Universität der Bundeswehr München

On the Path to High Precise IP Geolocation: A Self-Optimizing Model

Peter Hillmann, Matthias Schopp und Lars Stiemert

Faculty of Computer Science

Universität der Bundeswehr München

Peter.Hillmann@unibw.de

Agenda

Universität der Bundeswehr München

1. Introduction to active Geolocation
2. Our approach
3. Experiments and results
4. Extensions and improvements

1. Introduction

IP Geolocation

→ Determination of a geographical location of a logical IP address

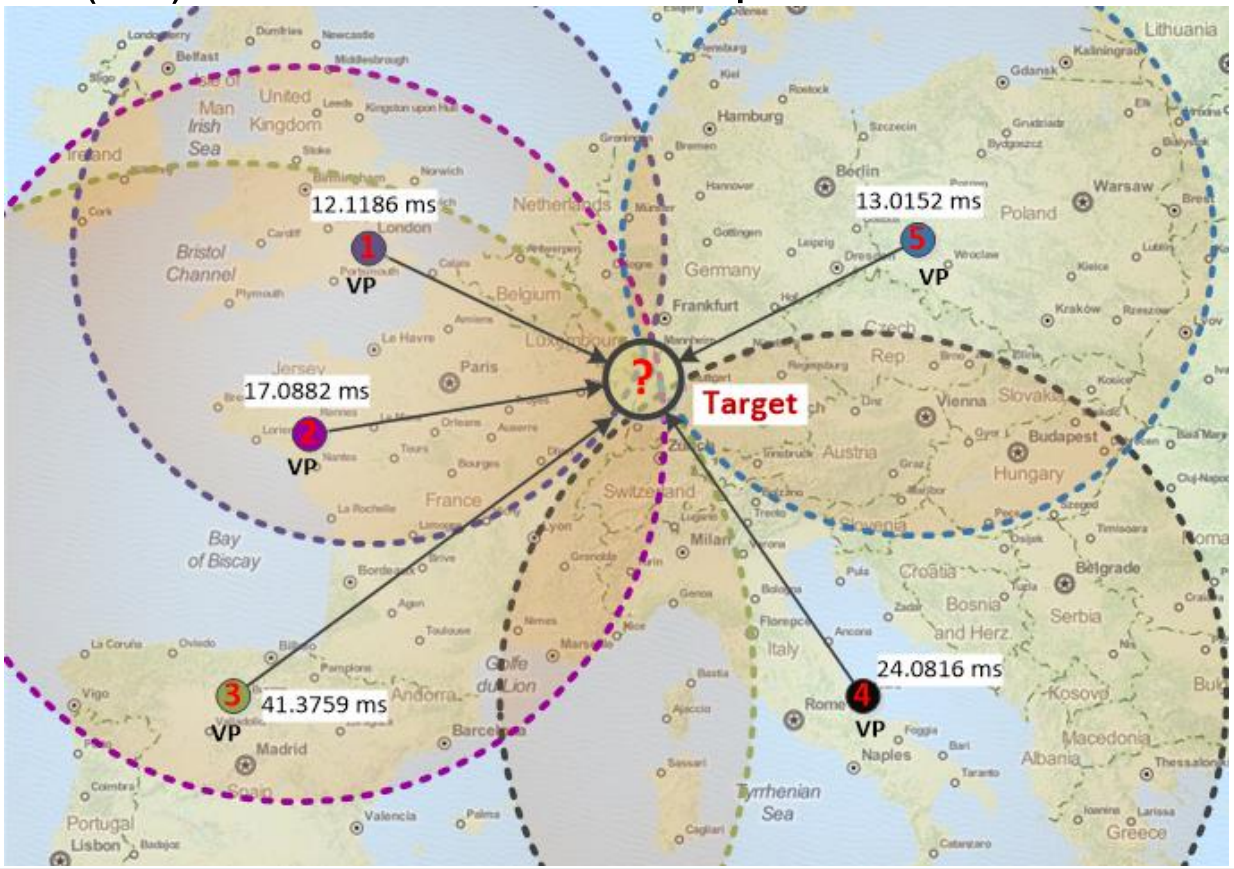
Taxonomie

- IP mapping based
  - Passive – No interaction with the target
  - Databases, …
- Measurement based
  - Active – Request-Response interaction with the target
  - „Loose“ correlation between network latency and geographical distance
  - Shortest Ping, GeoPing, CBG, TBG

1. Active Geolocation

Measurement Points

- Vantage Point (VP): Well-known location with measurement infrastructure
- Landmark (LM): Reactive network component with a known location



1. Scenario and infrastructure

der Bundeswehr Universität München

- **Target with unknown location**
- **Rough information about the infrastructure**
- **VP**
  - Uniform distribution
  - Near to the unknown target (The closer the measurement point is to the target, the lower the probability of interference)

1. Problems

1. Identification of Vantage Points

- Network nodes with fast connection (backbone)
- Number of measuring points (→ operating costs)
- Optimized position in the network (k-center, NP-hard)
- *Rough information about the infrastructure*

2. Modelling the reality

- *Approximation of the network path (Length of wire, Speed)*
- Delays on the transfer path
  - Delay processing VS time travel distance
  - Influence of other network load
- *Weak correlation of latency and distance*
- Configuration of measurements

1. Clarification

- **Speed of a signal in the wire:**
  - Literatur: 2/3 or 4/9 speed of light
  - Physics:
    - 2/3 speed of light in fiber optic with refractive index 1.5 – 200,000 km/s
    - 3/4 speed of light in copper – 225,000 km/s

  → Conclusion: Precise Measurement in Mircoseconds mandatory
  (1 us → 225 m)

  - In tests: Considerably less by delay during processing
  (> 30.000 km/s)

