Automated Behavioral Fingerprinting

Jérôme François, Humberto Abdelnur, Radu State and Olivier Festor

Outline

Outline

Device Fingerprinting

Definition:

- a given protocol
- software = protocol stack version
- hardware = vendor name, series, version

Applications:

- administration tool
- security assessment tool: to detect intruders or to locate automatically some devices for updating
- keep a complete and correct database of devices is unfeasible: user choices, various equipment...

Motivation

Current approaches:

- many active techniques
- based on the content of the messages: field values, general signature
- bad randomness of random value generators
- methods based on the protocol syntax

$\longrightarrow$ Our new technique $=$
passive $+$ no protocol knowledge $+$ devices behavior

Contributions

Device behavior ⇔ Interactions with other devices ⇔ Sequences of exchanged messages

- message identification:
  - = message type
  - no grammar
  - → protocol reverse engineering
- key idea
  - same protocol but different state machines
  - rich protocol syntax + freedom

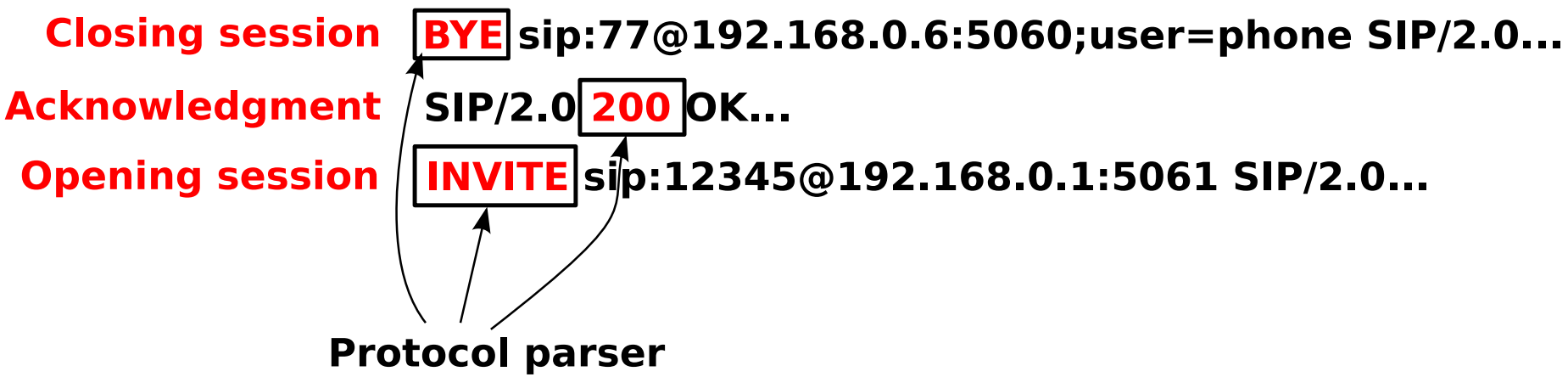
Two contributions:

- message type discovery
- behavioral fingerprinting

Outline

Problem definition

Objective: find the types of the messages = the semantic of the messages

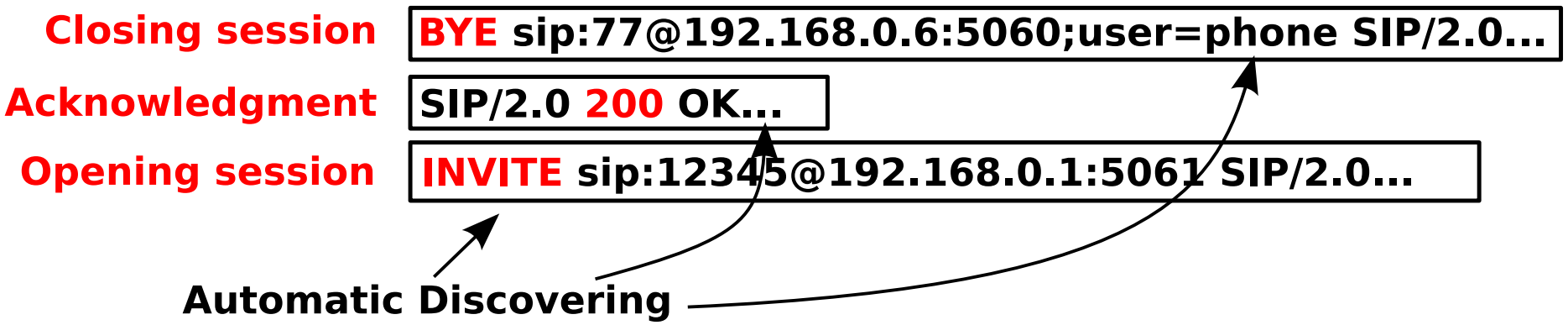


No protocol knowledge:

