A Passive Attack on the Privacy of Web Users Using Standard Log Information

Thomas Demuth
Department of Communication Systems
University of Hagen
Germany

What is this talk about?

How

- easy or
- difficult (?)

is it to identify

- computers (or persons) that use dynamically assigned IP addresses

in the World Wide Web
using log information of web servers?

Motivation

- To show the feasibility beyond a level of 'expert knowledge'
- To show how good (or how bad) it is possible
- To motivate the use of anonymising services (especially for non-experts/average WWW users)

Overview

- Privacy risks for WWW users
- Known privacy attacks
- HTTP
- Information retrieval
- Terminology
- Appropriate HTTP fields for identification
- Proposed Algorithm
- Experiments and results
- Countermeasures

Overview

- Known privacy attacks
- HTTP
- Information retrieval
- Terminology
- Proposed Algorithm
- Experiments and results
- Countermeasures

Privacy Risks for WWW Users

WWW users transmit personal information

implicitly

- via HTTP
  - additional information for influencing the reaction of the web server is transferred
  - language preference,
  - compression type,
  - authentication data.

Privacy Risks for WWW Users

explicitly

- postal address
  - for getting brochures, giveaways, or possible prizes in on-line lotteries
- individual personal information like
  - hobbies/personal preferences,
  - marital status,
  - their income,
  - or even other family members.

Privacy Risks for WWW Users

- Using this data, interested institutions are able to track web users.
- The mentioned information can additionally be extended by publicly available data (address directories, ...)
- Example: Attempted merging of DoubleClick and Abacus Online
- It can be assumed, that data brokers are interchanging their information.

Known Privacy Attacks

Active

- Cookies
- Webbugs

- Active elements in web pages
  - Active X
  - Java
  - JavaScript

Known Privacy Attacks

Passive

- Evaluating web server log files
- Assumption:
  - IP addresses are static
  - True for computers of
    - companies,
    - universities, ...
  - But most Internet users use ISPs (e. g. AOL):
    - IP address is dynamically assigned

Known Privacy Attacks

Common estimation:

- *Internet (WWW) users with dynamically assigned IP addresses are sufficiently secured against privacy attacks!*

True?

Can information of another OSI level be used for (re)identifying/tracking?

Known Privacy Attacks

Judgement of security experts:

- *User tracking by HTTP information is possible!*

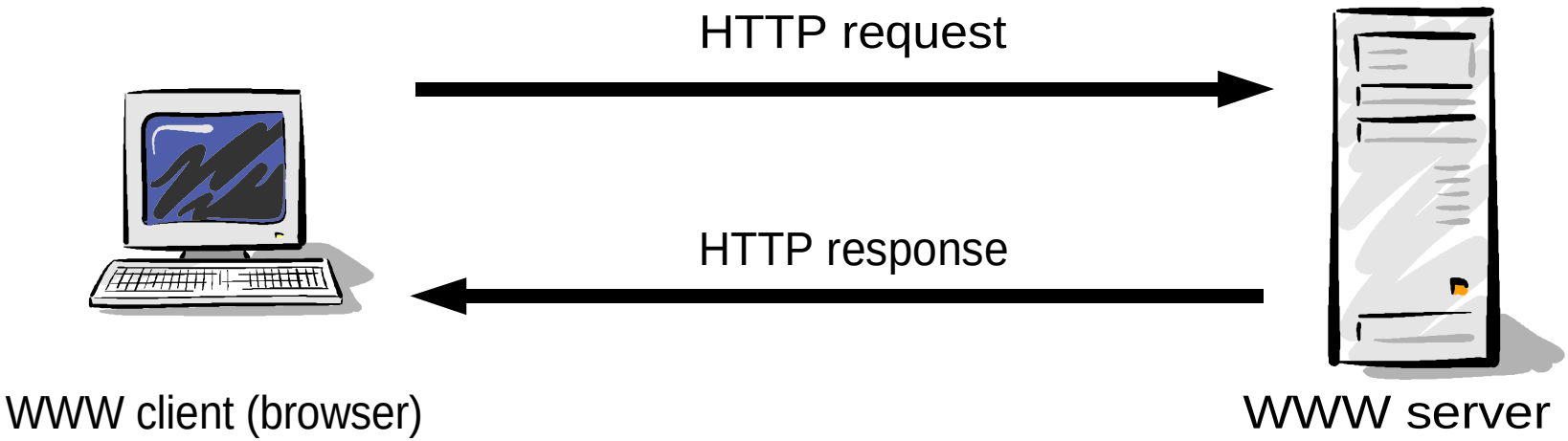
True?

How good (or bad)?
What degree of identification is possible?
What are the (computational) costs?

HTTP

HyperText Transfer Protocol:

- Standardised protocol for exchange of WWW objects.
- Client-server oriented
- Easily readable by humans



HTTP

Example HTTP request
(request for http://www.amazon.com/

```
GET http://www.amazon.com/
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
Accept: text/html, image/png, image/jpeg, image/gif,
image/x-xbitmap, */*
Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
Host: www.amazon.com
User-Agent: Opera/5.0 (Linux 2.2.16 i686; U)  [en]
```

Attacker's Situation

Has

- a huge database of web server log entries (consisting of selected HTTP fields)
- a smaller database of log entries with extended (personal) information

Wants

- to (re)identify users by identifying log entries as good as possible
- to track users for some time by tracing log entries (if possible)

Similar Situation

Information retrieval (library search)

- searching by keywords
- searching in large data bases of documents or articles
- good matching of terms and documents is desired

Advantages

- availability of search algorithms
- metrics for quality measurement of retrieval available

Information Retrieval

0. Database analysis

- Search for expressive keywords
- Elimination of redundancy
- Efficient storage

1. Formulation of a request

- Using keywords

2. Retrieval

- Match against each database entry

4. Presentation of the results

Process of Information Retrieval

0. Text analysis/indexing

- documents are parsed to find expressive keywords (indices/terms)
- each document is presented by a representation (index vector)

Goals

- elimination of redundancy
- performance reasons

Process of Information Retrieval

1. Formulation of request

- list of indices representing the desired documents as good as possible
- type of (boolean) concatenation

2. Retrieval

- matching of the request against each database entry (of representations)
- storing of the best *n* matches

Process of Information Retrieval

3. Presentation of the result

- as text, web page, etc.