2. Our approach

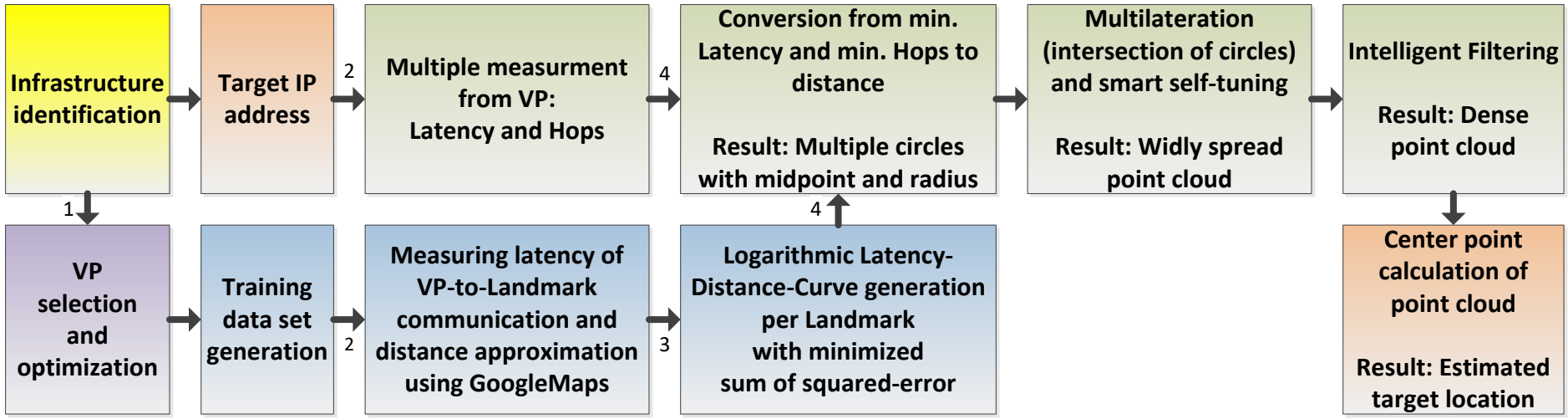
Idea

- Predefined set of vantage points (static)
- Send multiple requests to the target IP
- Measurement of latency, hop count and …

Steps to geolocate a target

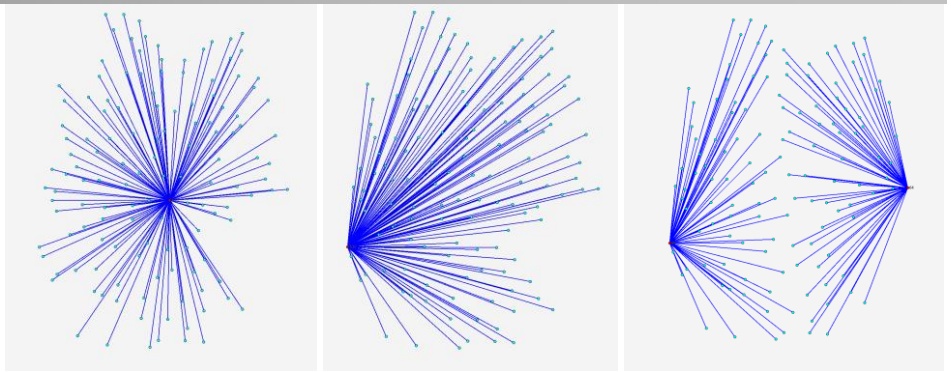
1. Identification of vantage points and its location
2. Measurement of latency and hop number between measuring points
3. Measurement of latency and hop number to the target
4. Conversion of the measurement results to a distance
5. Determination of the target location by lateration

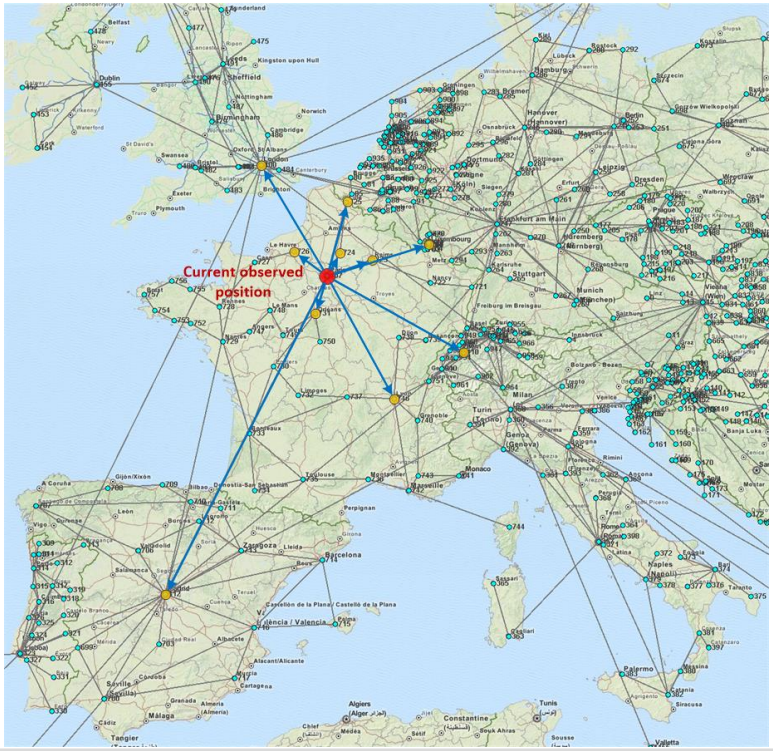
2. Overview of the process steps



2. Selection of VP

- **Novel algorithm: Dragoon** (Enhanced approach of 2-Approx and k-means)
- **Predefined amount of nodes initialized using the 2-Approx strategy with orientation node in the middle**
- **Checks locations around the observed position**
- **Objective: Maximum distance counted by hops**





2. Measurement

- **What should be measured? → Latency + Hop count**

- **Stochastic delay VS deterministic delay**
  - Different types of delays cause heavy variation:
    - Network load
    - Queuing
    - Different network paths

  → Several measurements to the same target (ICMP, UDP, TCP)

  → Select minimum value (No cached requests → Filter in V4)
    - Less side effects
    - Close to deterministic value of time for signal travelling

  - Hop count → processing network devices respected by average delay of 0.055 ms

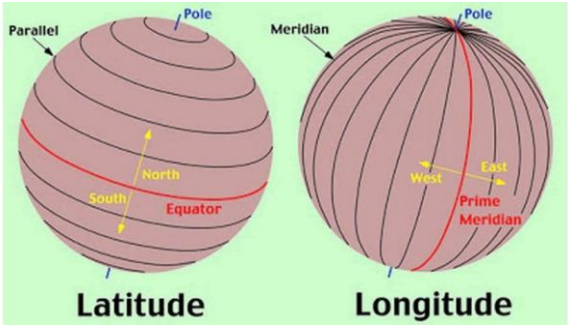
2. Distance approximation

1. Delay Measurement: VP-to-Landmark

- Locations are well-known → Distance is known?!