- no syntax
- no specific delimiters

Problem definition

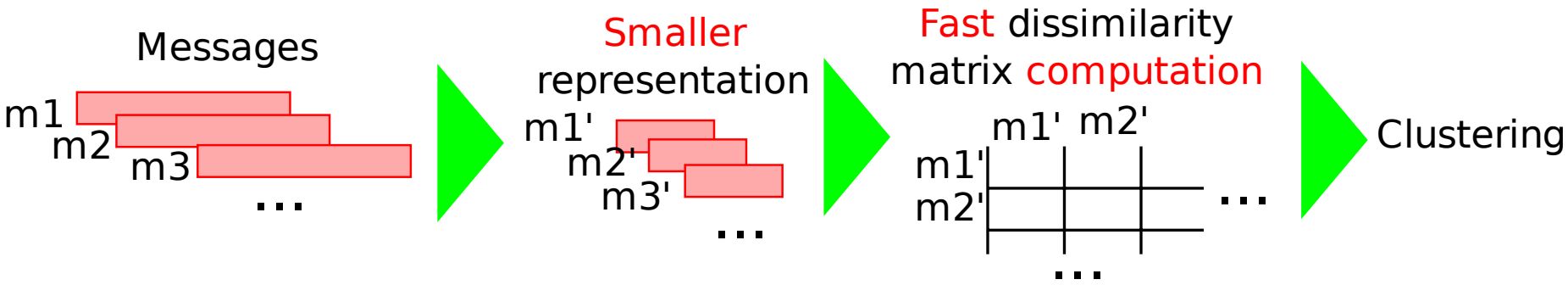
Objective: find the types of the messages = the semantic of the messages



No protocol knowledge:

- no syntax
- no specific delimiters

Framework



Advantages:

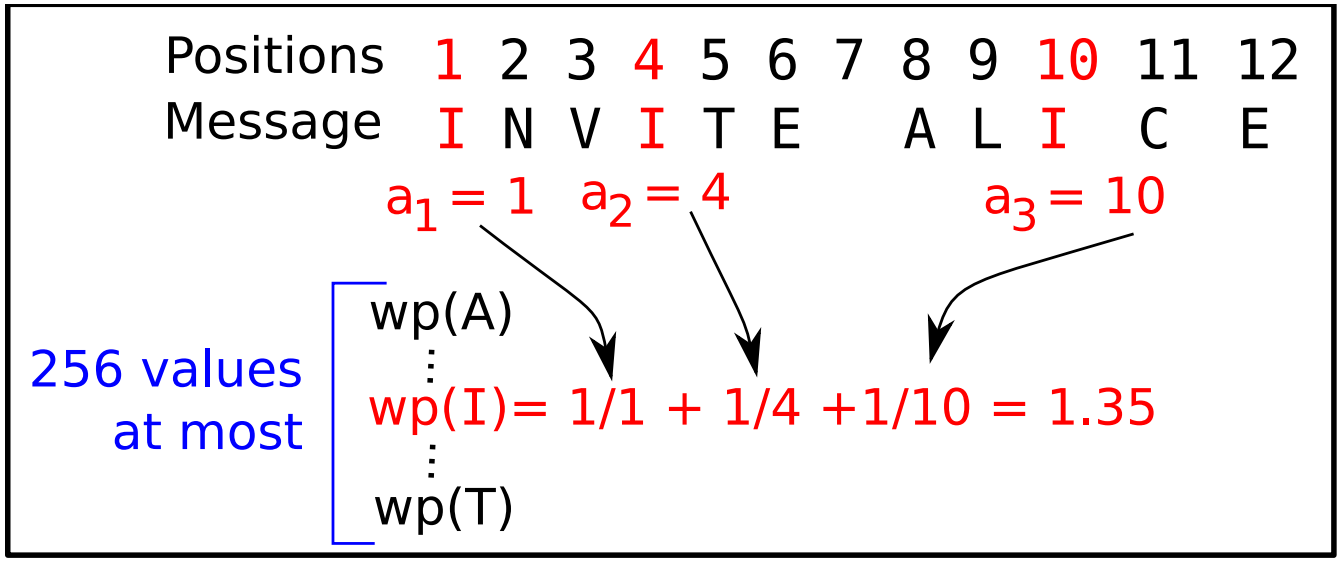
- no protocol knowledge
- no tainting analysis
- unsupervised (no training samples)

Close and complementary to work based on sequence alignement (high complexity)

Messages representations

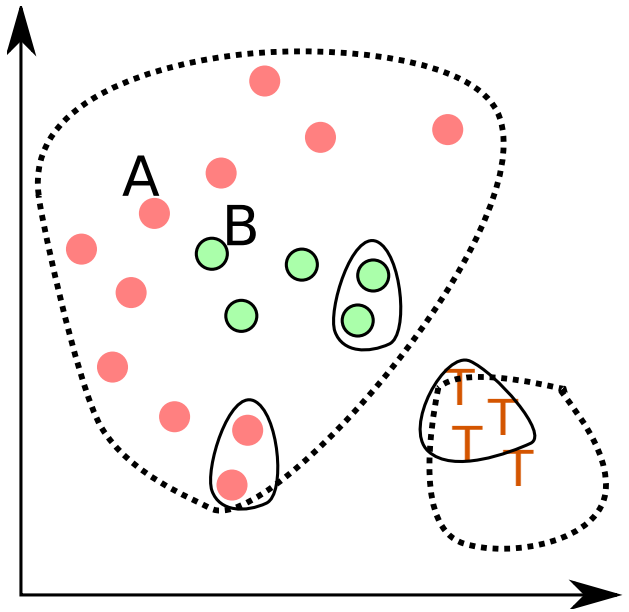
- length of messages, distribution of characters...
- protocol design $\longrightarrow$ type at the beginning of the message $\longrightarrow$ weighted character positions:

$$\forall\ character\ c,\ \ wp(c) = \frac{\sum_{i=1}^{k} pos(a_i)^{-1}}{k}$$



Nearest neighbors

- well known aggregative technique
- initialization: each message = one cluster
- algorithm: merge the closest clusters while the corresponding distance is higher than τ
- advantages: simple and a single parameter: τ
- drawback: unable to discover intertwined shapes

