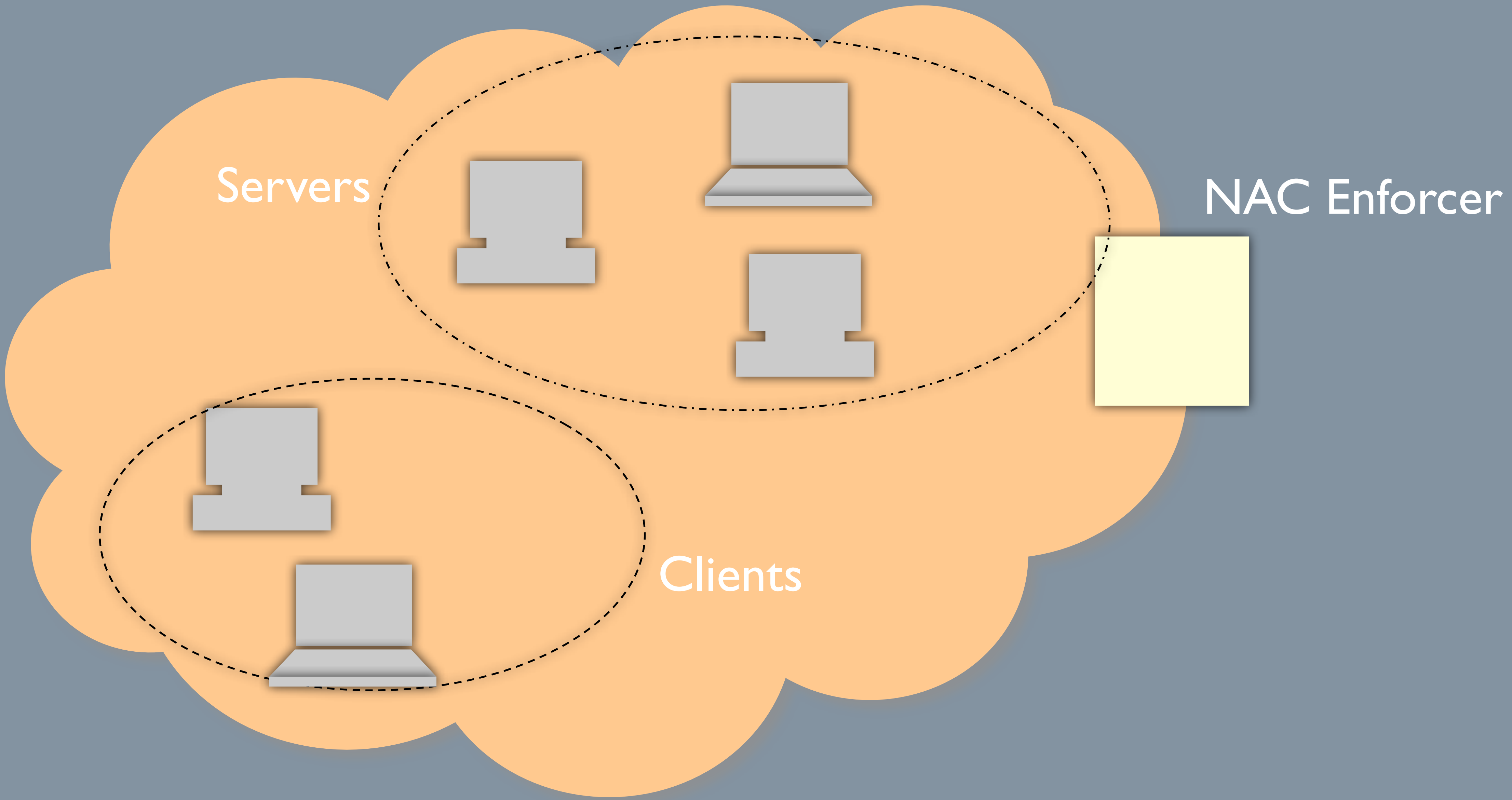
Behavior-based Access Control in Wired and Wireless Networks

Angelos D. Keromytis
Network Security Lab
Department of Computer Science

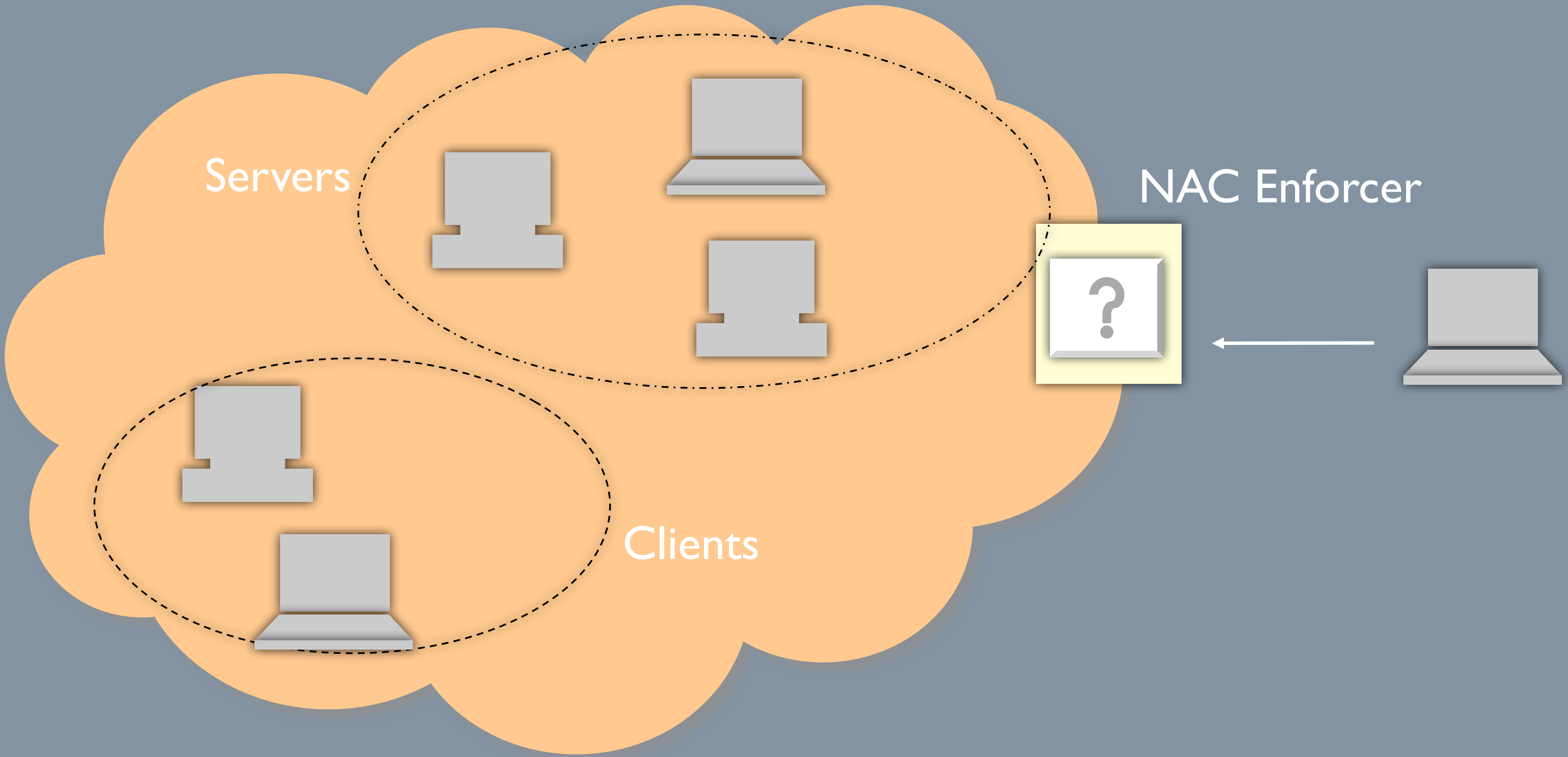
Network Access Control (NAC)

- Mechanism used to control access to a network
- Complements a firewall
  - firewalls keep untrusted nodes out
  - NAC allows trusted nodes to connect

