

In-Network Filtering of Distributed Denial-of-Service Traffic with Near-Optimal Rule Selection

Devkishen Sisodia, Jun Li, Lei Jiao
{dsisodia, lijun, jiao}@cs.uoregon.edu



UNIVERSITY
OF OREGON



CENTER FOR CYBER
SECURITY & PRIVACY

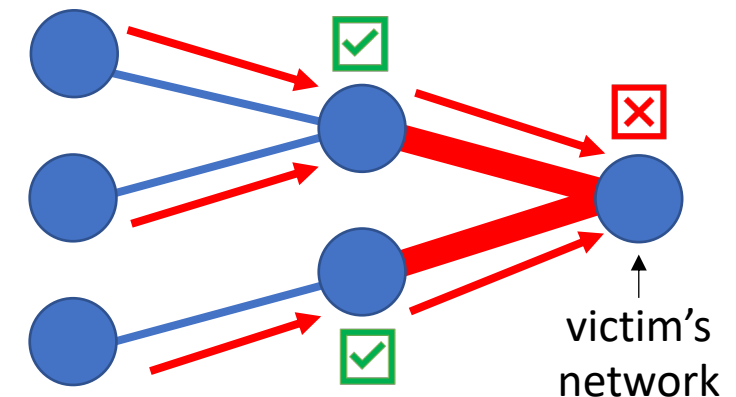
Outline

- Introduction
- Offer-Based Model
- Rule Selection Problem
- ACO-Based Rule Selection Algorithm
- Evaluation
- Conclusion

Introduction

Large-Scale DDoS Attacks

- Large scale distributed denial-of-service (DDoS) attacks are on the rise
 - Oct 2016: 1.2 Tbps (terabit per second)
 - Feb 2018: 1.3 Tbps
 - Mar 2018 : 1.7 Tbps
 - Jan 2019 : 500 Mpps (million packets per second)
 - Apr 2019: 580 Mpps
 - Feb 2020: 2.3 Tbps
- Victim-end defense approaches: insufficient in mitigating large volume attacks
 - Alternative: **in-network filtering** approaches



In-Network Filtering

- Filter traffic at multiple locations on the Internet
- General approach:
 - A **DDoS defense agent** generates DDoS-filtering rules
 - Places them at **DDoS-filtering networks** across the Internet
 - DDoS defense agent: victim
 - DDoS-filtering network: strategically located transit networks or scrubbing centers
- Plethora of papers on in-network filtering approaches
 - All surveyed papers follow the **directive-based model**

Mayday: Distributed Filtering for Internet Services

SOS: An Architecture For Mitigating DDoS Attacks

**Active Internet Traffic Filtering:
Real-Time Response to Denial-of-Service Attacks**

An Edge-to-Edge Filtering Architecture Against DoS

**To Filter or to Authorize: Network-Layer DoS Defense
Against Multimillion-node Botnets**

**DrawBridge—Software-Defined DDoS-Resistant Traffic
Engineering**

**MiddlePolice: Toward Enforcing Destination-Defined
Policies in the Middle of the Internet**

SENSS Against Volumetric DDoS Attacks

Stellar: Network Attack Mitigation using Advanced Blackholing

Christoph Dietzel
TU Berlin/DE-CIX

Matthias Wichtlhuber
DE-CIX

Georgios Smaragdakis
TU Berlin

Anja Feldmann
Max Planck Institute for Informatics

Directive-Based Model for In-Network Filtering

- Each DDoS-filtering network is obliged to deploy filtering rules
- Two main optimization problems:
 - Rule generation: How to generate filtering rules given incoming traffic?
 - Rule placement: Which DDoS-filtering networks to select to deploy generated rules?
- Assumptions:
 1. DDoS-filtering networks are willing and able to deploy generated rules
 2. Defense agent has complete knowledge of the filtering capabilities at the filtering networks
- Advantage: simplifies the defense agent's decision process
- Disadvantage: assumptions may not hold in the real-world

Questions

- Is there a better operational model for in-network DDoS filtering?
 - Yes: **offer-based model**
- If so, is there a new optimization problem associated with this model?
 - Yes: **rule selection problem**
- If so, how can we solve this problem?
 - **Ant Colony Optimization (ACO)-based rule selection algorithm**

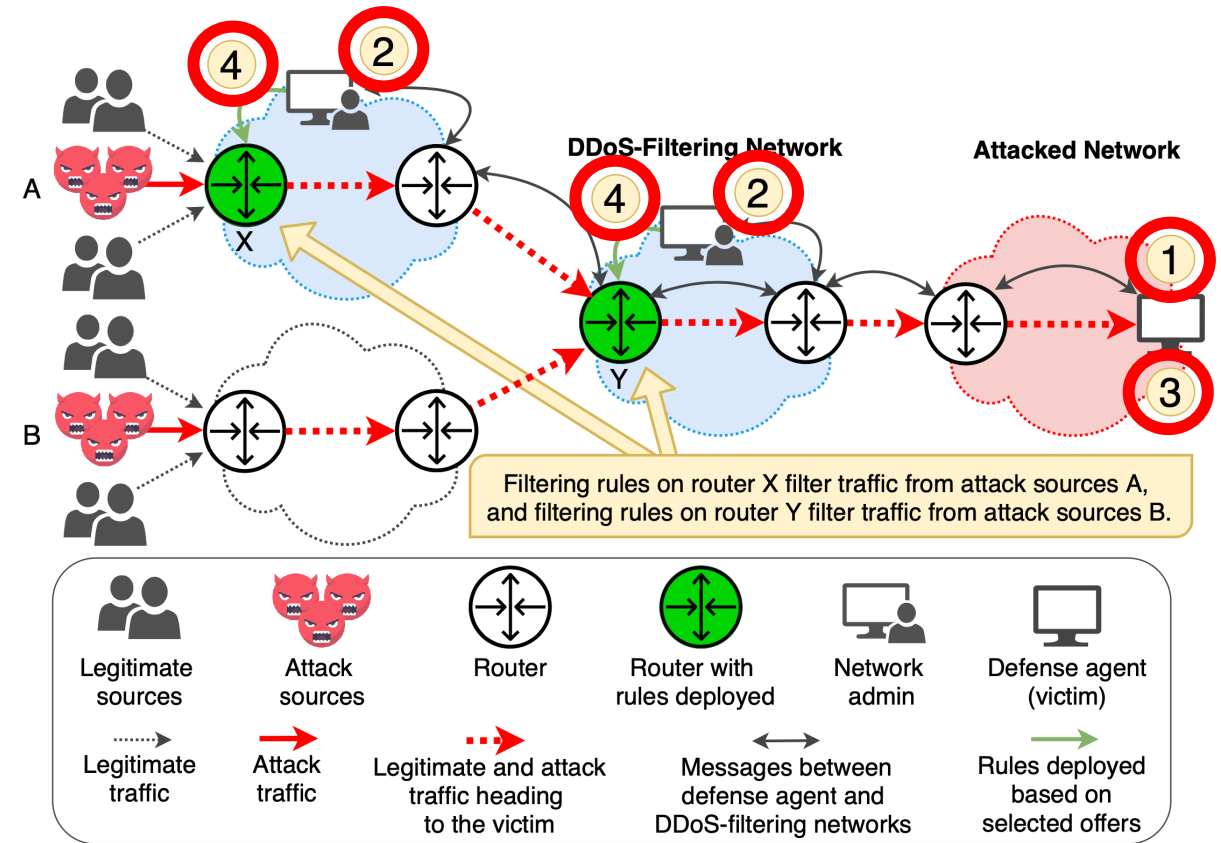
Offer-Based Model

Overview

- Allows the defense agent to express its filtering needs
- Plethora of mechanisms for filtering DDoS traffic:
 - Access control lists (ACLs)
 - Berkeley Packet Filters (BPFs)
 - Remotely Triggered Black Hole (RTBH) signals
 - BGP FlowSpec rules
 - SDN rules
- Focus of this paper: filtering rules based on source IP prefixes (e.g., 162.243.141.0/24)

Operational Model

- Step 1: defense agent generates rules
- Step 2: filtering networks create offers
 - Offer: a set of rules a filtering network is willing to deploy on behalf of the defense agent
- Step 3: defense agent selects offers
- Step 4: filtering networks deploy rules in selected offers



Offer-Based Model vs. Directive-Based Model

- Both models allow a defense agent to express filtering rules to all participating filtering networks
- However, only the offer-based model allows all participating DDoS filtering networks to decide which rules they deploy
- Offer-based model advantages:
 - Removes assumptions made by the directive-based model
 - More suitable for the real-world
- Offer-based model disadvantage:
 - A new optimization problem arises: rule selection

The Need for Rule Selection

- Significant drawback to source IP-based filtering: limited number of rules can be deployed at defending networks
 - Scarcity of memory space on routers/switches
 - Most high-end routers today only have enough TCAM space to support a few thousand filtering rules
- Case in point: Mirai
 - 50 million unique IP addresses spread all across the world
 - Infeasible to deploy a filtering rule for each /32 IP address -- very expensive!
- Therefore, defense agent must aggregate rules
 - Ex: multiple /32 -> one /24; multiple /24 -> one /16
 - Aggregation leads to collateral damage!

Rule Selection Optimization Problem

Maximize the amount of DDoS traffic filtered, while limiting the amount of collateral damage incurred and money spent on deploying filtering rules.

Rule Selection Problem

Overview

- Three main factors a defense agent must consider when selecting an offer:
 - Efficacy of the offer
 - Collateral damage incurred by the offer
 - Price of the offer
- In this paper, we focus on maximizing the defense efficacy, while keeping the maximum total collateral damage and the maximum amount of money spent on defense as constraints