General problems

- search for the best terms representing the documents
- matching function/algorithm

Process of Information Retrieval

Quality of results

- **Recall**

$$recall = \frac{\text{number of relevant found}}{\text{number of relevant available}}$$

- **Precision**

$$precision = \frac{\text{number of relevant found}}{\text{number of relevant found} + \text{number of irrelevant found}}$$

Terminology

Access Data Set (ADS)
contains

- a timestamp describing date and time of a web server's log entry, and
- a set of terms $\{t_{1,1}, .., t_{m,n}\}$, contents of a number of HTTP header fields ($\{h_1, ..., h_m\}$)

- ADS = web server log entry (=document)

extended ADS (eADS)

- an ADS extended by personal information of a user

Terminology

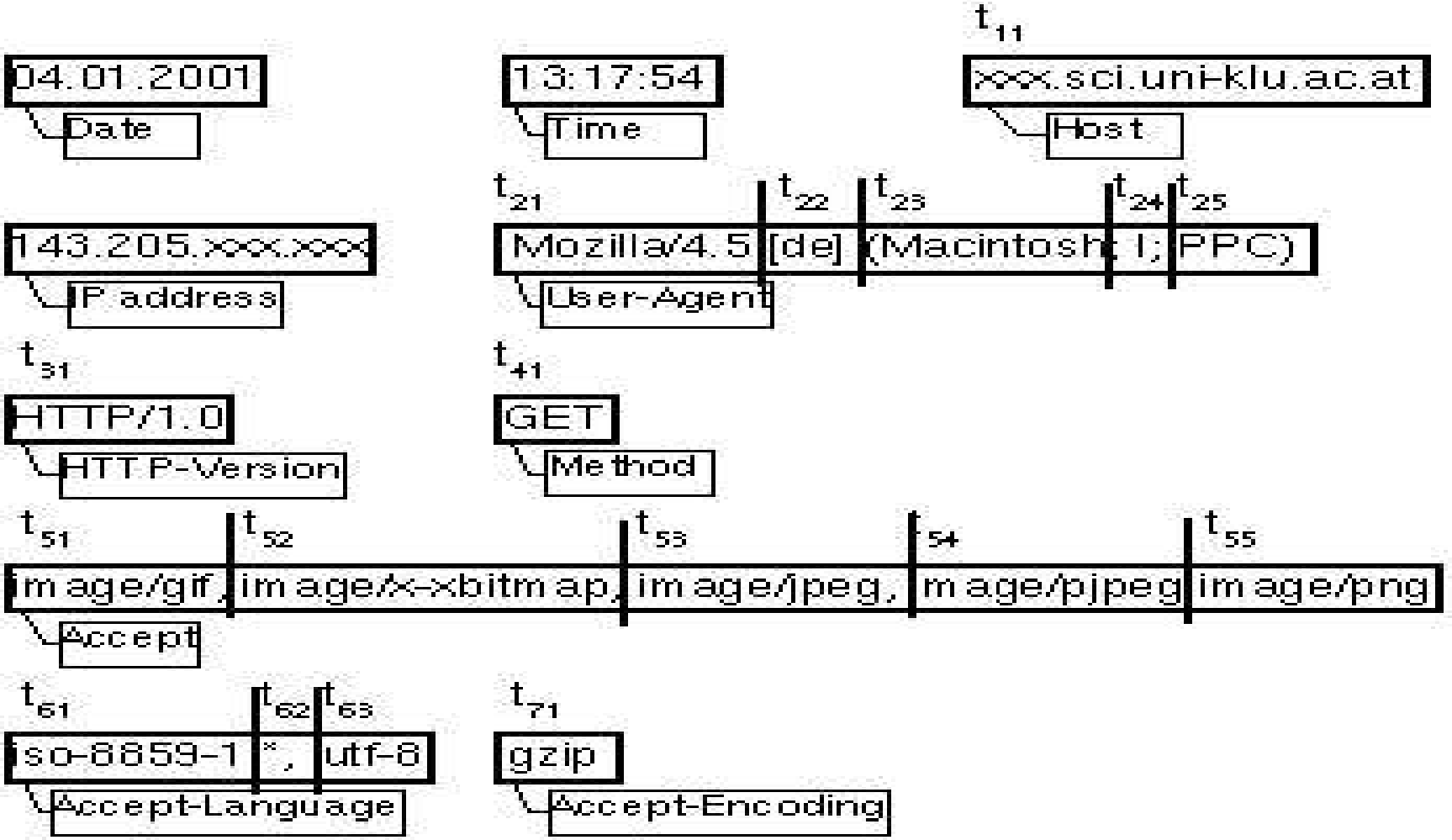
Instance
synonymously means

- a web browser and
- a person using this browser

- implicitly defined by the browser configuration

Terminology

Terms



Terminology / Adaption of Information Retrieval

IR	**Proposed attack**
index/keyword	term
document	ADS
document collection	ADS database
search request	existing ADS/ probe ADS