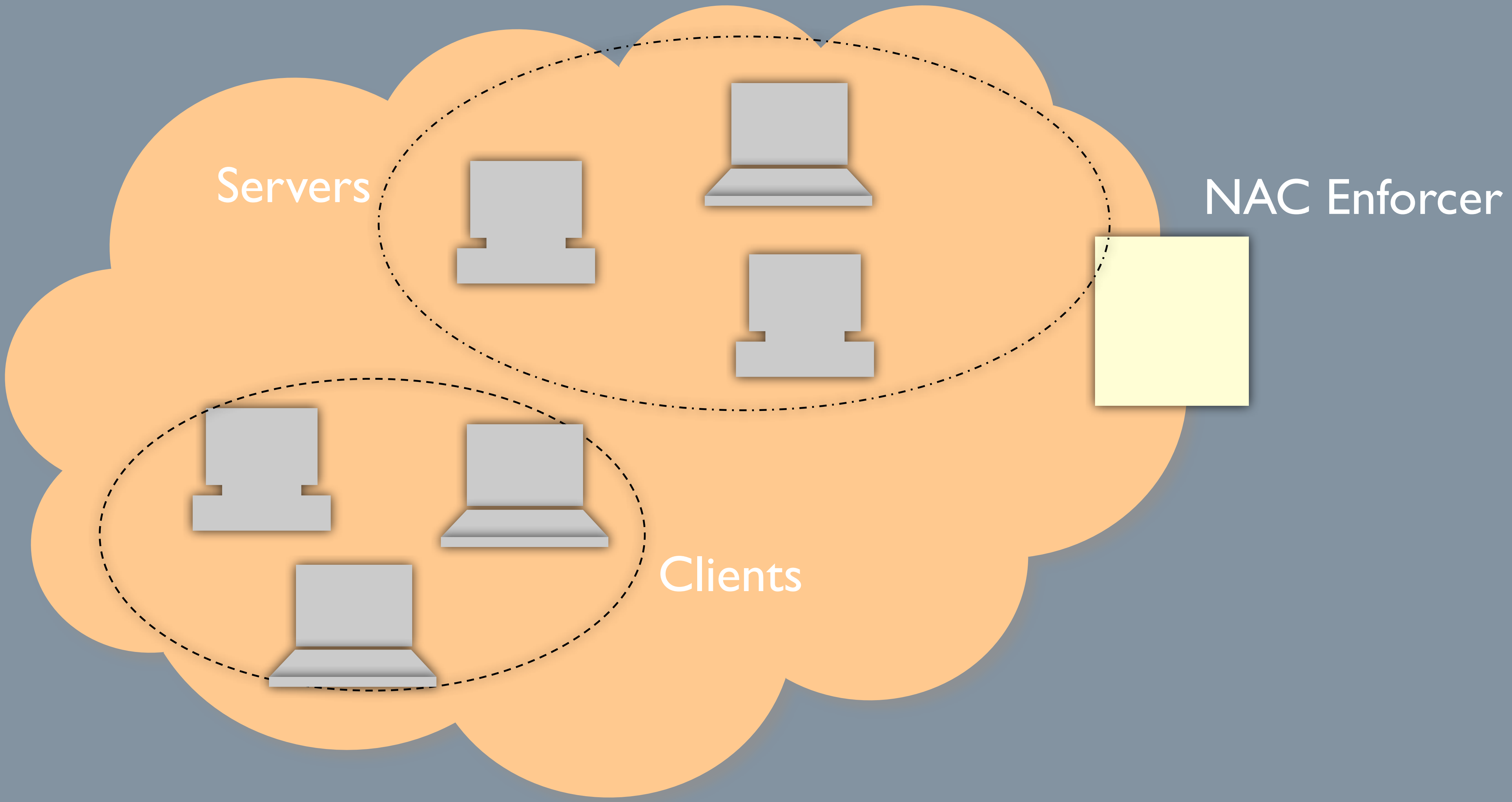
Motivation: Pre-connect Phase



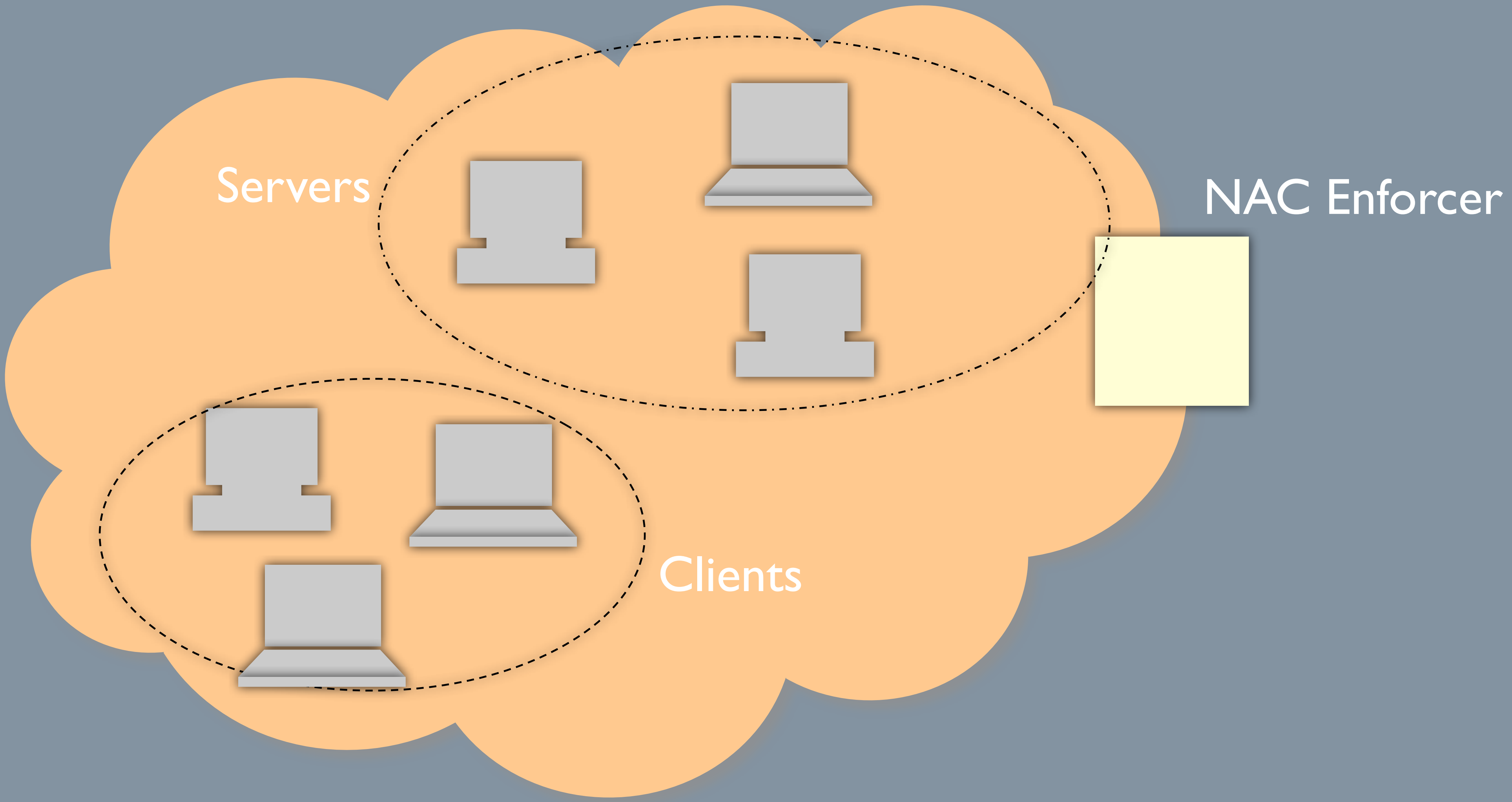
Motivation: Pre-connect Phase



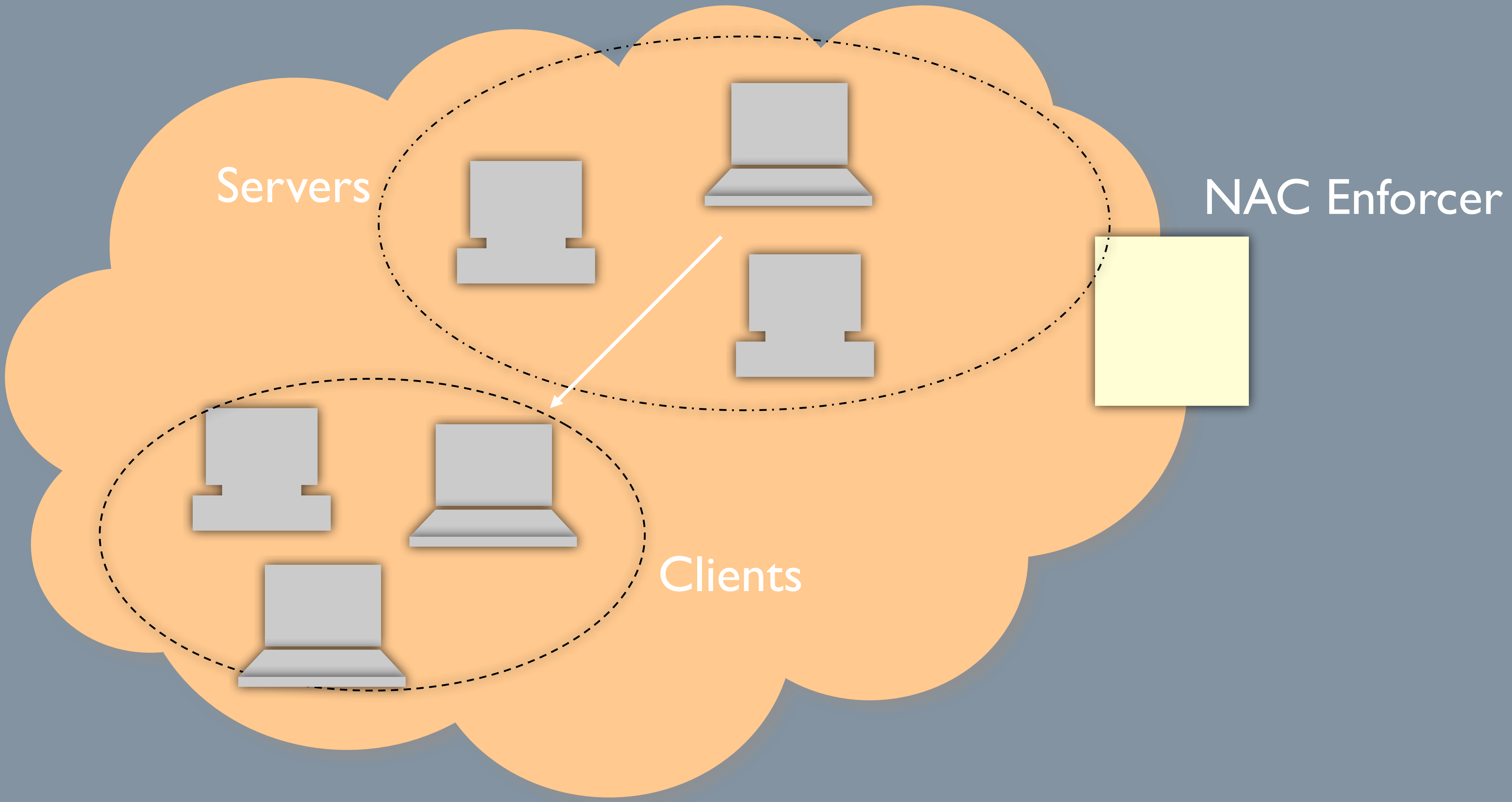
Motivation: Pre-connect Phase



Motivation: Post-connect Phase

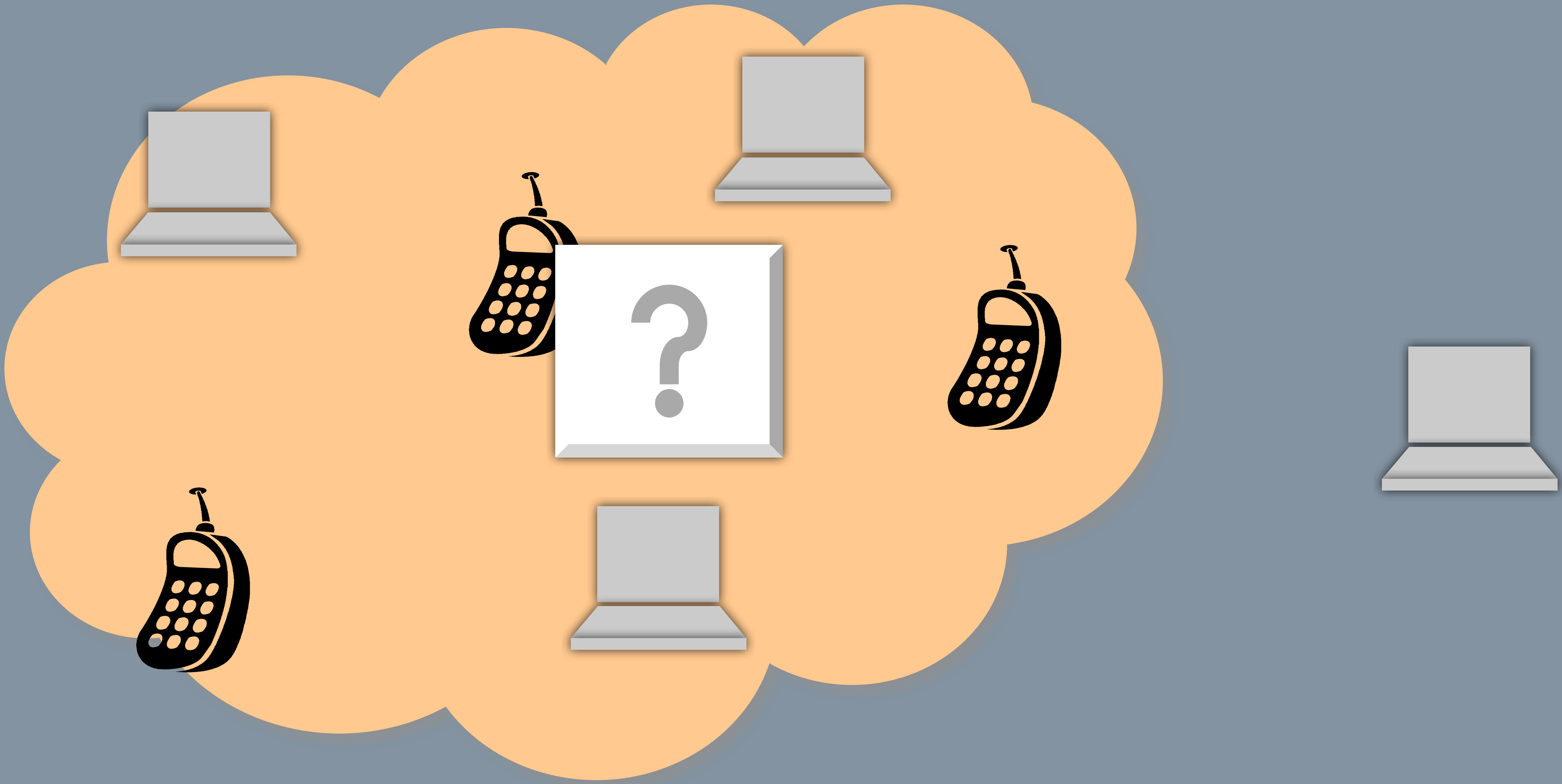


Motivation:
Post-connect Phase

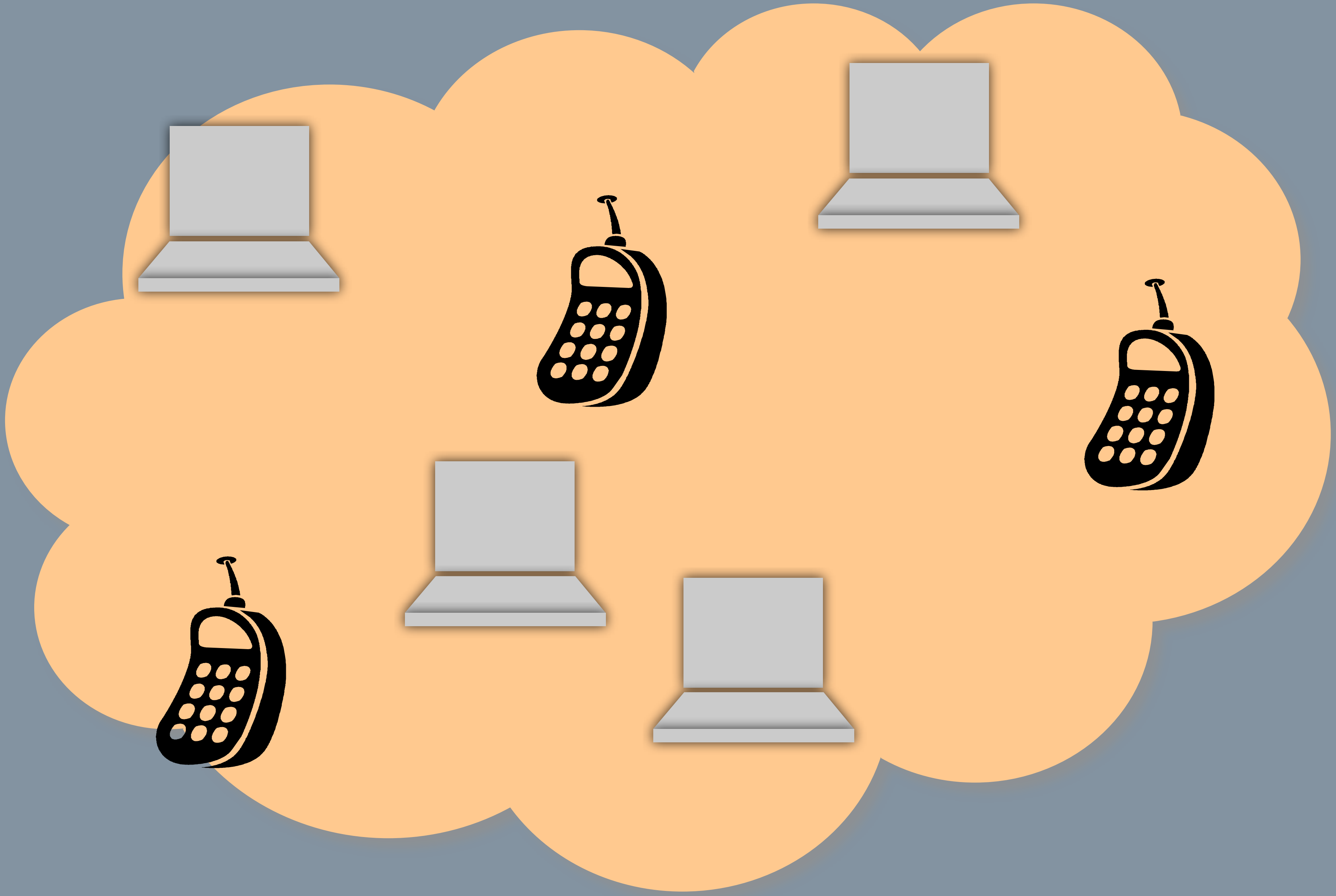


Motivation: Distributed Policies

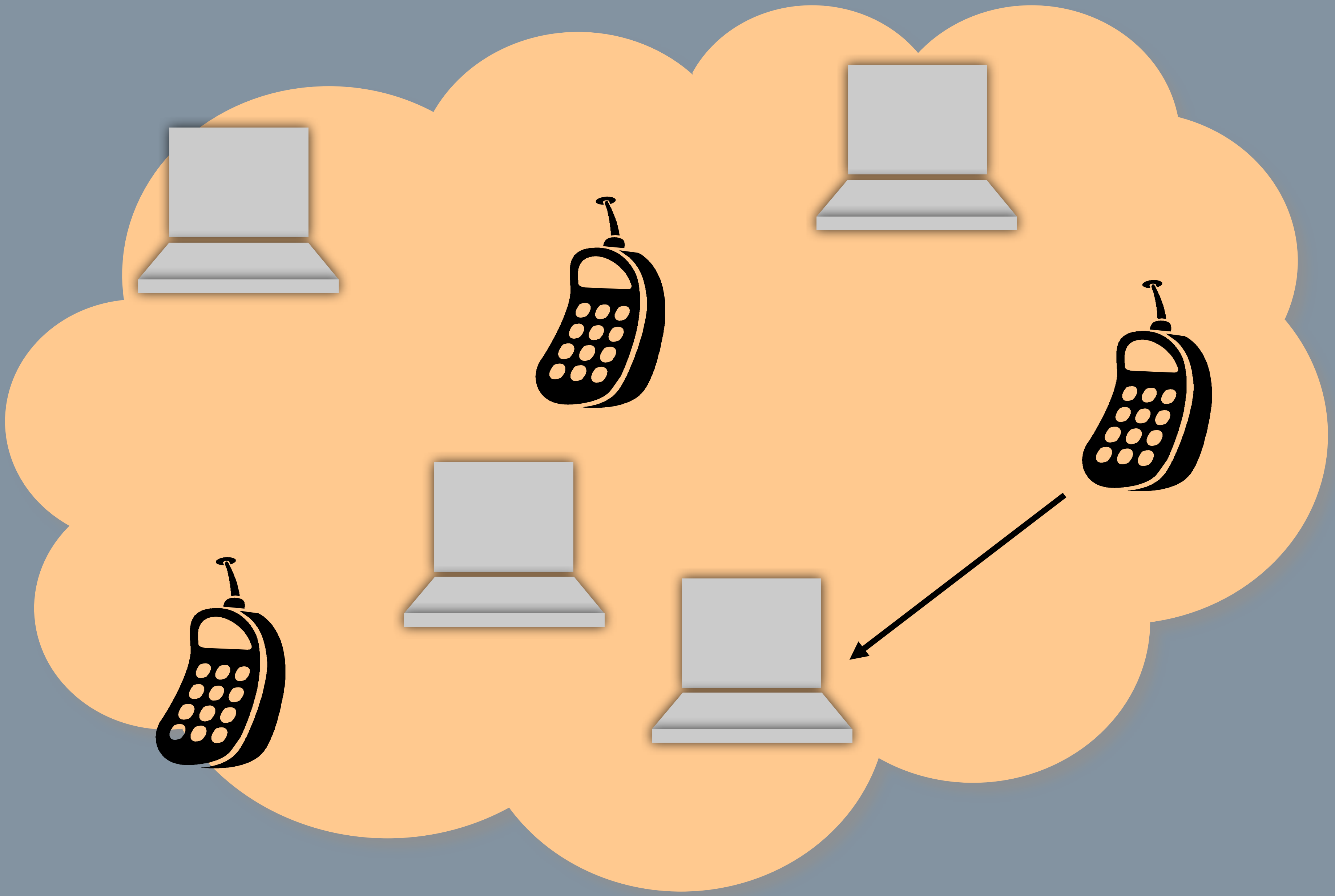
Motivation: Distributed Policies



Motivation: Distributed Policies



Motivation: Distributed Policies



Related Work

- Admission and Access control in NAC technologies
  - PacketFence, Cisco NAC, Microsoft NAP, Symantec
  - All pre- and post-connect policies are manually determined

- Admission and Access control in MANETS
  - Threshold cryptography: no specification of how decision is made (Narasimha et al. '03, Yin et al. '07)
  - IDS used for access control at a routing level (AODV, DSR) not application level (Parker et al. '04, Huang et al. '03)

Problem Statement

- *How to automate the creation and update of admission and access control policies in a network?*

- *We seek a solution that should be applicable to both centralized network architectures (NAC) and distributed network architectures like MANETs.*

Solution: Behavior-based Admission and Access Control

- *Derive admission and access control policies by* ***profiling the behavior of network devices****.*
- *Behavior profiles modeled by an AD sensor will be used to automatically identify what constitutes normal behavior within the network.*
- *Behavior is defined as the payload or volumetric characteristics of the network traffic exchanged.*

Solution: Behavior-based Admission and Access Control

- *Derive admission and access control policies by* ***profiling the behavior of network devices****.*
- *Behavior profiles modeled by an AD sensor will be used to automatically identify what constitutes normal behavior within the network.*
- *Behavior is defined as the payload or volumetric characteristics of the network traffic exchanged.*
    - ▶De Facto vs. De Jure policies

Contributions

- Enhancement by automating the creation of behavior-based policies for admission and access control
- Automatic and robust update of behavior-based policies as profiles evolve over time
- An approach that is applicable to both centralized and fully distributed networks

Outline

- Behavior-based NAC Mechanism
- Automatic Clustering and Policy Update
- Cluster-based AD sensor
- Behavior-based Mechanism for MANETs

Behavior-based NAC Mechanism

Behavior-based NAC Mechanism

- Each device has an AD sensor to model,
  - ▶Input behavior profile
  - ▶Output behavior profile
  - ▶Bad profile: malware knowledge
  - ▶All are automatically updated by the AD sensor

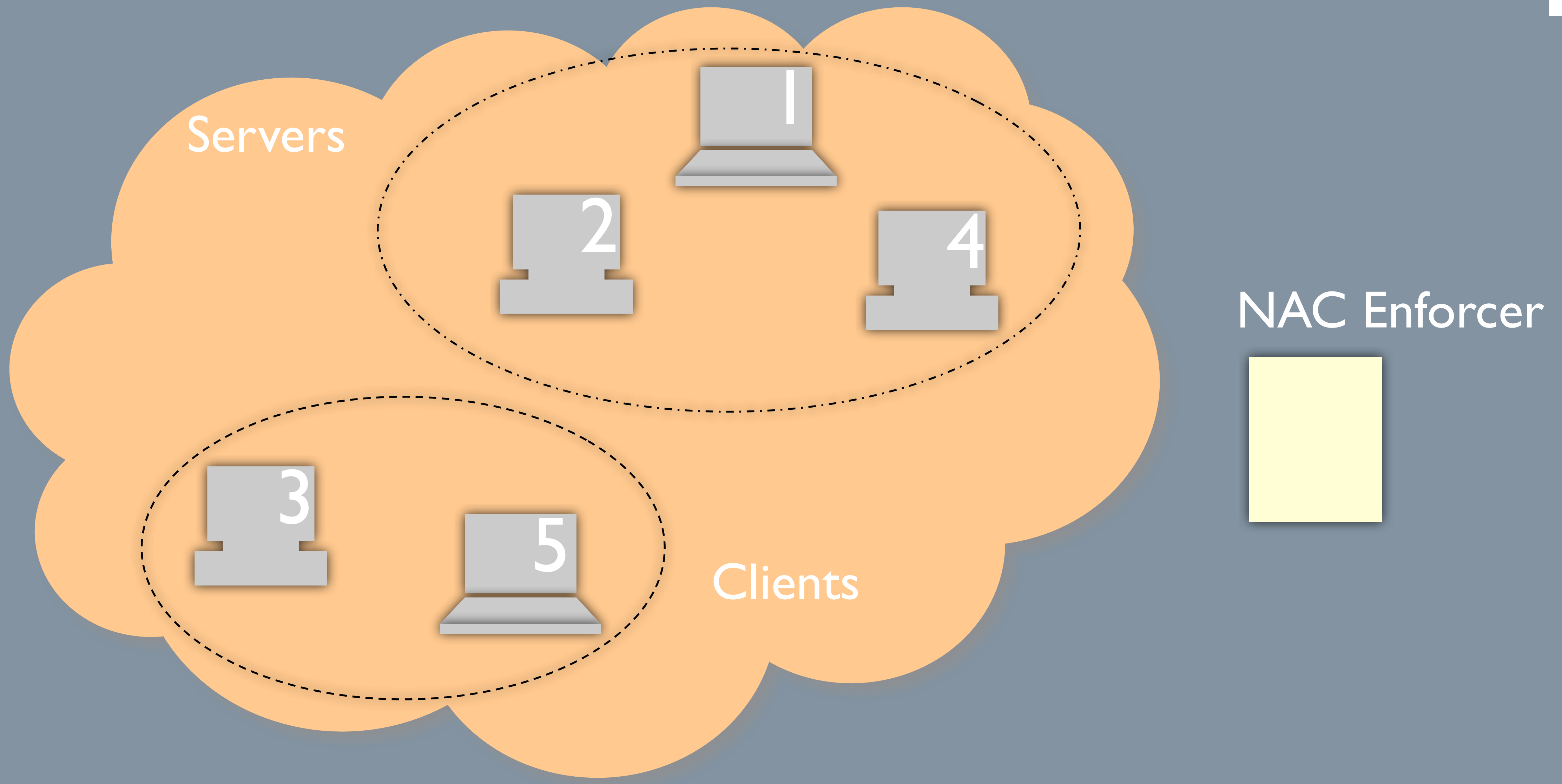
Behavior-based NAC Mechanism

- Each device has an AD sensor to model,
  - ▶Input behavior profile
  - ▶Output behavior profile
  - ▶Bad profile: malware knowledge
  - ▶All are automatically updated by the AD sensor
- BB-NAC Phases:
  - ▶ Initial Setup
  - ▶ Admission Control (Pre-connect Phase)
  - ▶ Access Control (Post-connect Phase)

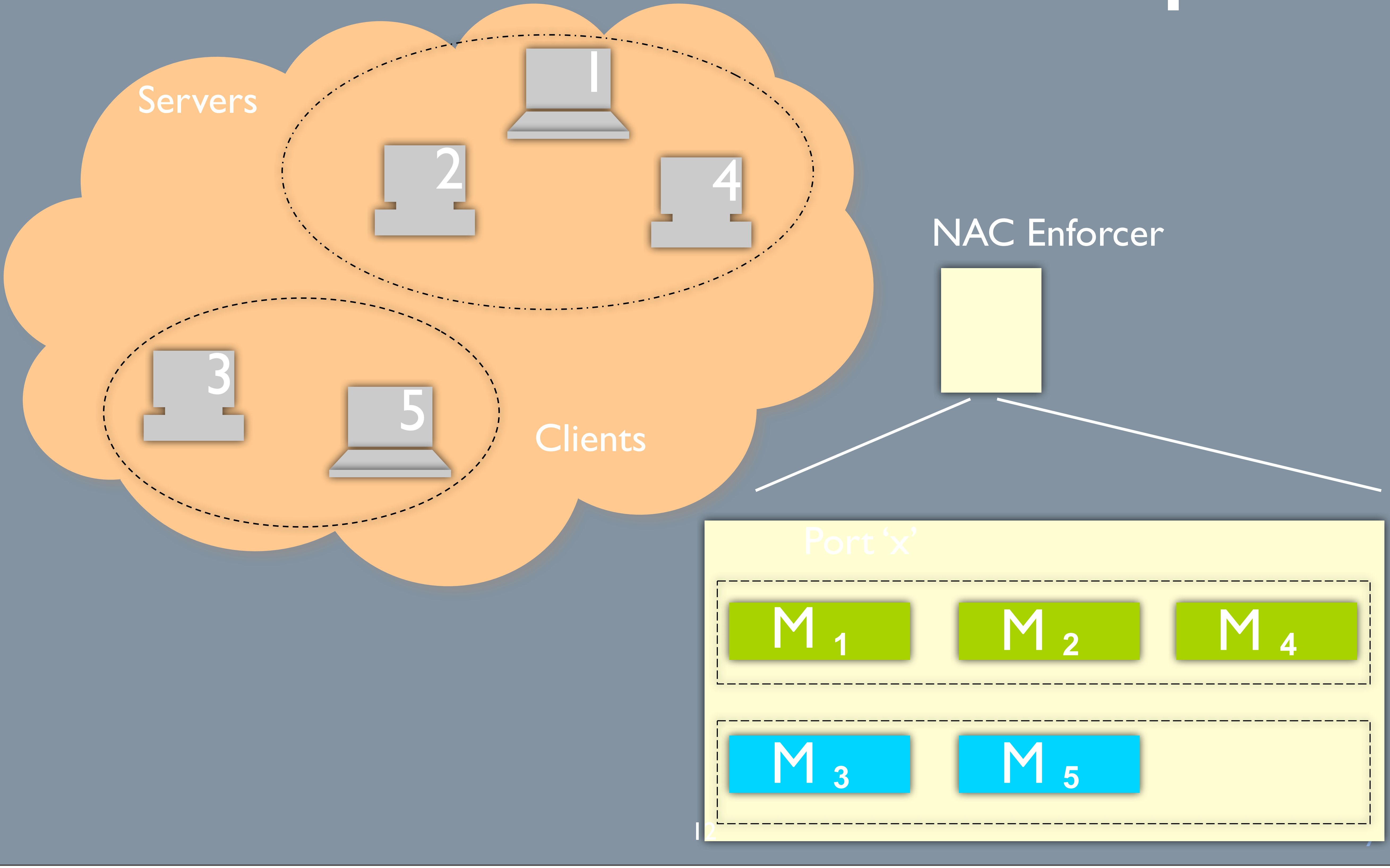
Behavior-based NAC Mechanism

- Each device has an AD sensor to model,
  - ▶Input behavior profile
  - ▶Output behavior profile
  - ▶Bad profile: malware knowledge
  - ▶All are automatically updated by the AD sensor
- BB-NAC Phases:
  - ▶ Initial Setup
  - ▶ Admission Control (Pre-connect Phase)
  - ▶ Access Control (Post-connect Phase)
- NAC enforcer and high-ranked monitors responsible for the admission and access control

BB-NAC: Initial Setup



BB-NAC: Initial Setup



Initial Setup

- Pre-determined clusters of behavior profiles:
  - Clients and servers (per port and direction)
  - Each cluster of behavior profiles represents a valid

    behavior for the network

- For each cluster member, NAC enforcer determines a threshold

  or local boundary of normal behavior,

  $t_{P_i} = \max_{j=0..n} d(P_i, P_j)$ for each profile P_i in cluster

- Each device is represented by $M_i = \{P_i, t_{P_i}, B_i\}$

Admission Control

P_{new} B_{new}

Admission Control

- New device with behavior profile P_{new} and bad profile B_{new}

Admission Control

- New device with behavior profile P_{new} and bad profile B_{new} solicits admission to network

Admission Control

- New device with behavior profile P_{new} and bad profile B_{new} solicits admission to network
- Behavior Profile Check
  - ▶Is the behavior of the device similar to any cluster of behavior in the network?

Admission Control

- New device with behavior profile P_{new} and bad profile B_{new} solicits admission to network
- Behavior Profile Check
  - ▶Is the behavior of the device similar to any cluster of behavior in the network?
- Bad Profile Check
  - ▶ Does the device have enough malware knowledge?

Admission Control

- New device with behavior profile P_{new} and bad profile B_{new} solicits admission to network
- Behavior Profile Check
  - ▶Is the behavior of the device similar to any cluster of behavior in the network?
- Bad Profile Check
  - ▶ Does the device have enough malware knowledge?
- If admission rejected multiple times, device IP is blacklisted

Behavior Profile Check

Behavior Profile Check

- Group decision based on the sum of individual decisions from each member of the closest cluster (percentage of agreement)

$$cluster = \min_{i=0..n} d(c[i], P_{new})$$

Behavior Profile Check

- Group decision based on the sum of individual decisions from each member of the closest cluster (percentage of agreement)
- Individual decisions evaluate the difference between their local behavior and the newcomer's behavior

$$cluster = \min_{i=0..n} d(c[i], P_{new})$$

$$v_i = 1 \text{ if } \; d(P_i, P_{new}) \leq t_{P_i} \text{ where } P_i \in cluster$$

Behavior Profile Check

- Group decision based on the sum of individual decisions from each member of the closest cluster (percentage of agreement)
- Individual decisions evaluate the difference between their local behavior and the newcomer's behavior

$$cluster = \min_{i=0..n} d(c[i], P_{new})$$

$$v_i = 1 \text{ if } d(P_i, P_{new}) \leq t_{P_i} \text{ where } P_i \in cluster$$

Behavior Profile Check

- Group decision based on the sum of individual decisions from each member of the closest cluster (percentage of agreement)
- Individual decisions evaluate the difference between their local behavior and the newcomer's behavior

▶ Closest Cluster:

$$cluster = \min_{i=0..n} d(c[i], P_{new})$$

$$v_i = 1 \text{ if } d(P_i, P_{new}) \le t_{P_i} \text{ where } P_i \in cluster$$

Behavior Profile Check

- Group decision based on the sum of individual decisions from each member of the closest cluster (percentage of agreement)
- Individual decisions evaluate the difference between their local behavior and the newcomer's behavior

▶ Closest Cluster:

$$cluster = \min_{i=0..n} d(c[i], P_{new})$$

▶ Individual decision:

$$v_i = 1 \text{ if } d(P_i, P_{new}) \le t_{P_i} \text{ where } P_i \in cluster$$

Behavior Profile Check

- Group decision based on the sum of individual decisions from each member of the closest cluster (percentage of agreement)

- Individual decisions evaluate the difference between their local behavior and the newcomer's behavior

▶ Closest Cluster:

$$cluster = \min_{i=0..n} d(c[i], P_{new})$$

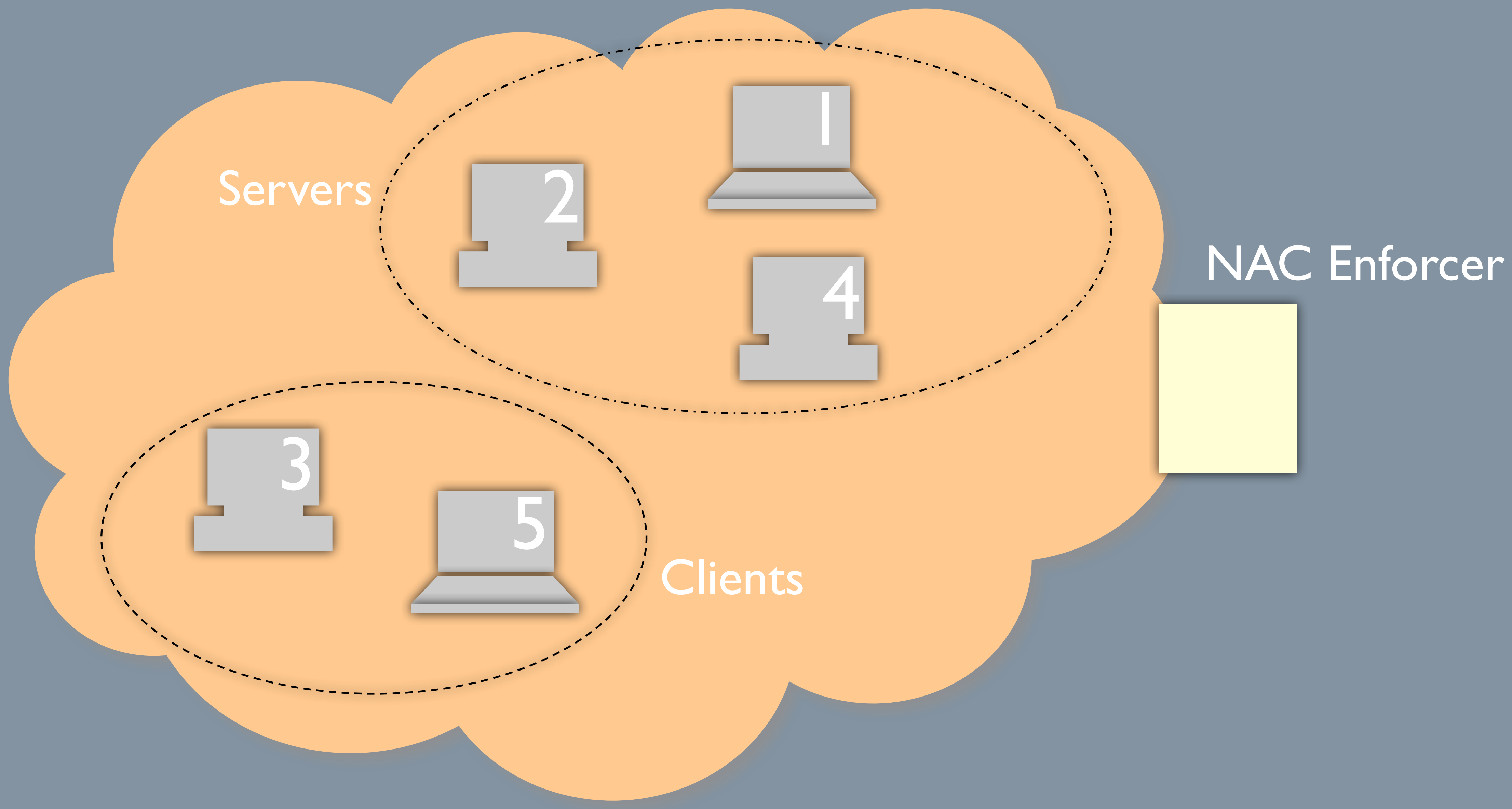
▶ Individual decision:

$$v_i = 1 \text{ if } d(P_i, P_{new}) \le t_{P_i} \text{ where } P_i \in cluster$$

