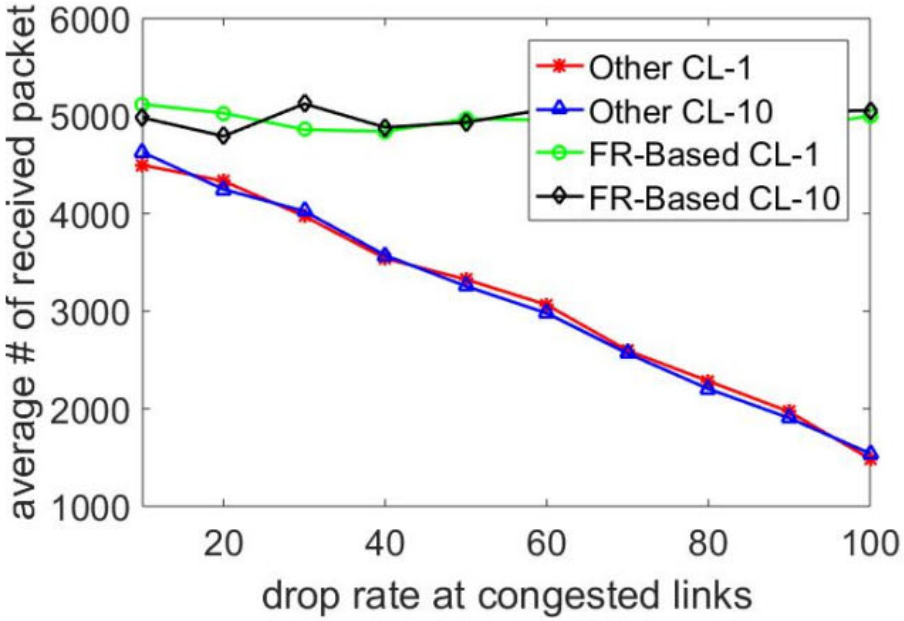
Links to block is much greater than possible congested links are much greater than

Comparison With Others

Topology I



Topology II

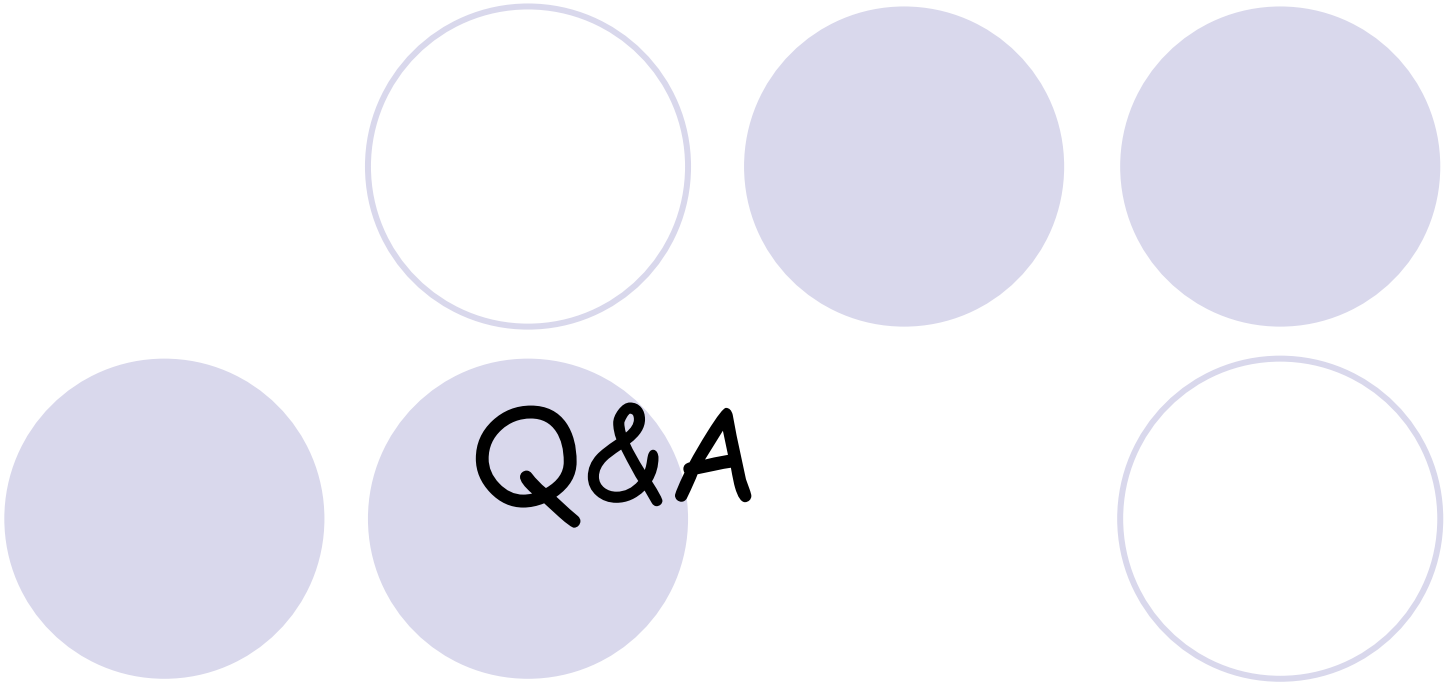


Number of blocked links are always less than SPIFFY

The average number of receied packets is not affected by drop rate at links

Summery

- The filter router-based link blockage can defense against transit-link ddos attack better than other approaches.
- The links needs to block for redirecting the user traffic through non congested way is much less than the possible congested links.


Q&A