PAS: A Packet Accounting System to Limit the Effects of DoS & DDoS

Debish Fesehaye & Klara Naherstedt
University of Illinois-Urbana Champaign

DoS and DDoS

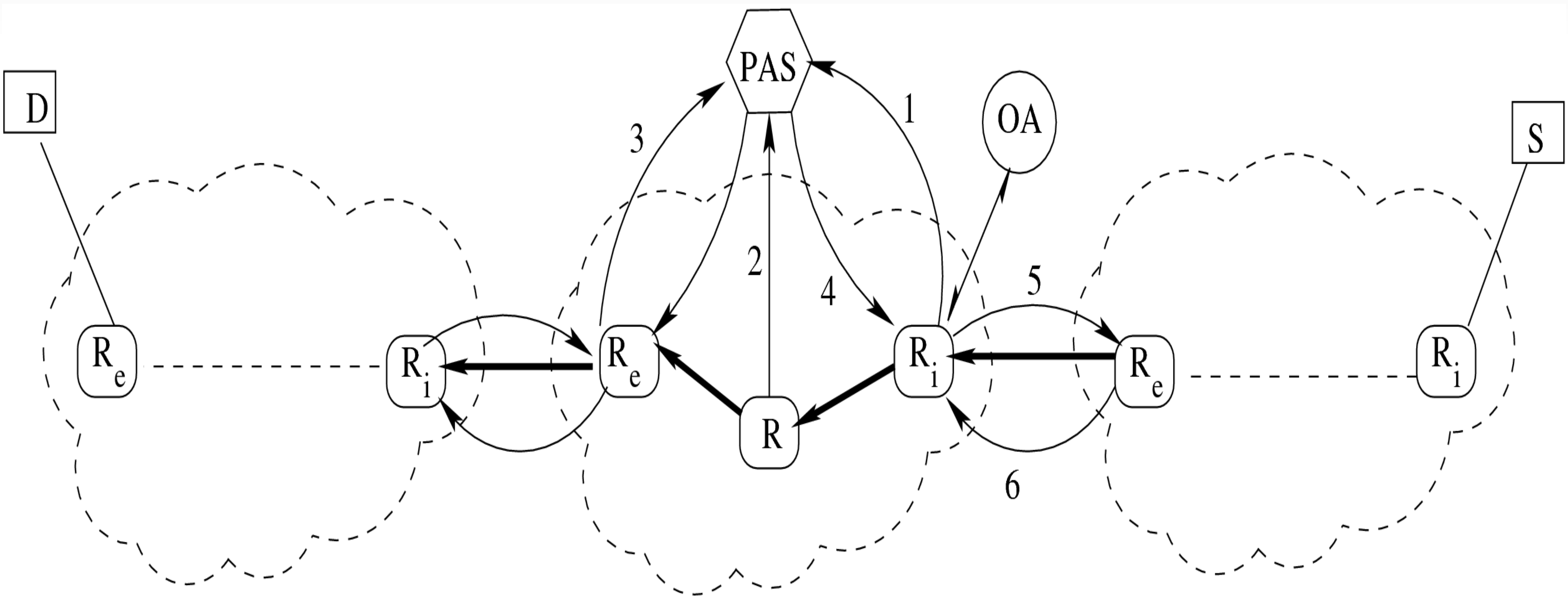
- DDoS attacks are increasing *threats* to our digital world.
- Existing *mitigation* techniques
  - *Capability* approach: to let a receiver explicitly authorize the traffic it wants to receive
    - Example: *TVA* (Traffic Validation Architecture)
    - computational, storage and traffic volume *overheads*
    - many sender and receiver nodes can *collude*
    - routes may *change* forcing routers to drop legit flows they do not know
  - *Filter* approach: to let a receiver install dynamic network filters that block the traffic it does not desire to receive.
    - DDoS may target *core links* where receivers can't see

PAS Design

- Main Idea
  - If every packet is accounted or paid for, then the DoS and DDoS problem reduces into a *congestion control* and *fairness problem*.
  - It can then be dealt with by finding *better routes* or *adjusting* the *sending rates*.
- PAS Design
  - Design 1: PAS uses *per flow* information from *all routers* or router-like boxes (Mostly *clean-slate*).
  - Design 2: Achieves Design 1 by a clever approaches using *only ingress routers* (Can be done in the *current Internet*).