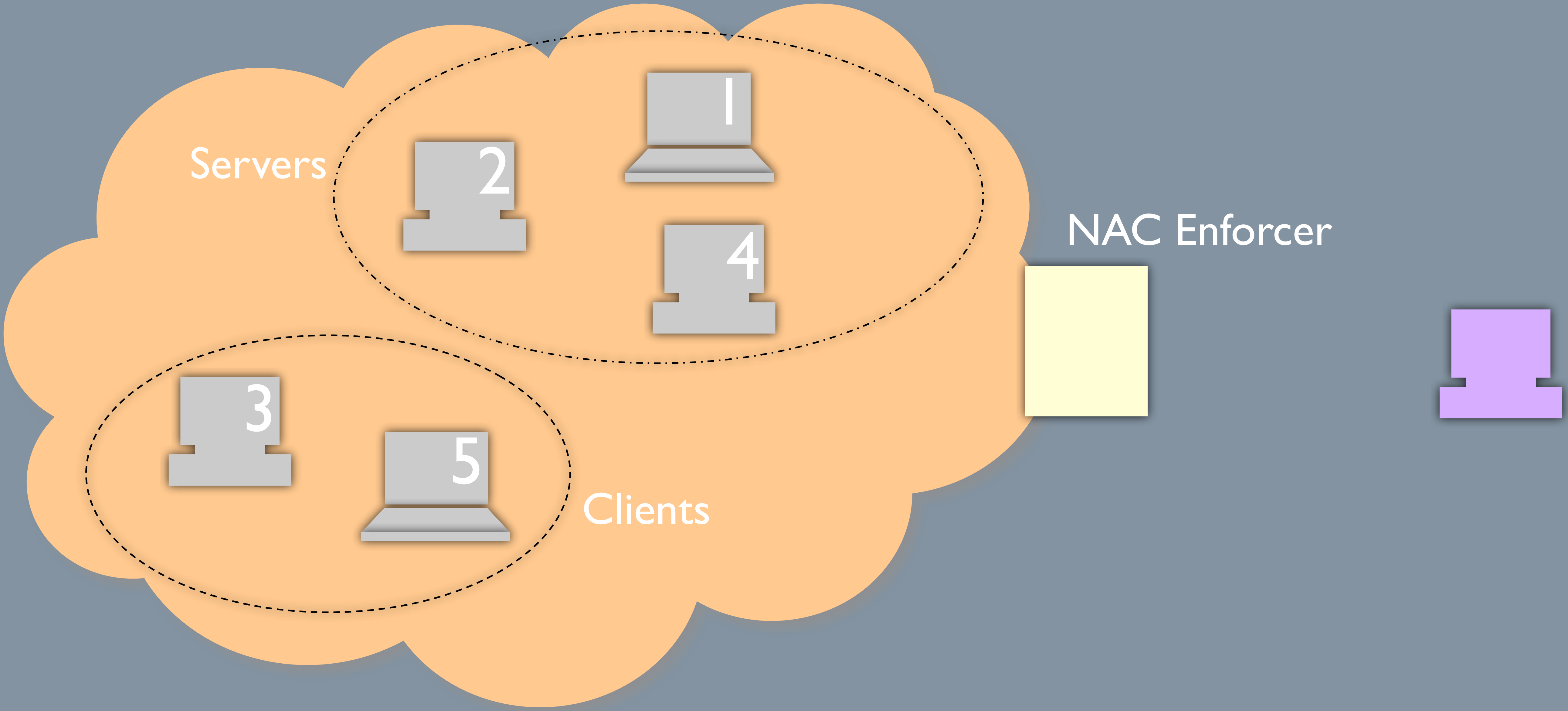
▶ Final Group Decision:

$$v = \frac{1}{n} \sum_{i=0..n} v_i \quad \text{if } v \succ \varepsilon \quad \text{ACCEPT}$$

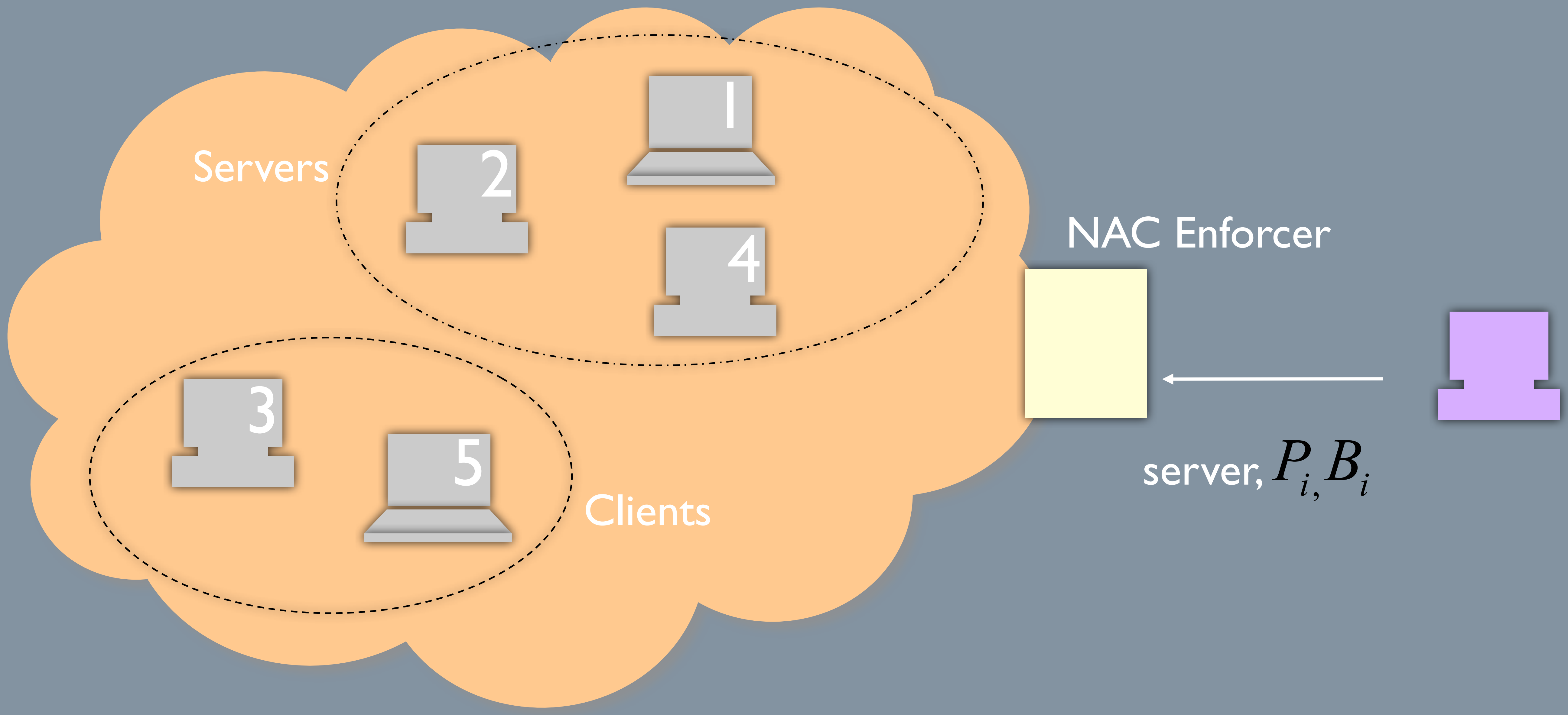
Pre-connect Phase: Example

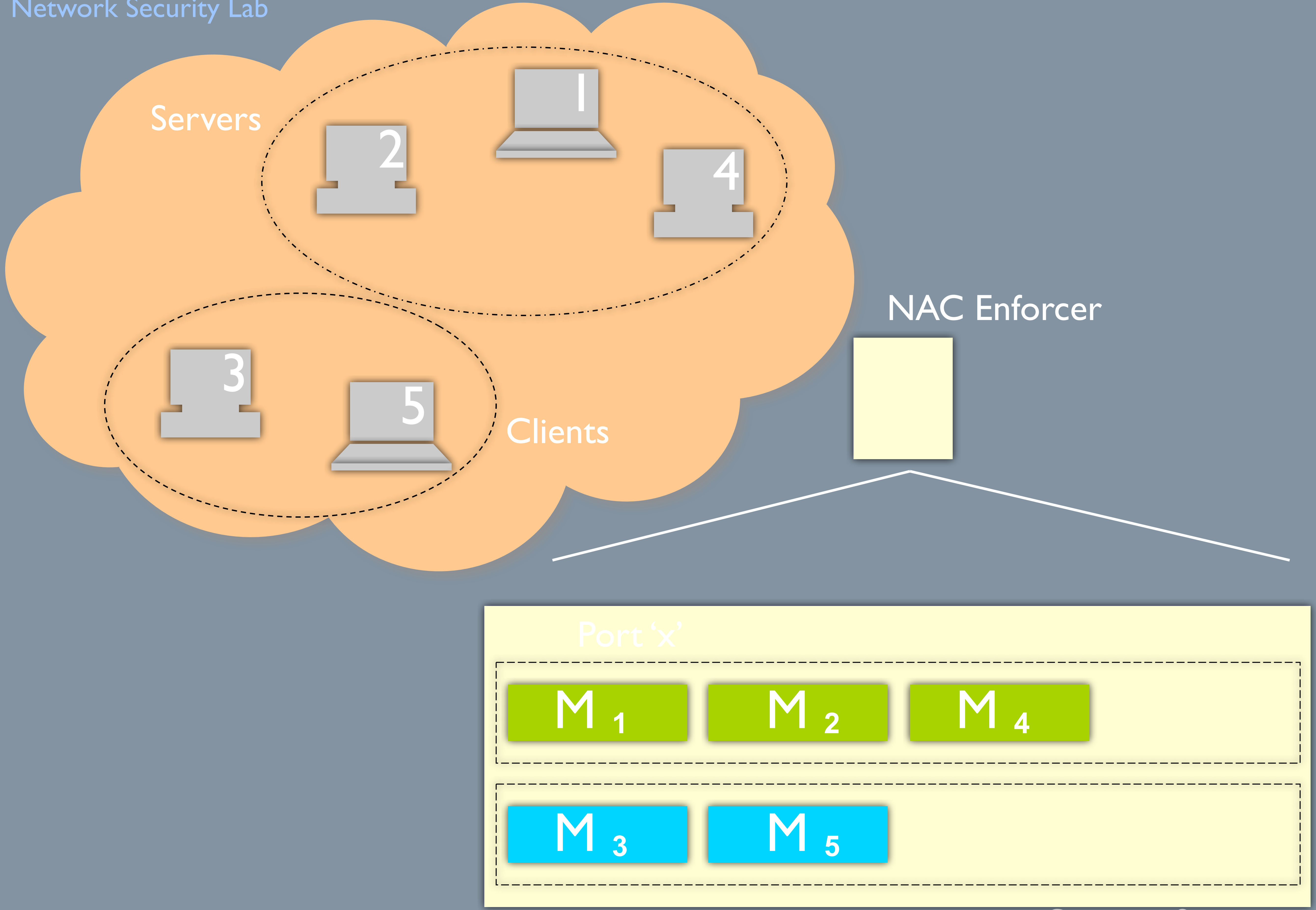


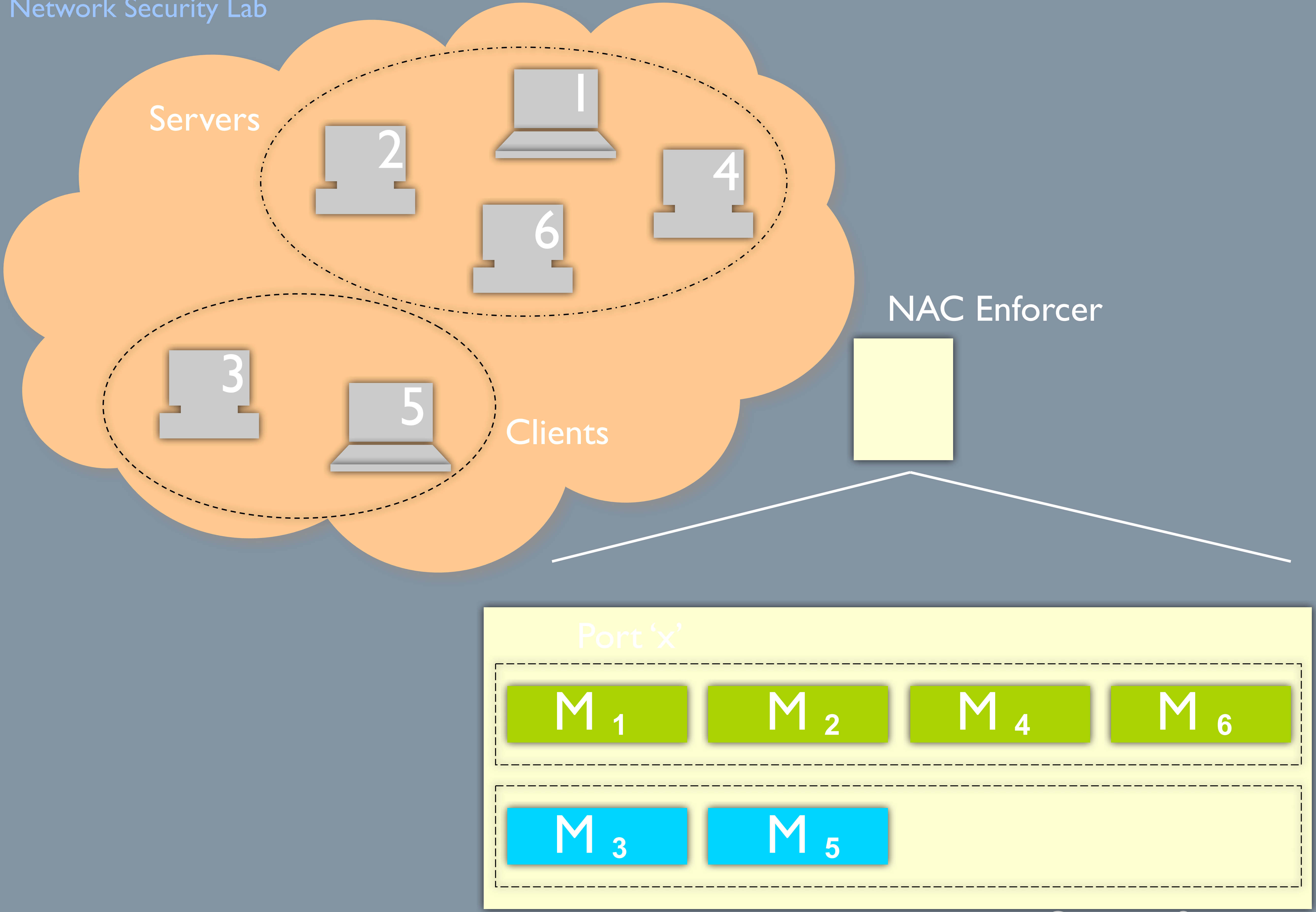
Pre-connect Phase: Example



Pre-connect Phase: Example




Servers
1
2
4
3
5
Clients
NAC Enforcer
Port ‘x’
M_1
M_2
M_4
M_3
M_5


Servers
1
2
4
6
3
5
Clients
NAC Enforcer
Port 'x'
M 1
M 2
M 4
M 6
M 3
M 5

Bad Profile Check

Bad Profile Check

- Group decision: each member evaluates malware knowledge of newcomer

$$v_i = 1 \text{ if } B_i \subset \mathrm{B}_{\mathrm{new}} \text{ where } B_i \in cluster$$

Bad Profile Check

- Group decision: each member evaluates malware knowledge of newcomer

▶ Individual decision:

$$v_i = 1 \text{ if } B_i \subset \mathrm{B}_{\text{new}} \text{ where } B_i \in cluster$$

▶ Final Group Decision:

$$v = \frac{1}{n} \sum_{i=0..n} v_i \quad \text{if } v \succ \varepsilon \quad \text{ACCEPT}$$

Access Control

- Traffic is deemed normal or anomalous based on a group-profile decision
- All members in a cluster vote, not only the ones exchanging traffic: group knowledge

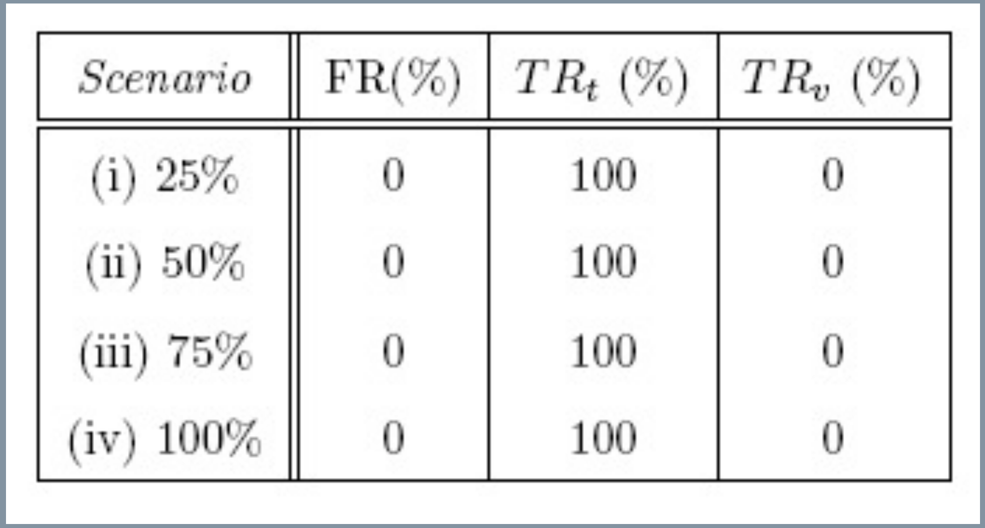
$$v = \frac{1}{n}\sum_{k=1}^{n} P_{k,d}(g) \quad \text{if } v \succ \varepsilon \quad \text{ANOMALY}$$

- If anomaly detected, generate alert / quarantine device
- Anomalous traffic is used to update bad profiles
- Normal traffic is used to train behavior profiles

Proof-of-concept Experiments

- Four CS webservers (servers cluster, port 80)
- ANAGRAM AD sensor: input content profiles (BFs)
- Initial Setup
  - Stable profiles computed with two weeks of traffic
- Admission Control (FR/TR)
  - Normal Testing Set
    - Stable profiles computed for the following two weeks
  - Malicious Testing Set
    - Profiles for 8 DNS servers computed for same two weeks
    - Normal profiles poisoned with one virus from *vxheavens*
- Access Control (FP/DR)
  - Traffic from worms*: CodeRed, Webdav, Mirela, php worms*

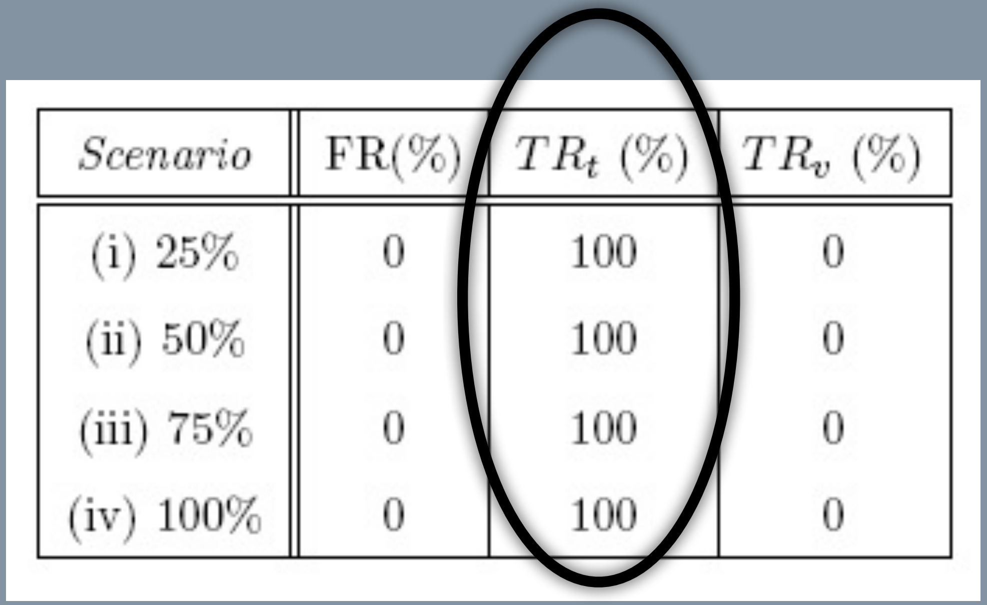
Experimental Results



Scenario	FR(%)	TR_t (%)	TR_v (%)
(i) 25%	0	100	0
(ii) 50%	0	100	0
(iii) 75%	0	100	0
(iv) 100%	0	100	0