Formulation

$$\max \sum_{k \in K} a_k \max_{j \in J_k, i \in I_j} \{x_{ij} b_{ijk}\}$$

maximize the total amount of attack traffic filtered

$$\text{s.t. } \sum_{u \in U} a_u \max_{j \in J_u, i \in I_j} \{x_{ij} b_{iju}\} \leq W_c$$

collateral damage constraint

$$\sum_{j \in J} \sum_{i \in I_j} P_{ij} x_{ij} \leq W_b$$

budget constraint

$$\sum_{i \in I_j} x_{ij} \leq 1, \quad \forall j \in J$$

limit to 1 selected offer per network

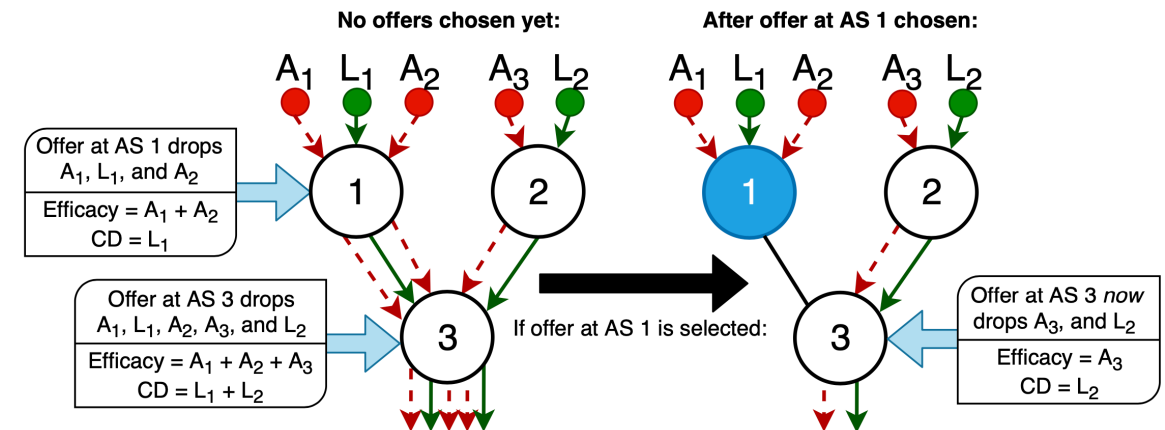
$$x_{ij} \in \{0, 1\}, \quad \forall j \in J, \forall i \in I.$$

offers are atomic

Notations	Definitions
$j \in J$	network j in set of all networks J
$i \in I$	offer i in set of all offers I
$I_j \subseteq I$	set of all offers provided by network j
$k \in K$	attack flow k in set of all attack flows K
$u \in U$	legitimate flow u in set of all legitimate flows U
$J_k \subseteq J$	set of networks that flow k passes through
$J_u \subseteq J$	set of networks that flow u passes through
a_k	amount of traffic belonging to attack flow k
b_{ijk}	binary: whether offer i from network j can filter k
a_u	amount of traffic belonging to legitimate flow u
b_{iju}	binary: whether offer i from network j can filter u
x_{ij}	binary: whether to select offer i from network j
P_{ij}	price of offer i from network j
W_c	defense agent's collateral damage threshold
W_b	defense agent's budget

Challenges

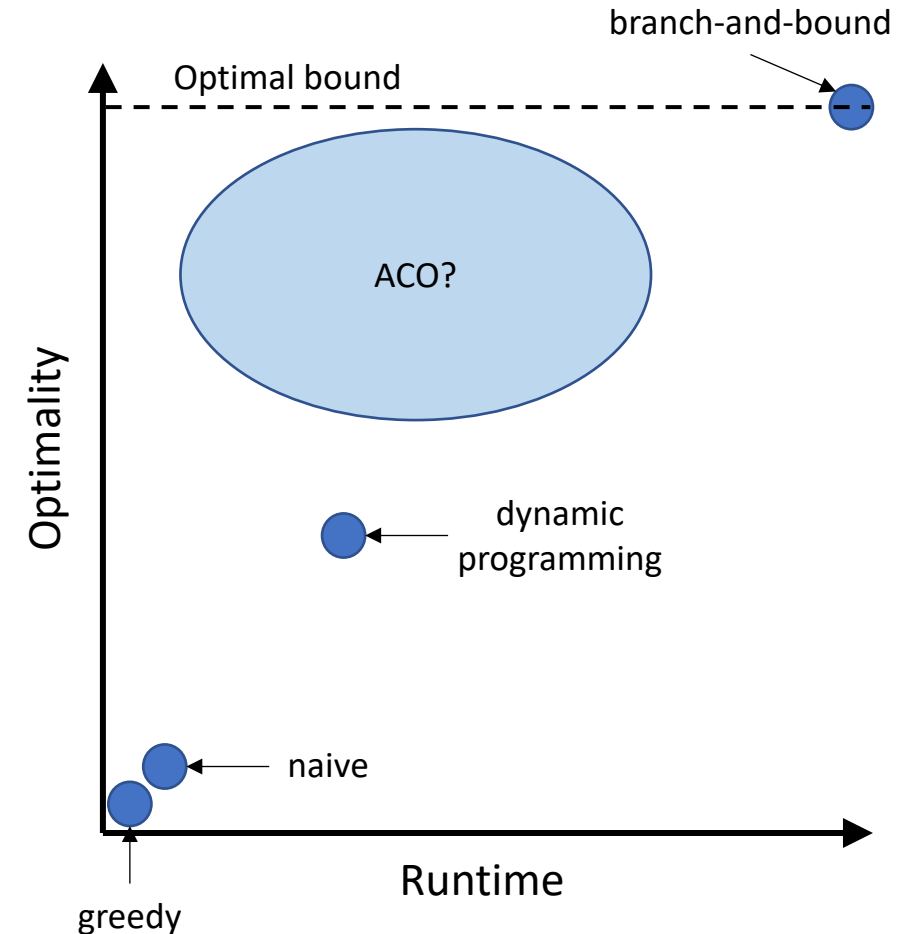
- NP-hard 0-1 multidimensional knapsack problem with value-dependent items
- Offers are value-dependent items
- Unlikely to be solved in pseudo-polynomial time
- Can use algorithms for the general 0-1 knapsack problem as bases
 - Greedy & Naïve
 - Dynamic Programming
 - Branch-and-Bound
 - Ant Colony Optimization



ACO-Based Rule Selection Algorithm

Analysis of Classical Algorithms



- Greedy & naive algorithms
 - Advantage: linear time complexity (short runtimes)
 - Disadvantage: perform relatively poorly in most cases
- Branch-and-bound-based algorithm
 - Advantage: optimal
 - Disadvantage: exponential time complexity (extremely long runtime)
- Dynamic programming-based algorithm
 - Advantage: outperforms greedy and naive, significantly better runtime than branch-and-bound
 - Disadvantage: suboptimal



Overview of the ACO Framework

- Inspired by the foraging behavior of some ant species
- Iterative algorithm
 - Each cycle, ants traverse a graph
 - Each ant builds a solution by walking from node to node
 - An ant chooses the next node partly based on the amount of pheromone laid on the path
 - At the end of a cycle, certain amount of pheromone is evaporated based on quality of the solution
 - Thus, ants in future cycles will be more attracted to solutions like the best ones previously constructed
 - Overall best solution is chosen at the end of last cycle
- Challenge: Cannot be directly applied to the rule selection problem
 - Why?: correlated nature of offers and their potential for overlapping
- Our contribution: develop an ACO-based algorithm for the rule selection problem
 - First time the classical ACO framework has been adapted and applied to the domain of in-network DDoS defense

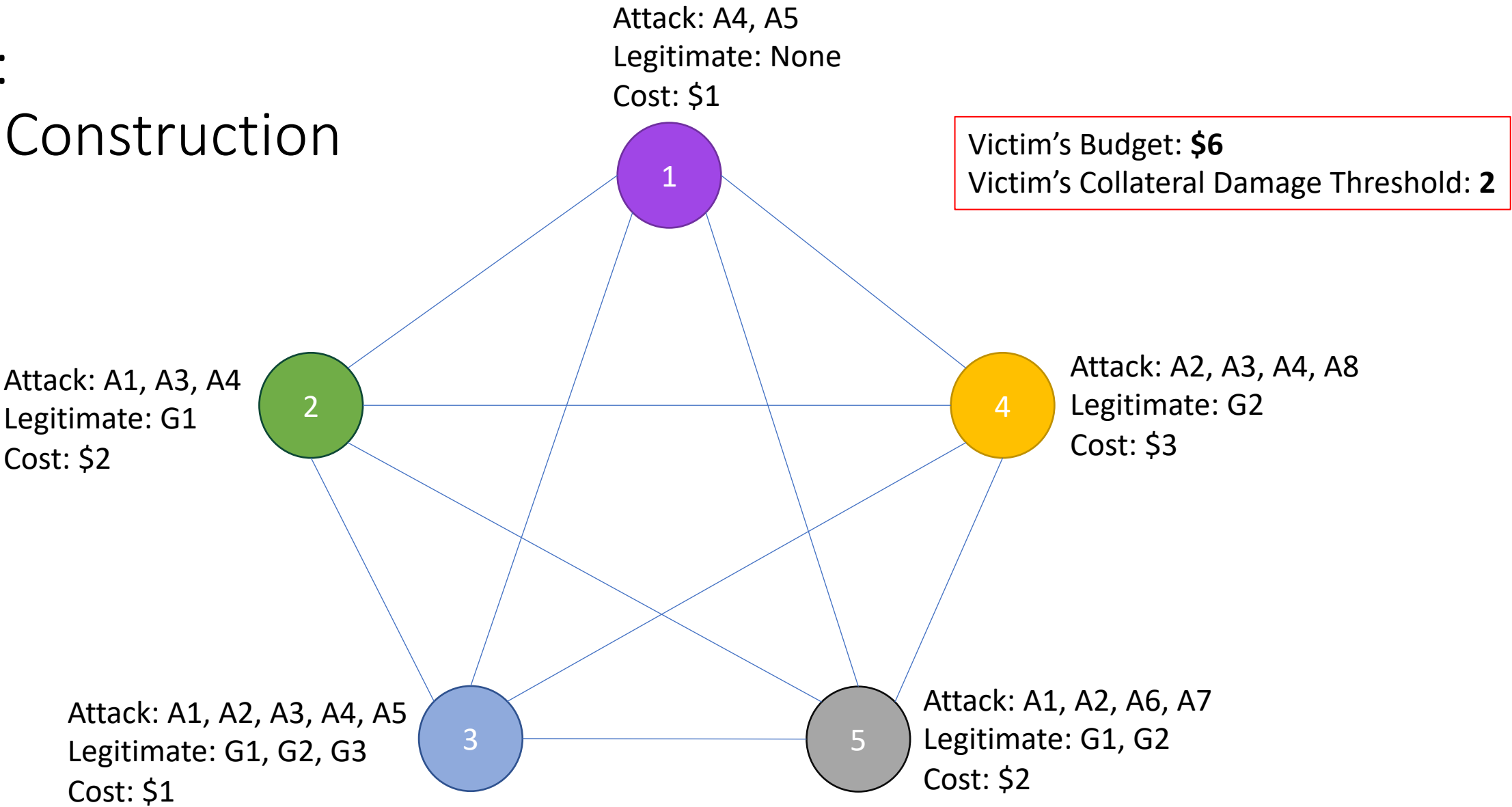
Example

-  Attack: A4, A5
Legitimate: None
Cost: \$1
-  Attack: A1, A3, A4
Legitimate: G1
Cost: \$2
-  Attack: A1, A2, A3, A4, A5
Legitimate: G1, G2, G3
Cost: \$1
-  Attack: A2, A3, A4, A8
Legitimate: G2
Cost: \$3
-  Attack: A1, A2, A6, A7
Legitimate: G1, G2
Cost: \$2