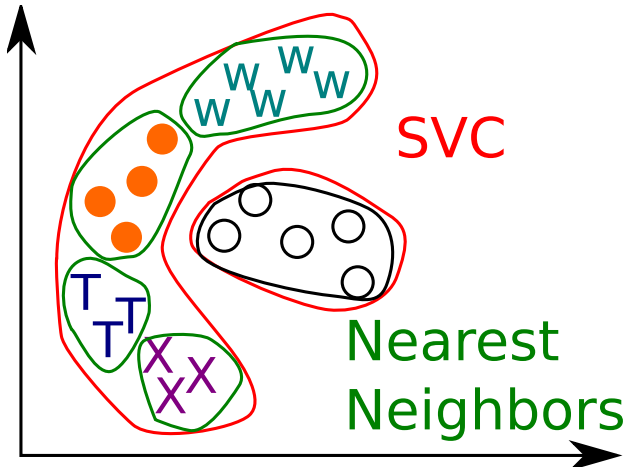


SVC

SVC = Support Vector Clustering[1]

- recent technique
- support vector machine paradigms → good performances + limited overhead in many domains
- advantage: irregular shapes discovered

- Experiences → global method with two passes: SVC + Nearest neighbors



[1] A. Ben-Hur, D. Horn, H.T. Siegelmann and V. Vapnik, A support vector clustering method, 2000

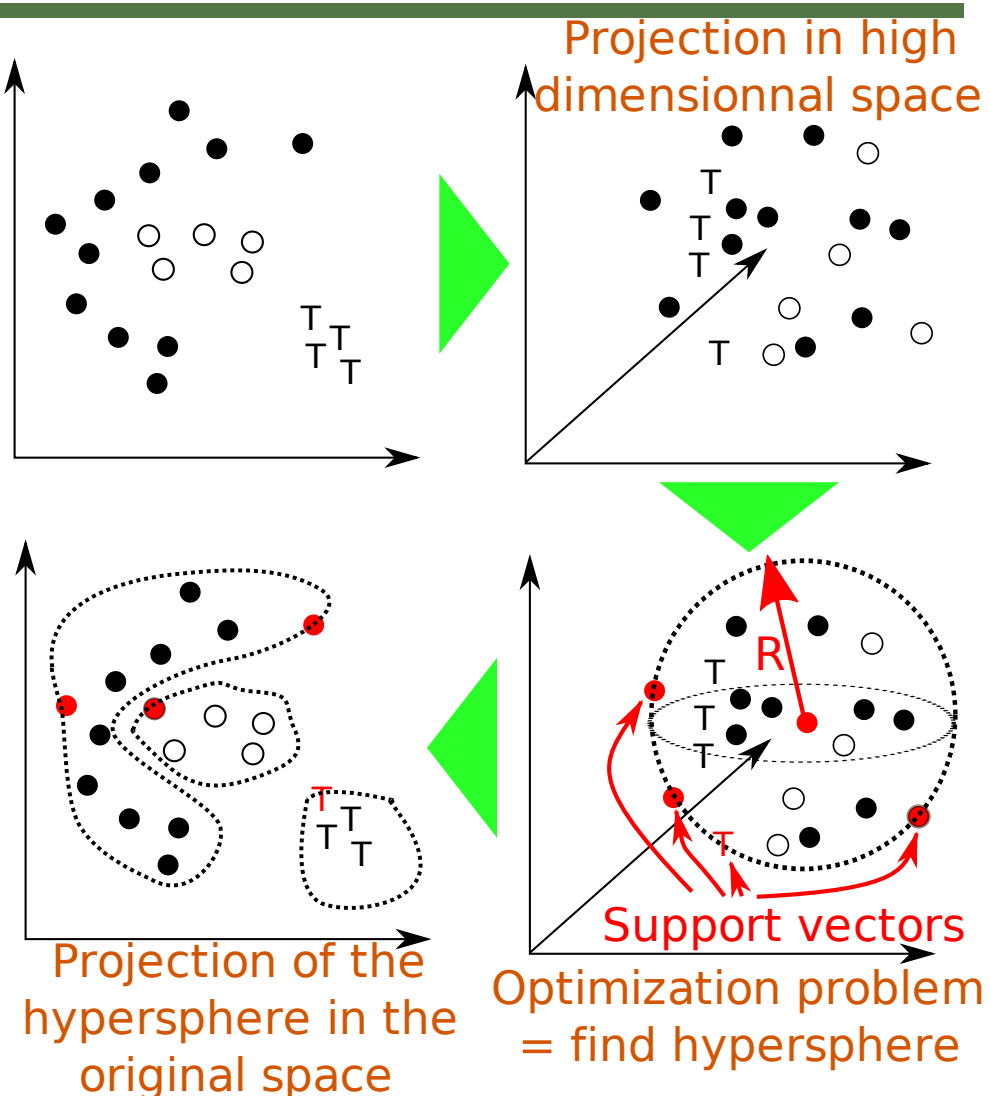
SVC

- projection function $\Phi \longrightarrow \langle \Phi(m_i).\Phi(m_j) \rangle \Leftrightarrow$ kernel function
- Gaussian kernel between two messages:

Parameter

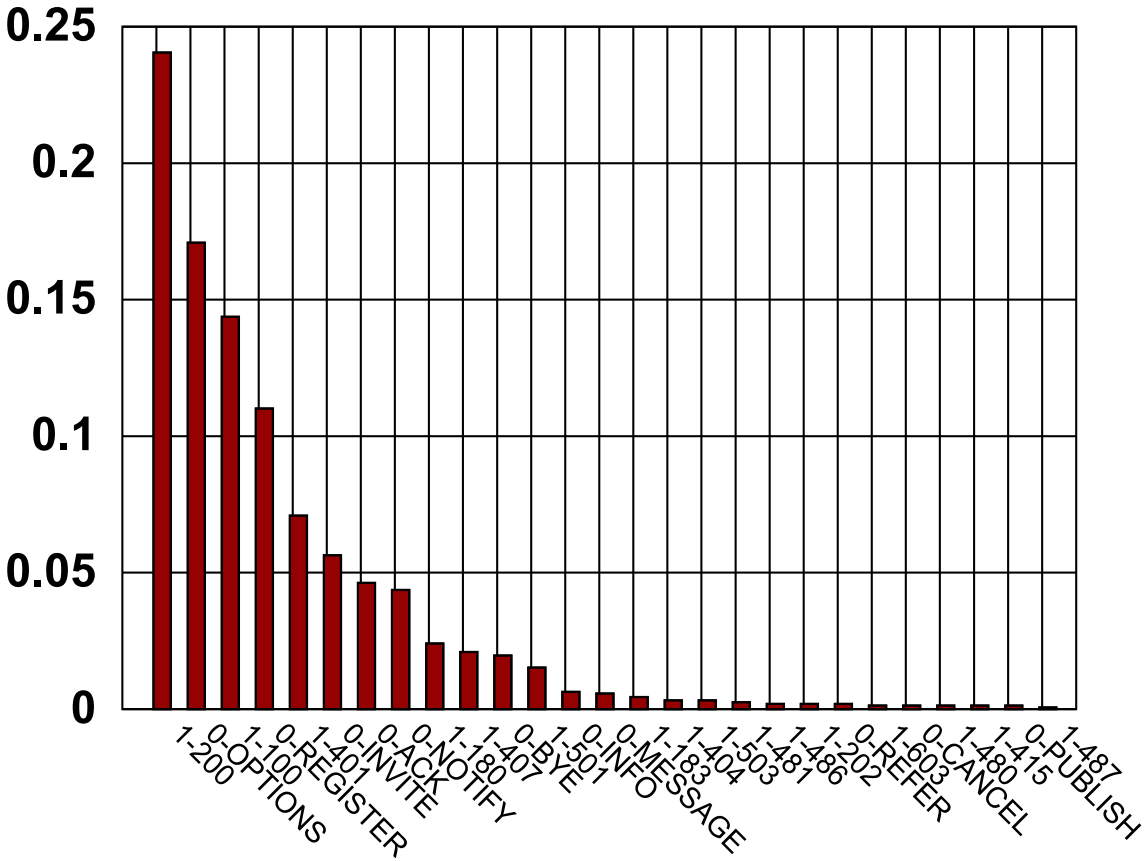
$$K(m_i, m_j) = e^{-q\|m_i - m_j\|^2}$$

- one additional parameter (C) allowing some points outside the sphere



Dataset

- SIP = Session Initiation Protocol
- management of VoIP communications: establishment, tear-down and negotiation of sessions