Experimental Results

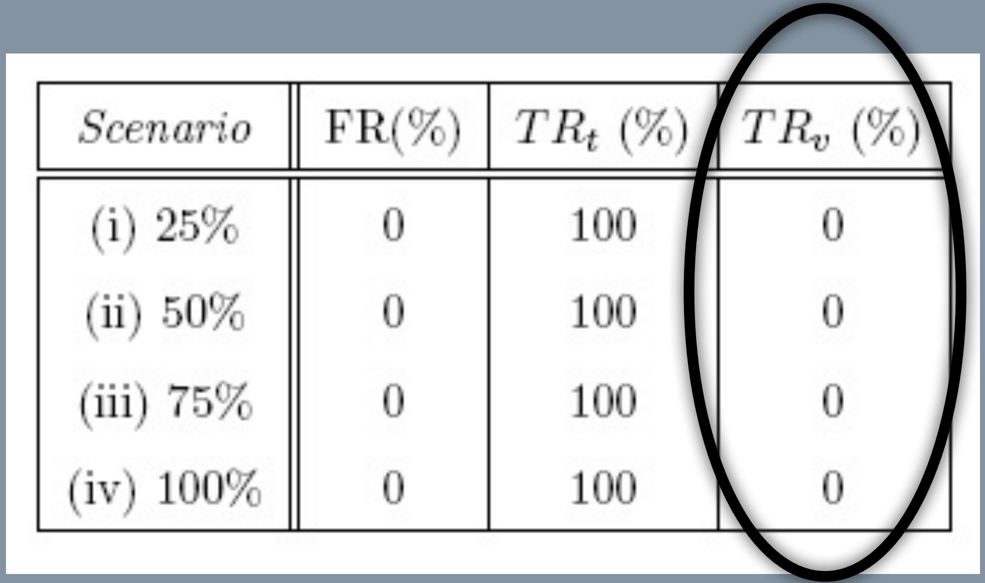
Profiles port 53



Scenario	FR(%)	TR_t (%)	TR_v (%)
(i) 25%	0	100	0
(ii) 50%	0	100	0
(iii) 75%	0	100	0
(iv) 100%	0	100	0

Experimental Results

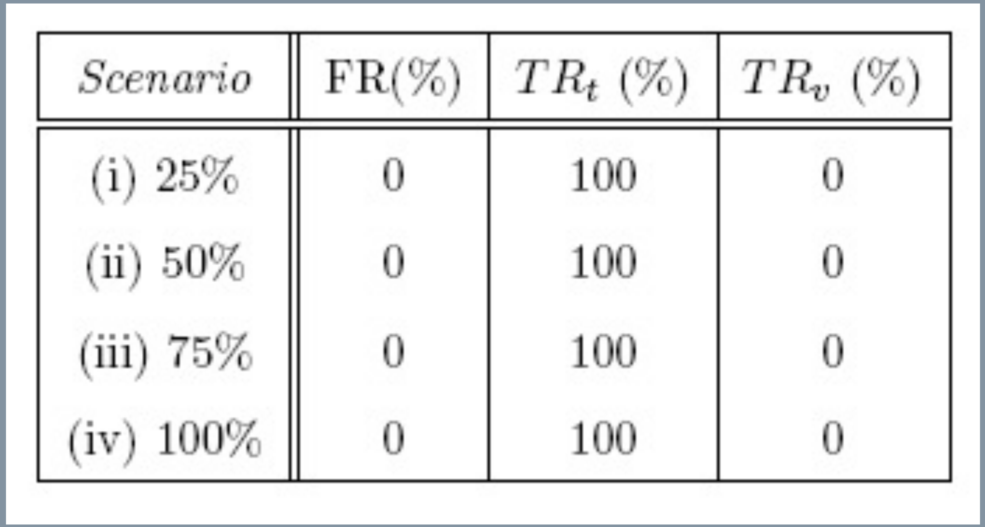
Profiles poisoned



Scenario	FR(%)	TR_t (%)	TR_v (%)
(i) 25%	0	100	0
(ii) 50%	0	100	0
(iii) 75%	0	100	0
(iv) 100%	0	100	0

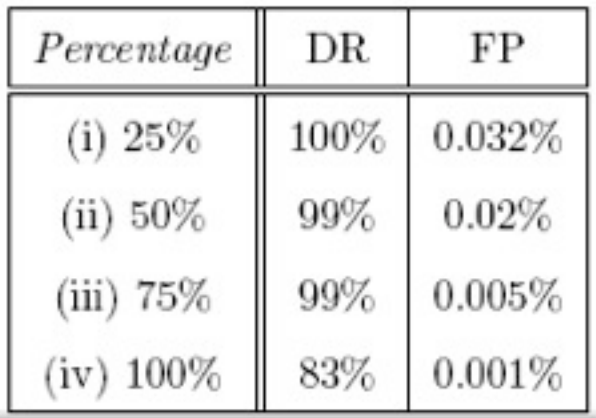
Experimental Results

- Admission Control



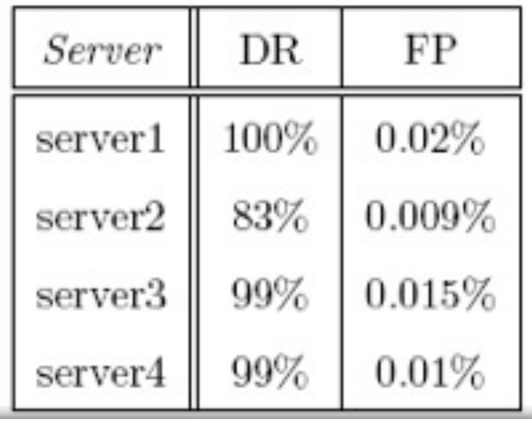
Scenario	FR(%)	TR_t (%)	TR_v (%)
(i) 25%	0	100	0
(ii) 50%	0	100	0
(iii) 75%	0	100	0
(iv) 100%	0	100	0

- Access Control



Percentage	DR	FP
(i) 25%	100%	0.032%
(ii) 50%	99%	0.02%
(iii) 75%	99%	0.005%
(iv) 100%	83%	0.001%

Group Rates

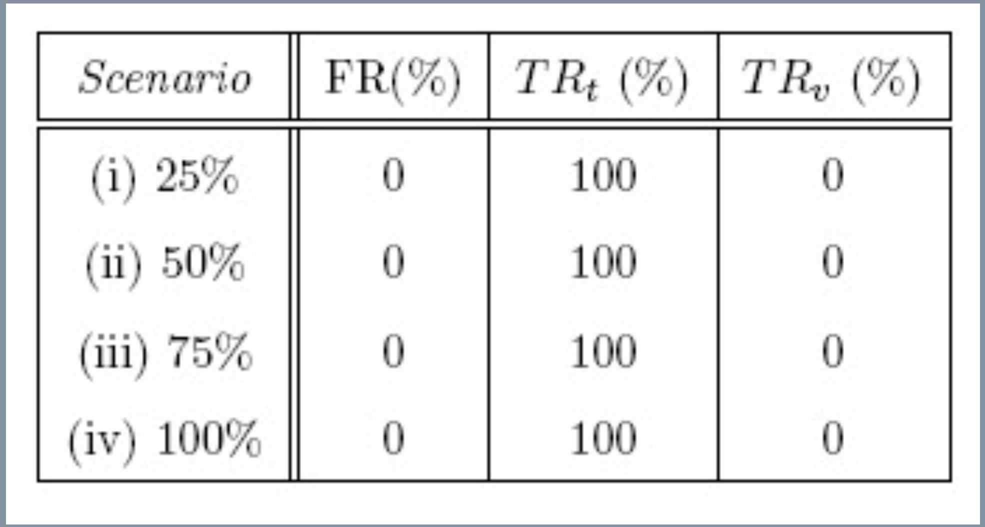


Server	DR	FP
server1	100%	0.02%
server2	83%	0.009%
server3	99%	0.015%
server4	99%	0.01%

Individual Rates

Experimental Results

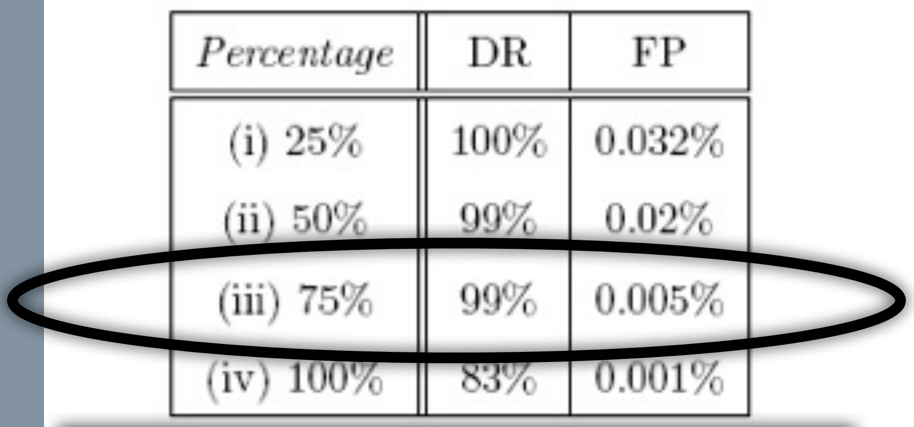
- Admission Control



Scenario	FR(%)	TR_t (%)	TR_v (%)
(i) 25%	0	100	0
(ii) 50%	0	100	0
(iii) 75%	0	100	0
(iv) 100%	0	100	0

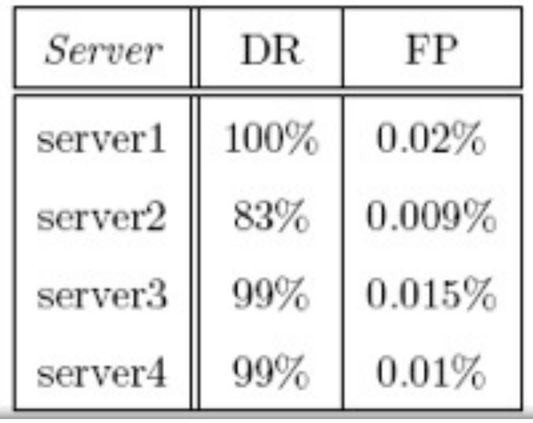
- Access Control

Best Collaborative Solution ε =75%



Percentage	DR	FP
(i) 25%	100%	0.032%
(ii) 50%	99%	0.02%
(iii) 75%	99%	0.005%
(iv) 100%	83%	0.001%

Group Rates



Server	DR	FP
server1	100%	0.02%
server2	83%	0.009%
server3	99%	0.015%
server4	99%	0.01%

Individual Rates

Concept Drift in Behavior Profiles

Concept Drift in Behavior Profiles

$$P_i = P_i \wedge P_{i-1} \wedge \ldots \wedge P_{i-q}$$

Concept Drift in Behavior Profiles

$$P_i = P_i \wedge P_{i-1} \wedge \ldots \wedge P_{i-q}$$

Concept Drift in Behavior Profiles

- AllModels: keep all previous knowledge

$$P_i = P_i \wedge P_{i-1} \wedge \ldots \wedge P_{i-q}$$

Concept Drift in Behavior Profiles

- AllModels: keep all previous knowledge

$$P_i = P_i \wedge P_{i-1} \wedge \ldots \wedge P_{i-q}$$

Concept Drift in Behavior Profiles

- AllModels: keep all previous knowledge

- ANDModels: keep only common knowledge between models

$$P_i = P_i \wedge P_{i-1} \wedge \ldots \wedge P_{i-q}$$

Concept Drift in Behavior Profiles

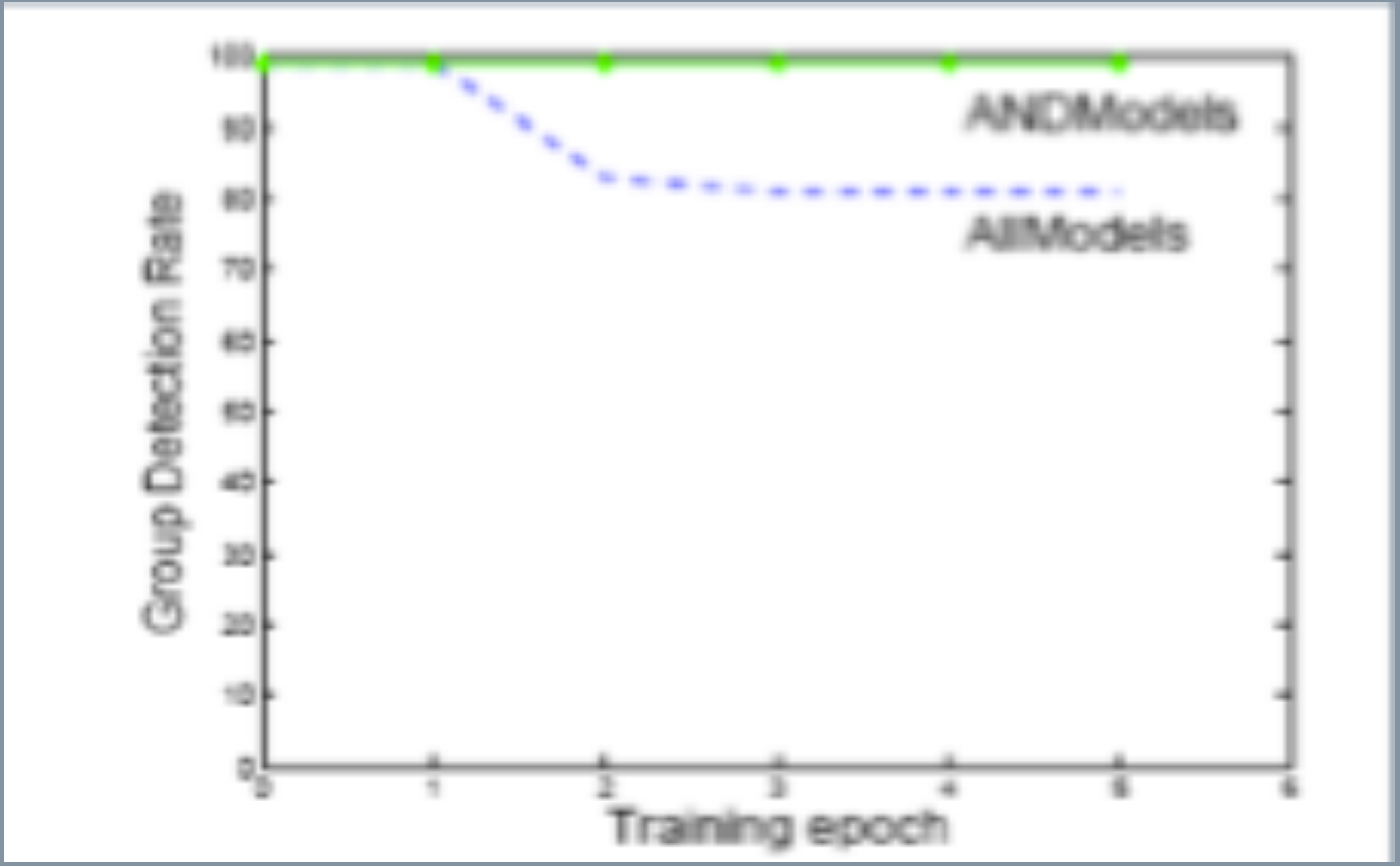
- AllModels: keep all previous knowledge
- ANDModels: keep only common knowledge between models

$$P_i = P_i \wedge P_{i-1} \wedge \ldots \wedge P_{i-q}$$

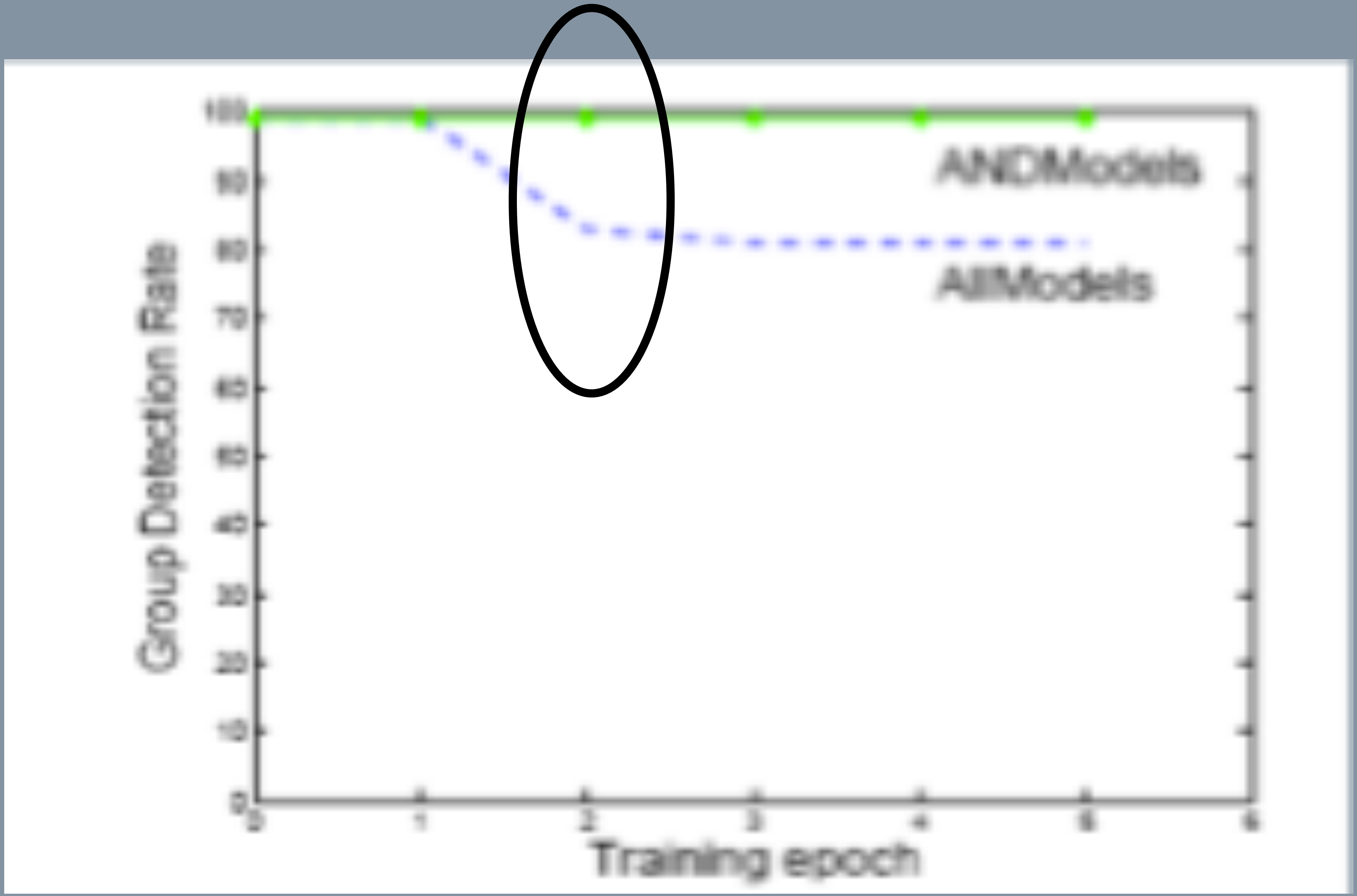
- Add new behaviors, OR ANDModels:

$$P_i = \vee_{t=0}^{s-1} P_{i-t} \wedge P_{i-1-t} \wedge \ldots \wedge P_{i-q-t}$$

Concept Drift in Behavior Profiles



Concept Drift in Behavior Profiles



BB-NAC Latency Analysis

BB-NAC Latency Analysis

- Pre-connect latency

$$l = l_a + (1 - \rho) \times l_q$$

BB-NAC Latency Analysis

- Pre-connect latency

$$l = l_a + (1 - \rho) \times l_q$$

$$l = (1 - FP) \times l_{BF} + FP \times l_q$$

BB-NAC Latency Analysis

- Pre-connect latency

$$l = l_a + (1 - \rho) \times l_q$$

- Post-connect latency

$$l = (1 - FP) \times l_{BF} + FP \times l_q$$

BB-NAC Latency Analysis

- Pre-connect latency

$$l = l_a + (1 - \rho) \times l_q$$

- Post-connect latency

$$l = (1 - FP) \times l_{BF} + FP \times l_q$$

BB-NAC Latency Analysis

- Pre-connect latency

$$l = l_a + (1 - \rho) \times l_q$$

- Post-connect latency

$$l = (1 - FP) \times l_{BF} + FP \times l_q$$

- BB-NAC values (cluster of 10 devices):
  - Pre-connect: 180~342ms
  - Post-connect (Best collaborative): 5,785~50.56ms

Summary of Results

- ☑ New mechanism to automatically create *De Facto* admission and access control policies

- ☑ A novel admission and access control based on a profile-group decision process which may outperform individual decision processes

- ☑ Proof-of-concept evaluation using content behavior profiles (hashed into BFs) from 4 webservers

Outline

- Behavior-based NAC technologies
  - –Automatic Clustering and Policy Update
- Cluster-based AD sensor
- Behavior-based Policies for MANETs
- Conclusions and Future Work

Automatic Clustering and Policy Update *

- Automatic computation of clusters of behavior
  - Initial setup complemented with *k-means clustering*
  - Other phases proceed normally

- Robust evolution of clusters of behavior over time
  - Incremental-learning algorithm: differentiate between *concept drift* and *attack*

- NAC enforcer is responsible for both tasks

* ACSAC'09

Clustering

- Initially all devices in network communicate behavior profile to NAC enforcer

- K-means clustering is performed on a per port, per direction basis to identify clusters of similar behavior

- Although computationally expensive, is performed only once during the setup

- Best number of clusters is selected via cross-validation based on best performance for the access control

Clustering

- Behavior Profiles are of the type $p_i = \{p_i[0], p_i[1], ...p_i[n]\}$
  - Each field $p_i[\ell]$ represents the average measure of any volumetric characteristic of the connections established by a user
- Distances between profiles: euclidean distance

$$d(p_i, p_j) = \sqrt{(}\sum_{\ell=0..n} (p_i[\ell] - p_j[\ell])^2)$$

- Normalization of measurements to avoid distance calculation dominated by measurements at different scales

Clustering

- Kmeans++: smart selection of initial seeds (non-deterministic)

- Cross-validation determines best values to be used during admission and access control, r = (1-FR) + TR

  - ▶ Best k distribution
  - ▶ Best percentage of agreement ε

$$v = \frac{1}{n} \sum_{i=0..n} v_i$$

  - ▶ Weighted or non-weighted voting (w)

$$w_i = \frac{d_{\max} - d_i}{d_{\max} - d_{\min}} \times v_i$$

BOOTSTRAP

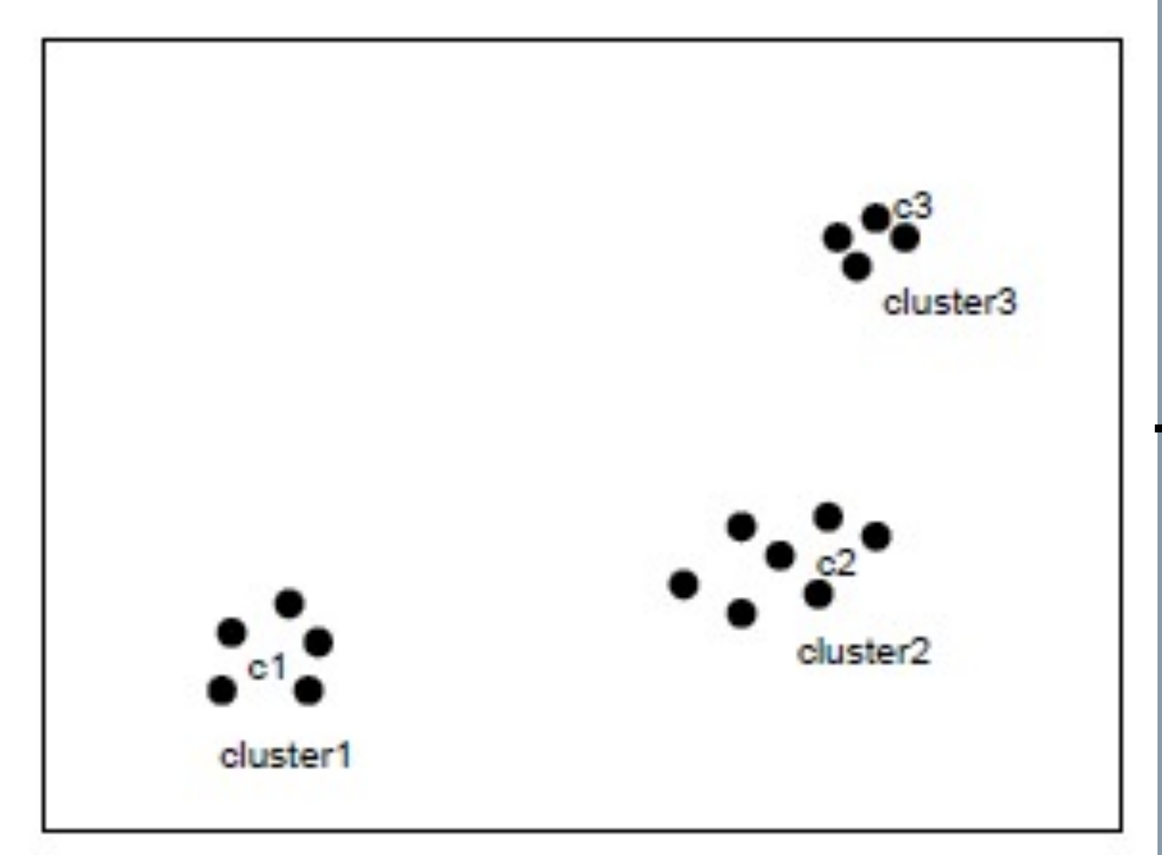
- NAC enforcer computes distance between each profile in the cluster and the others