Victim's Budget: **\$6**
Victim's Collateral Damage Threshold: **2**

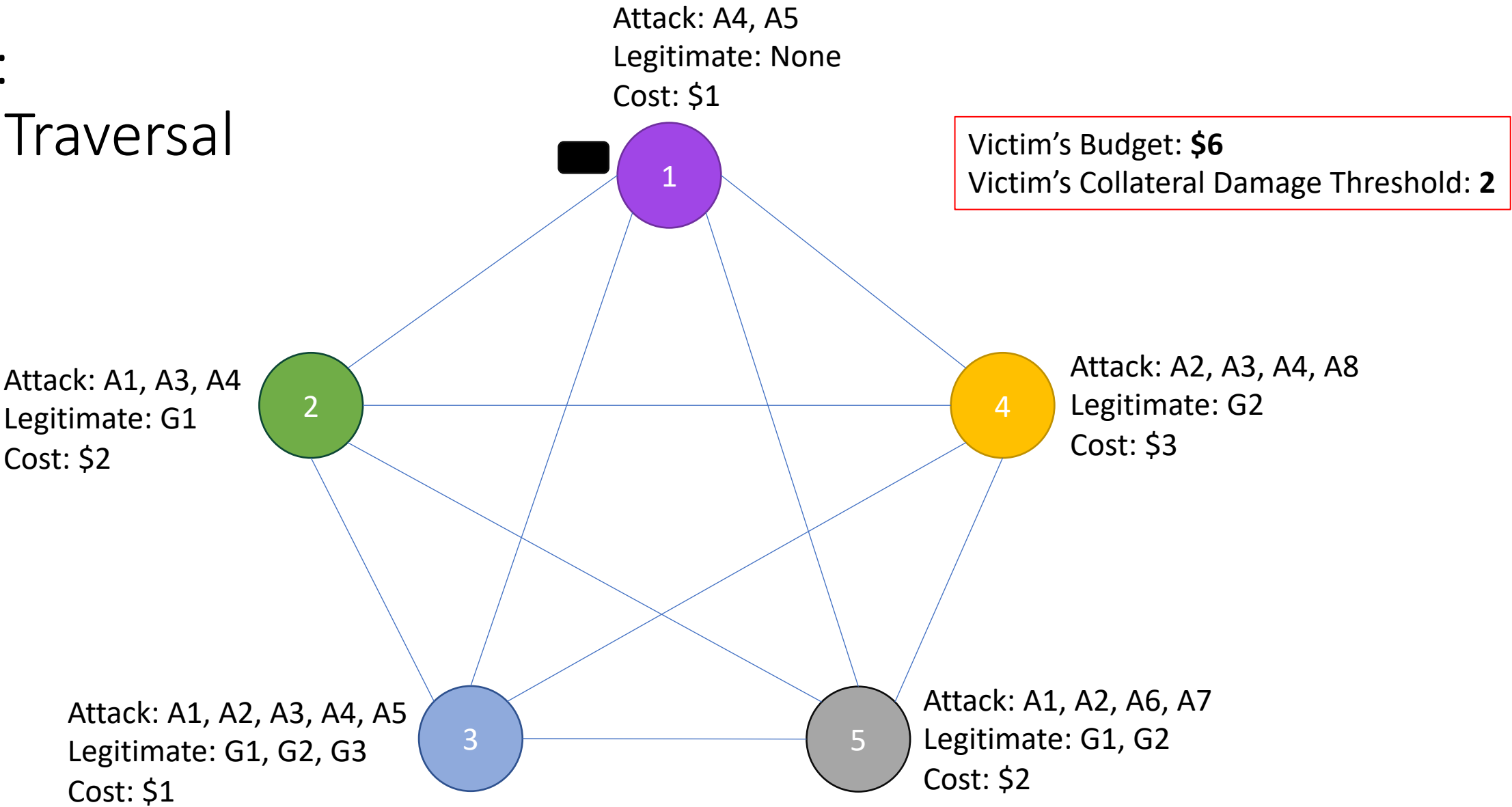
Total of 5 offers, each containing rules that filter certain attack and legitimate flows, and the deployment cost.

Step 1: Graph Construction



Construct a complete graph, where each node represents an offer. Initially all edges have an equal amount of pheromone.

Step 2: Graph Traversal



Ant begins at random offer and chooses subsequent offers based on budget, collateral damage threshold, and attractiveness (amount of pheromone).

Offers selected so far: 1
Cost so far: \$1
Collateral damage so far: None
Efficacy: 2 (A4, A5)

Attack: A4, A5
Legitimate: None
Cost: \$1

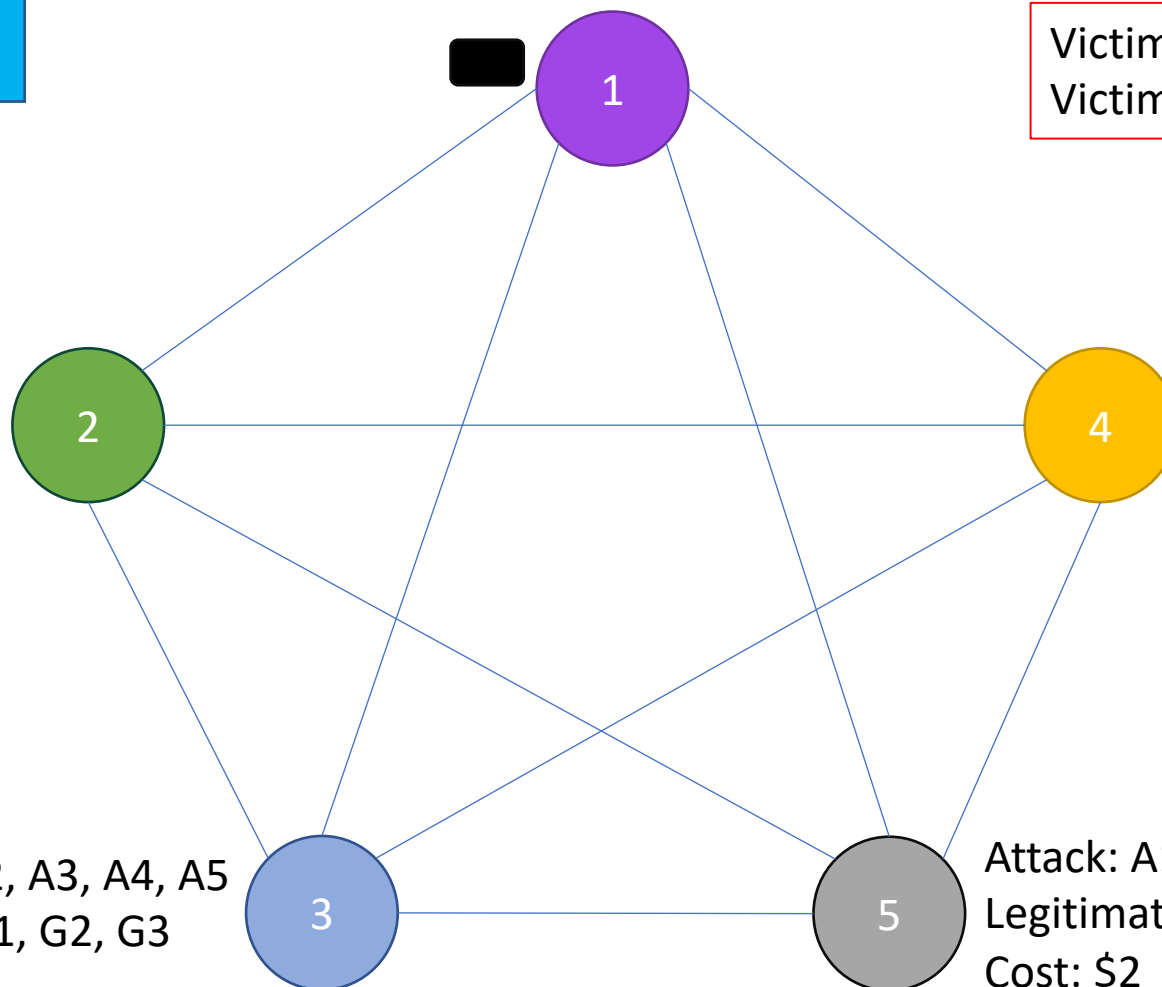
Victim's Budget: \$6
Victim's Collateral Damage Threshold: 2

Attack: A1, A3, A4
Legitimate: G1
Cost: \$2

Attack: A2, A3, A4, A8
Legitimate: G2
Cost: \$3

Attack: A1, A2, A3, A4, A5
Legitimate: G1, G2, G3
Cost: \$1

Attack: A1, A2, A6, A7
Legitimate: G1, G2
Cost: \$2



Ant begins at random offer and chooses subsequent offers based on budget, collateral damage threshold, and attractiveness (amount of pheromone).

Offers selected so far: 1 2
Cost so far: \$3
Collateral damage so far: 1 (G1)
Efficacy: 4 (A1, A3, A4, A5)

Attack: A4, A5
Legitimate: None
Cost: \$1

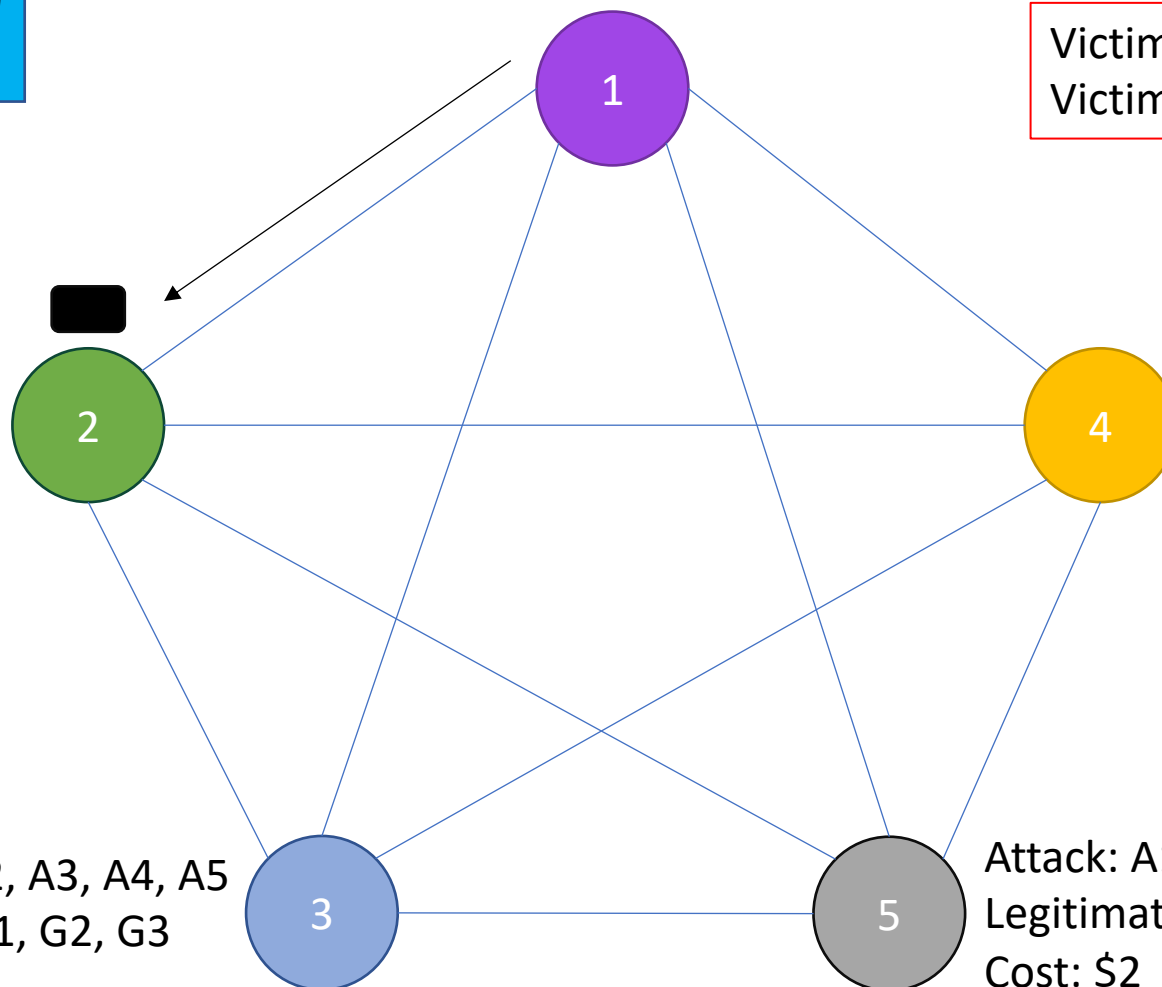
Victim's Budget: \$6
Victim's Collateral Damage Threshold: 2

Attack: A1, A3, A4
Legitimate: G1
Cost: \$2

Attack: A2, A3, A4, A8
Legitimate: G2
Cost: \$3

Attack: A1, A2, A3, A4, A5
Legitimate: G1, G2, G3
Cost: \$1

Attack: A1, A2, A6, A7
Legitimate: G1, G2
Cost: \$2



Ant begins at random offer and chooses subsequent offers based on budget, collateral damage threshold, and attractiveness (amount of pheromone).