Search Quality

precision
recall

Relevant HTTP/1.1 Header Fields

Irrelevant fields

- fields for transporting instances (e. g. caches) like Cache-Control
- fields that can contain only a few different terms like Method

In general

- The more terms a header field can contain, the more expressive it can be

Relevant HTTP/1.1 Header Fields

Method field

- can contain 1 out of 8 terms (GET, POST, ...)
- can “mark” 8 ADS uniquely

User-Agent field

- can contain *p* out of *n* terms
- *p*: only technical limits, normally between 4 and 12 (average: 8)
- *n*: depends on the available ADS database (e.g. 320)

$$\binom{n}{p} = \binom{320}{8}$$ different User-Agent fields possible

Relevant HTTP/1.1 Header Fields

Used HTTP header fields

Host
User-Agent
Server-Protocol
Accept
Accept-Language
Accept-Charset
Method
Expect
From
Trailer
Warning
Via
Range
If-Range
If-Match
If-None-Match
If-Modified-Since
If-Unmodified-Since

Problems

ADS by the same instance vary over time
(time dependent variance)
because of

- new preferences,
- new software installed, or
- updated browser software

or (worst case)

- new browser software
- new operating system

Problems

Consequence:
No matching on equality but on similarity

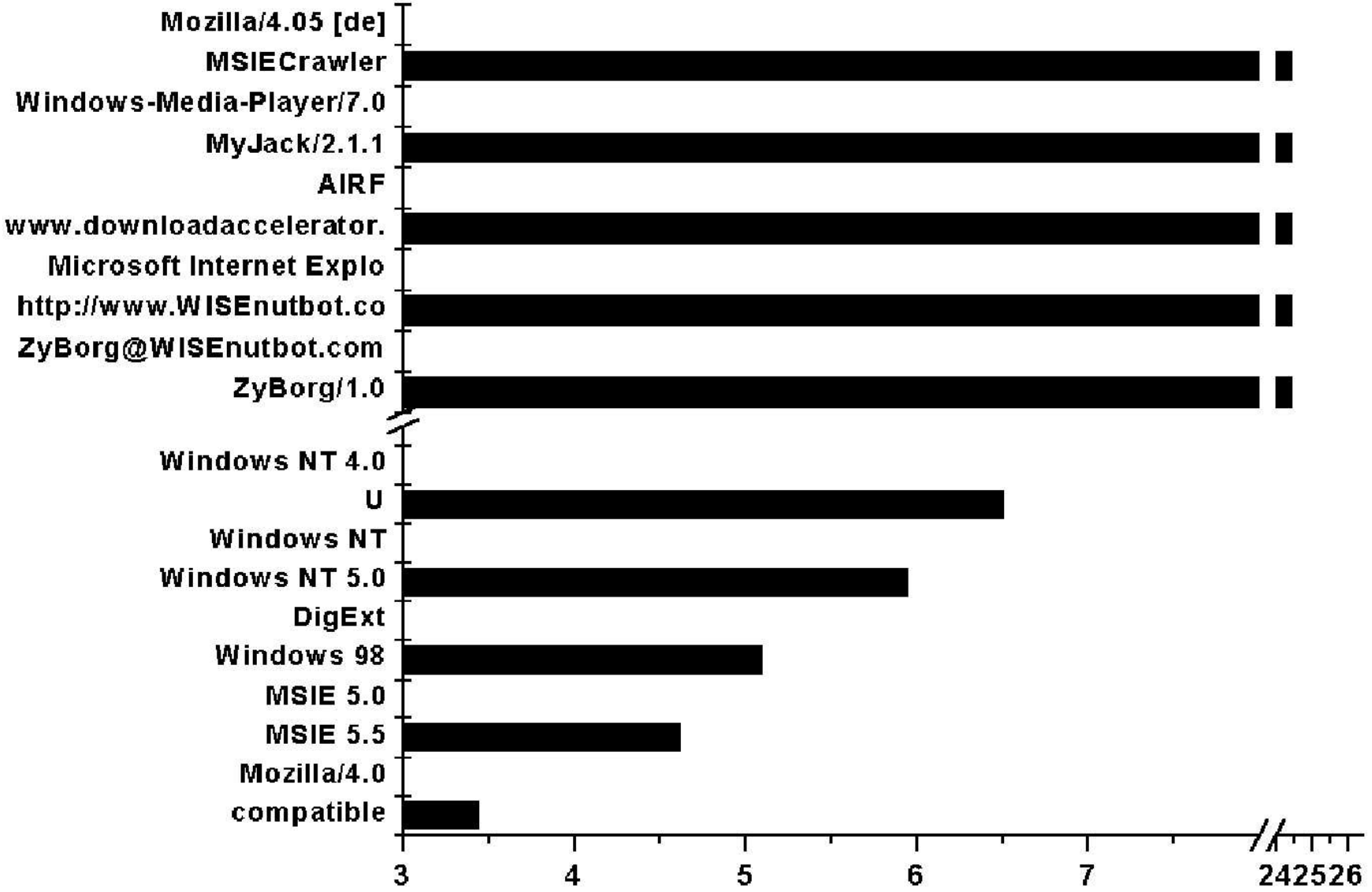
Each term has a significance (term weight):

$$weight(t_{j,k}) = -ld\left(\frac{cnt(t_{j,k})}{cnt(t)}\right)$$

Each ADS has a significance (ADS weight):

$$weight'(a) = \sum_{j=1}^{n} \frac{\sum_{k=1}^{l_j(a)} weight(t_{j,k})}{l_j(a)}$$