→Wire length approximation:

  - Euclidean
  - Orthodrom
  - Graph
  - GoogleMaps



- Reference curve (Latency-Distance)
- Location of VP is specific → Curve is individual for every VP
  (V1=common; V2= individual)

2. Delay Measurement: VP-to-Target

- Using reference curve to estimate distance from VP to target
- Multiple circles with midpoint as VP and radius

2. Delay conversion (1)

- **Physic:**

  v · t = s → lineare correlation

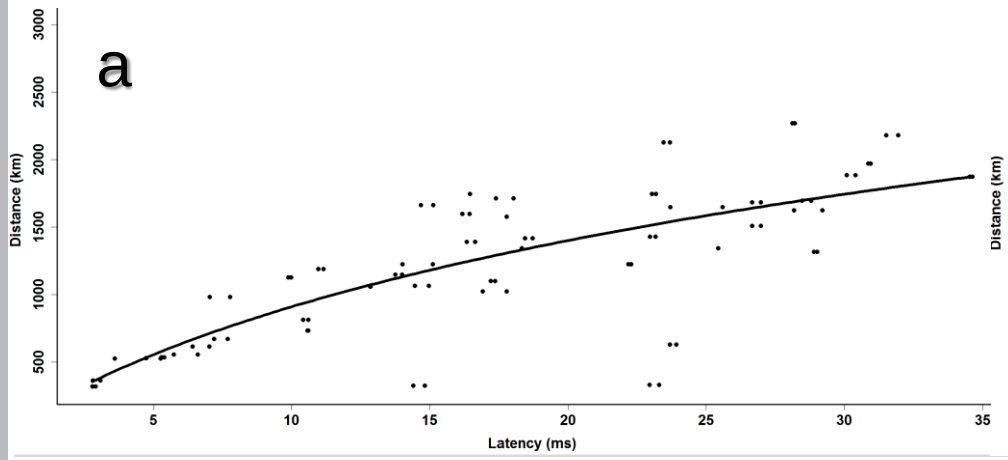
- **Reality:**

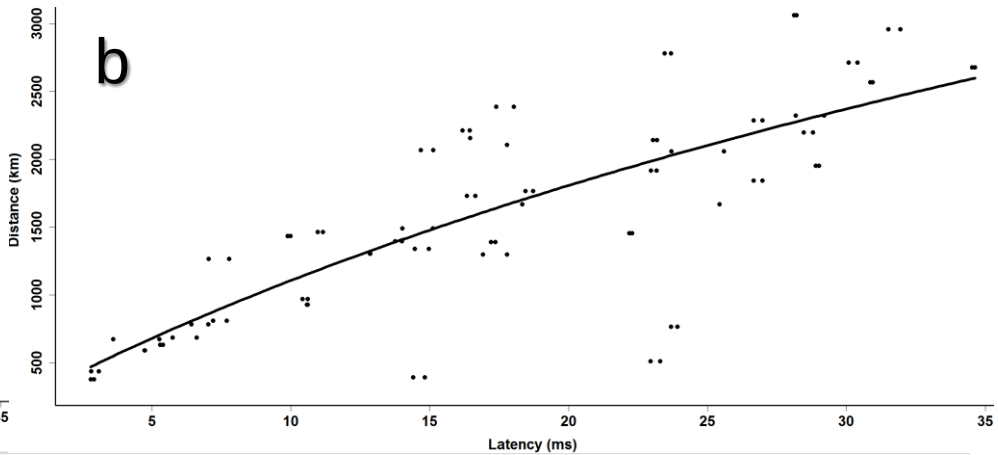
  logarithmic curve = a **· ln (** b **· Latency +** c **) +** d

- **Causes:**
  - Rough information, still imprecise model
  - Measurement deviation
  - Effect of “last mile”

2. Delay conversion (2)

- **Latency = (RTT – 0.11 ms) / 2 – 0.055 ms · hops**
  - Curve construction based on inter VP measurements
  - Reference distances:
    - Based on Earth model WGS84 (Orthodrome) → a
    - Road network by Google Maps → b
  - Minimal sum of the least squares
  - Variation of hop latency from 0.045 ms to 0.070 ms (V2=fix; V3=dynamic)

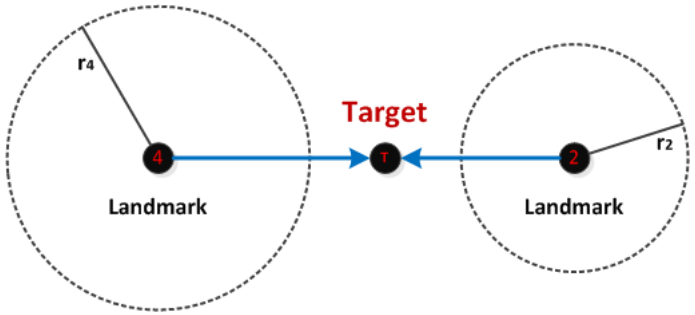




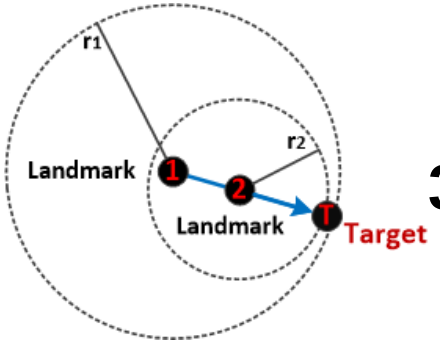
2. Lateration

- **Converted delay from Landmark to target in a distance**
  → Circle with radius and known mid point
- **Intersection point of circles estimates the target location**
  → 4 cases

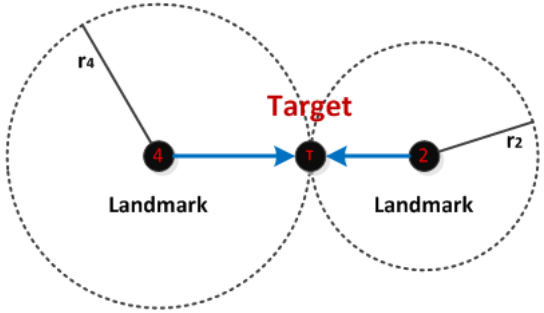
1.