Offers selected so far: 1 2 4
Cost so far: \$6
Collateral damage so far: 2 (G1, G2)
Efficacy: 6 (A1, A2, A3, A4, A5, A8)

Attack: A4, A5
Legitimate: None
Cost: \$1

Victim's Budget: \$6
Victim's Collateral Damage Threshold: 2

Attack: A1, A3, A4
Legitimate: G1
Cost: \$2

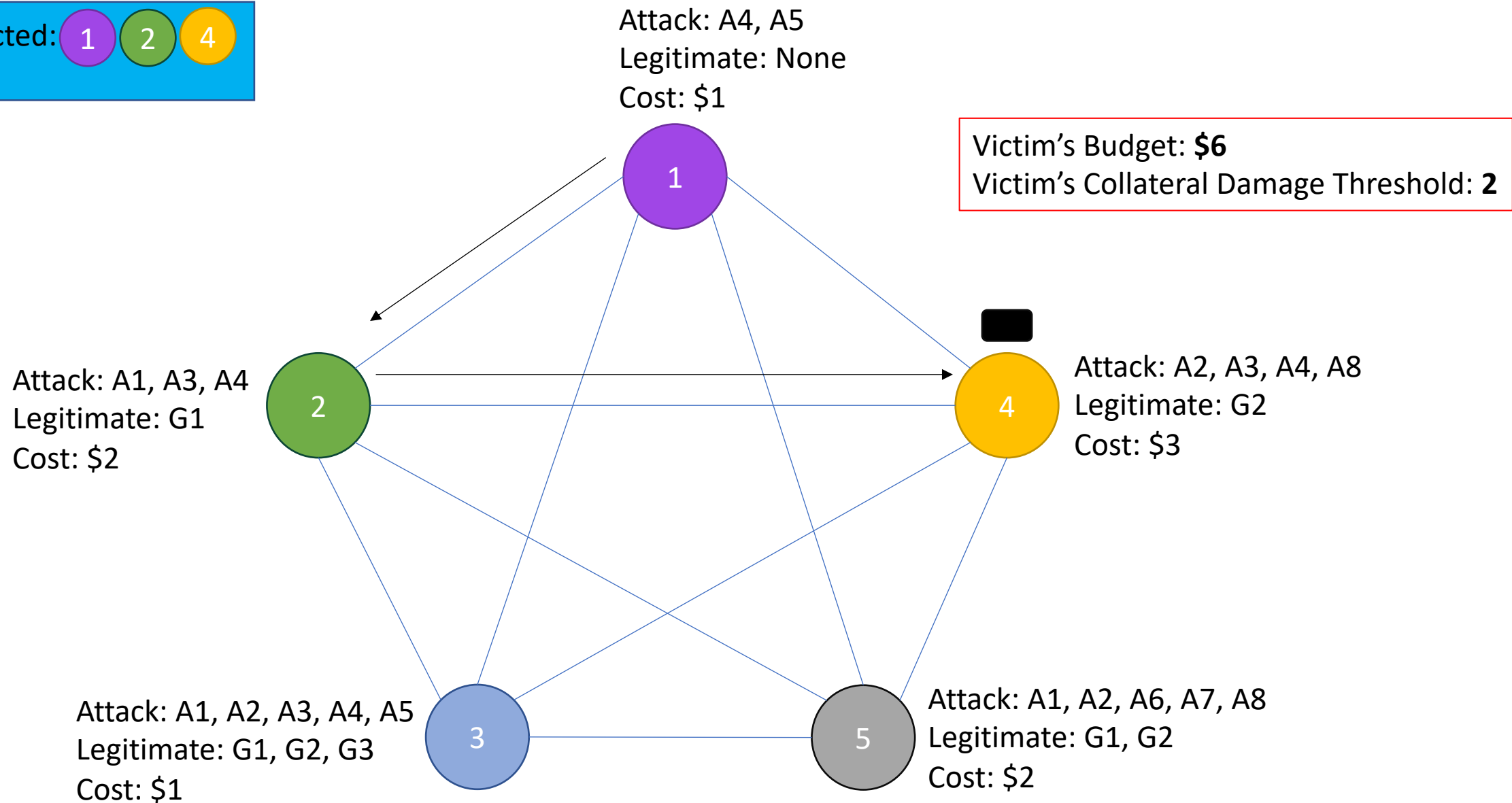
Attack: A2, A3, A4, A8
Legitimate: G2
Cost: \$3

Attack: A1, A2, A3, A4, A5
Legitimate: G1, G2, G3
Cost: \$1

Attack: A1, A2, A6, A7, A8
Legitimate: G1, G2
Cost: \$2

Ant begins at random offer and chooses subsequent offers based on budget, collateral damage threshold, and attractiveness (amount of pheromone).

Ant #1 selected: 1 2 4
Efficacy: 6



Ant begins at random offer and chooses subsequent offers based on budget, collateral damage threshold, and attractiveness (amount of pheromone). Stops once it can no longer choose another offer due to constraints.

Ant #1 selected: 1 2 4
Efficacy: 6

Offers selected so far: 5
Cost so far: \$2
Collateral damage so far: 2 (G1, G2)
Efficacy: 5 (A1, A2, A6, A7, A8)

Attack: A4, A5
Legitimate: None
Cost: \$1

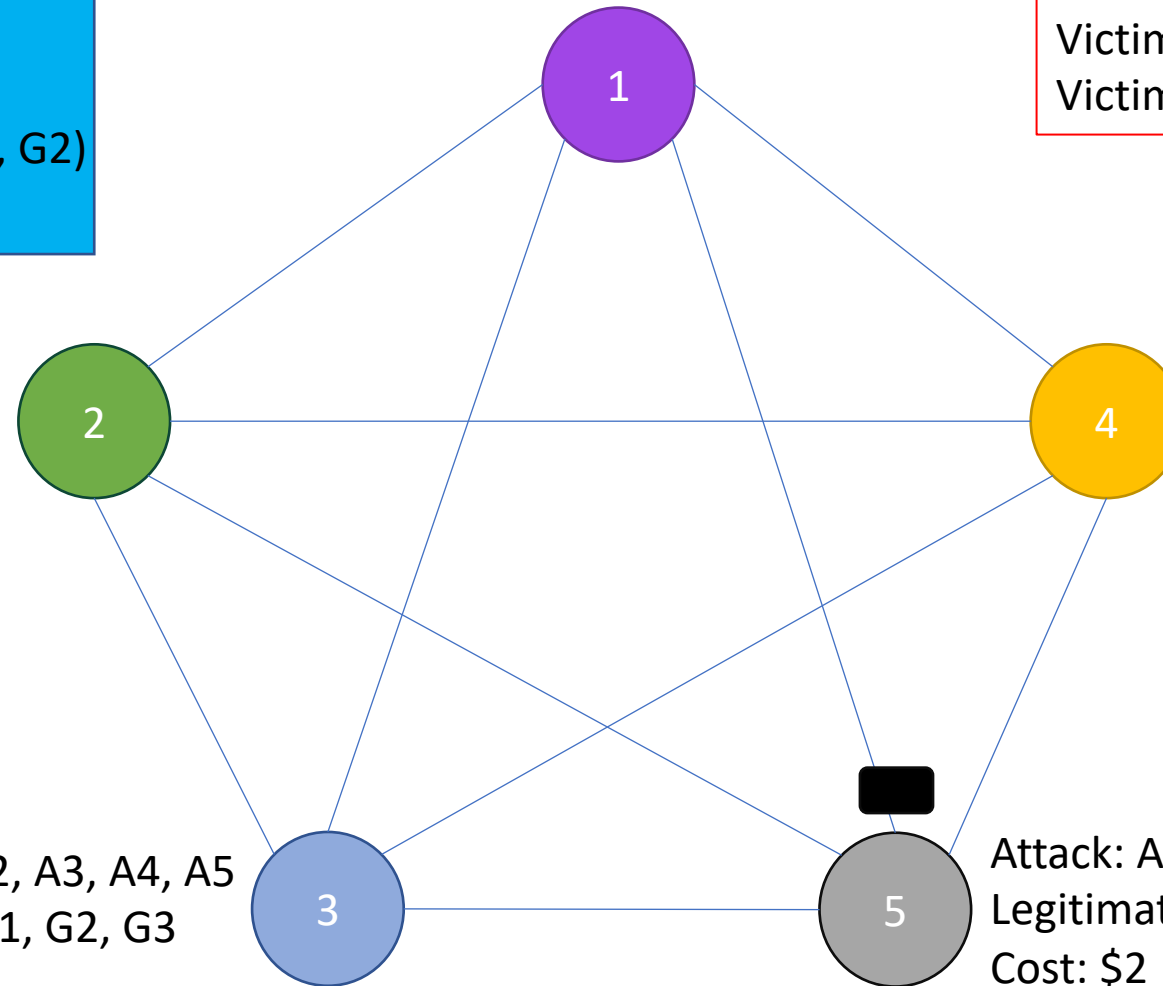
Victim's Budget: \$6
Victim's Collateral Damage Threshold: 2

Attack: A1, A3, A4
Legitimate: G1
Cost: \$2

Attack: A2, A3, A4, A8
Legitimate: G2
Cost: \$3

Attack: A1, A2, A3, A4, A5
Legitimate: G1, G2, G3
Cost: \$1

Attack: A1, A2, A6, A7, A8
Legitimate: G1, G2
Cost: \$2



Next ant in cycle begins its journey at random offer and chooses subsequent offers based on budget, collateral damage threshold, and attractiveness (amount of pheromone).