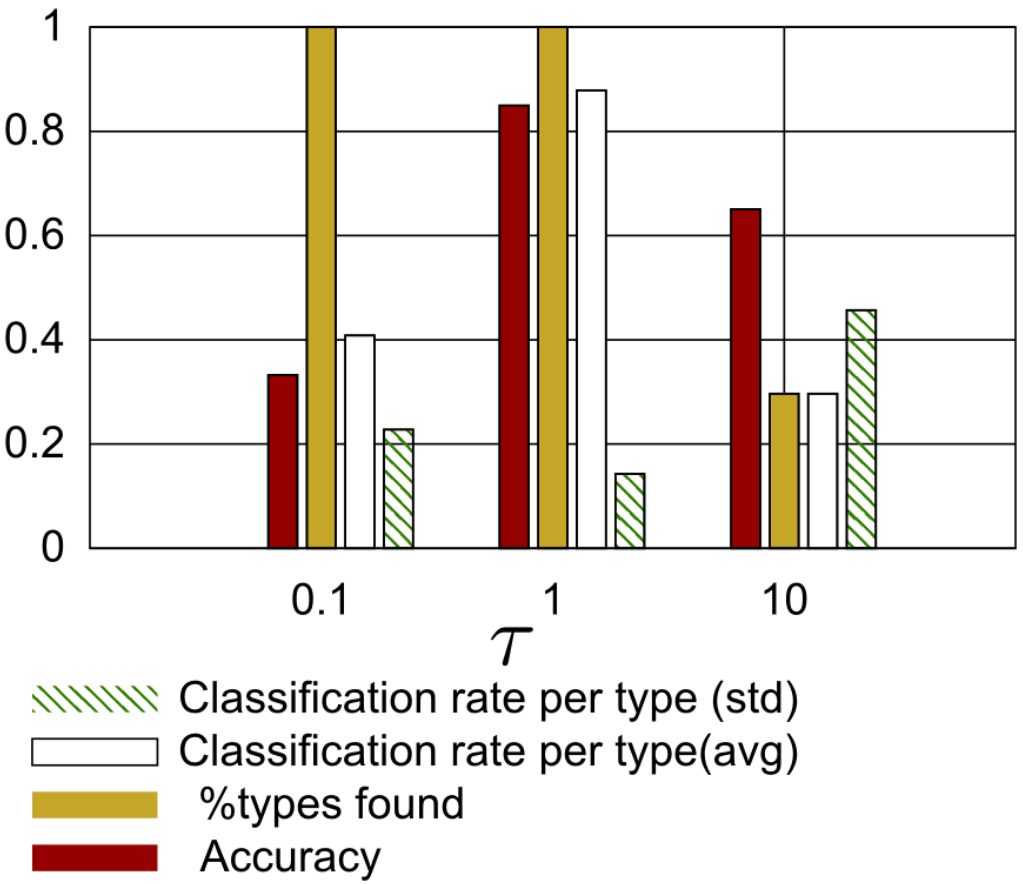


- 1580 messages
- 27 types
- unbalanced distribution

Evaluation metrics

- classification rate / accuracy = proportion of messages correctly identified
- classication rate per type = proportion of messages of a certain types correctly identified
- proportion of different message types discovered