PAS Design1

- Each Autonomous System (AS) maintains a *PAS* as shown in Figure 1 below.
  – The PAS setup can be *hierarchical*, *cloud* or *distributed* for better resource management



PAS Design1

- Each *router* (router-like box) of the AS *reports* a packet count w_j per round for each flow *j* to its *PAS*.
- The PAS *calculates* the rate R^i and per packet price p^i on the behalf of each of its router *i* as shown next.
- The PAS *keeps* the *ID* of each ingress router and a *list* of the flow IDs the ingress router receives.
- The PAS also *receives* the rate R_j and the price P_j^d from each of its egress routers for each of the flows it sends to each *downstream* AS.

PAS Design1 … cont'd

- The PAS calculates the *local path*, path *rate* R and cost of each path P for *each* of its ingress routers as
  - The path with the *maximum* of the *minimum* R^i of each path
  - The cost is the *sum* of the p^i of each router in the selected path
- The PAS can also obtain the best local path, path rate R_j and price P_j for *each flow j* on the behalf of each ingress router accepting flow j and caches these values as
  - The *minimum* R_j of the rates of its *local AS* and that of the *downstream AS* for each flow
  - The *sum* P_j of the prices of the links P_j^L traversed by flow j in its *local AS* and price P_j^d of the *downstream AS* for flow j where $P_j^L = w_j$ x *sum of the* p^i *in the local AS path of flow j*
- These two steps can also be done by the *ingress* routers.

PAS Design1 … cont'd

- The PAS then *forwards* the *path*, the path *rate* and the path cost of each flow j path to the corresponding *ingress* router accepting flow j.
- The ingress router *serves* flow j in the selected path at R_j and *sends* the R_j and P_j to the egress router of the *upstream* AS from which it receives flow j.
- The ingress router then *filters* packets of flow j based on the response it gets from the upstream AS from which it receives flow j as follows:
  - If flow j sends at rate higher than R_j , its packets are sent to a *lower* priority queue at the *ingress* router.
  - If the AS which generates flow j *doesn't pay (account)* for the packets of flow j, flow j is blocked.
  - If it makes *partial payment (accountability)*, packets of flow j are served in the corresponding lower rate.
- An Offline Analyser (*OA*) can also be attached to the *Ingress* router or to the *PAS* to do offline historic analysis of flow traffic and recommend strategies to the ingress routers and the ISP operators.

Computation of fair rate

- *Notations*: C^i, Q^i, N^i and d^i are the capacity, the queue length, number of flows and control interval at router i.
- The *fair rate* at router i is then

$$R^i = \frac{C^i - Q^i/d^i}{N^i}$$

- The fair *packet count* (cwnd) at router i is $w^i = R^i d^i$ and $w_p^{\ i}$ is the w^i of the previous round (control interval d^i).
- But some flows *may not* have enough data to send to utilize their share of the bandwidth.
- This may result in link *under-utilization* while other legitimate flows which have more data to send could use the bandwidth.
- So count some flows as less than one flow as follows:

Computation of fair rate

- The we have

$$n_j^i = \begin{cases} 1 & \text{if } w_j^i > w_p^i \\ \frac{w_j^i}{w_p^i} & \text{otherwise} \end{cases}$$

$$A^i = \sum_{j=1}^{N^i} n_j^i$$

Where n_i^j is a flow indicator and A_i is the actual flow count

- The rate is then $R^i = \frac{(C^i - \frac{Q^i}{d^i})}{A^i}$ and

$$w^i = R^i \cdot d^i$$

Computation of packet price

- The *per packet price* p^i is a funciton of the fair rate R^i.
- If R^i *increases* there is *less demand* and hence cheaper price.
- To capture this we use current rate R^i and price p^i and previous round values $R_p^{\,i}$ and $p_p^{\,i}$ and calculate the current per packet price as

$$p^i = p_p^i \frac{R_p^i}{R^i}$$

- Other more sophisticated pricing funcitons can be used
- The total flow price of flow j at link i in a given round is $w_j p^i$.
- Different R^i values can also be obtained for each flow based on some *priorities* (weights).

PAS Design2

- Each *ingress* router sends the *current packet count* w_j of flow j in the current control interval to its PAS.
- The PAS *aggregates* these values from each ingress router and finds the rate R^i and p^i on the behalf of router i of its AS as follows:
  - For each *ingress* router
    - For each *flow* of the ingress router
      - For each *link* crossed by the flow
        - ➔ Obtain A^i for the R^i calculation as the sum of all actual flow counts A_k^i of each ingress router k
  - For each ingress router in the network
    - Obtain R^i
    - Find the *best path* for each flow of of each ingress
- The *rest* of the design is the same as *Design1*.