Example: Term Weights of Header Field User-Agent



Example: Variety in Header Field User-Agent

Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
Mozilla/4.0 (compatible; MSIE 5.0; MSN 2.5; Windows 98; PKBL008; DigExt)
Mozilla/4.61 [en] (Win95; I)
Mozilla/4.0 (compatible; MSIE 5.0; Windows NT; DigExt)
Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)
Mozilla/4.51 [en] (X11; I; Linux 2.2.15 i686)
Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; DigExt)
Mozilla/4.0 (compatible; MSIE 5.5; Windows 98; Win 9x 4.90)
Mozilla/4.0 (compatible; MSIE 5.01; Windows 98)
Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; DigExt)
Mozilla/4.0 (compatible; MSIE 5.0; MSN 2.5; Windows 98; PKBL008; DigExt)
Mozilla/4.0 (compatible; Powermarks/3.5; Windows 95/NT4)
Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; DigExt)
Mozilla/4.7 [en] (Win95; U)
Mozilla/4.51 [en] (X11; I; Linux 2.2.15 i686)
Mozilla/4.0 (compatible; MSIE 5.5; Windows 98; BitWise Systems)
Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Mozilla/4.7 [de] (WinNT; I)
Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Mozilla/4.0 (compatible; MSIE 5.01; Windows 95)
Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Mozilla/5.0 (X11; U; Linux 2.2.16 i686; en-US; Preview) Gecko/20001101 Beonex/0.6-pre

Up to 23.4 % of ADS are unique within the User-Agent header

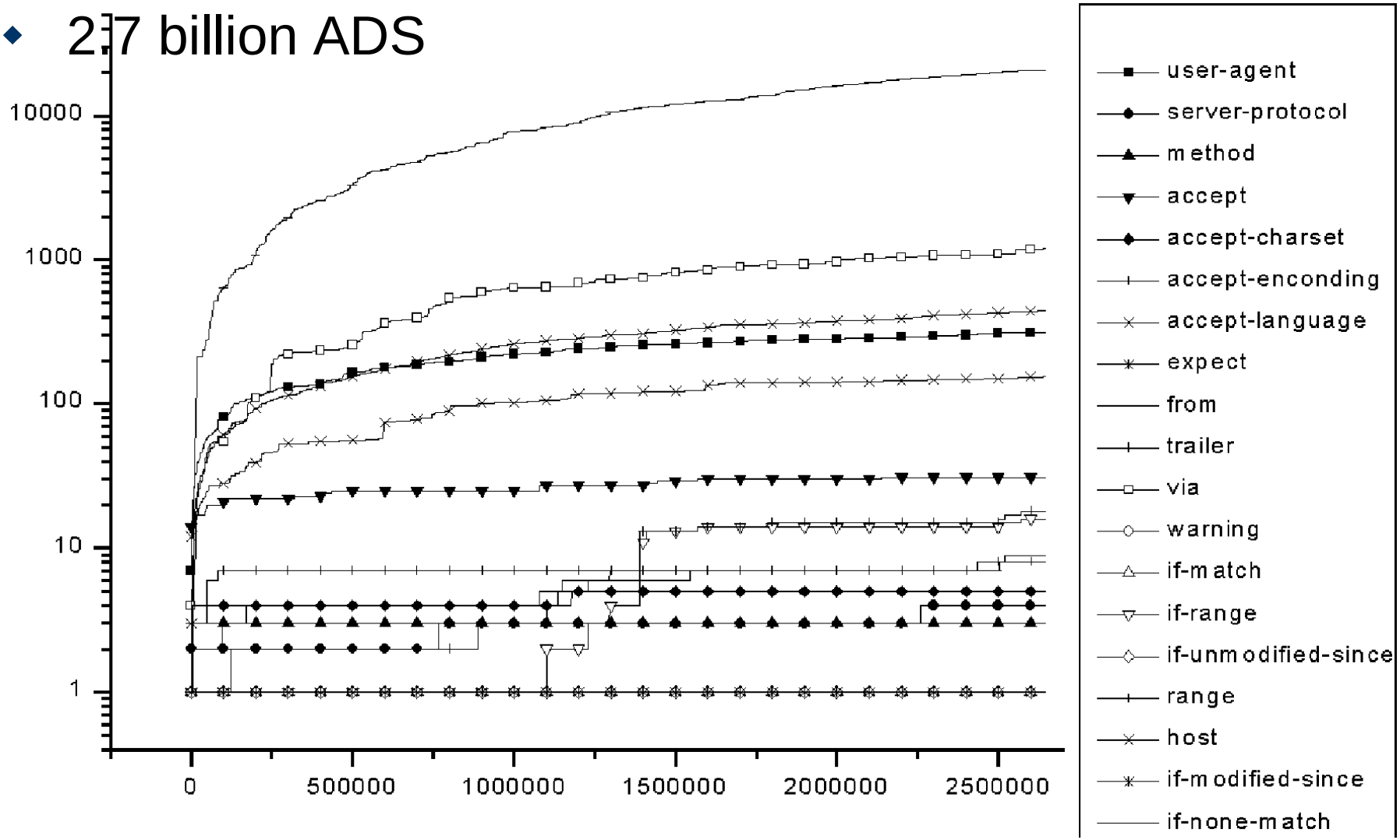
Example: Variety in Header Field User-Agent

Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
Mozilla/4.0 (compatible; MSIE 5.0; MSN 2.5; Windows 98; PKBL008; DigExt)
Mozilla/4.61 [en] (Win95; I)
Mozilla/4.0 (compatible; MSIE 5.0; Windows NT; DigExt)
Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)
Mozilla/4.51 [en] (X11; I; Linux 2.2.15 i686)
Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; DigExt)
Mozilla/4.0 (compatible; MSIE 5.5; Windows 98; Win 9x 4.90)
Mozilla/4.0 (compatible; MSIE 5.01; Windows 98)
Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; DigExt)
Mozilla/4.0 (compatible; MSIE 5.0; MSN 2.5; Windows 98; PKBL008; DigExt)
Mozilla/4.0 (compatible; Powermarks/3.5; Windows 95/NT4)
Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; DigExt)
Mozilla/4.7 [en] (Win95; U)
Mozilla/4.51 [en] (X11; I; Linux 2.2.15 i686)
Mozilla/4.0 (compatible; MSIE 5.5; Windows 98; BitWise Systems)
Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Mozilla/4.7 [de] (WinNT; I)
Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Mozilla/4.0 (compatible; MSIE 5.01; Windows 95)
Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Mozilla/5.0 (X11; U; Linux 2.2.16 i686; en-US; Preview) Gecko/20001101 Beonex/0.6-pre

Analysis of ADS database

Parameters of used ADS data base

- 2.7 billion ADS



Algorithm

Step -1:

- For each ADS in the ADS database (new) terms are determined and stored:

$$\vec{t} = (t_{1,1}, \ldots, t_{1,\,cnt(h_1)}, t_{2,1}, \ldots, t_{2,\,cnt(h_2)}, t_{3,1}, \ldots)$$

Step 0:

- For each ADS *a* in the ADS database, the (binary) index vector is determined:

$$iv(a) = (b_1, \ldots, b_{|\vec{t}|})$$

Result: Database of representations of all ADS

Algorithm

For each search:

Step 1:

- ADS a_{probe} to be tracked, $weight(a_{probe})$ and index vector are calculated

Step 2:

- For each ADS a_i in the ADS database the similarity to a_{probe} is calculated

$$similarity(a_{probe}, a_i) = \sum_{r=1}^{l_{iv}} \sum_{s=1}^{l_{iv}} iv_r(a_{probe}) * iv_s(a_i) * weight(t_r) * weight(t_s)$$

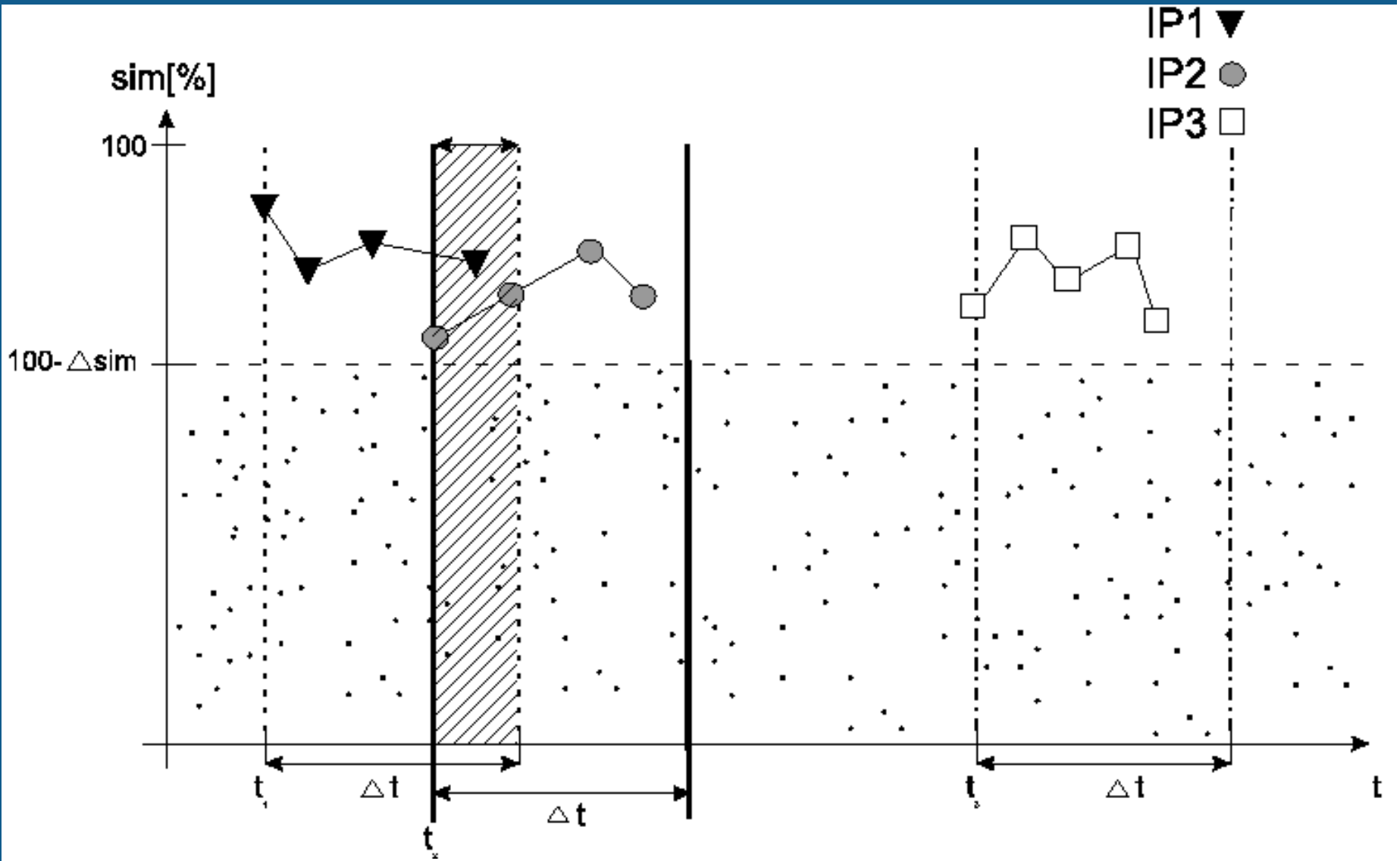
Evaluation

Dynamically assigned IP addresses normally does not vary during an Internet session