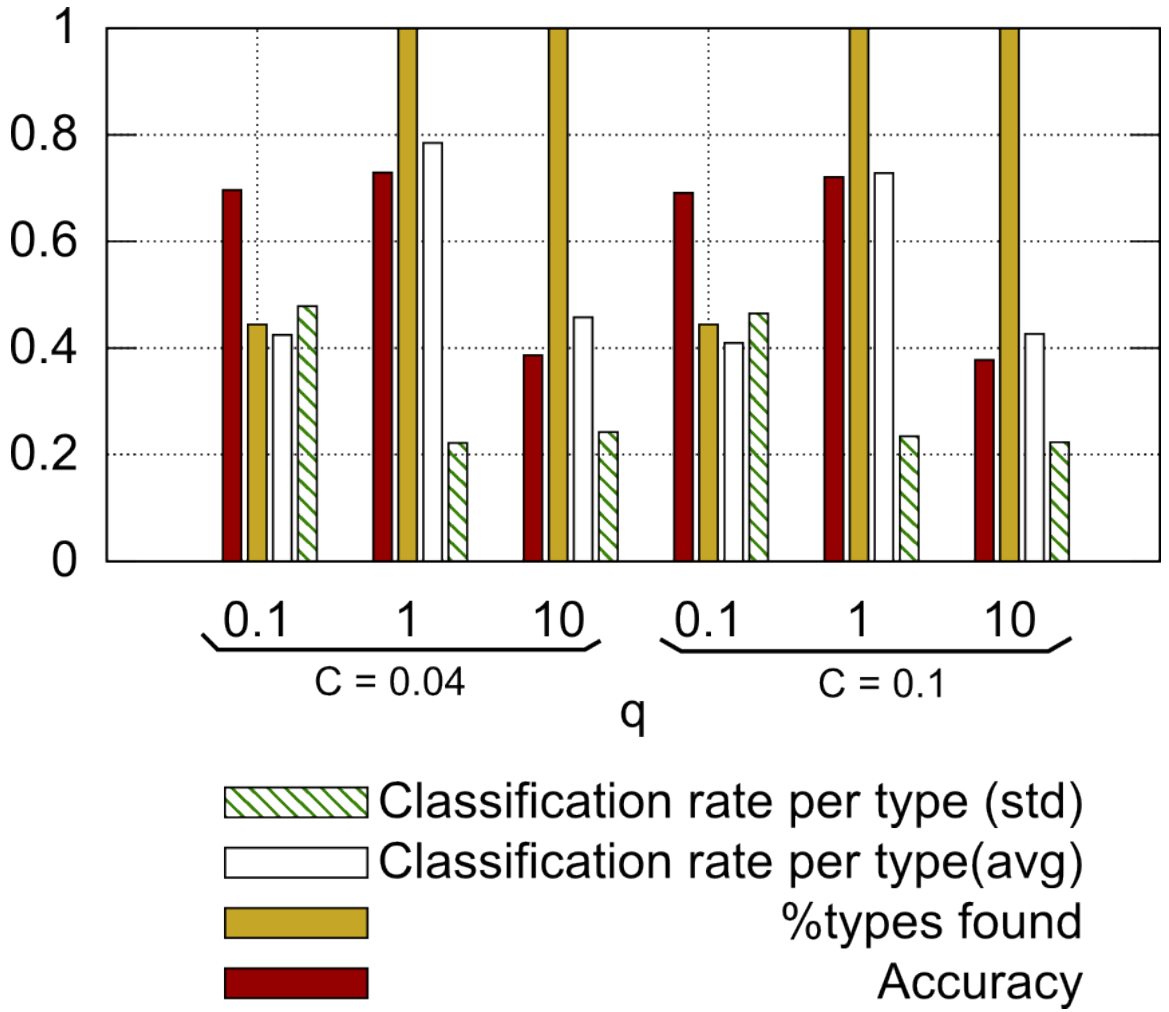
Nearest neighbors only results



- τ increases
  - larger clusters
  - begin: group similar messages
  - end: group different messages
- low standard deviation per type + manual analysis $\rightarrow$ mixed types

- best case: accuracy = 0.85, all message types found

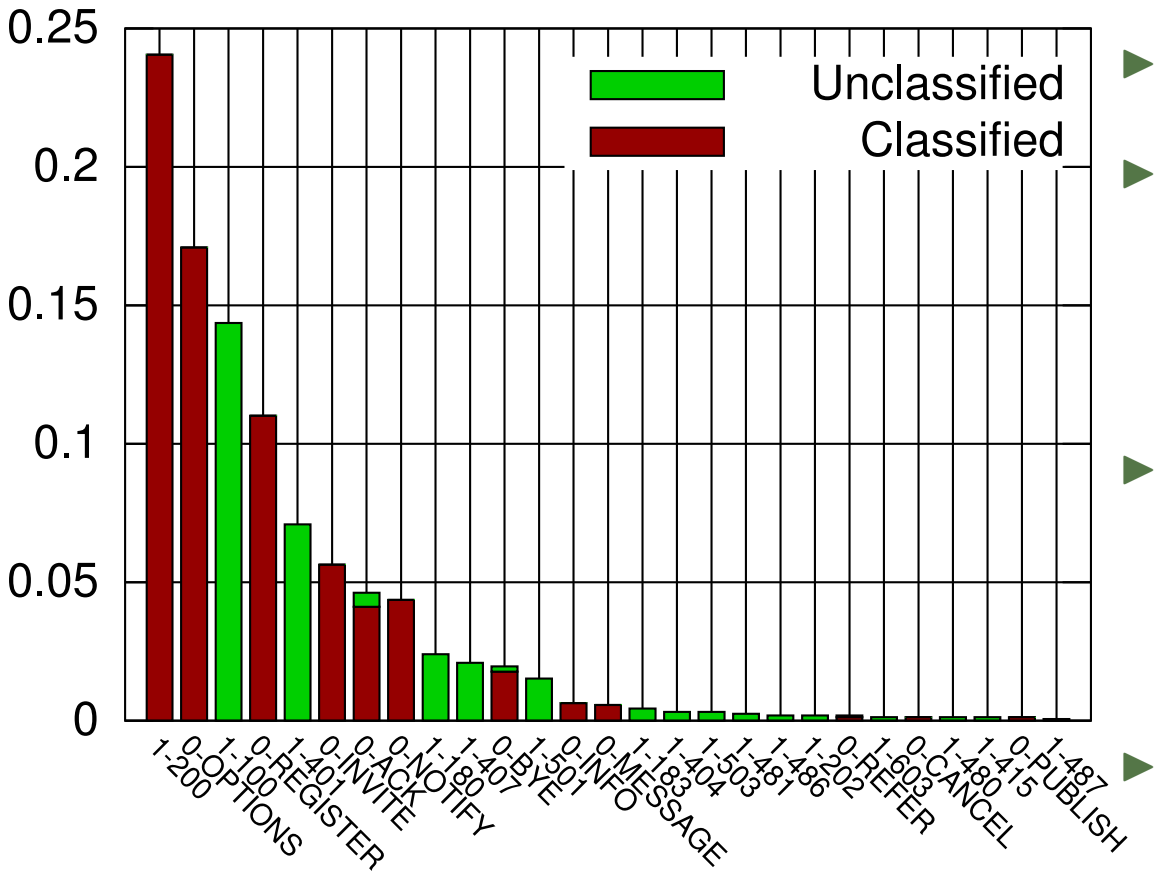
SVC only results



- no impact of C
- q increases $\longrightarrow$ differences emphasized between messages $\longrightarrow$ more clusters