Ant #1 selected: 1 2 4
Efficacy: 6

Offers selected so far: 5 4
Cost so far: \$5
Collateral damage so far: 2 (G1, G2)
Efficacy: 7 (A1, A2, A3, A4, A6, A7, A8)

Attack: A4, A5
Legitimate: None
Cost: \$1

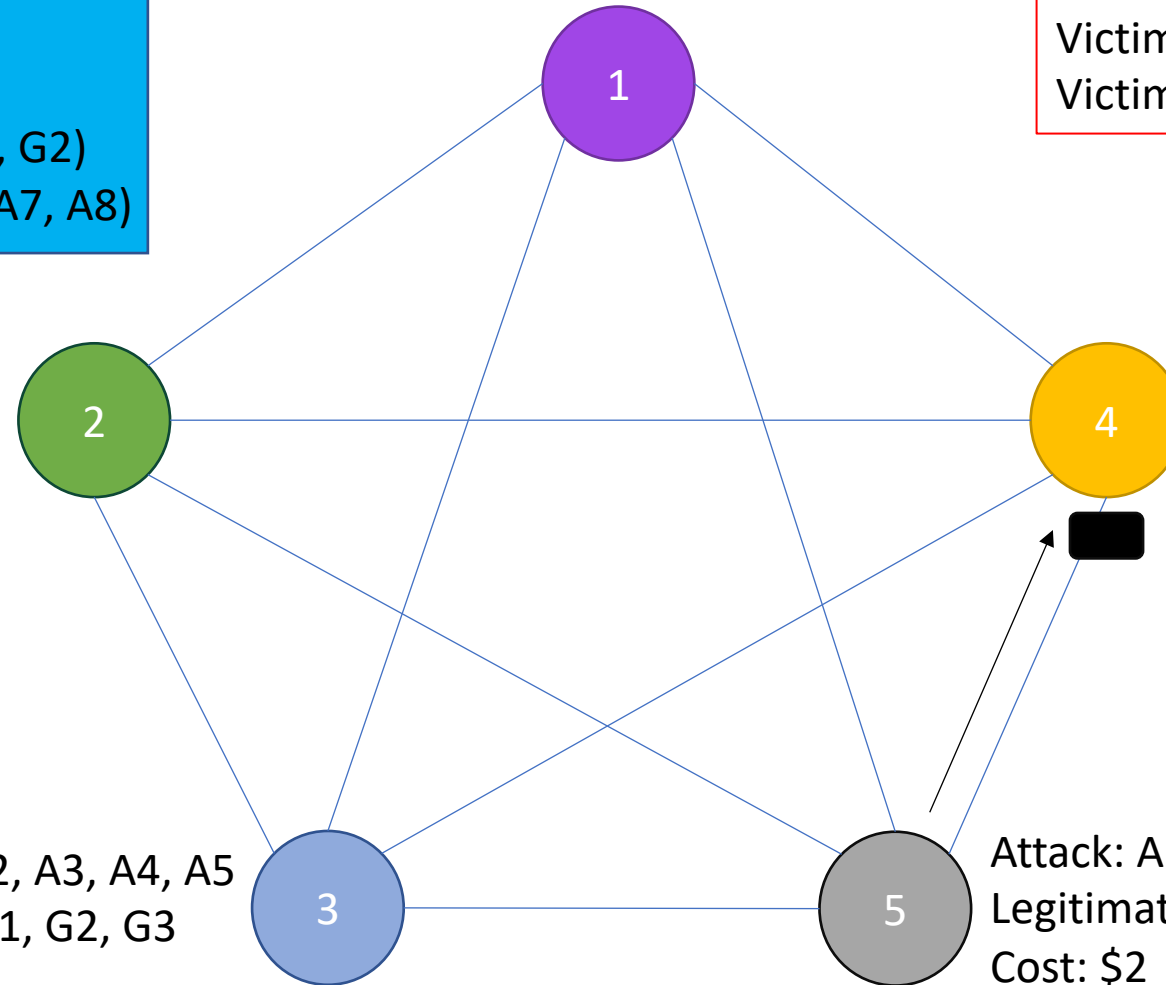
Victim's Budget: \$6
Victim's Collateral Damage Threshold: 2

Attack: A1, A3, A4
Legitimate: G1
Cost: \$2

Attack: A2, A3, A4, A8
Legitimate: G2
Cost: \$3

Attack: A1, A2, A3, A4, A5
Legitimate: G1, G2, G3
Cost: \$1

Attack: A1, A2, A6, A7, A8
Legitimate: G1, G2
Cost: \$2



Ant begins at random offer and chooses subsequent offers based on budget, collateral damage threshold, and attractiveness (amount of pheromone).

Ant #1 selected: 1 2 4
Efficacy: 6

Offers selected so far: 5 4 1
Cost so far: \$6
Collateral damage so far: 2 (G1, G2)
Efficacy: 7 (A1, A2, A3, A4, A5, A6, A7, A8)

Attack: A4, A5
Legitimate: None
Cost: \$1

Victim's Budget: \$6
Victim's Collateral Damage Threshold: 2

Attack: A1, A3, A4
Legitimate: G1
Cost: \$2

Attack: A1, A2, A3, A4, A5
Legitimate: G1, G2, G3
Cost: \$1

Attack: A2, A3, A4, A8
Legitimate: G2
Cost: \$3

Attack: A1, A2, A6, A7, A8
Legitimate: G1, G2
Cost: \$2

Ant begins at random offer and chooses subsequent offers based on budget, collateral damage threshold, and attractiveness (amount of pheromone).

Ant #1 selected: 1 2 4
Efficacy: 6

Ant #2 selected: 5 4 1
Efficacy: 7

Attack: A4, A5
Legitimate: None
Cost: \$1

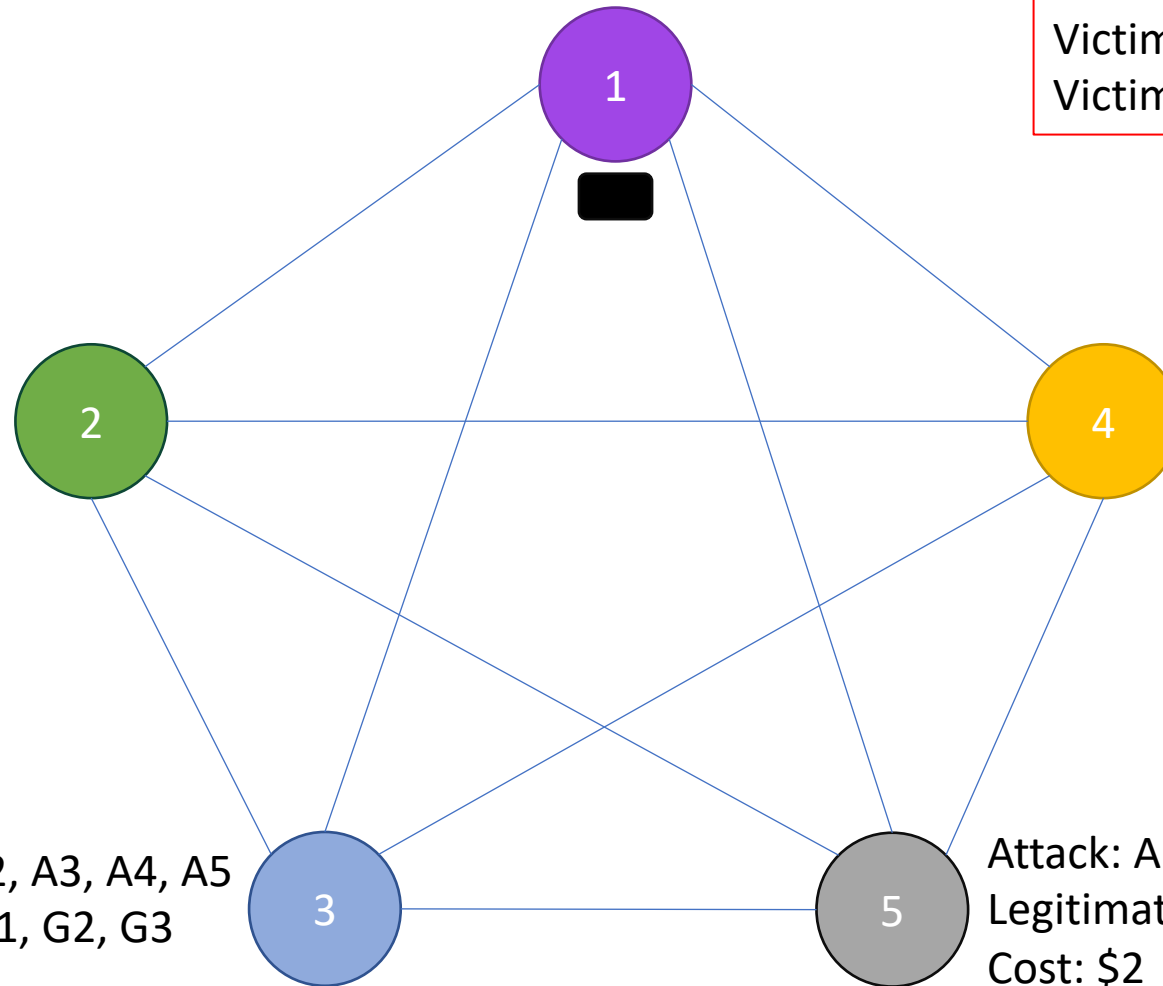
Victim's Budget: \$6
Victim's Collateral Damage Threshold: 2

Attack: A1, A3, A4
Legitimate: G1
Cost: \$2

Attack: A2, A3, A4, A8
Legitimate: G2
Cost: \$3

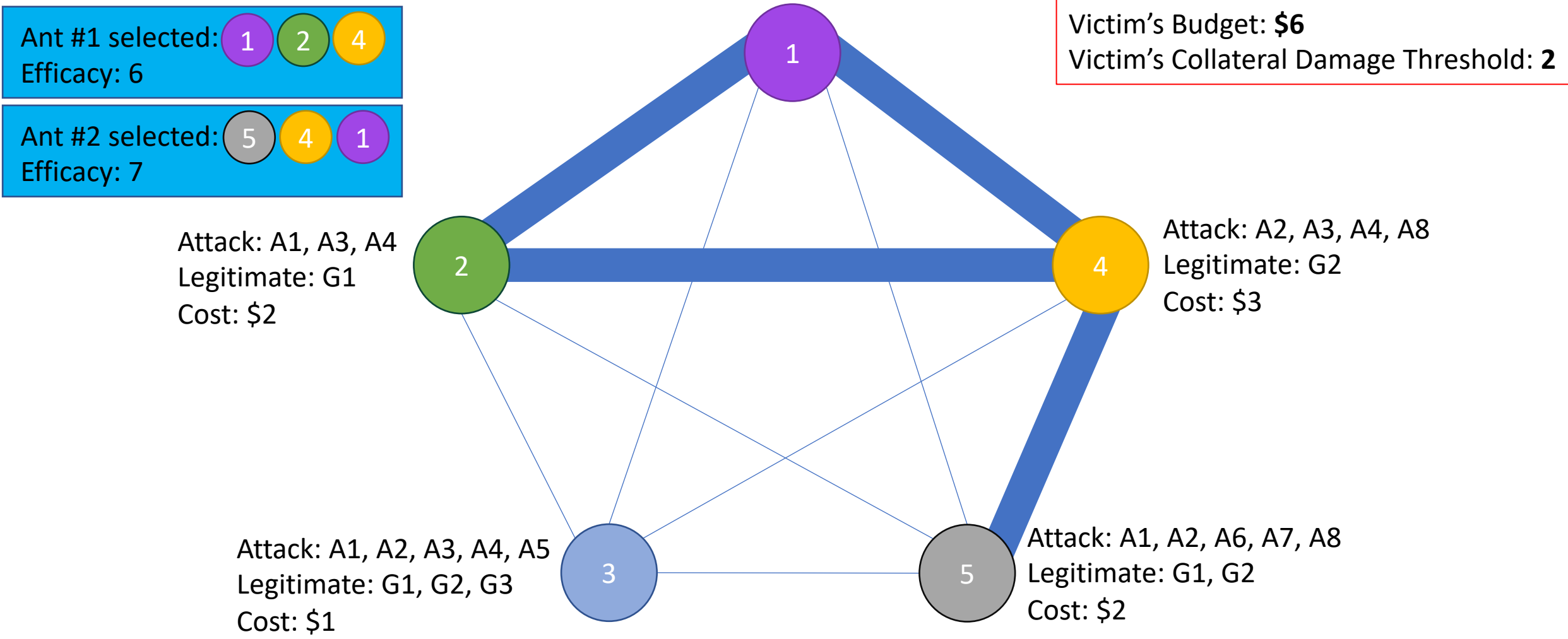
Attack: A1, A2, A3, A4, A5
Legitimate: G1, G2, G3
Cost: \$1