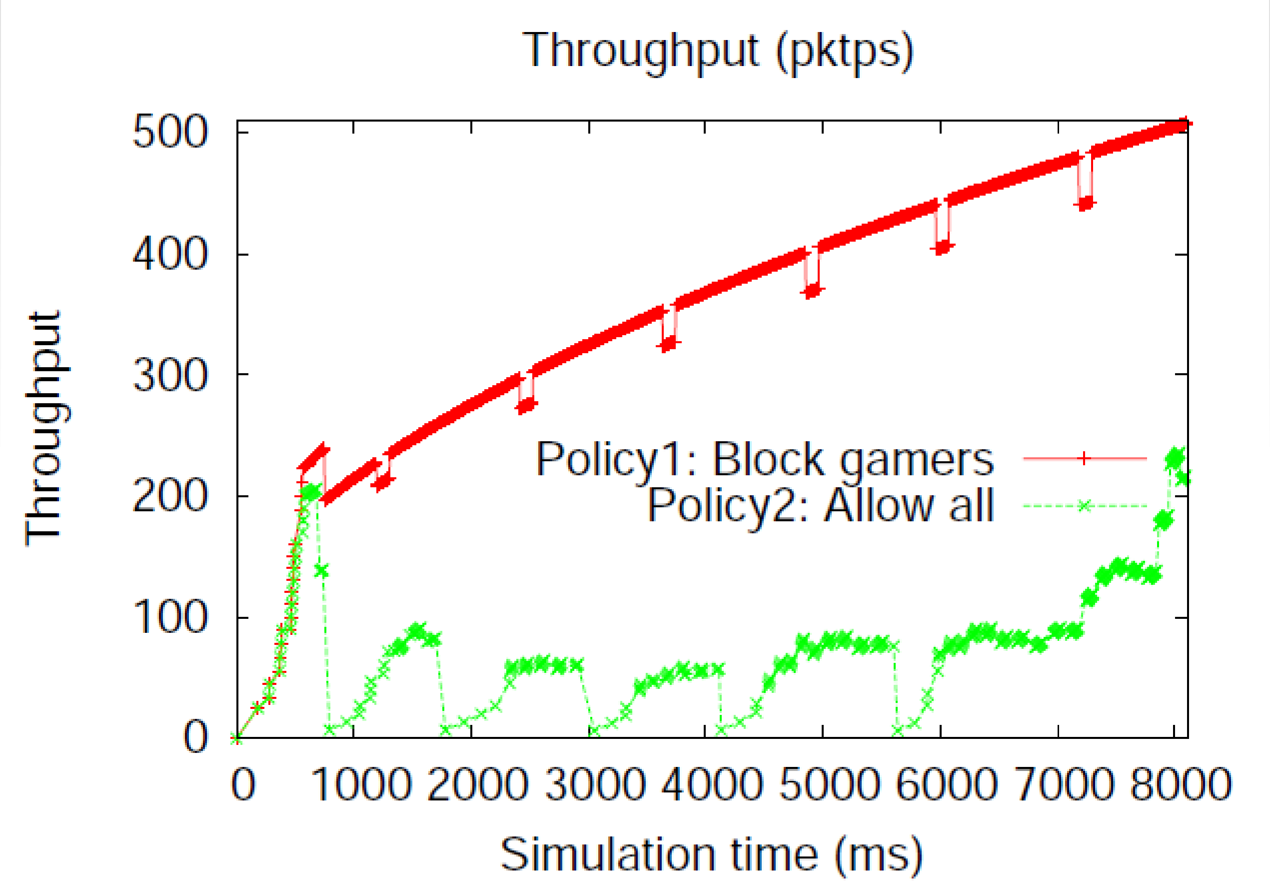
PAS Cloud Computing

- To gain on *processing* speed the PAS Design2 rate computation for each link can be done using *map-reduce* cloud computing framework.
- A *maper* can be used for each *ingress* router and
- A *reducer* aggregates the A_k^i of ingress router k to link i
- After all A_k^i are added, R^i and p^i of link i are computed and the *max-min* algorithm is run to find the *best path* and *rate* for each flow j along with the total price P_j to be sent to the egress router at the upstream AS.
- A *map-reduce* framework can also be used for the *max-min* path and rate computation as shown in the next slide.

Cloud computing to find the best path

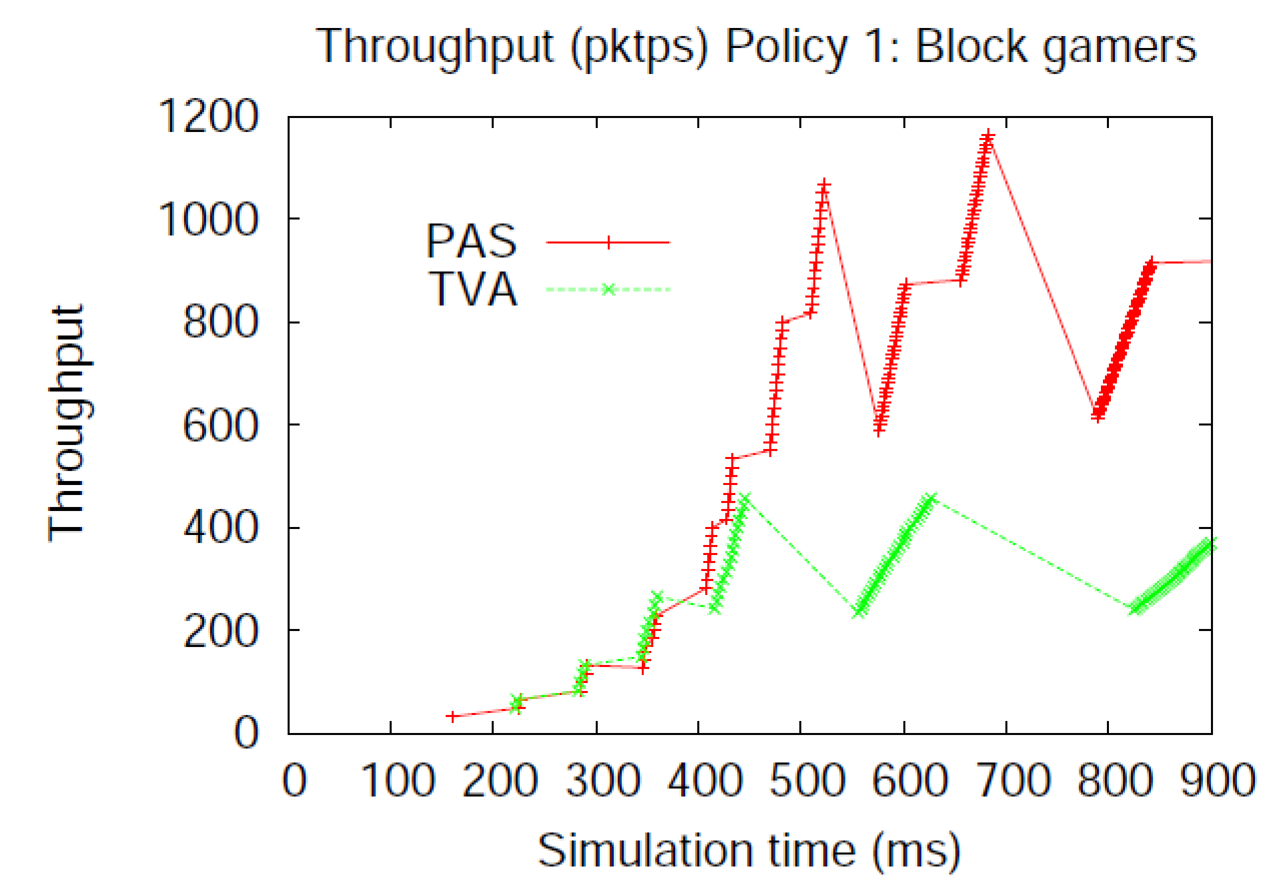
- Cloud computing may be used to find the *max-min* or shortest path in a graph and its weight (rate)
- Given a *source* *s* with degree (number of outgoing links) *n* and *destination* *d* in a graph G,
- A *mapper* can be associated with each *neighbor* of *s* to find the max-min or shortest path to *d*
- A *reducer* can then choose the path with the *max-min* or shortest of each of its *outgoing* links
- A *heirarchical* map-reduce framework may be associated with the neighbors of the neighbors of a node.
- The best path computation for each ingress router can also be computed in *parallel* using a cloud compute node.