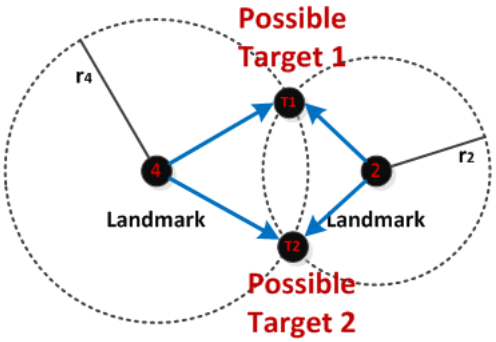
2.



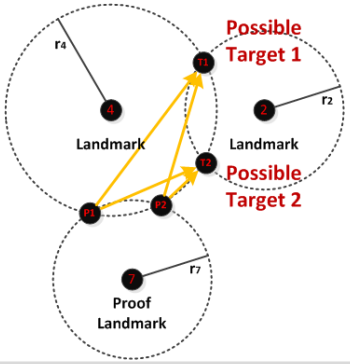
3.



4a.



4b.



2. Calculation intersection points

- X,Y: Coordinates of VP (Latitude, Longitude)
- R: radius (Distance from VP to target)

- Circle: $(X_{Target} - X_{VP})^2 + (Y_{Target} - Y_{VP})^2 = R_{VP}$

- Equation Method

$$\mathbf{Y'^2} - \frac{(a+b)*d}{c^2+d^2} * \mathbf{Y'} - R_1^2 - \left(\frac{a+b}{2*c}\right)^2 = 0 \qquad \text{(F.1)}$$

$$Y = Y' + Y_1 \qquad \text{(F.2)}$$

$$a = X_1^2 + Y_1^2 - R_1^2 - X_2^2 - Y_2^2 + R_2^2 \qquad \text{(F.3)}$$

$$b = -2(X_1 - X_2) * X_1 - 2(Y_1 - Y_2) * Y_1 \qquad \text{(F.4)}$$

$$c = (X_1 - X_2) \qquad \text{(F.5)}$$

$$d = (Y_1 - Y_2) \qquad \text{(F.6)}$$

2. Multilateration (after V4)

der Bundeswehr Universität München

- **Multiple circles:**
  - Weigthed-Least-Squares
  - Hyperbolic Positioning
- **i=1 Anchor (x,y=0)**

$$d_i^2 = (x_i - x)^2 + (y_i - y)^2$$

$$d_i^2 - d_1^2 = x_i^2 + -2xx_i + y_i^2 - 2yy_i$$

$$\underbrace{\begin{bmatrix} 2x_2 & 2y_2 \\ \vdots & \vdots \\ 2x_N & 2y_N \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\underline{x}} = \underbrace{\begin{bmatrix} x_2^2 + y_2^2 - d_2^2 + d_1^2 \\ \vdots \\ x_N^2 + y_N^2 - d_N^2 + d_1^2 \end{bmatrix}}_{b}$$

$$\hat{x} = (H^T H)^{-1} H^T \tilde{b} \rightarrow \hat{x} = (H^T S^{-1} H)^{-1} H^T S^{-1} \tilde{b}$$

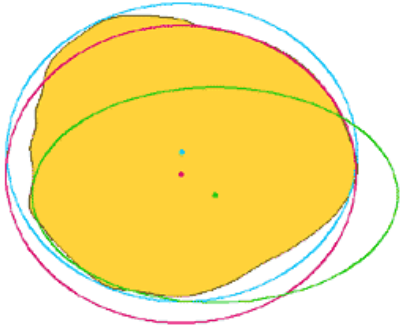
$$S = \begin{bmatrix} Var(\tilde{d}_1^2) + Var(\tilde{d}_2^2) & Var(\tilde{d}_1^2) & \dots & Var(\tilde{d}_1^2) \\ Var(\tilde{d}_1^2) & Var(\tilde{d}_1^2) + Var(\tilde{d}_3^2) & \dots & Var(\tilde{d}_1^2) \\ \vdots & \vdots & \ddots & \vdots \\ Var(\tilde{d}_1^2) & Var(\tilde{d}_1^2) & \dots & Var(\tilde{d}_1^2) + Var(\tilde{d}_N^2) \end{bmatrix}$$

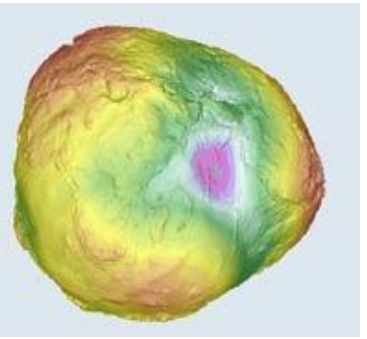
- **Problem: Difficult to filter; Good for verification**

2. Transformation of coordination system (after V2)

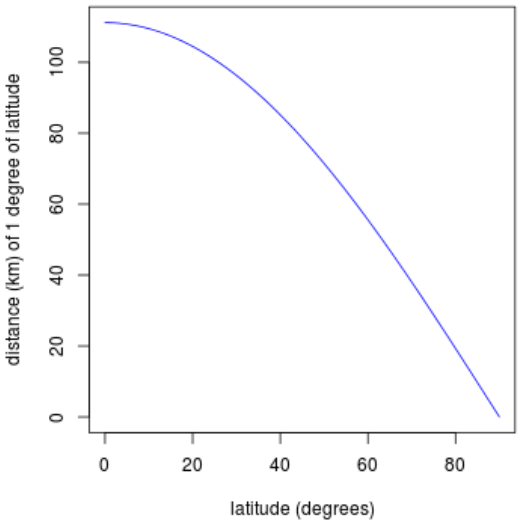
der Bundeswehr Universität München

- **The earth is not a flat disc**
  → **more like a potato** (hills, vale, sea, ..)