Attack: A1, A2, A6, A7, A8
Legitimate: G1, G2
Cost: \$2



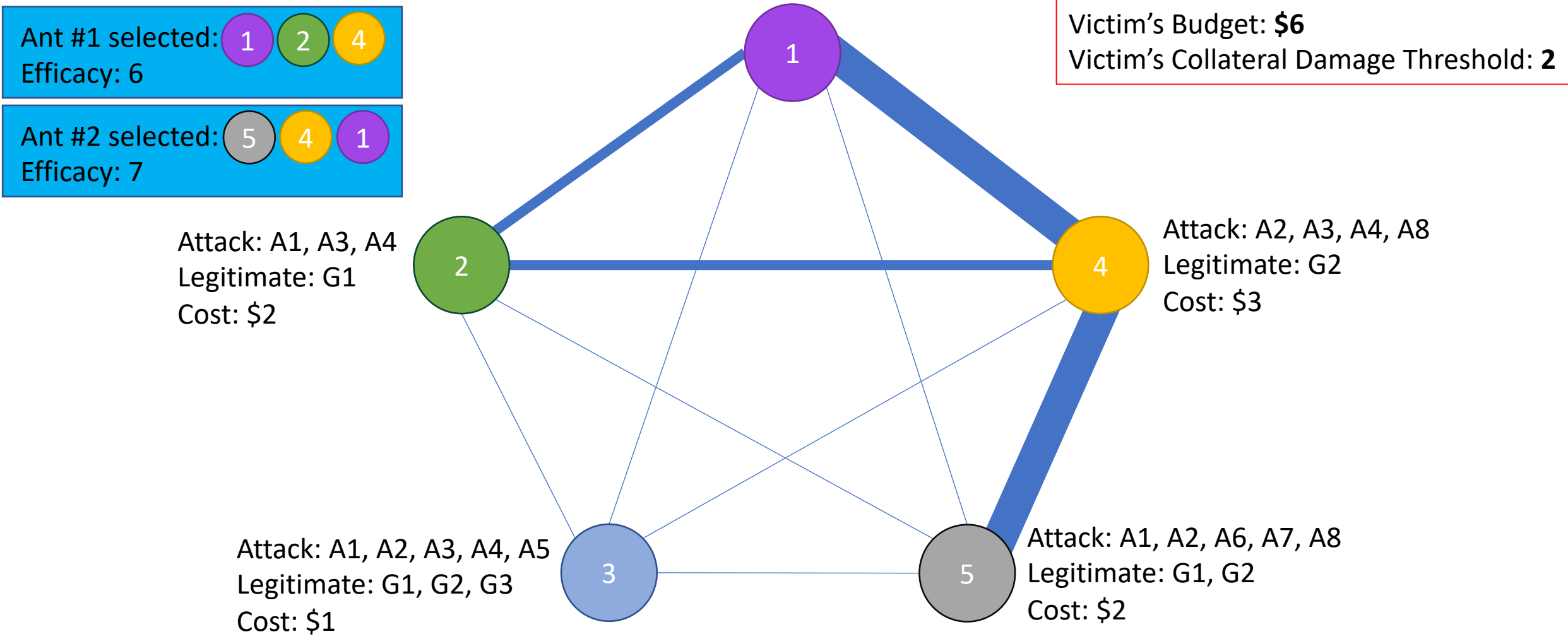
Cycle continues until all ants in the colony have finished traversing the graph.

Step 3: Updating Pheromone



After each cycle, pheromone is dropped along path traversed by ants.

Step 3: Updating Pheromone



Pheromone is evaporated only from paths that do not make up the best solution so far (best solution so far: 5 4 1).