Numerical Results:
Budget Exceeded Block Gamers



- Policy1: As the price reported by downstream PAS exceeds the budget of the AS, the AS sends *block gamers* signal and hence *increases* the throughput of its other *legit* flow.
- Policy2: All flows are allowed and hence the gamers reduce the throughput of the other legit flow

Numerical Results: With Colluding Attackers



Simulation Attributes	TVA	PAS
Num. of downloads	27	37
Avg. Download size (bytes)	81115.38	91861.11
Avg. Download time (sec)	5.11	4.42

- Top figure: 50% colluding
- Bottom Figure: 1/3 colluding
- Both figures show that PAS can outperform TVA, a well known DoS mitigation approach.

Numerical Results:
With Route Changes

Percentage of route change	unfinished downloads	download times
50	56	34.06
25	35	55.83
15	21	102.87
5	0	153.05

TABLE II
FILE DOWNLOAD TIME (SEC) OF TVA WITH DIFFERENT PERCENTAGE OF ROUTE CHANGES

Percentage of route change	unfinished downloads	download times
50	0	154.90
25	0	153.92
15	0	155.46
5	0	153.84

h

TABLE III
FILE DOWNLOAD TIME (SEC) OF PAS WITH DIFFERENT PERCENTAGE OF ROUTE CHANGES

- As shown in both tables PAS can outperform TVA interms of download times of legitimate traffic when route of packets changes

Summary

- We have presented a noble packet accounting system (*PAS*) to deal with DoS and DDoS
- PAS can also serve as a *congestion* control and *routing* scheme with packet *pricing*.
- Our scheme (Design2) can be implemented in the *current Internet* with few additional features to the current network infrastructure.
- Preliminary numerical NS2 simulation results show that our scheme can *outperform TVA*.
- We are working on real *implementation* of PAS in linux.