- best case: accuracy $= 0.73$, all message types found

Global method application



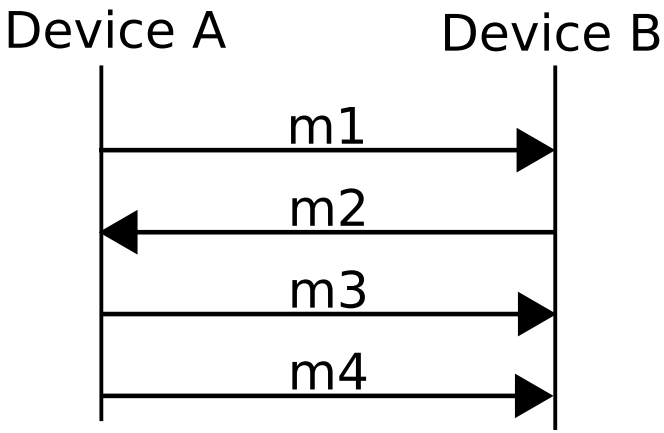
- $q = 0.1$
- types undiscovered not scattered among clusters
- meta-clusters containing several types
- $\longrightarrow$ splitting the obtained clusters again

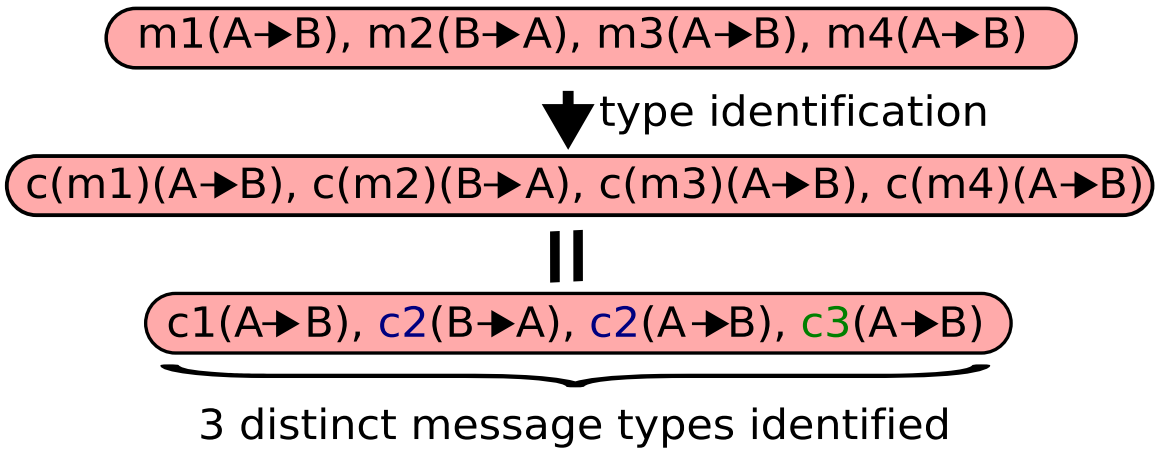
Gobal method: accuracy $= 91\%$ with 96% types discovered

Outline

Session representation

- session = messages exchanged between several entities (IP addresses and ports) ended by a long inactivity period
- behavior = sequence of messages

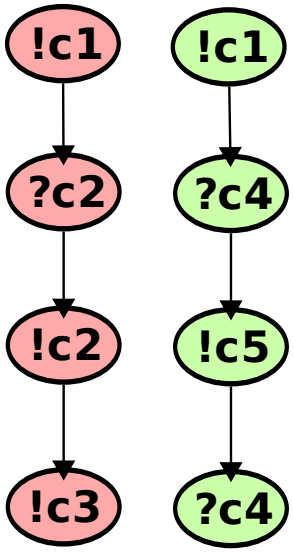




Tree representation

c1(A→B), c2(B→A), c2(A→B), c3(A→B)

c1(A→B), c4(B→A), c5(A→B), c4(A→B)

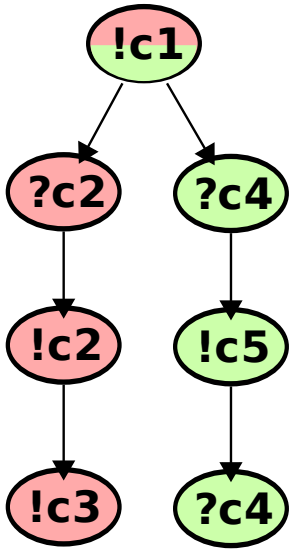


- 1 sequence = 1 branch

Tree representation

c1(A→B), c2(B→A), c2(A→B), c3(A→B)

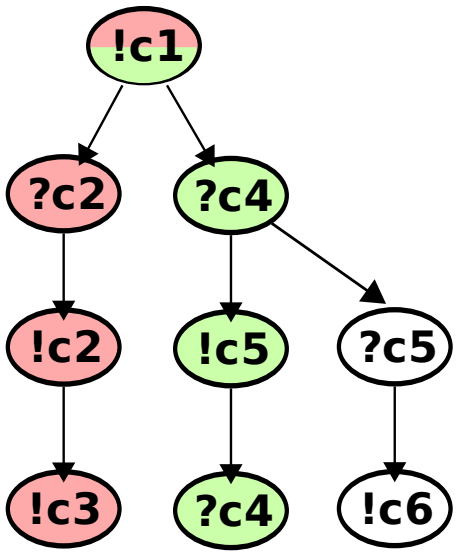
c1(A→B), c4(B→A), c5(A→B), c4(A→B)