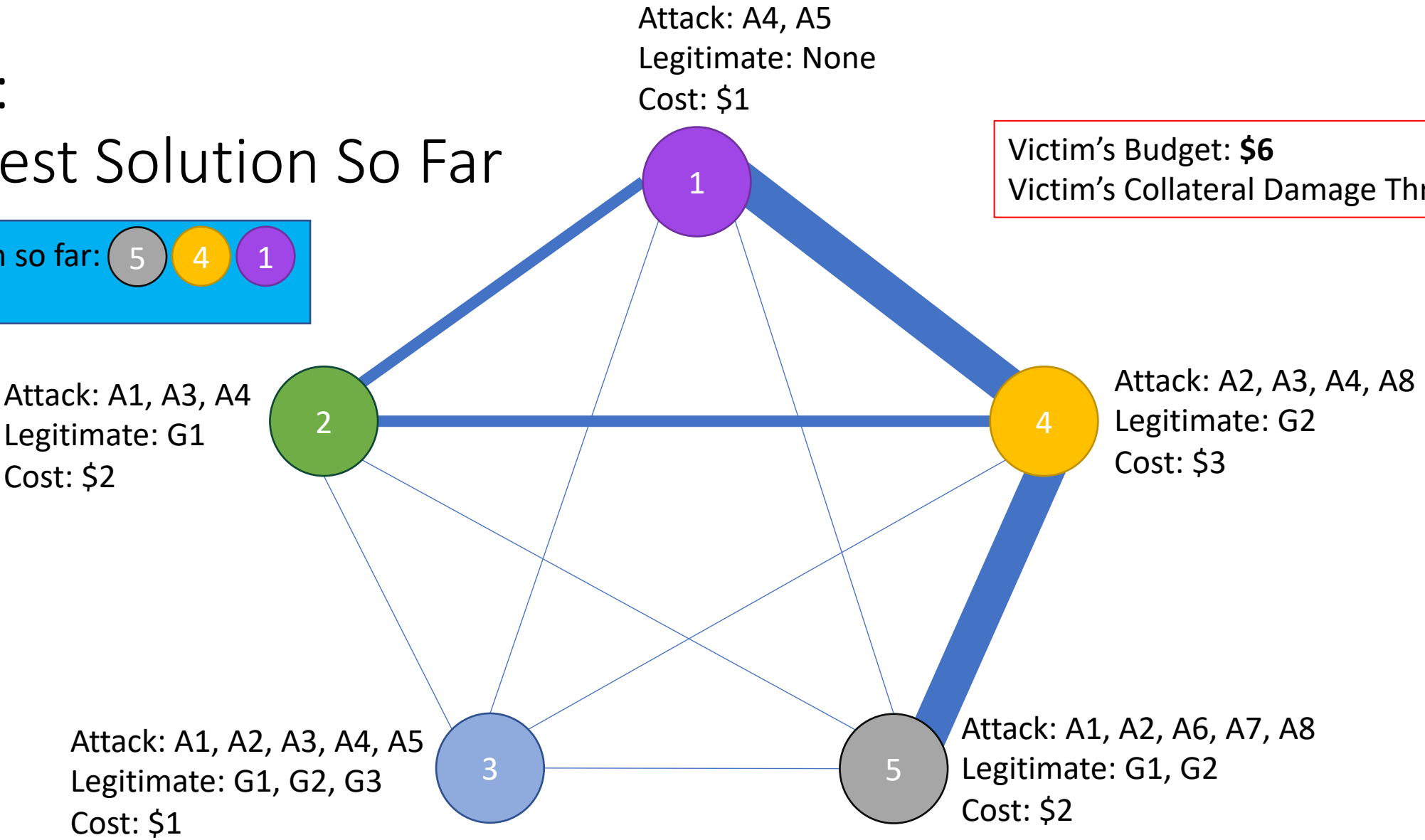
Step 4: Save Best Solution So Far

Best solution so far: 5 4 1

Efficacy: 7

Victim's Budget: \$6

Victim's Collateral Damage Threshold: 2

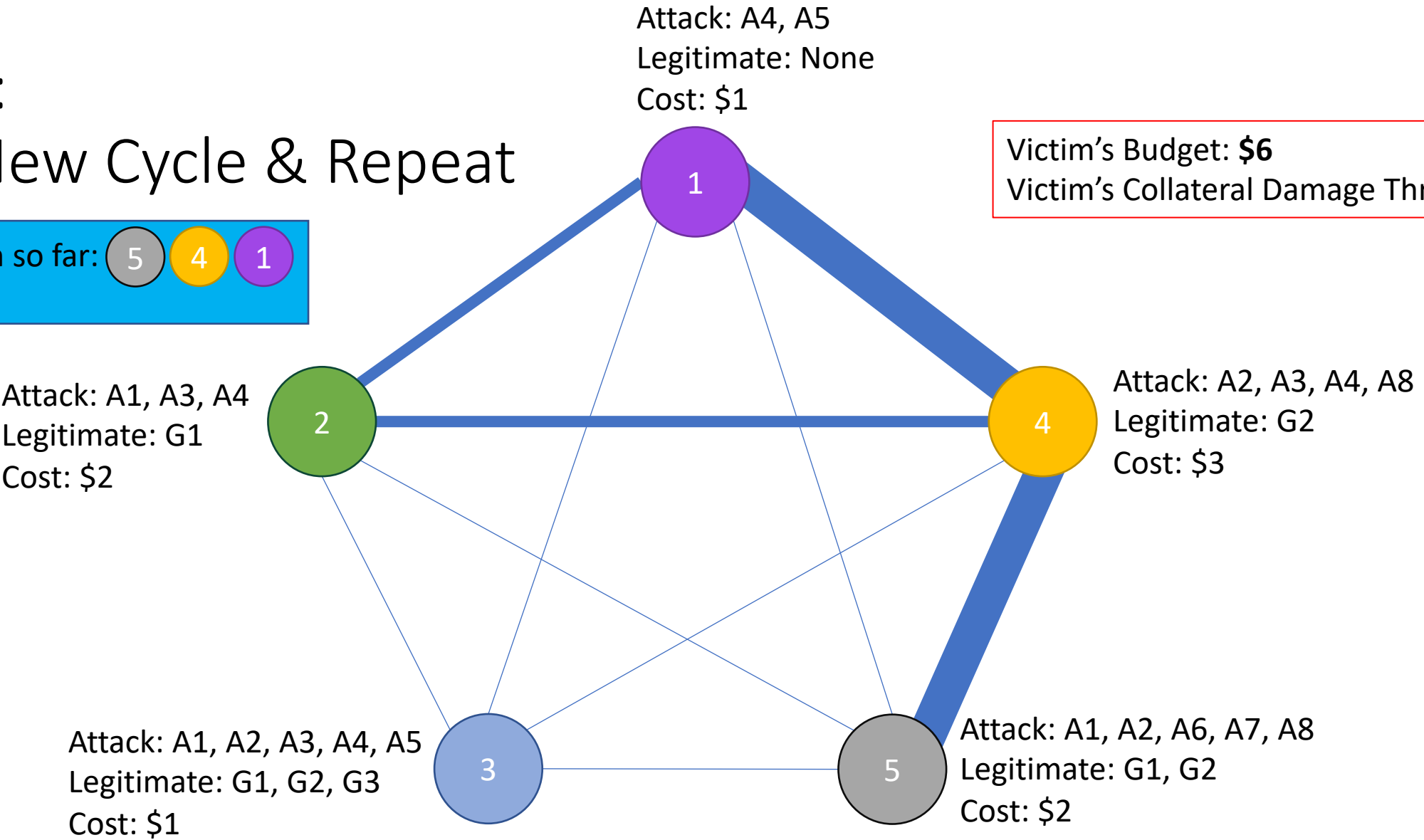


Before the start of a new cycle, save the best solution of the current cycle.

Step 5: Start New Cycle & Repeat

Best solution so far: 5 4 1
Efficacy: 7

Victim's Budget: \$6
Victim's Collateral Damage Threshold: 2



Start a new cycle, repeat the process until all cycles complete. Finally, select the best solution of all the cycles.

Approaching Optimality

- As the number of cycles approaches infinity, the overall best solution approaches the optimal solution
- ACO-based algorithm will eventually find the optimal solution (albeit not in polynomial time)

Evaluation

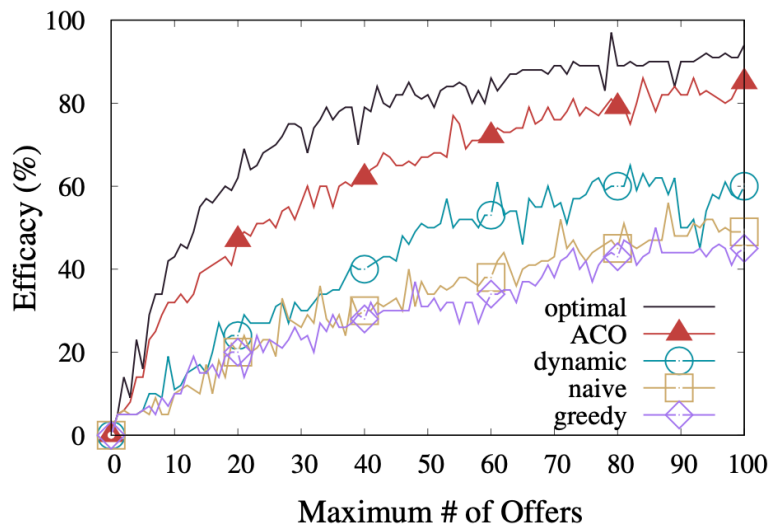
Methodology

- Main two metrics:
 1. Efficacy
 2. Runtime
- Compare ACO-based algorithm with greedy, naive, dynamic programming, and branch-and-bound-based algorithms

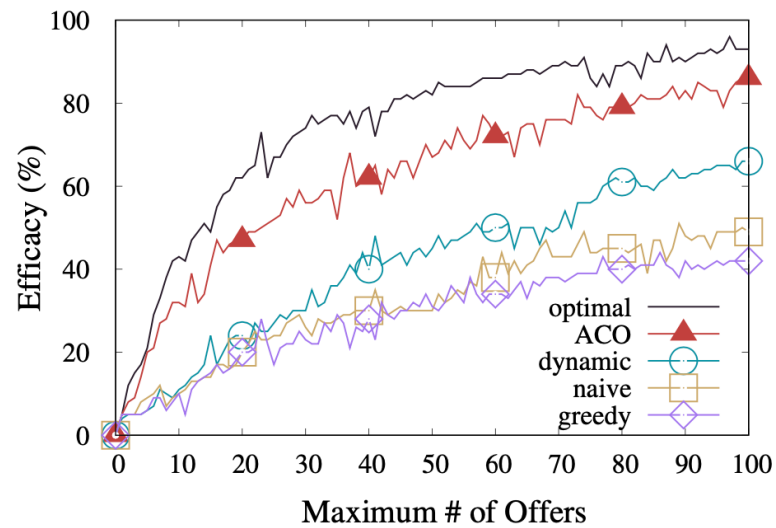
Setup

- Construct an AS-level Internet topology from RouteViews data on July 16, 2019
- We use three different attack traces:
 - CAIDA 2007 DDoS attack trace
 - ~4,700 attack sources
 - ~1,400 source ASes
 - Merit's RADb 2016 DDoS attack trace
 - ~2,300 attack sources
 - ~1,300 source ASes
 - Synthetic trace that follows the attack distribution of the September 2016 DDoS attack launched by the Mirai botnet on Krebs on Security