- This measurement represents a threshold used during access control to determine which devices can be accepted into the network

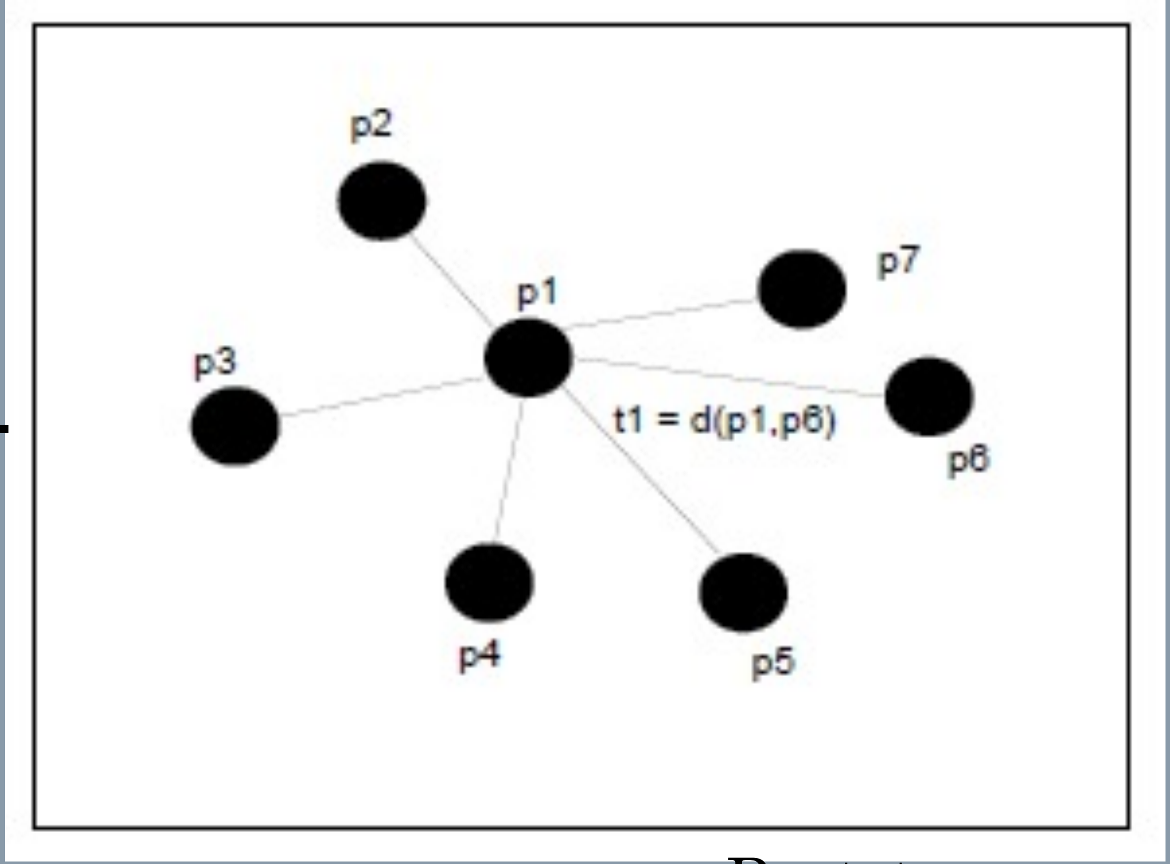
ACCESS CONTROL

- Upon arrival, newcomer presents its profile
- NAC enforcer computes closest behavioral cluster
- NAC enforcer computes a “voting process” across profiles in closest cluster
  - Members vote for or against with thresholds computed during bootstrap phase
  - Simple voting: all votes count equally. If 50% or more agree on normalcy, device is accepted
  - Weighted voting: profiles closer to newcomer have a stronger vote

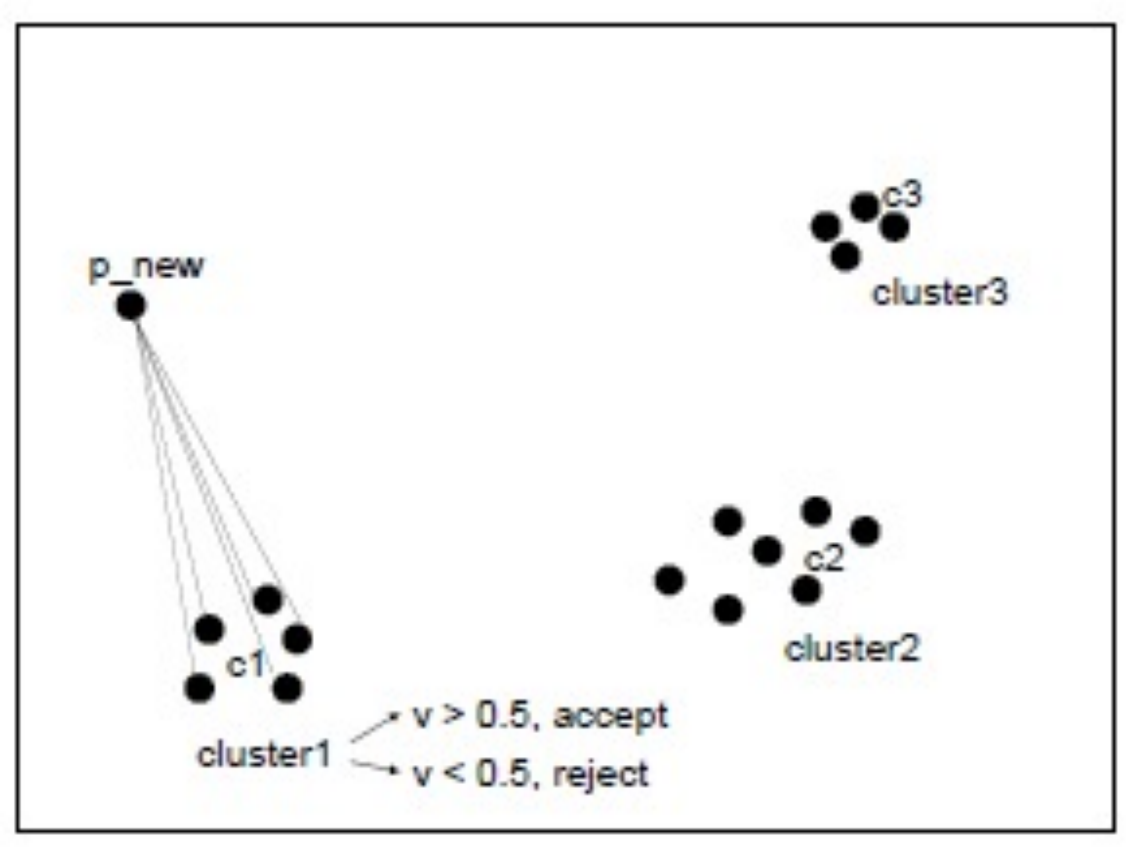
EXAMPLE



Clustering



Bootstrap



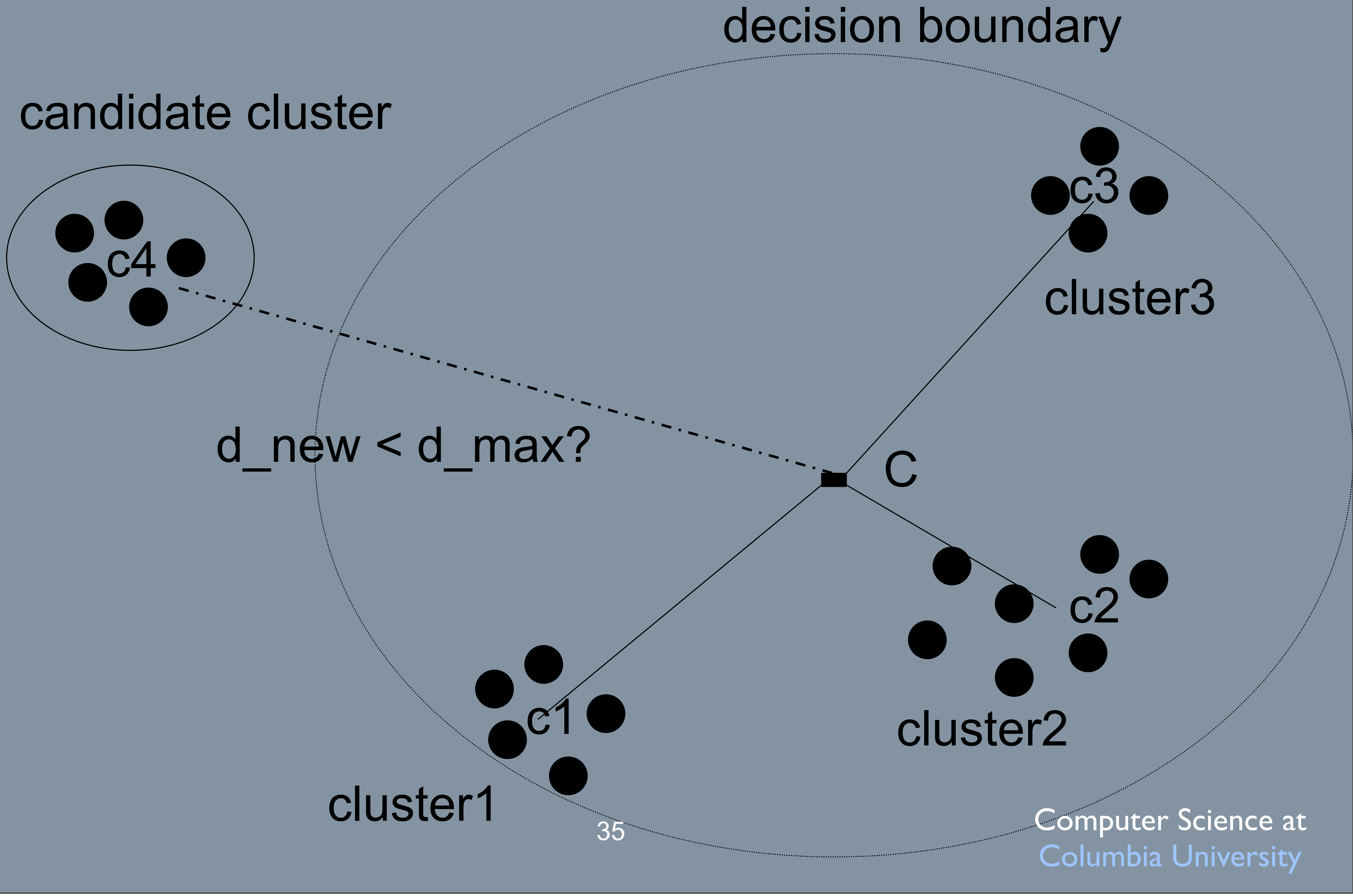
Access Control

Policy Update

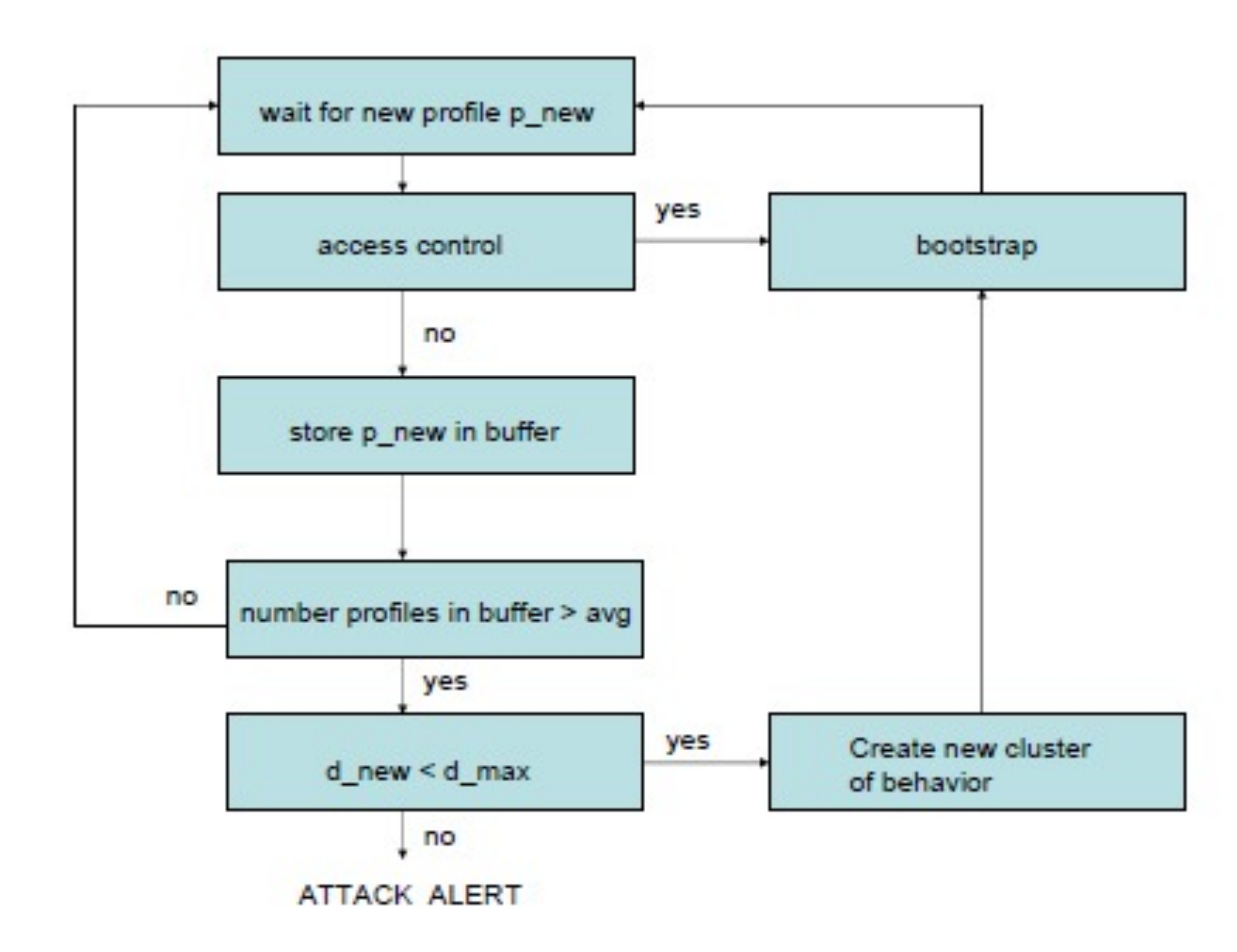
- Over time, new behavior profiles may be computed
  - Admitted as normal by one of the existing clusters
  - New behavior evolved from old behavior (concept drift)
  - Malicious behavior profile (attack)

- Behaviors are defined as clusters
  - New behavior accepted if defines a cluster within the boundaries of existing behaviors

Incremental-Learning Algorithm



INCREMENTAL LEARNING ALGORITHM 2



wait for new profile p_new

access control

yes

bootstrap

no

store p_new in buffer

no

number profiles in buffer > avg

yes

d_new < d_max

yes

Create new cluster of behavior

no

ATTACK ALERT

Attacks against the Mechanism

- Collusion Attacks
  - Attackers try to craft malicious behavior profiles to create a new cluster
- Threshold Attacks
  - Attackers try to modify thresholds to change dimensions of own cluster
  - Unified: stretch in one direction
  - Diversified: stretch in multiple directions
- Experimental results show robustness against both types of attacks

Experimental Evaluation

} Ground Truth

Experimental Evaluation

- Cisco NetFlow logs from router at Columbia University
- 300 Columbia IPs (hosts) were randomly selected for *port 80*
  - 300 Profiles computed with their flows (week1)
  - 300 Profiles computed with their flows (week2)
  - Each behavior characterized 7 flow parameters:
    - total number of flows
    - average flow size

Ground Truth

Experimental Evaluation

- Cisco NetFlow logs from router at Columbia University
- 300 Columbia IPs (hosts) were randomly selected for *port 80*
  - 300 Profiles computed with their flows (week1)
  - 300 Profiles computed with their flows (week2)
  - Each behavior characterized 7 flow parameters:
    - total number of flows
    - average flow size
    - average flow duration

Ground Truth

Experimental Evaluation

- Cisco NetFlow logs from router at Columbia University
- 300 Columbia IPs (hosts) were randomly selected for *port 80*
  - 300 Profiles computed with their flows (week1)
  - 300 Profiles computed with their flows (week2)

    } Ground Truth
  - Each behavior characterized 7 flow parameters:
    - total number of flows
    - average flow size
    - average flow duration
    - total number of packets contained in all flows

Experimental Evaluation

- Cisco NetFlow logs from router at Columbia University
- 300 Columbia IPs (hosts) were randomly selected for *port 80*
  - 300 Profiles computed with their flows (week1)
  - 300 Profiles computed with their flows (week2)
  - Each behavior characterized 7 flow parameters:
    - total number of flows
    - average flow size
    - average flow duration
    - total number of packets contained in all flows
    - average number of packets per flow

Ground Truth

Experimental Evaluation

- Cisco NetFlow logs from router at Columbia University
- 300 Columbia IPs (hosts) were randomly selected for *port 80*
  - 300 Profiles computed with their flows (week1)
  - 300 Profiles computed with their flows (week2)

    } Ground Truth
  - Each behavior characterized 7 flow parameters:
    - total number of flows
    - average flow size
    - average flow duration
    - total number of packets contained in all flows
    - average number of packets per flow
    - total number of unique IP addresses contained in all flows

Experimental Evaluation

- Cisco NetFlow logs from router at Columbia University
- 300 Columbia IPs (hosts) were randomly selected for *port 80*
  - 300 Profiles computed with their flows (week1)
  - 300 Profiles computed with their flows (week2)

    Ground Truth
  - Each behavior characterized 7 flow parameters:
    - total number of flows
    - average flow size
    - average flow duration
    - total number of packets contained in all flows
    - average number of packets per flow
    - total number of unique IP addresses contained in all flows
    - average packet size

Experimental Evaluation

Ground Truth

- average flow duration
- total number of packets contained in all flows
- average number of packets per flow
- total number of unique IP addresses contained in all flows
- average packet size

Experimental Evaluation

} Ground Truth

- total number of packets contained in all flows
- average number of packets per flow
- total number of unique IP addresses contained in all flows
- average packet size

Experimental Evaluation

Ground Truth

- average number of packets per flow
- total number of unique IP addresses contained in all flows
- average packet size

Experimental Evaluation

Ground Truth

- total number of unique IP addresses contained in all flows
- average packet size

Experimental Evaluation

Ground Truth

- average packet size

Experimental Evaluation

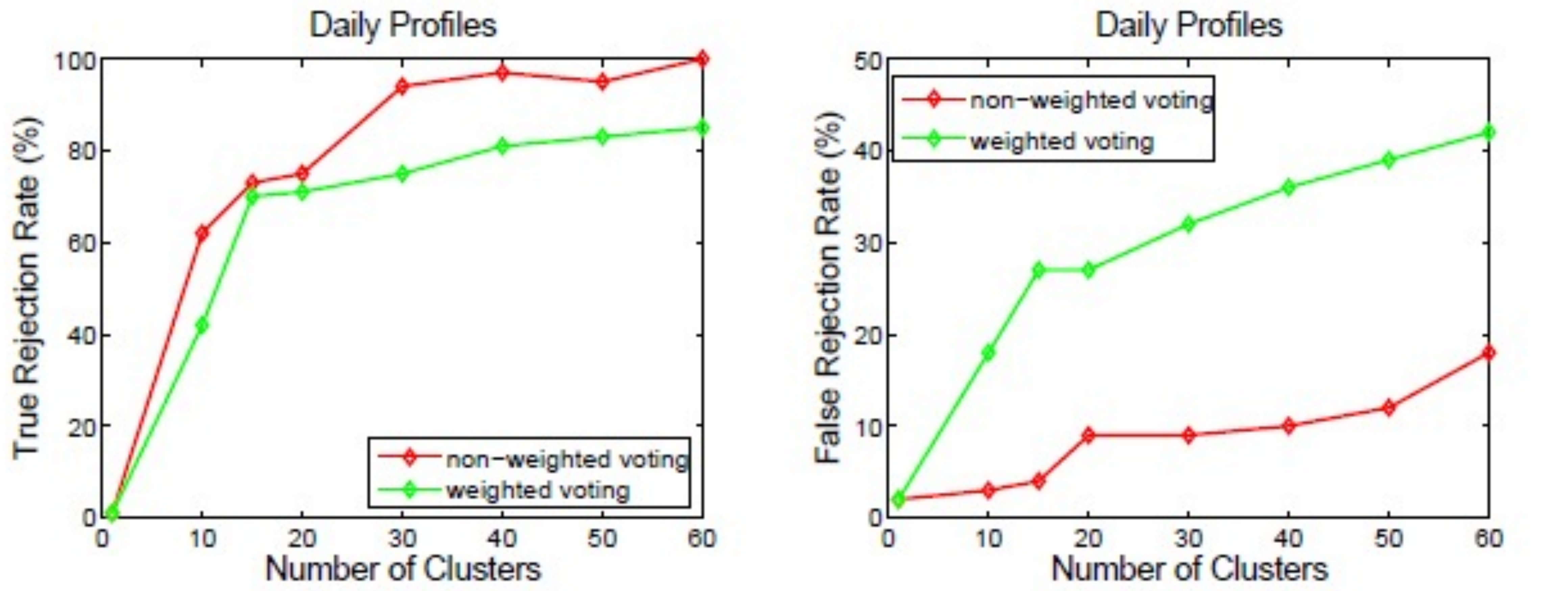
Ground Truth

- Simulated NAC with 300 members, other 300 attempt admission
  - Training Set: 300 profiles week1
  - Cross-validation Set: 75 profiles week2 (FR)
  - Testing Set: 225 profiles week2 (FR)
  - TR: profiles at one, two and three σ away from individual clusters

VALIDATION EXPERIMENTS: PHASES

- Simulated NAC with 300 members, other 300 attempt admission
- Training Set: 300 profiles week1
- Cross-validation Set: 75 profiles week2 (FR)
- Testing Set: 225 profiles week2 (FR)
- TR: create artificial profiles at one, two and three standard deviations away from individual clusters

VALIDATION EXPERIMENTS:
CLUSTERING AND CROSS-VALIDATION



(a) Best TR as a function of k using K-*means++* method.