- **Problem:**
  - Lateration → Euclidean
  - Measurment results → Geocoordinates; Earth; degree, km

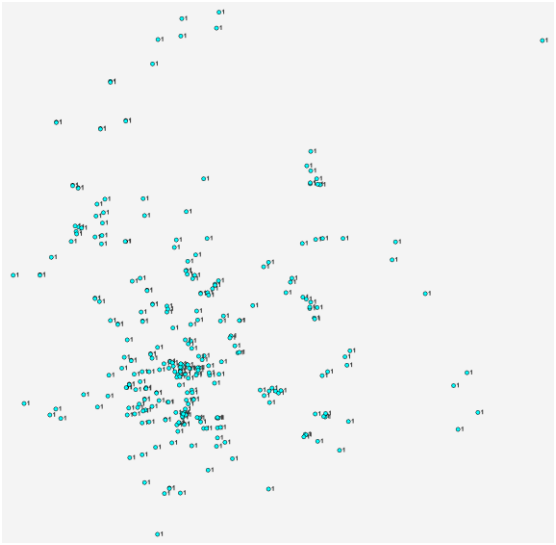


- **Spacing of 1° :**
  - Latitude: 113,325 km
  - Longitude:

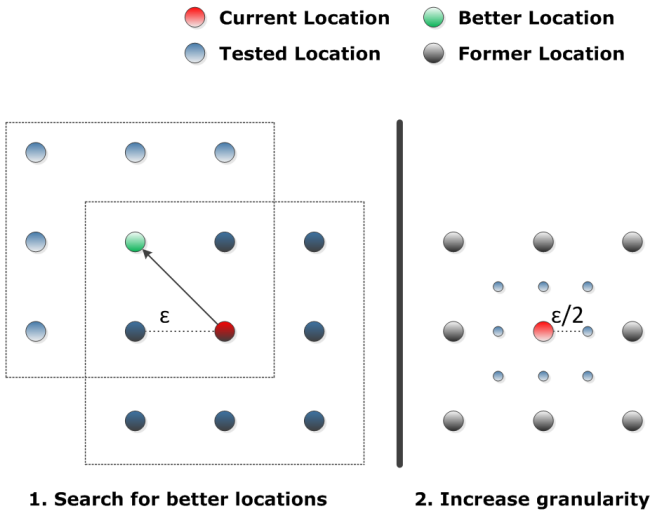
  →Lateration in uniform space of degree

2. Location estimation

- Multiple lateration result to multiple locations → Point cloud



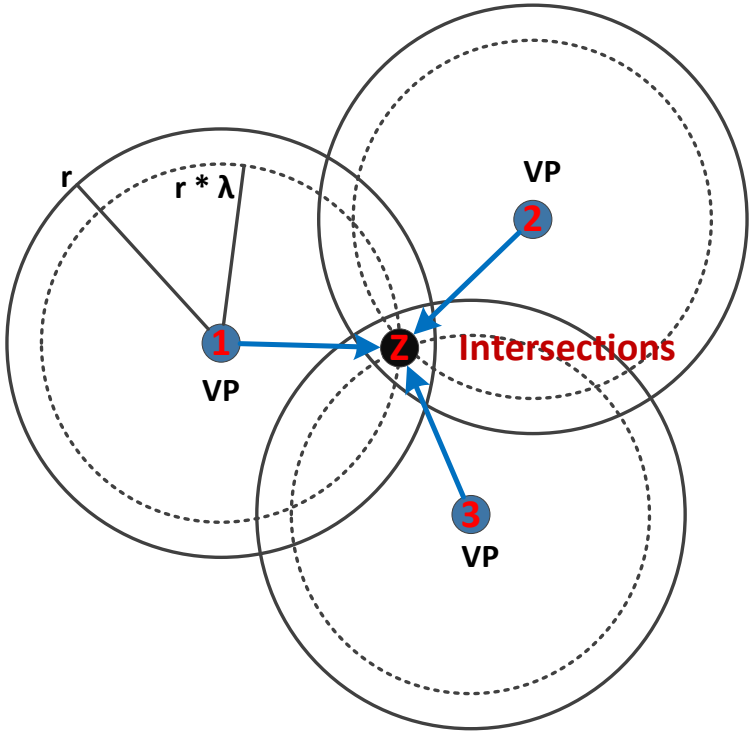
- Calculate center location with adapted Dragoon and minimized <u>average</u> distance



- Filter outliers (after V2)

2. Self-optimization (after V3)

- Latency-Distance curve is too optimistic → Overestimation (VP with fast network connection)
- Reduce all radii / lower logarithmic curve by multipy with factor < 1
- Calculate distance between intersection points
- Filtering → Identification of dense cloud