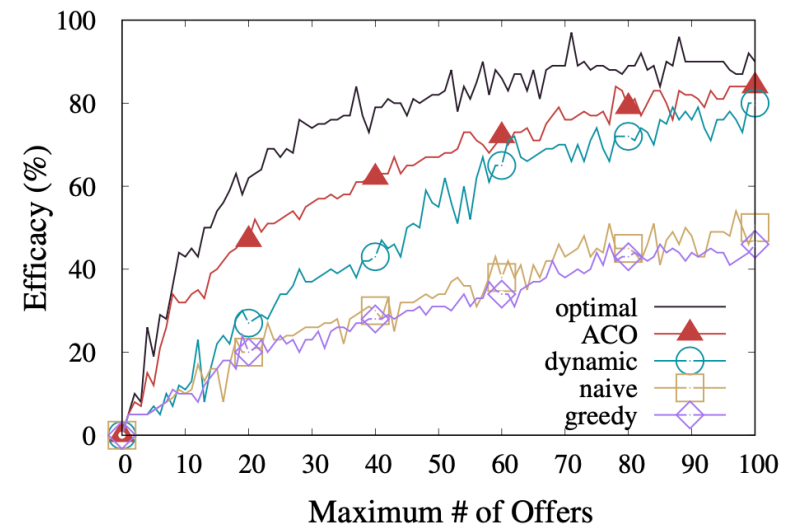
Efficacy



(a) CAIDA 2007

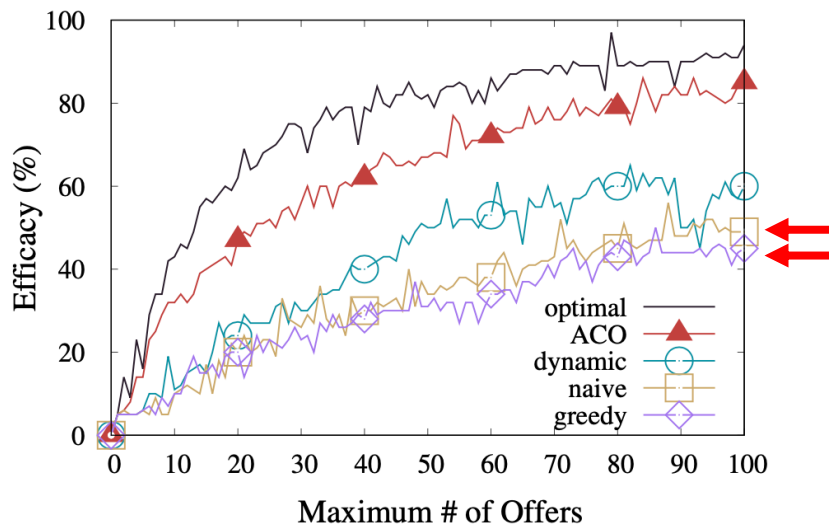


(b) RADb 2016

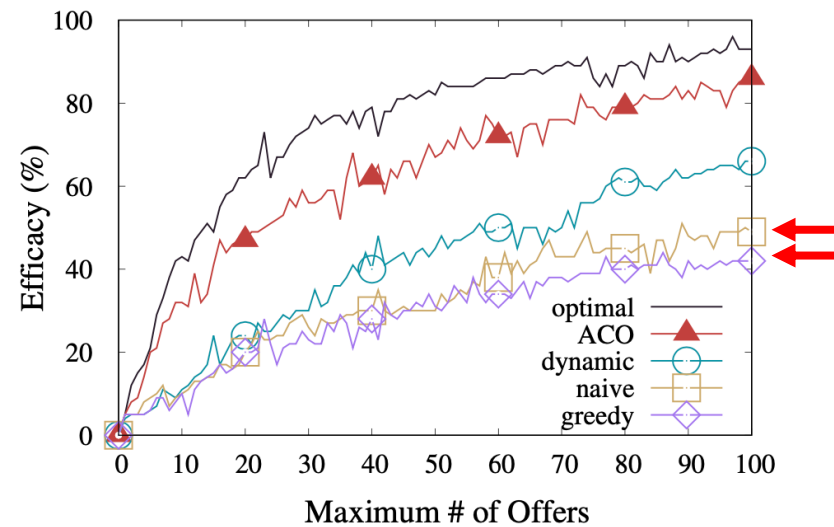


(c) Mirai 2016

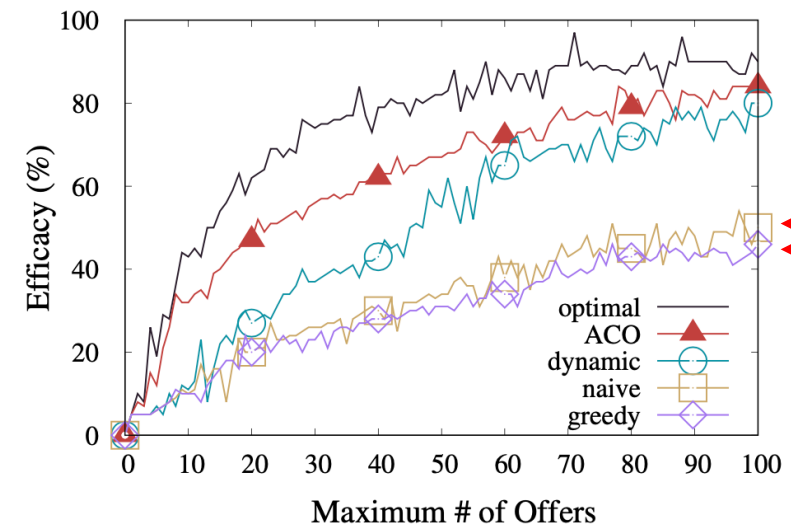
Efficacy



(a) CAIDA 2007



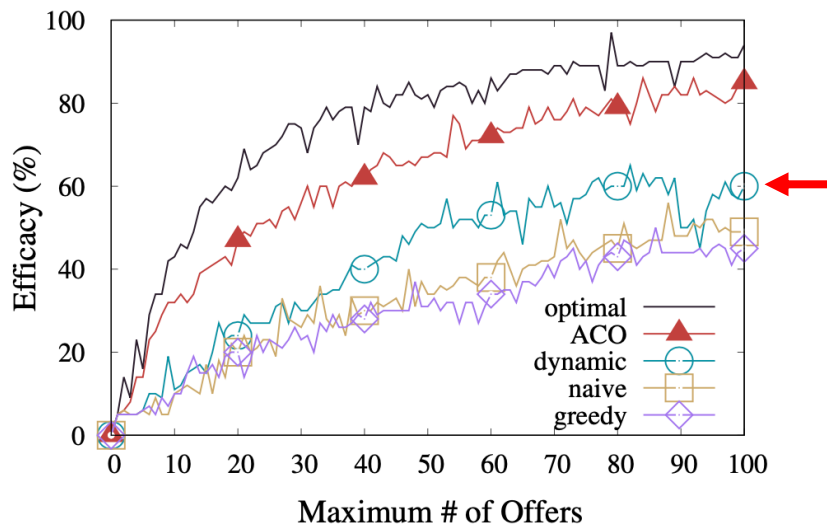
(b) RADb 2016



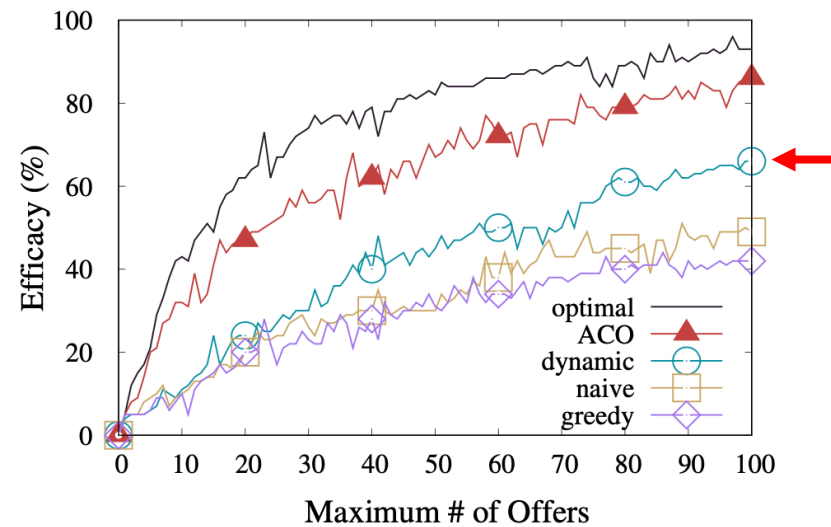
(c) Mirai 2016

Both the greedy and naive algorithms perform underwhelmingly in all three attacks mainly due to uneven distribution of attack sources.

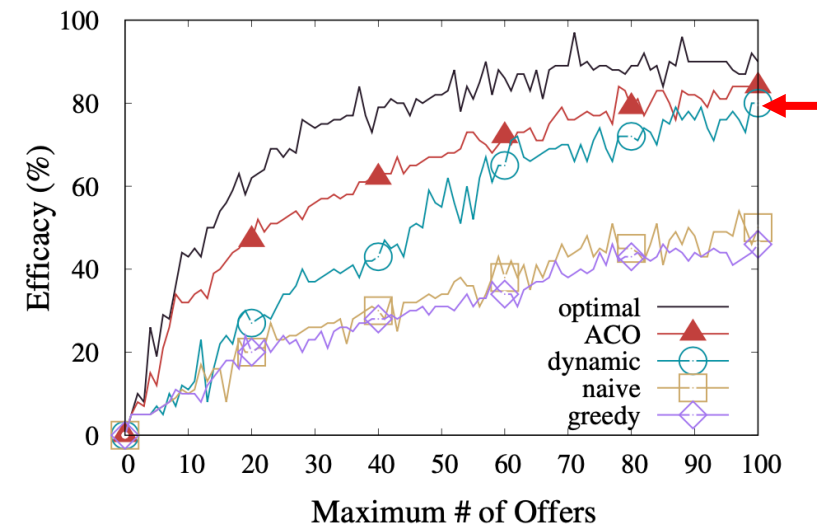
Efficacy



(a) CAIDA 2007



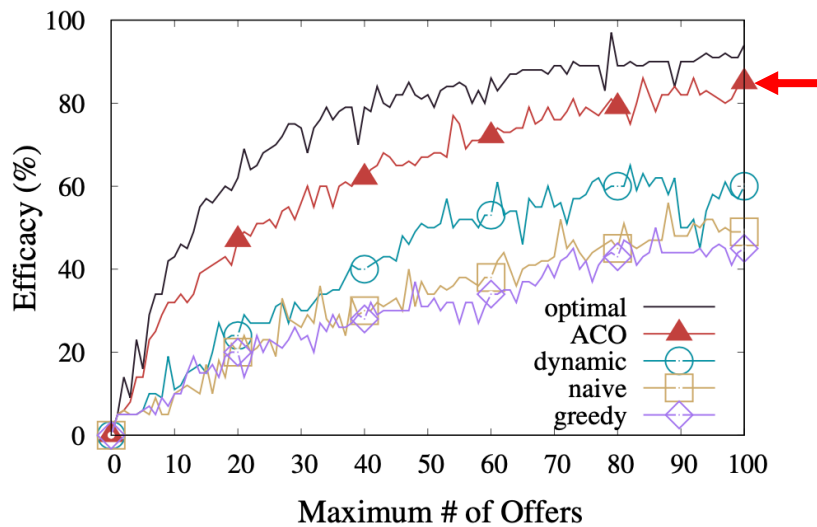
(b) RADb 2016



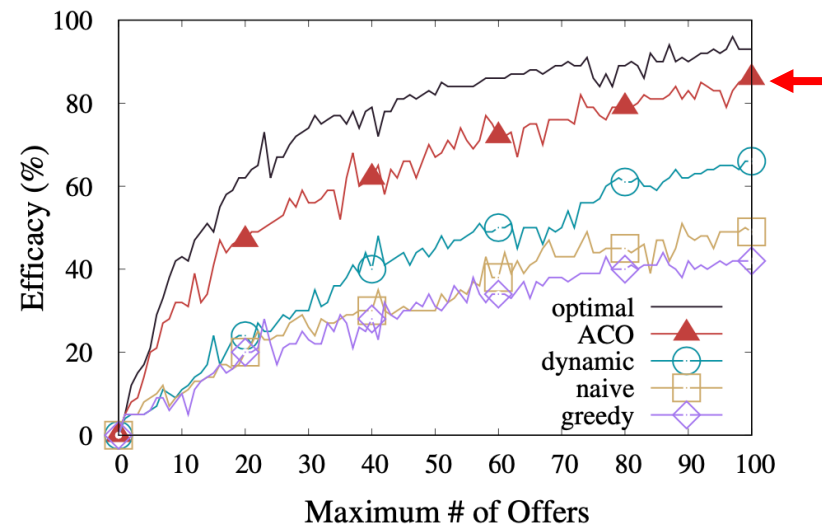
(c) Mirai 2016

While the dynamic programming algorithm achieves significantly better results than the greedy and naive algorithms, it performs worse than the ACO-based algorithm (in most cases).

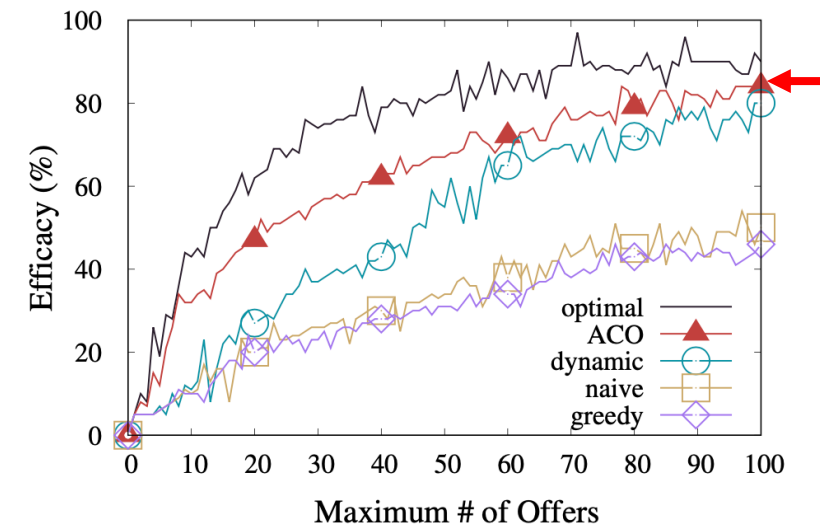
Efficacy



(a) CAIDA 2007



(b) RADb 2016



(c) Mirai 2016

In conclusion, the ACO-based algorithm achieves the best results among the sub-optimal algorithms, and is relatively close to the optimal solution, regardless of the attack.

Runtime

6-Core Intel i7 Processor			
Algorithms	CAIDA 2007	RADb 2016	Mirai 2016
optimal	~9 hrs	~9 hrs	~9 hrs
ACO	11.33 s	10.72 s	12.20 s
dynamic	5.79 s	5.75 s	4.32 s
naive	0.37 s	0.71 s	0.60 s
greedy	0.42 s	0.55 s	0.43 s
24-Core Intel Xeon Processor			
ACO	0.83 s	0.74 s	0.91 s

Conclusion

Conclusion

- Effective in-network DDoS defense is increasingly necessary
- Fundamental dilemma: how to generate, select, and place rules effectively
- This paper tackles the problem of rule selection for in-network DDoS defense
- Contributions:
 - Introduce a new, offer-based operational model for in-network DDoS defense
 - Formulate the NP-hard rule selection problem for this model
 - Design a near-optimal algorithm for the rule selection problem
 - Evaluate our algorithm using a real-world-based Internet routing topology along with real and synthetic DDoS traffic traces
- ACO-based algorithm outperforms the other rule selection algorithms under real-world attacks and performs only slightly worse than the optimal solution even at a large scale

Acknowledgments



This project is in part the result of funding provided by the Science and Technology Directorate of the United States Department of Homeland Security under contract number D15PC00204. The views and conclusions contained herein are those of the authors and should not be interpreted necessarily representing the official policies or endorsements, either expressed or implied, of the Department of Homeland Security or the US Government. We further thank Ann Cox and anonymous reviewers of this paper for their comments.



UNIVERSITY
OF OREGON



CENTER FOR CYBER
SECURITY & PRIVACY

A special thanks to my collaborators Dr. Jun Li and Dr. Lei Jiao, and the rest of the Center for Cyber Security and Privacy at the University of Oregon!

And thank you all for listening! Please feel free to direct any questions to my email address: dsisodia@cs.uoregon.edu