(b) Best FR as a function of k using K-*means++* method.

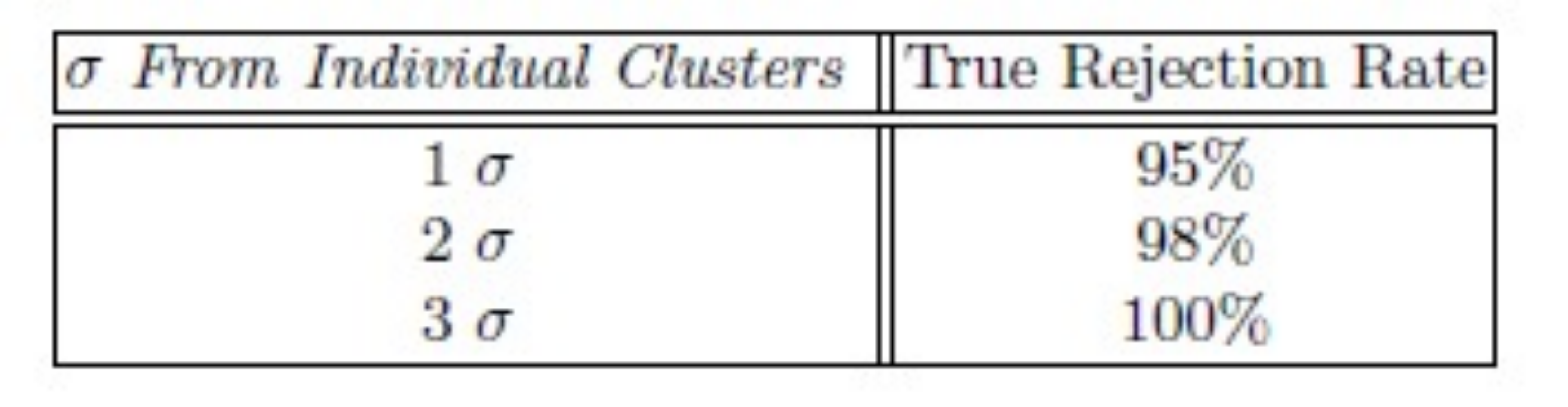
Best Performance: K=40 clusters (TR=95%, FP=10%)

VALIDATION EXPERIMENTS:

ACCESS CONTROL

- @one and two standard deviation some anomalous profiles go undetected due to problems with the profile generation itself

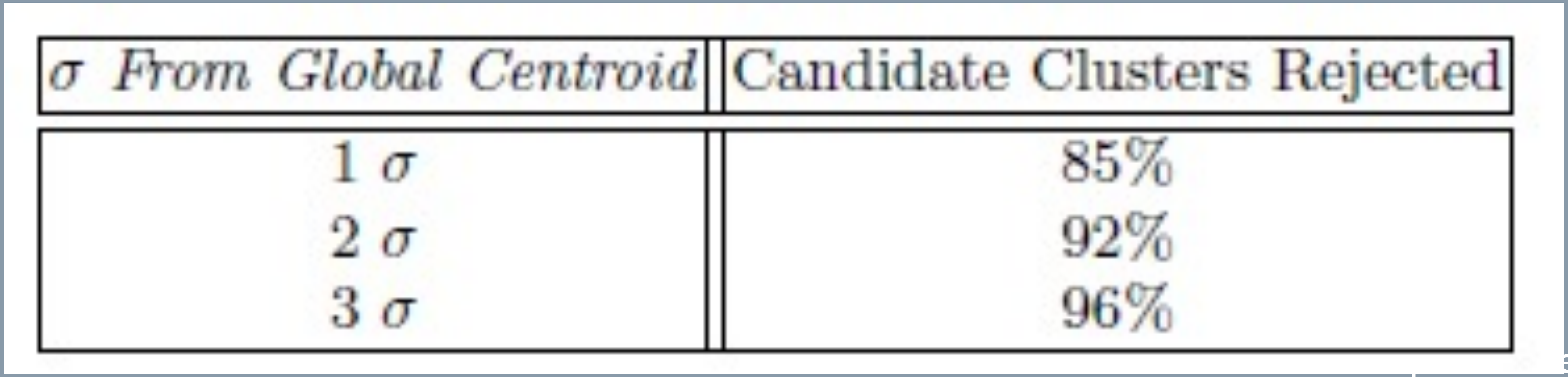
- Future work will evaluate real anomalous profiles and not artificially generated



σ *From Individual Clusters*	True Rejection Rate
$1\ \sigma$	95%
$2\ \sigma$	98%
$3\ \sigma$	100%

VALIDATION EXPERIMENTS:
CONCEPT DRIFT AND COLLUSION ATTACKS

- Create anomalous profiles at one, two and three standard deviations away from the global behavioral centroid
- Idea: create outlier profiles to determine the boundary the algorithm creates between concept drift and attacks
- Important: the creation of new clusters is limited by the algorithm so that the damage an attacker can infer is also limited



σ *From Global Centroid*	Candidate Clusters Rejected
1 σ	85%
2 σ	92%
3 σ	96%

VALIDATION EXPERIMENTS: THRESHOLD ATTACKS

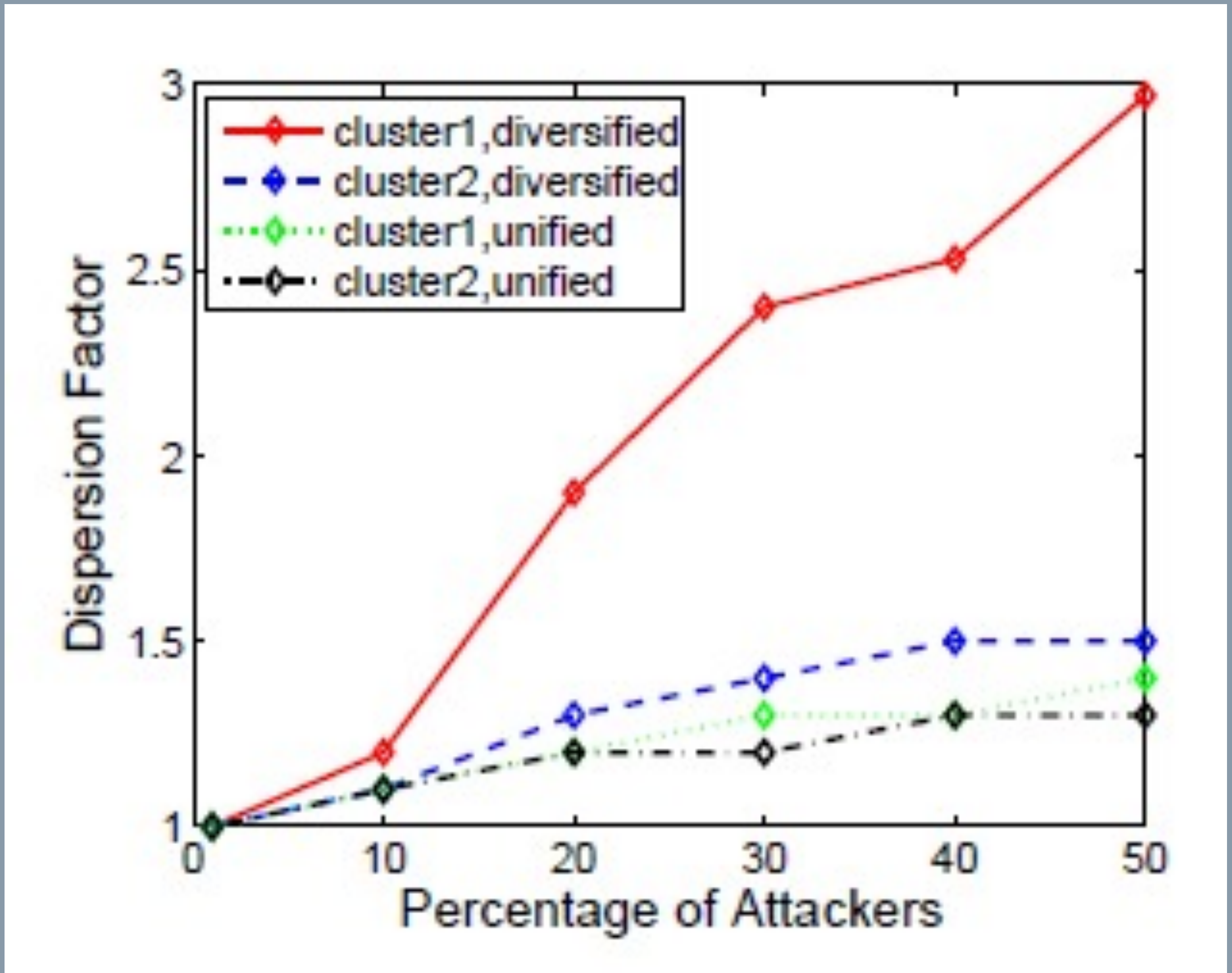
- Selected two clusters from clustering phase: lowest and highest spread (distance across behaviors)

- Attacks: each attacker generates a different profile and iteratively increases feature values by 10%
  - Diversified attack: each attacker generates a different profile
  - Unified attack: all initial profiles are equal

- Dispersion: ratio between initial average threshold and final average threshold after attack

VALIDATION EXPERIMENTS:
THRESHOLD ATTACKS

- All attacks are eventually detected by the mechanism
- Dispersion factor increased at maximum by three times



Summary of Results

- ☑ Mechanism to automatically determine clusters of behavior and other parameters specific to the network
- ☑ Mechanism is robust over time while allowing for creation of new behaviors (concept drift)
- ☑ Mechanism can detect collusion attacks and limits the damage incurred by threshold attacks
- ☑ Extensive evaluation using volumetric profiles (flows) of 300 hosts at Columbia University

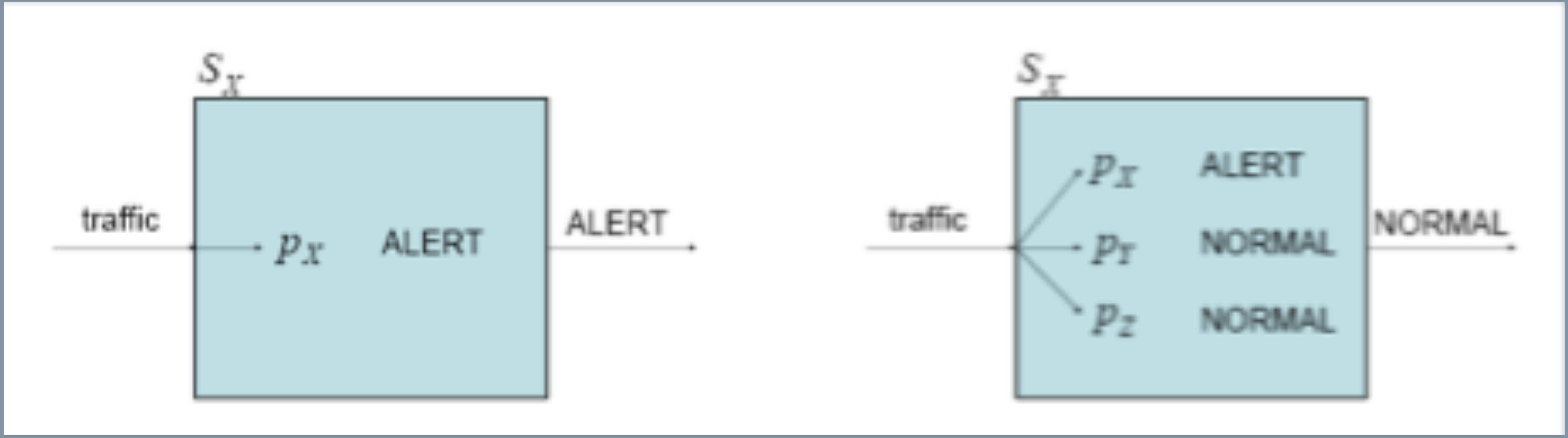
Outline

- Behavior-based NAC technologies
- Automatic Clustering and Policy Update
    - Cluster-based AD sensor
- Behavior-based Policies for MANETs
- Conclusions and Future Work

Sensor Details

Single-Profile AD sensor

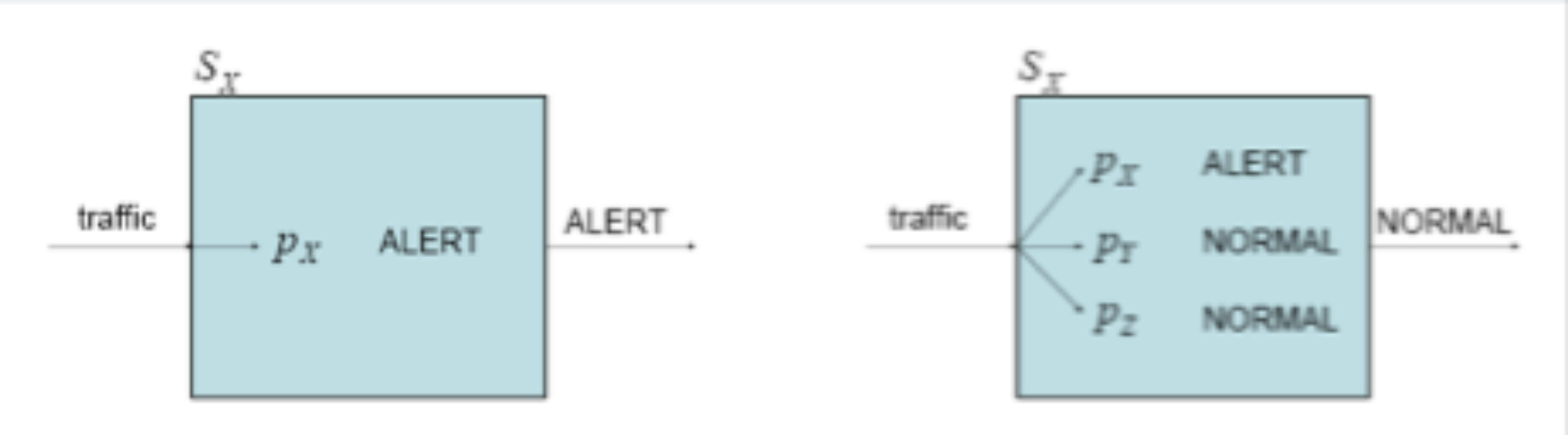
Cluster-based AD sensor



Sensor Details

Single-Profile AD sensor

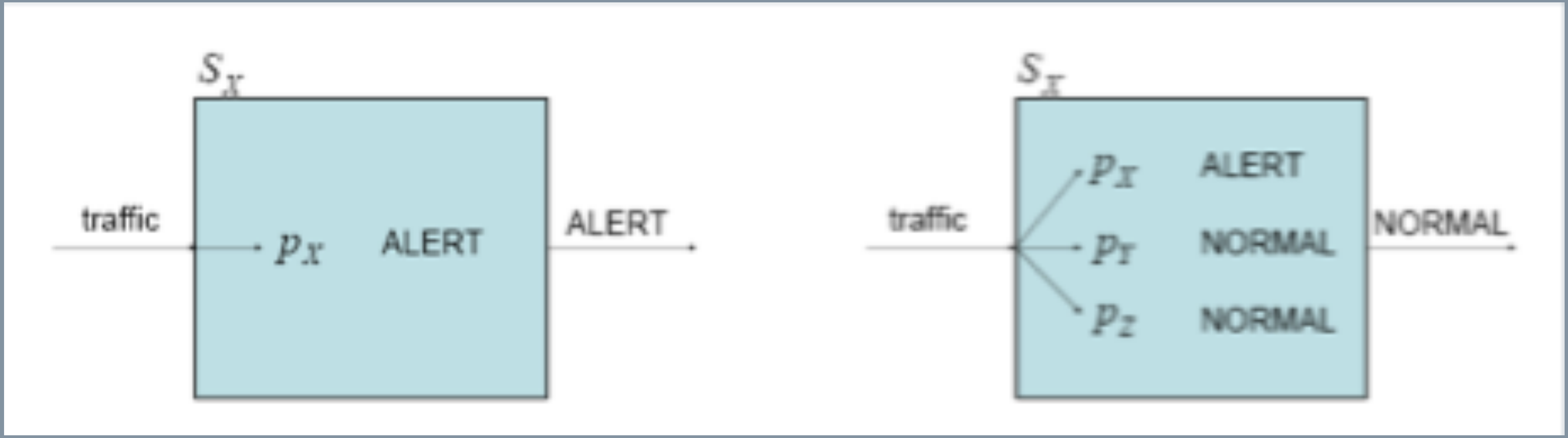
Cluster-based AD sensor



Sensor Details

Single-Profile AD sensor

Cluster-based AD sensor



$$D_{P_x}(t) = \begin{cases} 0, & \text{if } P_x - \sigma \prec t \prec P_x + \sigma \\ 1, & \text{otherwise} \end{cases}$$

$$S_x = \begin{cases} 1, & \text{if } \dfrac{\sum_{i}^{x,y,z} D_{P_i}(t)}{3} \geq \varepsilon \\ 0, & \text{otherwise} \end{cases}$$

Experimental Evaluation

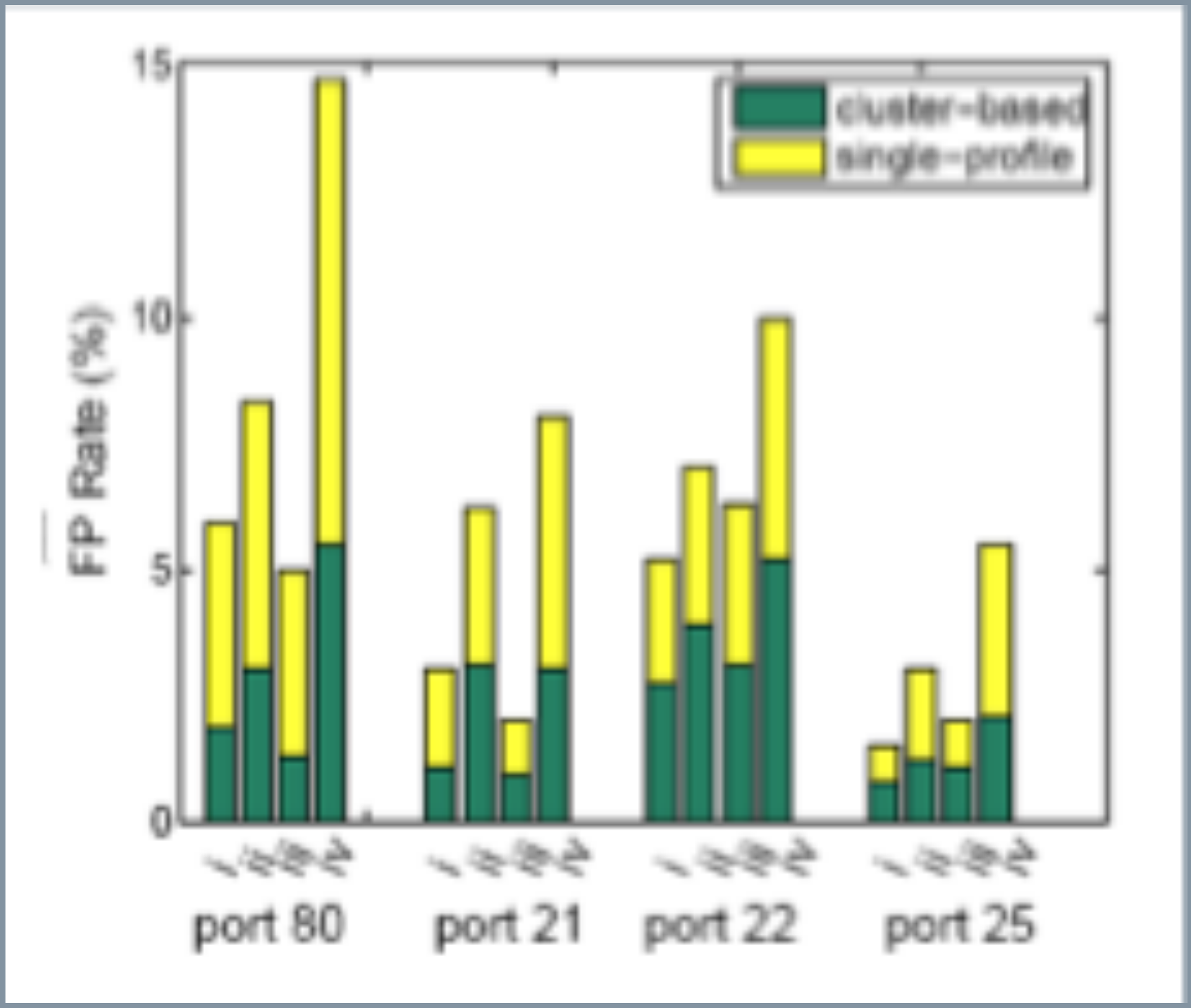
- CRAWDAD repository: 1 month of tcpdump wireless traffic (no content)
- Ports: 21 (FTP), 22 (SSH), 25 (SMTP), 80 (HTTP)
- For each port, we identified 100 different MAC addresses with output traffic to the services
- Behavior profiles for each user/service were computed as daily histograms of usage for the first week of data

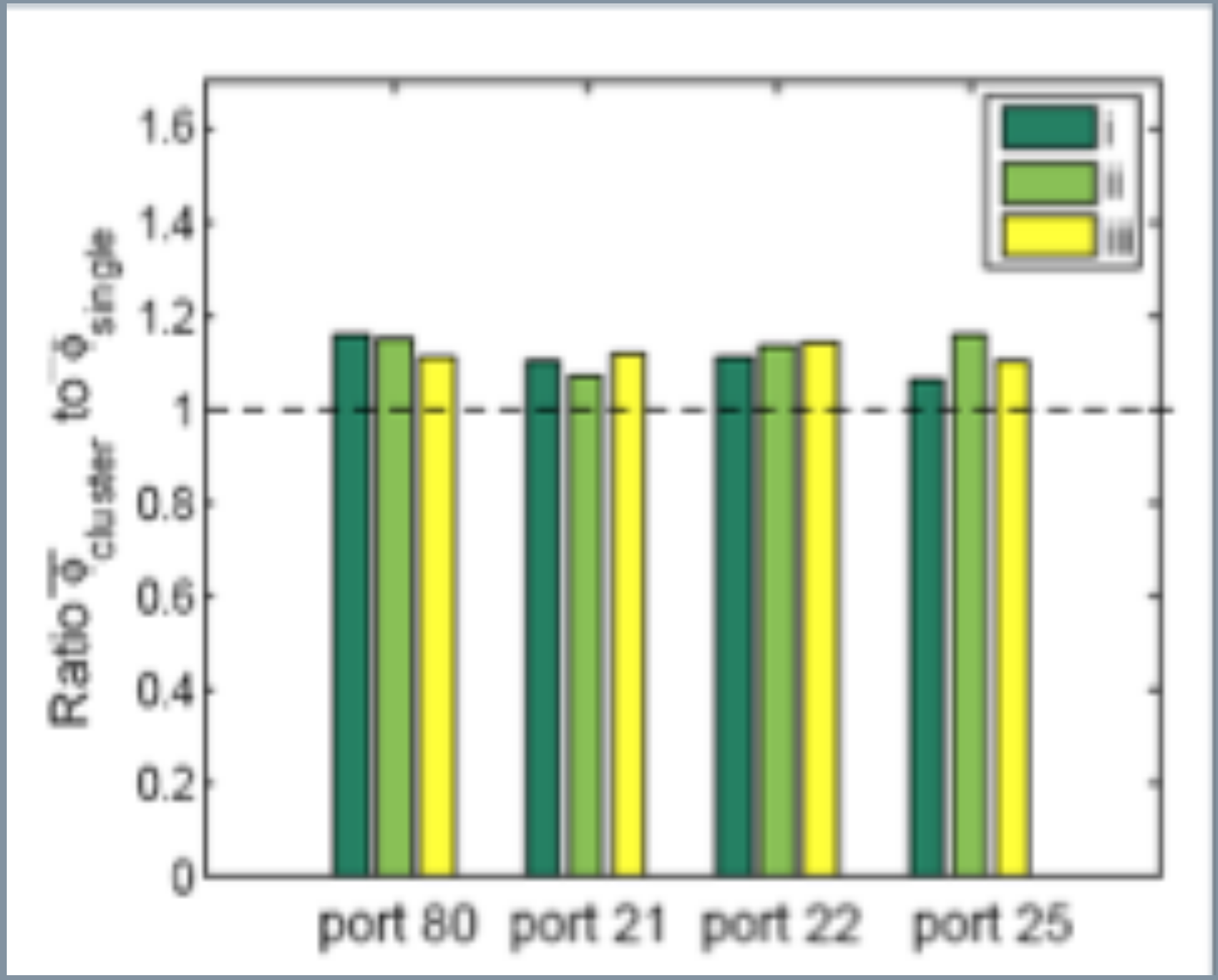
$$P_i = \{h_{f_1},\ldots,h_{f_n}\} \text{ where } h_{f_n} = \{(\alpha_0,\sigma_0),(\alpha_1,\sigma_1),\ldots,(\alpha_{23},\sigma_{23})\}$$

- Parameters modeled:
  - average number of unique users contacted per hour (i)
  - average number of packets exchanged per hour (ii)
  - average length of the packets exchanged per hour (iii)

- Performance measured with average false positive $\overline{FP}$ and average detection threshold $\overline{\phi}$

Experimental Evaluation





Summary of Results

☑ Cluster-based AD sensor provides a broader definition of normal behavior that compensates for poor or insufficient training of individual sensors and reduces the volume of false alerts

Outline

- Behavior-based NAC technologies
- Automatic Clustering and Policy Update
- Cluster-based AD sensor

► Behavior-based Policies for MANETs

- Conclusions and Future Work

Behavior-based Policies for MANETs

Behavior-based Policies for MANETs

- Does the BB-NAC mechanism work for MANETs?