PAP (Potential Activity Period)

- A group of ADS assumed to be generated by the same instance
- Fulfil criteria:
  - Same IP address (as initialising ADS)
  - Similarity to a_{probe} is high enough (threshold Δsim)
  - Lies within a given time window Δt

Evaluation
PAP and PAP Intersections



Grade of Anonymity of an ADS

Two PAP intersecting (with the same IP address) build a PAP intersection:

- The more intersections, the more anonymous the probe ADS is
- The more common the configuration of the instance, the more common the generated ADS are
- The more common the ADS, the more intersections occur

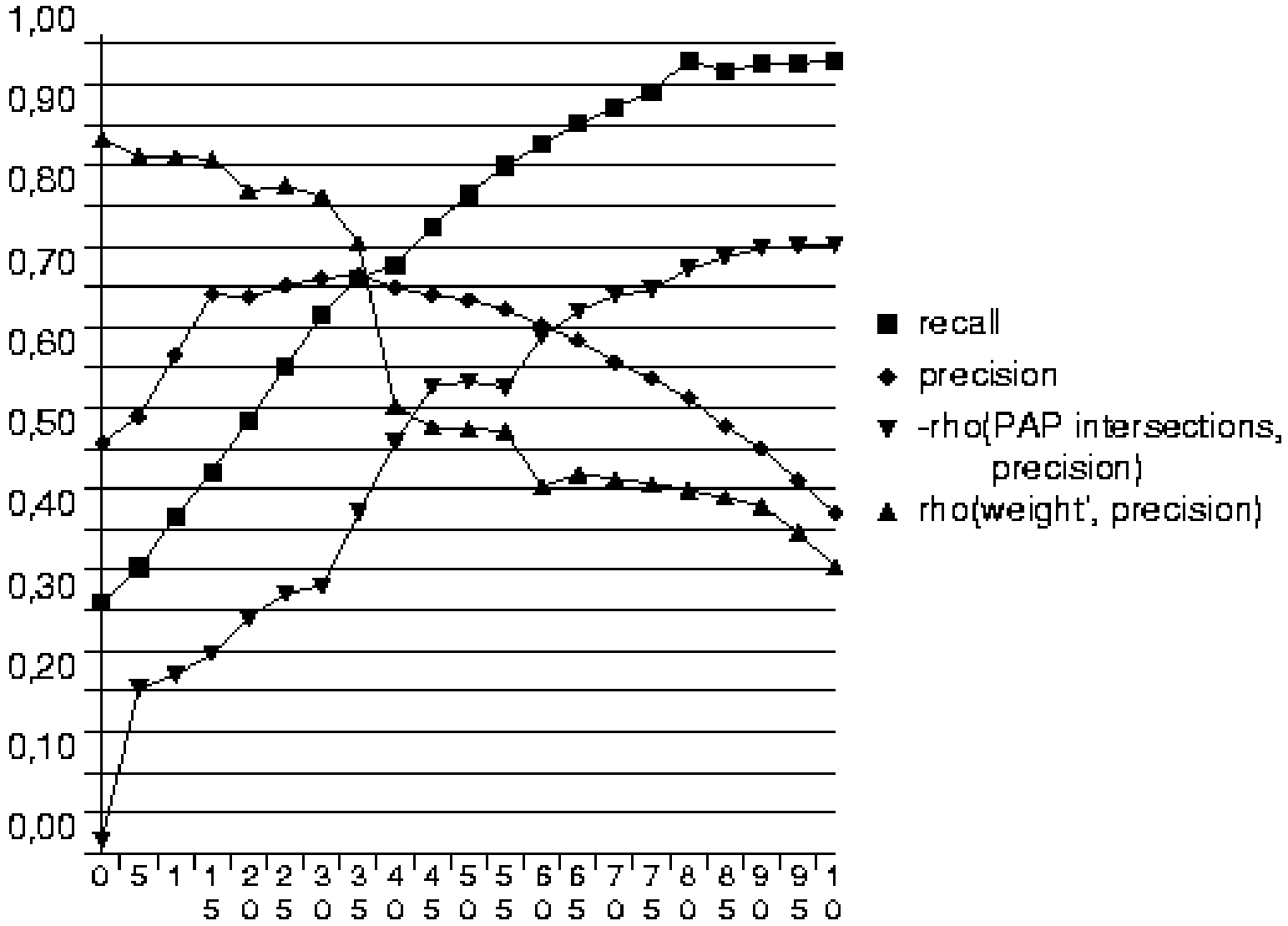
→ The PAP intersections of a probe ADS form an **Anonymity Set** for the probe