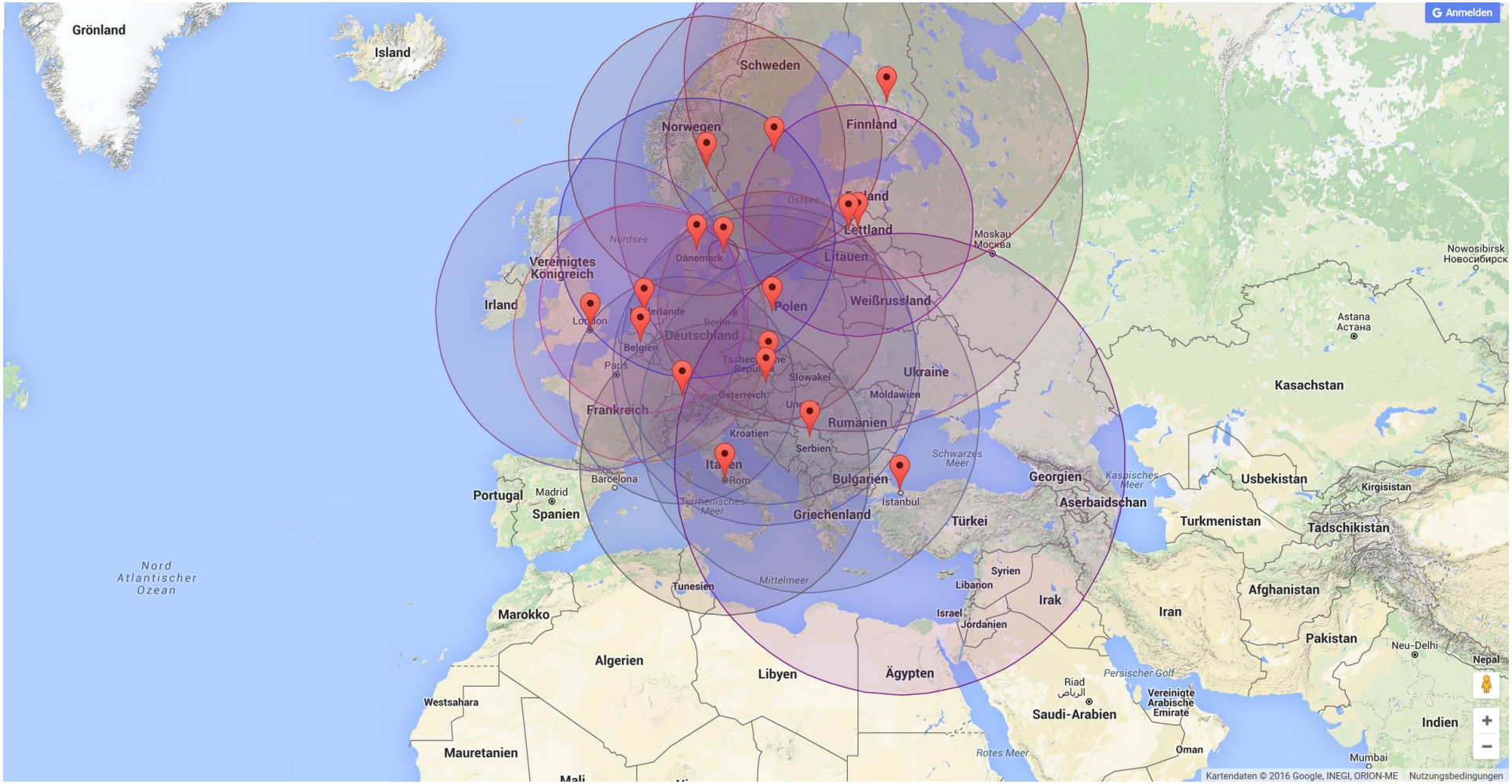
1. Start


Grönland
Island
Schweden
Norwegen
Finnland
Ostsee
Estland
Lettland
Litauen
Moskau
Москва
Nordsee
Dänemark
Vereinigtes Königreich
Irland
London
Niederlande
Berlin
Polen
Weißrussland
Deutschland
Belgien
Paris
Tschechische Republik
Slowakei
Ukraine
Österreich
Moldawien
Ungarn
Frankreich
Rumänien
Kroatien
Serbien
Italien
Rom
Barcelona
Bulgarien
Schwarzes Meer
Georgien
Kaspisches Meer
Usbekistan
Portugal
Madrid
Spanien
Tyrrhenisches Meer
Istanbul
Griechenland
Türkei
Aserbaidschan
Turkmenistan
Tadschikistan
Kirgisistan
Nord Atlantischer Ozean
Tunesien
Mittelmeer
Syrien
Libanon
Irak
Afghanistan
Marokko
Israel
Jordanien
Iran
Pakistan
Neu-Delhi
Nepal
Algerien
Libyen
Ägypten
Persischer Golf
Riad
الرياض
Vereinigte Arabische Emirate
Saudi-Arabien
Indien
Westsahara
Mauretanien
Rotes Meer
Oman
Mumbai
Nowosibirsk
Новосибирск
Astana
Астана
Kasachstan
Anmelden
Kartendaten © 2016 Google, INEGI, ORION-ME
Nutzungsbedingungen