Behavior-based Policies for MANETs

- Does the BB-NAC mechanism work for MANETs?
  - ▶ Automatically derive policies from behavior profiles

Behavior-based Policies for MANETs

- Does the BB-NAC mechanism work for MANETs?
  - ▶ Automatically derive policies from behavior profiles
  - ▶ Fully distributed approach to profile computation and alert generation/analysis

Behavior-based Policies for MANETs

- Does the BB-NAC mechanism work for MANETs?
  - ▶ Automatically derive policies from behavior profiles
  - ▶ Fully distributed approach to profile computation and alert generation/analysis

- BARTER: adaptation of BB-NAC to fully distributed environments

Behavior-based Policies for MANETs

- Does the BB-NAC mechanism work for MANETs?
  - ▶ Automatically derive policies from behavior profiles
  - ▶ Fully distributed approach to profile computation and alert generation/analysis

- BARTER: adaptation of BB-NAC to fully distributed environments
  - ▶ Threshold cryptographic layer (t,n) for fully distributed management

Behavior-based Policies for MANETs

- Does the BB-NAC mechanism work for MANETs?
  - ▶ Automatically derive policies from behavior profiles
  - ▶ Fully distributed approach to profile computation and alert generation/analysis

- BARTER: adaptation of BB-NAC to fully distributed environments
  - ▶ Threshold cryptographic layer (t,n) for fully distributed management
  - ▶ Light-weight version. Instead of clusters of behavior, MANET devices derive thresholds from their top t-1 most similar profiles

Behavior-based Policies for MANETs

- Does the BB-NAC mechanism work for MANETs?
  - ▶ Automatically derive policies from behavior profiles
  - ▶ Fully distributed approach to profile computation and alert generation/analysis

- BARTER: adaptation of BB-NAC to fully distributed environments
  - ▶ Threshold cryptographic layer (t,n) for fully distributed management
  - ▶ Light-weight version. Instead of clusters of behavior, MANET devices derive thresholds from their top t-1 most similar profiles
  - ▶ Members only exchange output behavior profiles to avoid

Behavior-based Policies for MANETs

- Does the BB-NAC mechanism work for MANETs?
  - ▶ Automatically derive policies from behavior profiles
  - ▶ Fully distributed approach to profile computation and alert generation/analysis

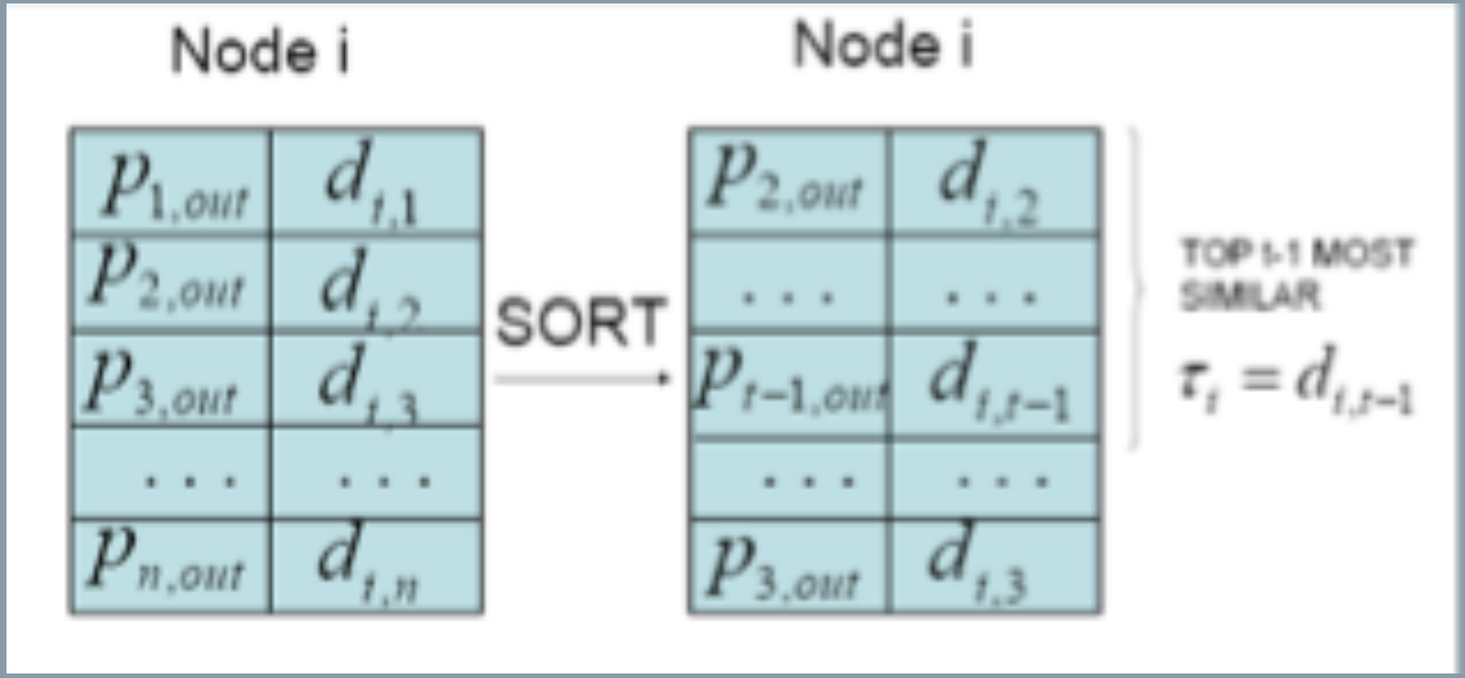
- BARTER: adaptation of BB-NAC to fully distributed environments
  - ▶ Threshold cryptographic layer (t,n) for fully distributed management
  - ▶ Light-weight version. Instead of clusters of behavior, MANET devices derive thresholds from their top t-1 most similar profiles
  - ▶ Members only exchange output behavior profiles to avoid crafted attacks adapted to input profiles

BARTER SETUP

- Devices initially exchange their output behavior profiles and compute their local thresholds:

$$t_{P_i} = \max_{j=0..n} d(P_{i,in}, P_{j,out}) \text{ where } P_j \in \text{top t-1 most similar}$$

- Each member computes its own threshold as,



Node i

$p_{1,out}$	$d_{i,1}$
$p_{2,out}$	$d_{i,2}$
$p_{3,out}$	$d_{i,3}$
. . .	. . .
$p_{n,out}$	$d_{i,n}$

SORT

Node i

$p_{2,out}$	$d_{i,2}$
. . .	. . .
$p_{t-1,out}$	$d_{i,t-1}$
. . .	. . .
$p_{3,out}$	$d_{i,3}$

TOP t-1 MOST SIMILAR

$\tau_i = d_{i,t-1}$

- Cross-Validation to determine best t/n for the network

BARTER CRYPTO SETUP

- Shamir Secret Sharing to divide the secret into n shares
- Each member computes its partial shared key such that any *t* members will be able to recover the secret and generate new keys for new members

BARTER Admission Control

- Individual admission control decision:

$$v_i = 1 \text{ if } d(P_{i,in}, P_{new,out}) \leq \tau_i$$

- Group decision: if *t* MANET members agree, newcomer accepted
- Newcomer receives output profiles from all other members and creates its local table
- To avoid recalculating t/n upon every new accepted member:
  - t is updated only when the ratio exceeds the range (t/n+w, t/n-w)

BARTER Crypto Admission

- Newcomer broadcasts its public key certificate and its behavior profile
- Each individual member that accepts the newcomer as normal, generates a partial signature.
- Newcomer computes signature by summing *t* partial signatures received upon acceptance.

BARTER ACCESS CONTROL

- Each member AD is continuously screening traffic
- If an anomaly is detected, it is shared with its *top t-1* most similar members
- If the *t-1* members agree on the anomaly, the device is expelled
- Assumption: existence of scheme to avoid data tampering and prevent devices from falsifying alerts or replay attacks

BARTER Crypto Access

- Proactive Key Generation is used to eliminate a device from accessing further communications
- The “bad” device is added to a Certificate Revocation List (CRL) which is broadcasted to all members
- Devices that receive t CRLs renew keys through point-to-point encrypted communication channels with members not in the CRL

BARTER ATTACKS

- DDoS costs: robustness against DDoS attacks
- Normalized value (0..1)

$$DDoS\begin{cases} 0.5, \text{ if } \dfrac{t}{n} = 0.5 \\ \dfrac{t}{n}, \text{ if } \dfrac{t}{n} \prec 0.5 \\ 1 - \dfrac{t}{n}, \text{ if } \dfrac{t}{n} \succ 0.5 \end{cases}$$

BARTER CRYPTOGRAPHIC COSTS

- Cryptographic costs incurred in key (re)generation upon admission and access control (when t is (in/de)cremented)

$$CC = K \times \sum_{n_0}^{n_{final}} (update \times n) - 1$$

$$update = \begin{cases} 1, & \text{if } \dfrac{t}{n} \prec \left(\dfrac{t_0}{n_0} - w \right) \quad \text{or} \quad \text{if } \dfrac{t}{n} \succ \left(\dfrac{t_0}{n_0} + w \right) \\ 0, & \text{otherwise} \end{cases}$$

Experimental Evaluation Content Profiles

- 140 users from ENRON dataset, email-like MANET application
- Computed input and output profiles using Shanner’s Algorithm

$$W(i) = \log \frac{x_i}{N_g} \times \frac{1}{\log N_g} \sum_{j=1}^{N_g} p_{ij} \log \frac{1}{p_{ij}} \times \left(1 - \frac{1}{\log L}\right) \sum_{j}^{goodS,badS} p_{ij} \log \frac{1}{p_{ij}}$$

- ► 3-grams, top 5000

- Good samples: ENRON emails
- Bad samples: Signature content of Snort rules (58) and 600 virus samples from vxheavens

EXPERIMENTAL EVALUATION
CONTENT PROFILES

- 140 users from ENRON dataset, email-like MANET application
- Computed input and output profiles using Shanner’s Algorithm

$$W(i) = \boxed{\log \frac{x_i}{N_g}} \times \frac{1}{\log N_g} \sum_{j=1}^{N_g} p_{ij} \log \frac{1}{p_{ij}} \times \left(1 - \frac{1}{\log L}\right) \sum_{j}^{goodS,badS} p_{ij} \log \frac{1}{p_{ij}}$$

- ► 3-grams, top 5000

- Good samples: ENRON emails
- Bad samples: Signature content of Snort rules (58) and 600 virus samples from vxheavens

EXPERIMENTAL EVALUATION CONTENT PROFILES

- 140 users from ENRON dataset, email-like MANET application
- Computed input and output profiles using Shanner's Algorithm

$$W(i) = \boxed{\log \frac{x_i}{N_g}} \times \boxed{\frac{1}{\log N_g} \sum_{j=1}^{N_g} p_{ij} \log \frac{1}{p_{ij}}} \times \left(1 - \frac{1}{\log L}\right) \sum_{j}^{goodS,badS} p_{ij} \log \frac{1}{p_{ij}}$$

- ► 3-grams, top 5000

- Good samples: ENRON emails
- Bad samples: Signature content of Snort rules (58) and 600 virus samples from vxheavens

Experimental Evaluation Content Profiles

- 140 users from ENRON dataset, email-like MANET application
- Computed input and output profiles using Shanner's Algorithm

$$W(i)=\boxed{\log\frac{x_i}{N_g}}\times\boxed{\frac{1}{\log N_g}\sum_{j=1}^{N_g} p_{ij}\log\frac{1}{p_{ij}}}\times\boxed{\left(1-\frac{1}{\log L}\right)\sum_{j}^{goodS,badS} p_{ij}\log\frac{1}{p_{ij}}}$$

  - ▶ 3-grams, top 5000

- Good samples: ENRON emails
- Bad samples: Signature content of Snort rules (58) and 600 virus samples from vxheavens

Experimental Evaluation
Content Profiles

- Behavior Profiles are saved as Bloom Filters
- BFs are one-way structures that preserve the privacy of the output content of the users
- BARTER saves its behavior profiles (input and output) as BFs
- Distance between profiles is computed as XOR that quantifies the amount of entries that differ across profiles:

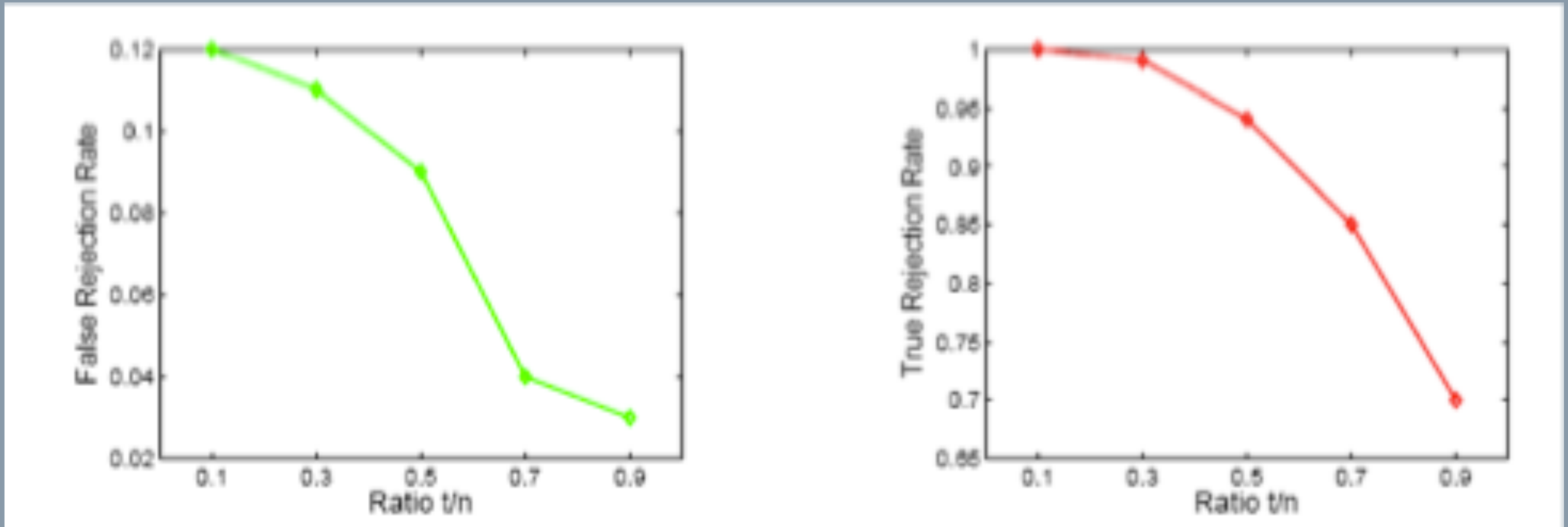
$$d(P_i, P_j) = |P_i \oplus P_j|$$

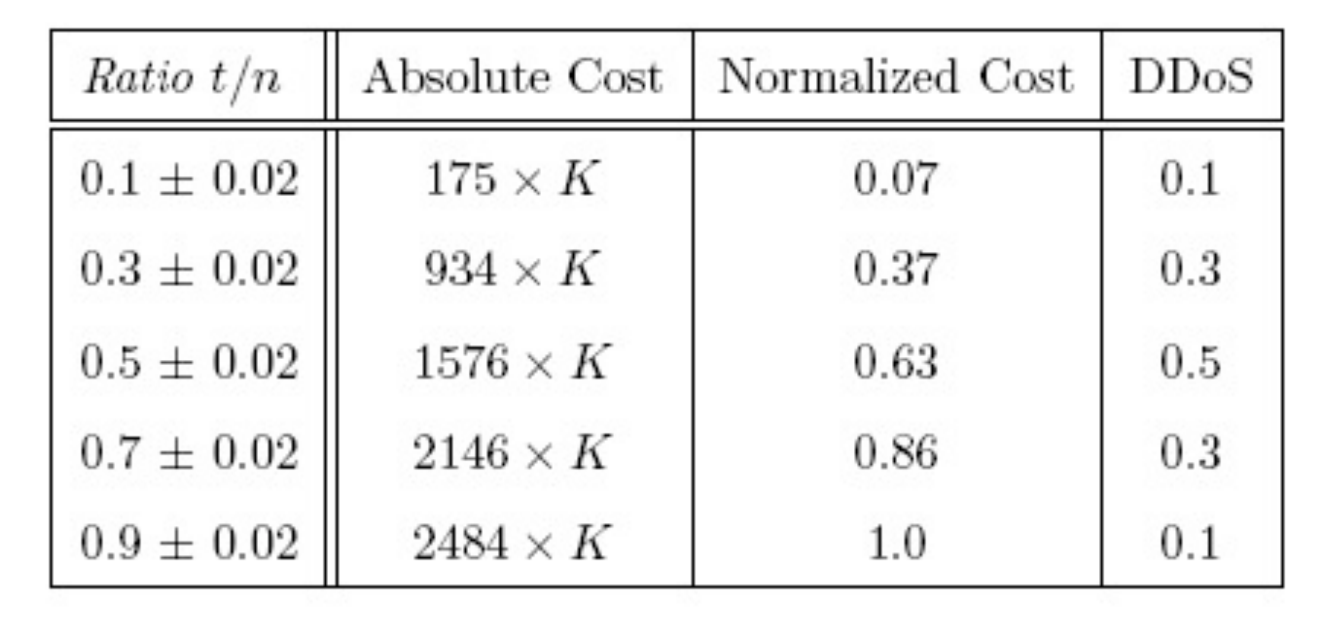
Experimental Evaluation Content Profiles

- Training Set: 80 ENRON behavior profiles → Ground truth
- Cross-validation Set:
  - 30 ENRON behavior profiles (ground truth)
  - 30 code/executables-based behavior profiles
- Testing Set:
  - 30 behavior ENRON profiles (ground truth)
  - 30 code/executables-based behavior profiles
- Admission Control measured in terms of TR and FR rates
- Cross-validation
  - For each t/n we compute r = (1-FR) + TR + (1-CC) + DDoS
  - The highest ranked t/n is selected for admission and access control
  - Experiments run multiple times and results are averaged

Experimental Evaluation Content Profiles

r = (1- FR) + TR + (1 – CC) + DDoS

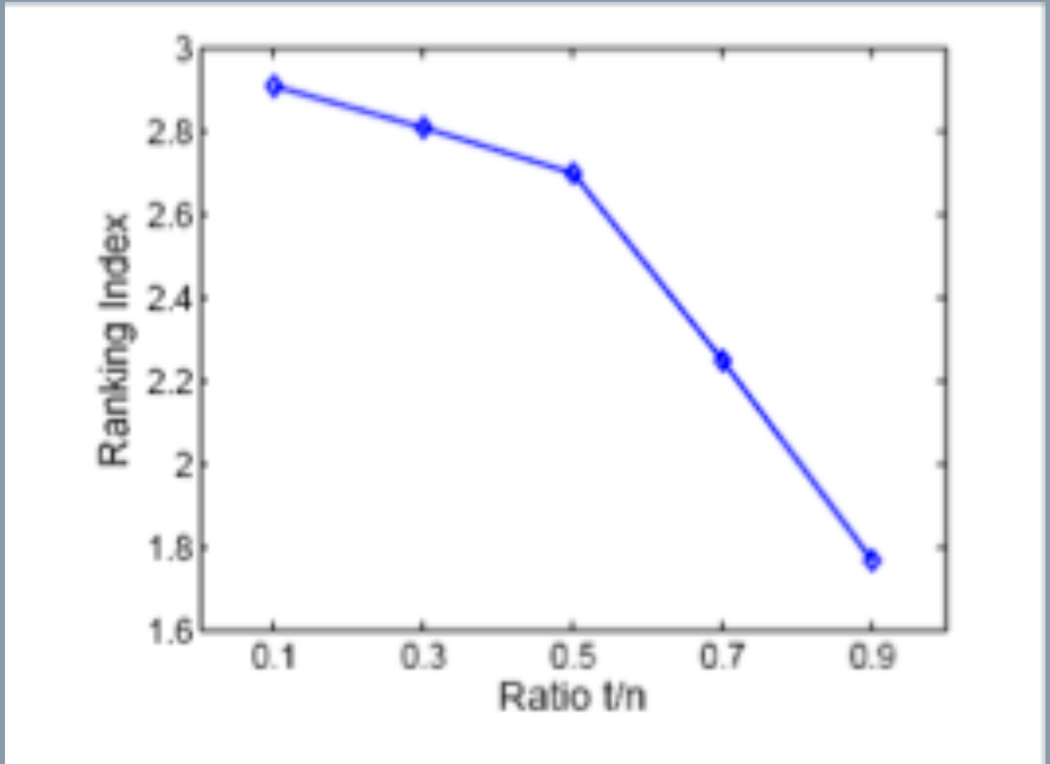




Ratio t/n	Absolute Cost	Normalized Cost	DDoS
0.1 ± 0.02	$175 \times K$	0.07	0.1
0.3 ± 0.02	$934 \times K$	0.37	0.3
0.5 ± 0.02	$1576 \times K$	0.63	0.5
0.7 ± 0.02	$2146 \times K$	0.86	0.3
0.9 ± 0.02	$2484 \times K$	1.0	0.1