- 1 sequence = 1 branch
- Merge common prefix

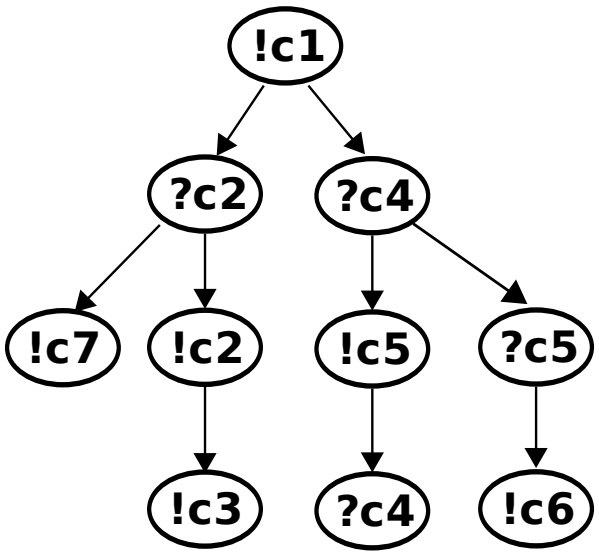
Tree representation

c1(A→B), c2(B→A), c2(A→B), c3(A→B)



Device A

c1(A→B), c4(B→A), c5(A→B), c4(A→B)

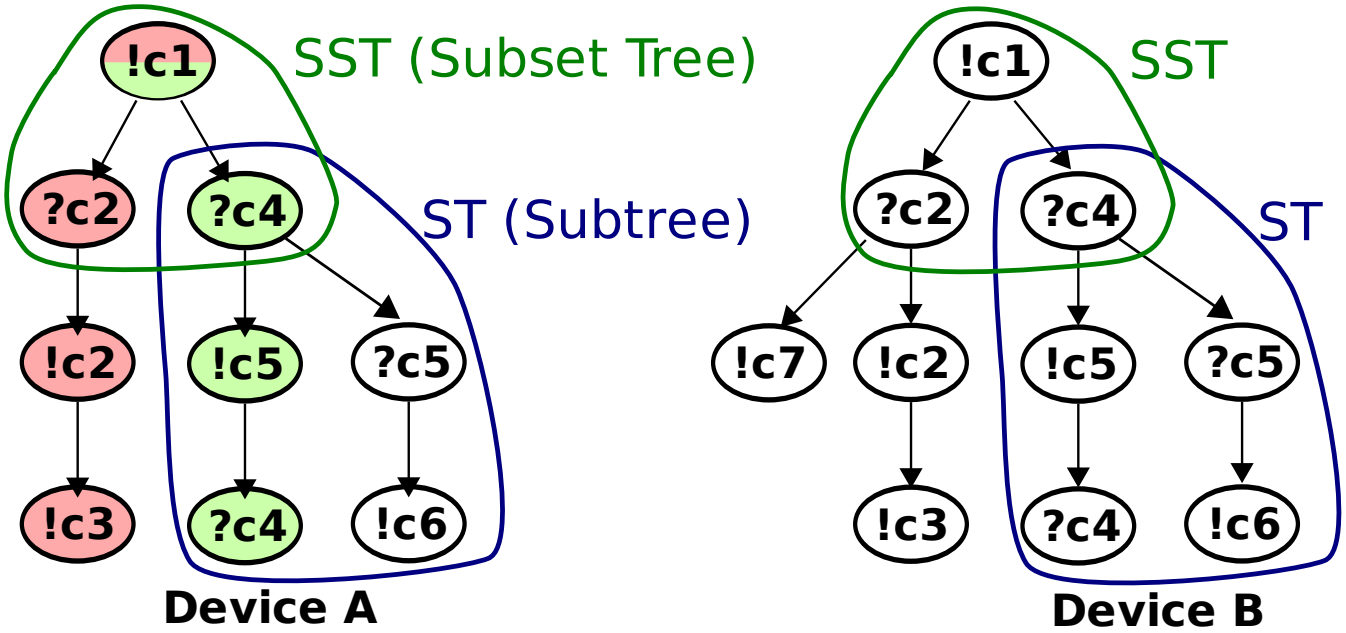


Device B

- 1 sequence = 1 branch
- Merge common prefix

- Construct complete trees from different machines

Trees comparison



- goal = estimate the similarity between two trees[2]
  - subtree = a set of connected nodes (perfect match)
  - subset tree = a "cut" in the tree (partial match)
- multi-class Support Vector Machine (SVM)

[2] A. Moschitti, Making tree kernels practical for natural language learning, 2006

Results

- SIP $\longrightarrow$ VoIP $\longrightarrow$ many threats (DoS, toll fraud, SpIT...)
- dataset details:
  - 40 different phones and proxies
  - 8 GB (training set = 20%)
  - real VoIP network
- assumption: device type = user-agent banner
- promising result: 80% of devices correctly identified

Outline

Conclusion

- message type discovery:
  - compressed message representation
  - recent classification technique (SVC)
- device fingerprinting:
  - message types based only
  - behavior = sequence of messages grouped in a tree
  - recent classification technique: SVM + tree kernel
- main constraint = no protocol syntax knowledge
- future directions:
  - other protocols
  - other representations / classification algorithms

Automated Behavioral Fingerprinting

Jérôme François, Humberto Abdelnur, Radu State and Olivier Festor