2. Selection of VP

der Bundeswehr Universität München

3. Measurement to the target IP

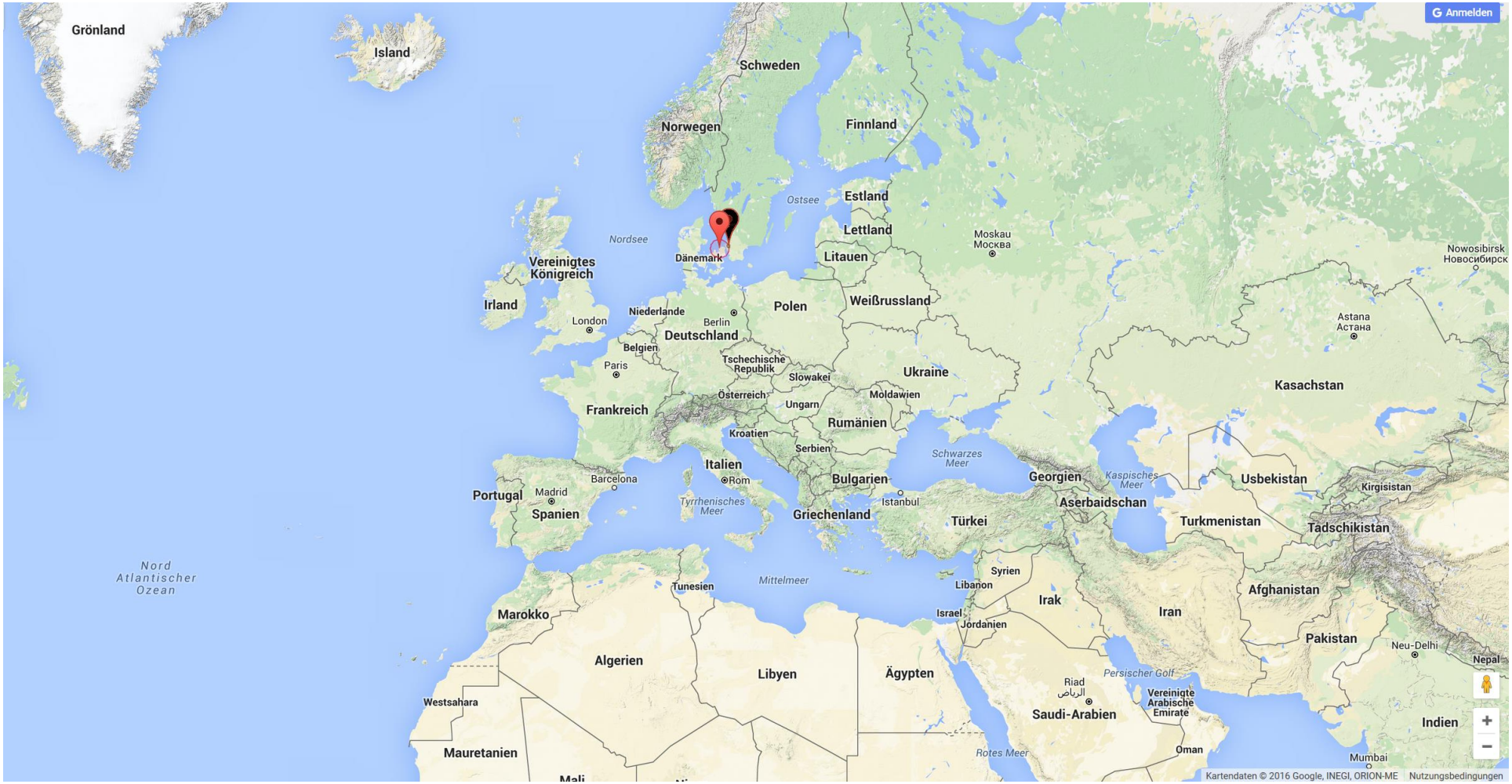


4. Aggregation of measurements

5. Determination of location

6. Target location

7. Evaluation of reality



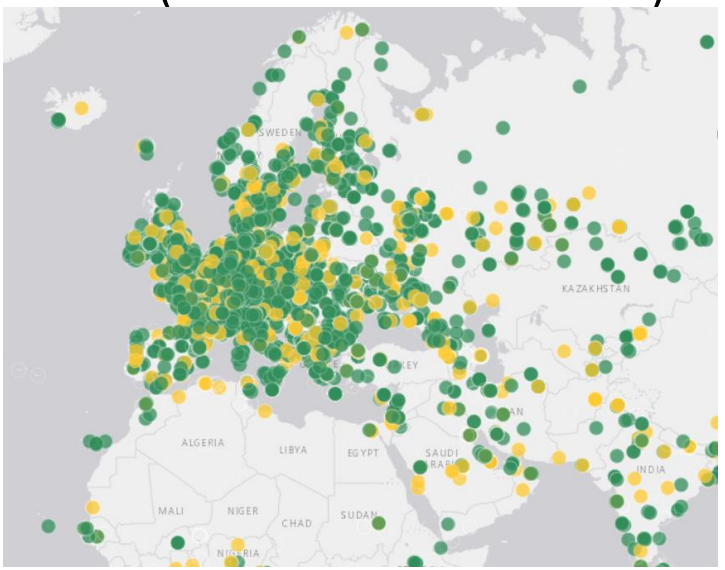
3. Experiments and Results

Experiments focus on Europe

- Optimized Landmarks matched to possible hosts provided by RIPE Atlas
- Paris Traceroute with ICMP requests
- Tool R with curve fitting method nls
- Which RIPE Node to choose? (V4 vs V5)

Anchor (Powerful – no direct match)

Probe (Small – direct match)



3. Test dataset

- **Originally more than 25 targets → stable + fixed**

- **Reduction to 16**
  - Obtained always a close estimated location
  - No comparable results from other IP Geolocation services

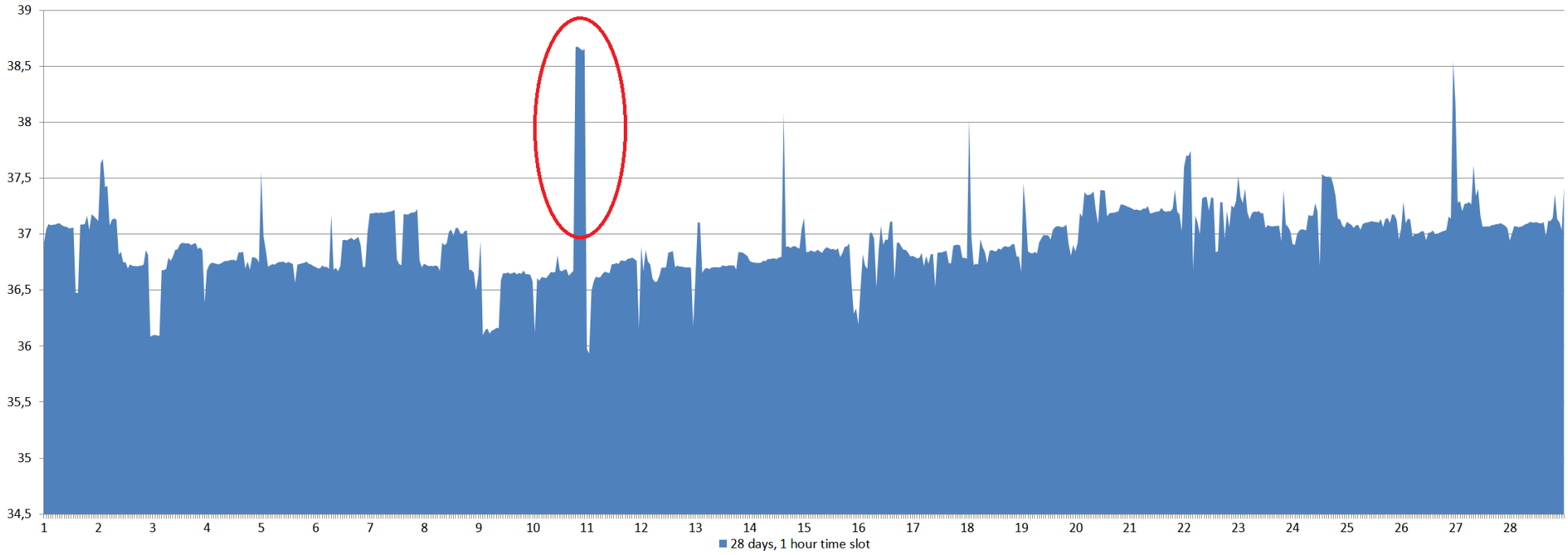
Stadt	Land	Latitude	Longitude	ASN	ID	Name	IP-Adresse
Amsterdam	NL	52,345432	4,832429	12041	6082	nl-ams-as12041	199.10.66.231
Ballerup	DK	55,728092	12,377482	39839	6103	dk-blp-as39839	193.163.102.206
Berlin	DE	52,500122	13,401483	25291	6017	de-ber-as25291	37.49.152.204
Calw	DE	48,712210	8,747244	39702	6073	de-cal-as39702	89.106.219.238
Delft	NL	51,950000	4,225000	31019	6022	nl-dft-as31019	91.228.151.10
Ettlingen	DE	48,952507	8,390881	202040	6078	de-ett-as202040	193.141.27.220
Frankfurt	DE	50,120030	8,735270	5580	6023	de-fra-as5580	80.94.66.74
Glattbrugg	CH	47,435177	8,559989	20612	6115	ch-gtg-as20612	194.242.34.190
Gothenburg	SE	57,729977	12,002948	50168	6063	se-got-as50168	185.47.192.253
Hamburg	DE	53,550000	10,048330	12731	6014	de-ham-as12731	213.128.137.33
Kiel	DE	54,366245	10,108079	13101	6052	de-kel-as13101	213.178.92.102
Prag	CZ	50,099430	14,392700	2852	6068	cz-prg-as2852	195.113.161.50
Stuttgart	DE	48,721450	9,128540	48918	6108	de-str-as48918	94.186.178.253
Turin	IT	45,091300	7,660600	12779	6011	it-trn-as12779	213.212.129.68
Wien	AT	48,208176	16,373820	1120	6042	at-vie-as1120	193.171.255.2
Zürich	CH	47,380123	8,545130	559	6015	ch-zrh-as559	130.59.80.2