Experiments and Results

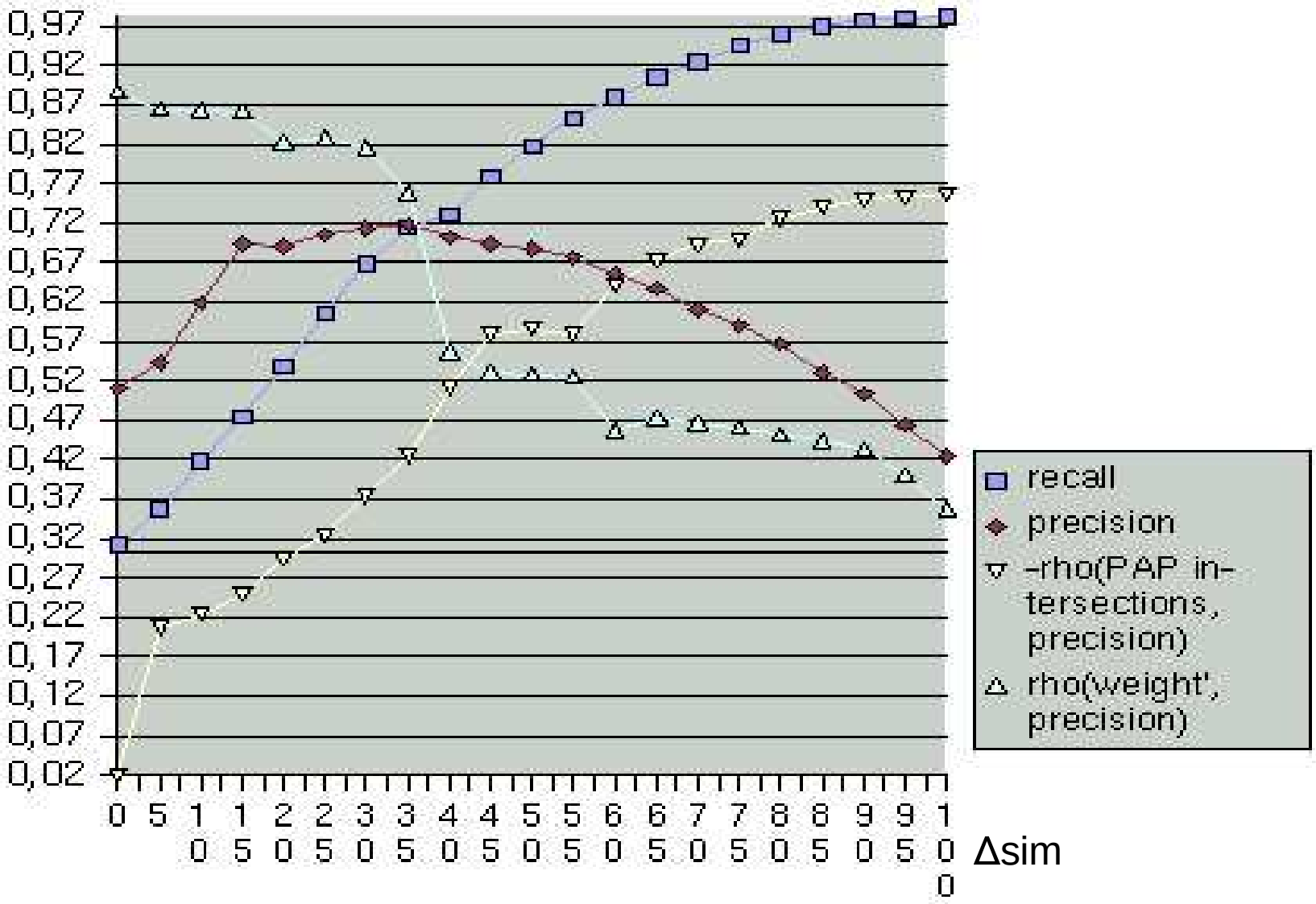
- 300 ADS of the ADS database have been “mutated” resulting in ca. 13.000 test ADS
- The mutated (and marked) ADS have been spread over the database