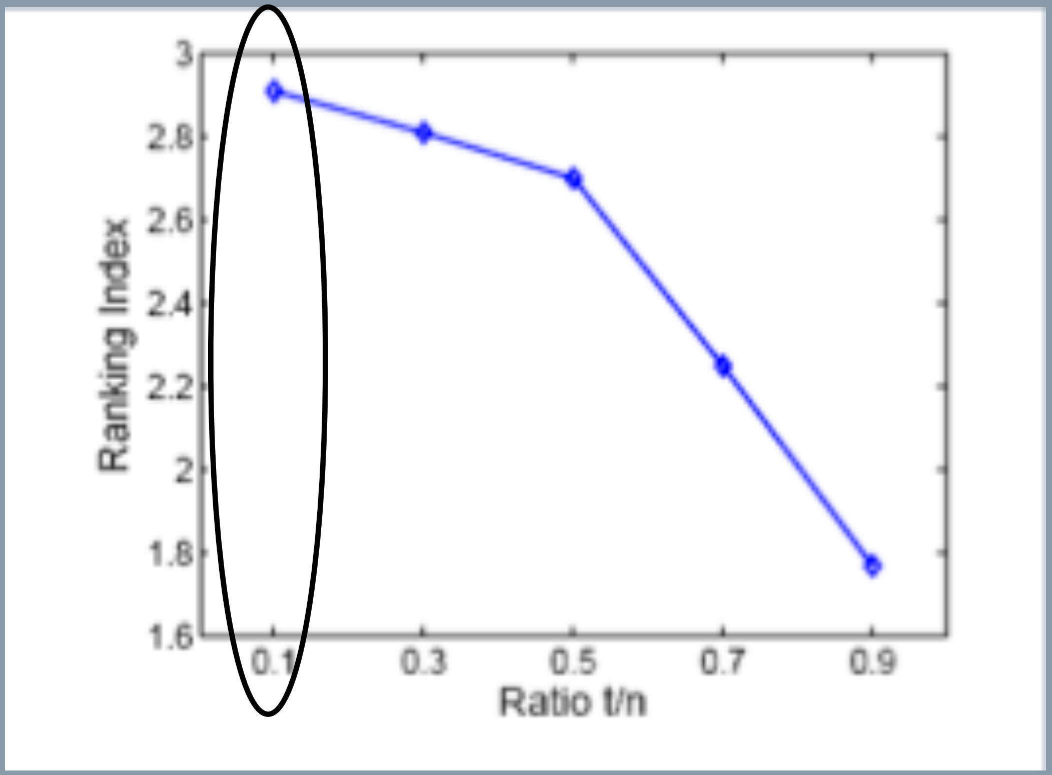
EXPERIMENTAL EVALUATION CONTENT PROFILES

$r = (1 - FR) + TR + (1 – CC) + DDoS$



Experimental Evaluation Content Profiles

r = (1- FR) + TR + (1 – CC) + DDoS



Best ratio t/n=10%

▶ Admission Control Results:

FR= 0.13

TR= 1

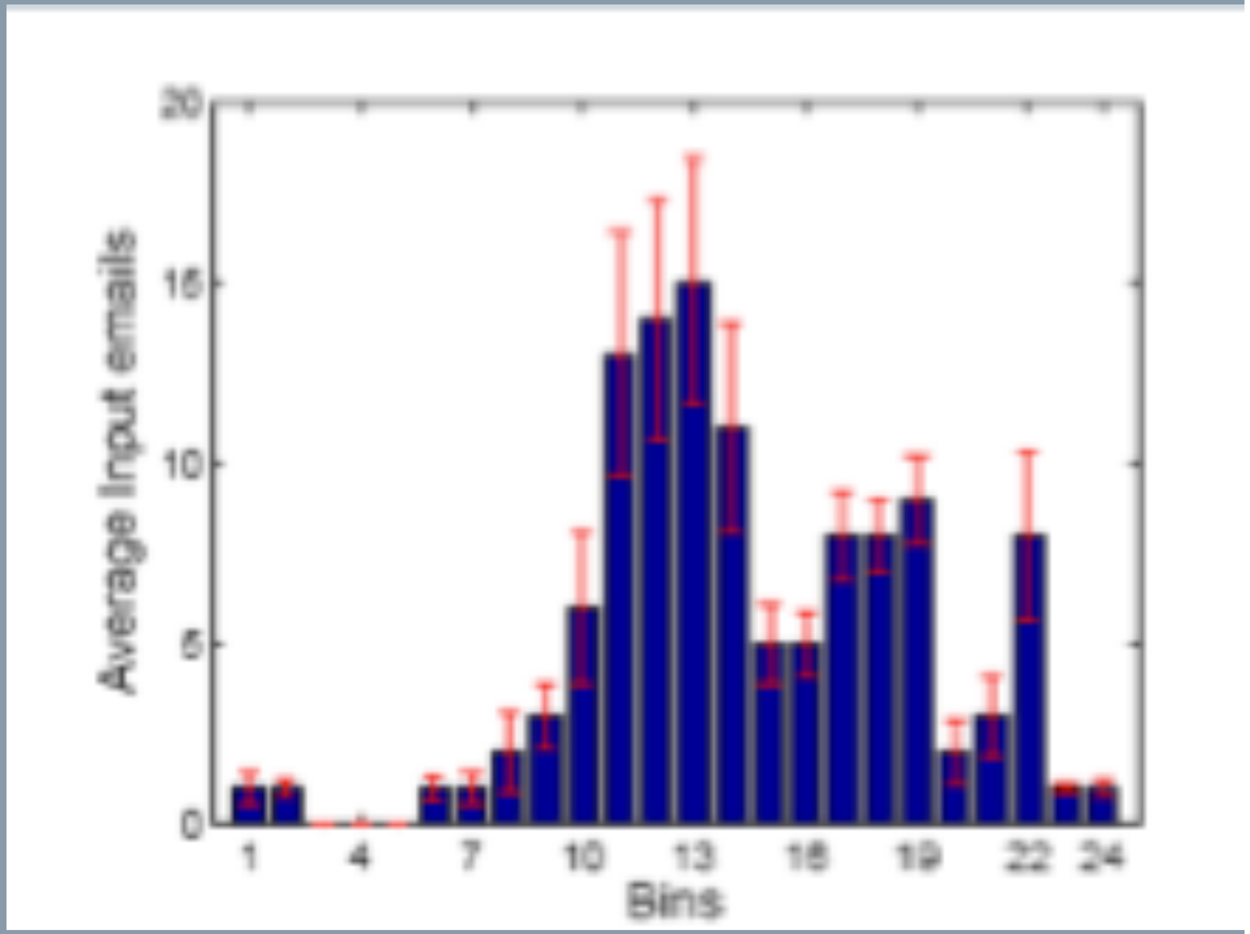
Experimental Evaluation Volumetric Profiles

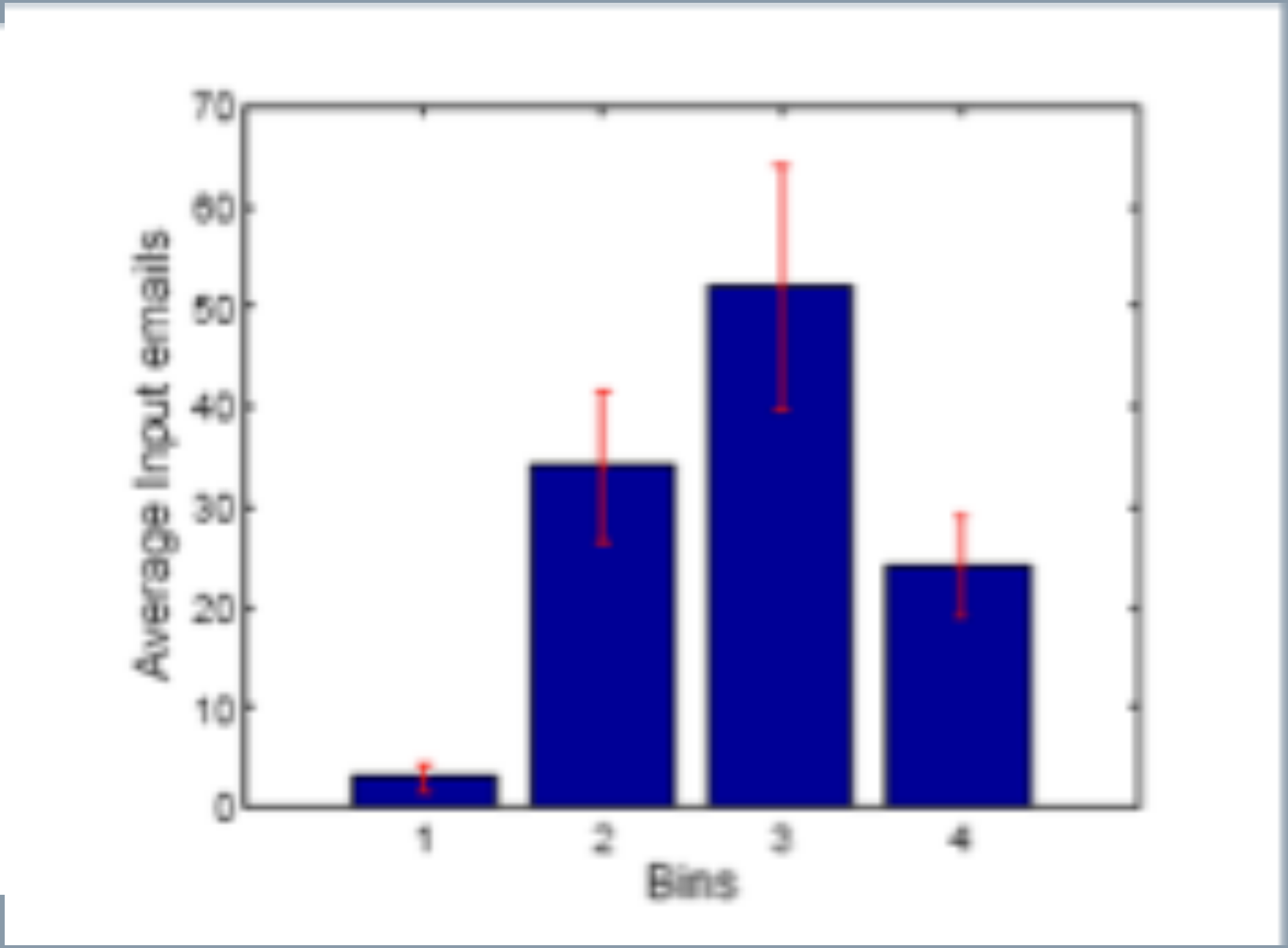
- 140 users from ENRON dataset
- Computed input and output profiles as daily histograms on the
- number of emails exchanged (EMT tool),

$$P_{i,d} = \{b_1, \ldots, b_{bg}\} \text{ where } b_j = \{(\alpha, \sigma)\}$$

- Two types of volumetric profiles:
  - Hourly Histrograms: bg=24
  - Grouped Histograms: bg=4
  - Profile distance computed using Euclidean distance
  - Grouped Histograms save more bandwidth during its exchange
  - Best profiling technique is selected during cross-validation, together with best t/n

EXAMPLES OF A VOLUMETRIC PROFILE





Hourly Profile bg=24

Grouped Profile bg=4

EXPERIMENTAL EVALUATION VOLUMETRIC PROFILES

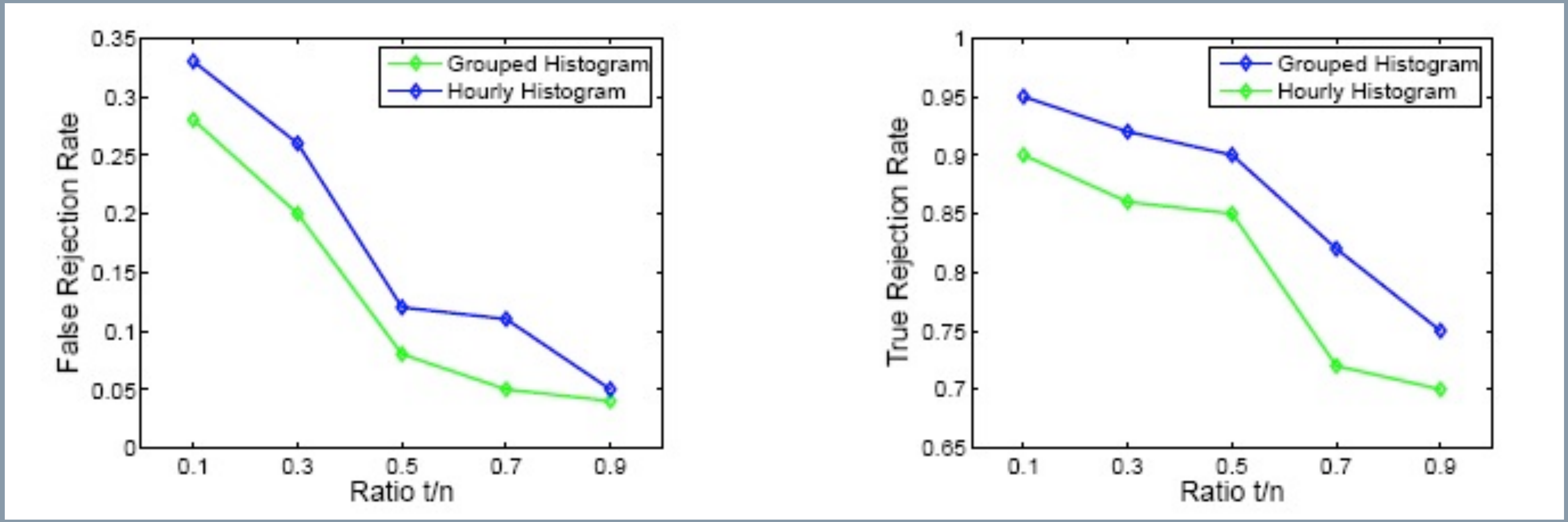
- Simulated MANET running an email-like application
- Admission Control quality measured in terms of TR and FR

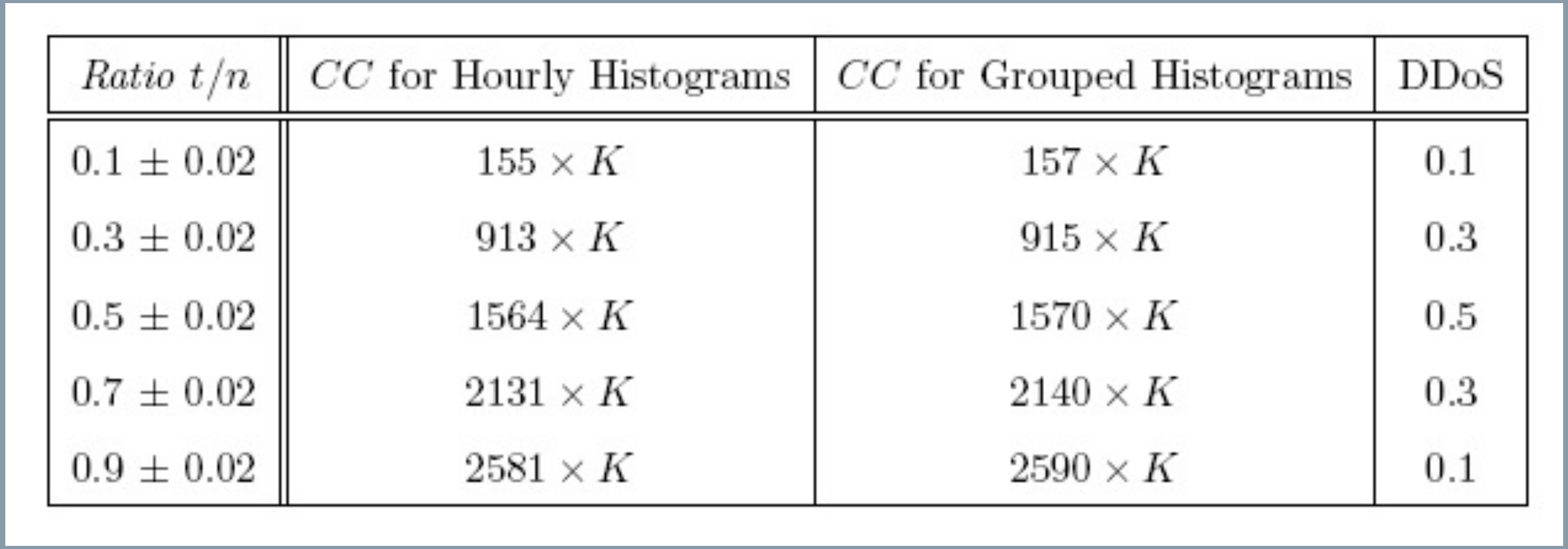
- Good Samples: ENRON volumetric profiles
- Bad Samples: Synthetically computed as profiles that are one, two and three standard deviations away from the top t-1 entries in the local table of each MANET member.

- Training Set (80), Cross-validation Set (30) and Testing Set (30)
- Cross-validation determines ratio t/n, type of profile

Experimental Evaluation

r = (1- FR) + TR + (1 – CC) + DDoS

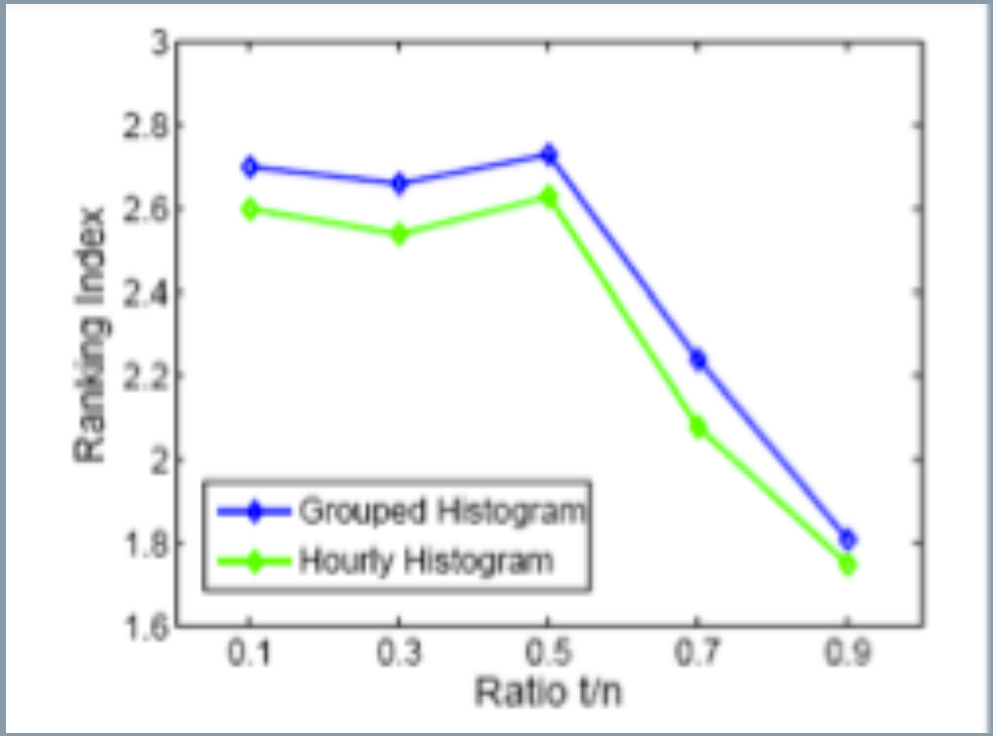




Ratio t/n	*CC* for Hourly Histograms	*CC* for Grouped Histograms	DDoS
0.1 ± 0.02	$155 \times K$	$157 \times K$	0.1
0.3 ± 0.02	$913 \times K$	$915 \times K$	0.3
0.5 ± 0.02	$1564 \times K$	$1570 \times K$	0.5
0.7 ± 0.02	$2131 \times K$	$2140 \times K$	0.3
0.9 ± 0.02	$2581 \times K$	$2590 \times K$	0.1

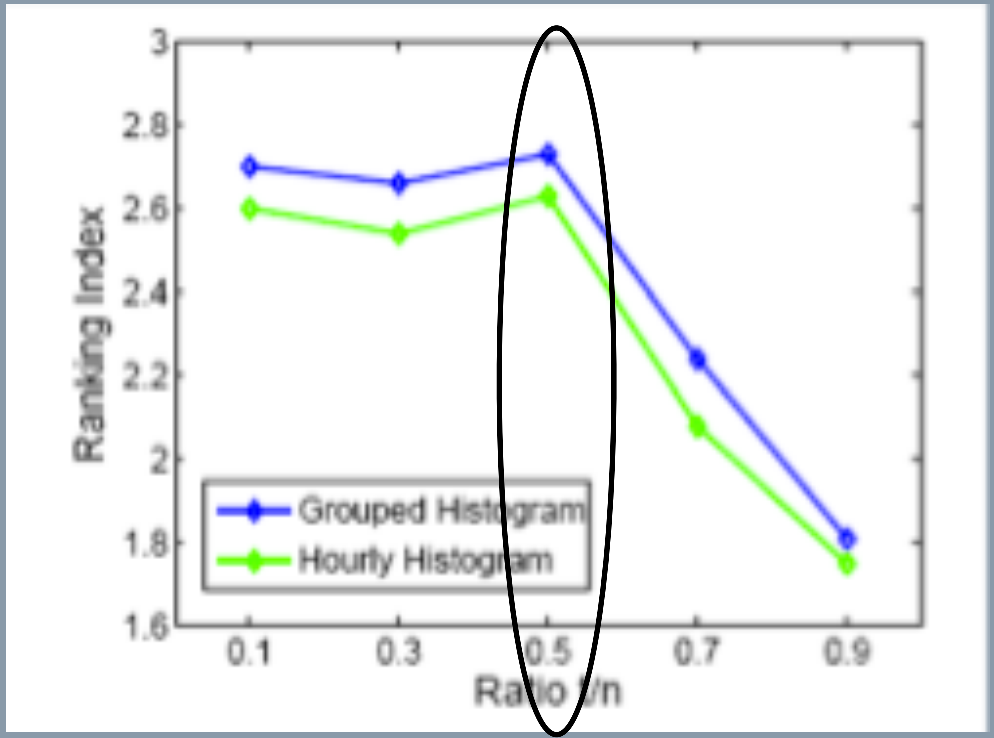
Experimental Evaluation

$$r = (1 - FR) + TR + (1 - CC) + DDoS$$



Experimental Evaluation

r = (1- FR) + TR + (1 – CC) + DDoS



Best ratio t/n=50%, and grouped histogram

▶ Admission Control Results:

FR= 0.08

TR= 0.9

Summary of Results

☑ Adaptation of BB-NAC to MANETs

☑ Integration of the mechanism with a threshold cryptographic layer to achieve fully distributed decisions

☑ Evaluation of BARTER with content and volumetric profiles from ENRON dataset

Outline

- Behavior-based NAC technologies
- Automatic Clustering and Policy Update
- Cluster-based AD sensor
- Behavior-based Policies for MANETs

► Conclusions and Future Work

Conclusions / Contributions

- ☑ Novel mechanism to automatically create and update behavior-based policies for NAC technologies and MANETs
- ☑ Incremental-Learning Algorithm that makes the mechanism robust against attacks from network members
- ☑ Cluster-based AD sensor that reduces volume of false alerts by sharing the detection with similar behavior profiles
- ☑ Extensive experimental evaluation of BB-NAC and BARTER with different types of behavior profiles and datasets