- Precision: How good is the algorithm in finding relevant ADS

Experiments and Results



Experiments and Results

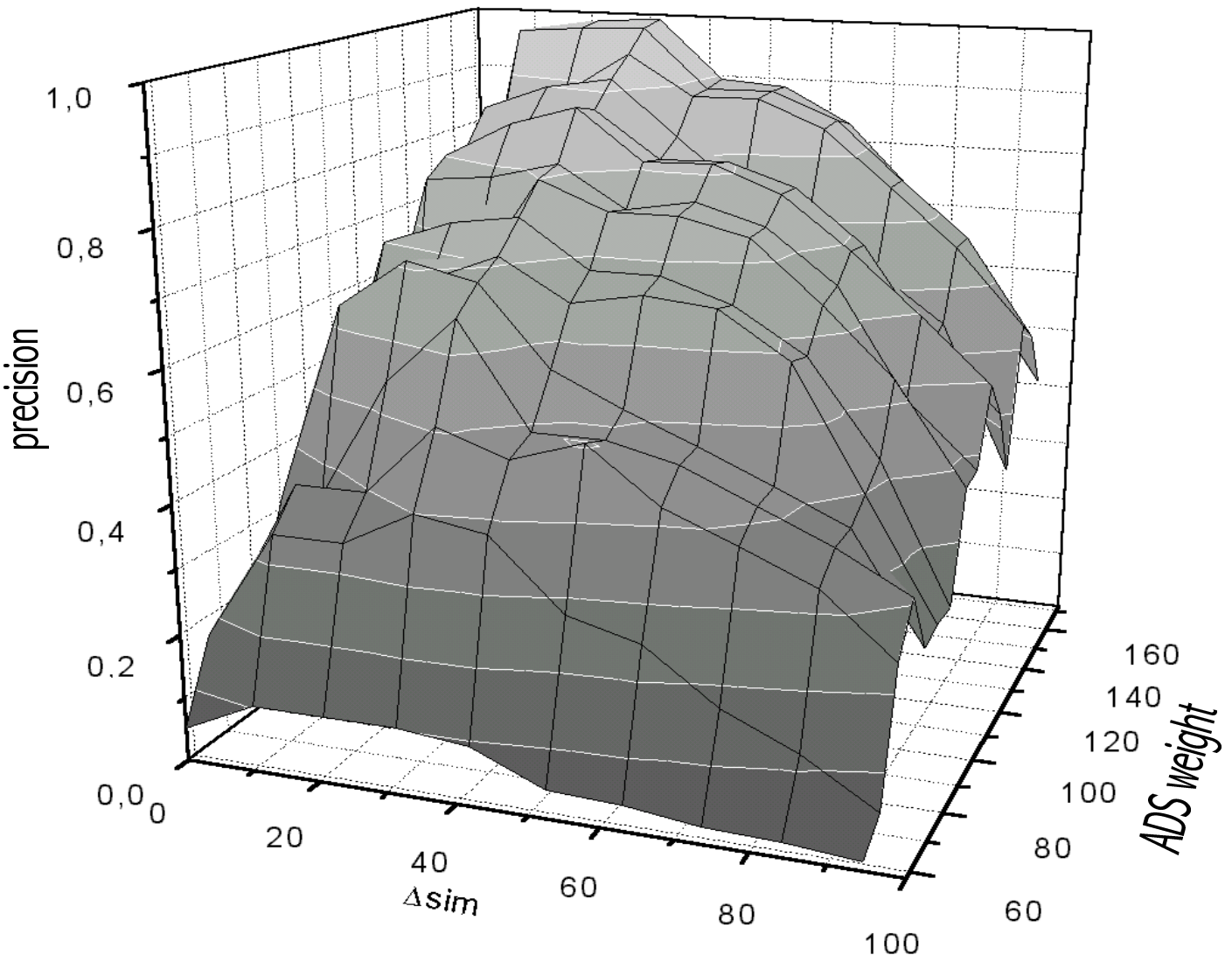


Experiments and Results

Algorithm shows

- on average
  - precision up to 0.71 (desired)
  - recall up to 0.98
  - local optimum at $\Delta sim = 35\ \%$
- at maximum
  - precision up to 1.0 (desired)
  - recall up to 1.0
- Correlation between weight and precision is identical to an “a priori” assertion/ predication

Experiments and Results



Experiments and Results

ADS examples

ADS with weight 87.58 and precision 0.82

```
<DATE> <TIME> <HOST>.dip.t-dialin.net <IP>
  Mozilla/4.0 (compatible; MSIE 5.5 Windows
  98; Win 9x 4.90) HTTP/1.0 GET */*
```

ADS with weight 156.01 and precision 0.97

```
<DATE> <TIME> <HOST>.uni-hamburg.de <IP>
  Mozilla/4.76 [de](X11; U; Linux 2.2.10 i686)
  HTTP/1.0 GET image/gif, image/x-xbitmap,
  image/jpeg, image/pjpeg, image/png
  iso-8859-1,*,utf-8 gzip de, ex-MX, es, en
```

Experiments and Results

Algorithm shows

- The higher the weight of an ADS, the higher the precision of the retrieval is
- The higher the number of PAP intersections, the lower the precision of the retrieval is

Countermeasures

In general

- Increasing the anonymity set
  - Producing more PAP intersections
- Decreasing the relative similarity to the probe ADS
  - Stronger variation of the instance's configuration

Countermeasures

1. Anonymising proxies

- anonymizer.com

- Acting as an intermediary
- Transforming (unifying) the HTTP request
- More ADS with the same weight are a found
- More PAP intersections occur

Countermeasures

2. Local proxies

- extending HTTP header fields by random and/or valid terms
- different for each access
- possible, because HTTP header fields are considered from left to the right
- intended header fields are interpreted correctly
- can be performed by simple software on each computer

Conclusion

Attack/algorithm

- shows, how to measure web accesses generated by an instance
- shows, how to compare accesses
- shows, that it is possible to identify instances (and therefore people)
  - depending on
    - the instance/browser configuration
    - the desired precision
- tracking is also possible with little more expenses

Thank you

for your attention!

Reviewer's Remark (Part 1)

Reviewer:

Some of the header fields listed are not relevant for identifying users (e.g. if-None-Match, Host, Range, If-Modified-Since), because they are more kind of identifications of the requested page or server. Almost every browser sends these special headers in order to request a certain page. Without these header fields the number of distinguishable requests gets substantially smaller.

Reviewer's Remark (Part 1)

1. Analysis of the probe ADS shows the usage of header fields:

if-match	0%
if-modified-since	0%
if-none-match	2.2%
if-range	0%
if-unmodified-since	0%
if-range	0%

Conclusion: not used in most cases.

Reviewer's Remark (Part 1)

2. Etags as identification mechanism:

- Etags are “strong validators”
- HTTP/1.1, 13.3.2: “... reliable validation in situations where ... the one-second resolution of HTTP date values are not sufficient”
- Can be unique because of very short lifetime (< 1 s)
- Etag could be misused to “mark” users

Reviewer's Remark (Part 2)

Reviewer:

> *I'm anyway in doubt about the idea of identifying users by the remaining header fields, because users mostly use a standard windows with standard IE, i.e. all users with the same windows version have the same fingerprint. (Nevertheless this method of identification by header fingerprinting only works for exotic configurations.)*

Right!

- But that is one statement of the article/presentation.

And

- Configurations don't have to be exotic, but they must not be trivial.