3. Delay overview

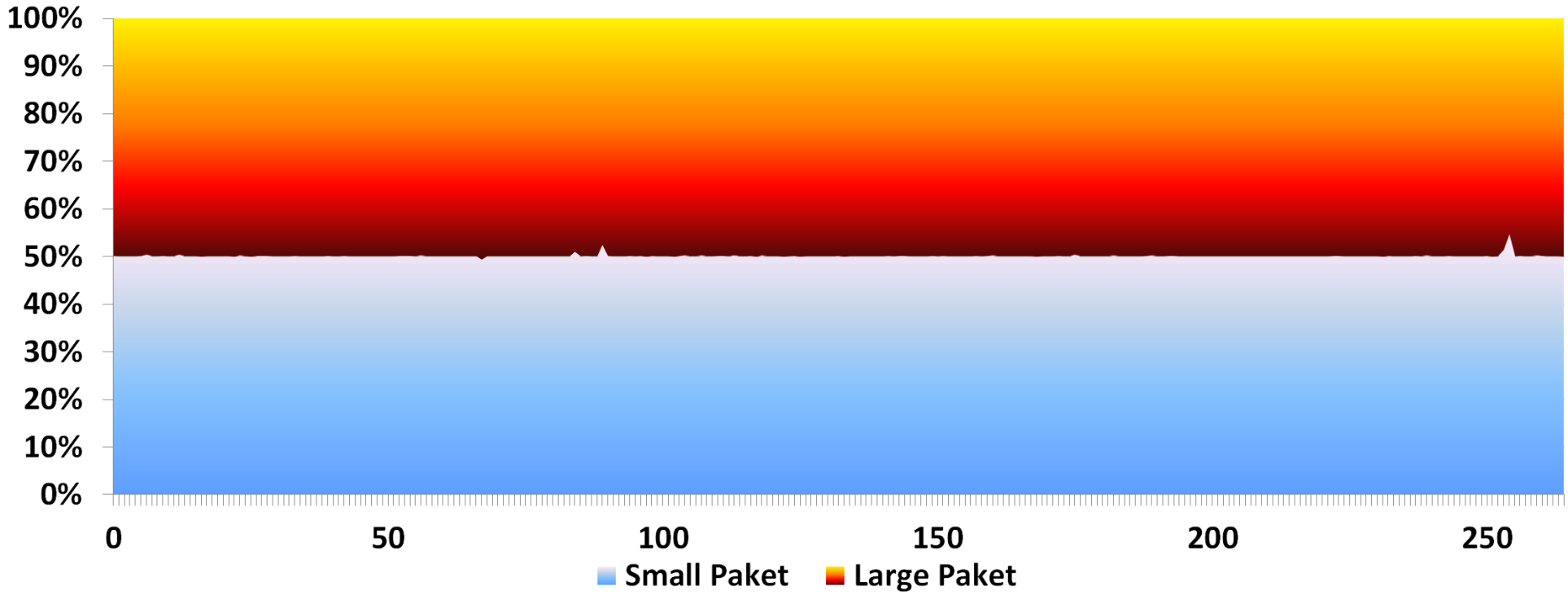
der Bundeswehr Universität München

- Impact of Mircosoft Patch Day to network load

→ Dense and fast measurments



3. Impact of paket size



- As long as no fragmentation → Impact less than 2 %
- ToS more important

3. Impact of Modelling

Universität der Bundeswehr München

Ziel	Dragoon (GM)	Dragoon (OD)	ILP (GM)	ILP (OD)	Zwei-Approx (GM)	Zwei-Approx (OD)	HYP(GM)	HYP(OD)
Amsterdam	59	4	67	799	60	60	769	840
Ballerup	48	55	109	38	71	121	339	414
Berlin	244	288	215	312	36	175	438	570
Calw	56	221	242	205	537	229	636	938
Delft	65	65	144	632	667	799	730	815
Ettlingen	126	462	225	210	318	153	743	313
Frankfurt	45	153	588	803	94	93	236	803
Glattbrugg	34	7	6	6	275	159	228	494
Gothenburg	328	120	370	333	324	470	295	333
Hamburg	16	245	226	197	266	131	361	340
Kiel	109	242	129	476	254	294	728	388
Prag	132	152	217	218	383	365	595	210
Stuttgart	322	254	718	804	666	547	247	895
Turin	23	268	17	223	851	544	381	674
Wien	0	0	251	180	177	82	605	240
Zürich	7	1	260	297	277	146	287	306
⌀ Abw.	101	159	237	358	329	273	476	536

GM…Google Maps

OD…Orthodrome

ILP…Integer Linear Programming

HYP…Hyperbolic mit LM nach Dragoon

3. Impact of number of Landmarks

Target	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Avg.
20 LM	0	3	11	23	28	38	43	87	95	148	176	194	205	268	268	307	390	134
30 LM	214	77	198	87	169	70	165	143	353	135	198	243	207	155	107	222	277	178

- **Amount of Landmarks has an impact.**

→ More precise measurment results is more important !!!

3. Comparision to others

der Bundeswehr Universität München

Ziel	Model	Whois (P)	MaxMind (P)	Spotter (A)	CBG (A)	TBG (A)	Geoplugin (A)	FreeGeoIP (A)
Amsterdam	59	*5481*	*5474*	477	172	163	5390	5389
Ballerup	48	*14*	*13*	738	806	795	13	22
Berlin	56	*2*	*341*	401	866	866	347	347
Calw	244	*1*	*13*	46	46	46	13	6
Delft	65	*16*	*65*	495	205	205	65	65
Ettlingen	126	*0*	*1*	111	4	4	1	1
Frankfurt	45	*379*	*370*	136	101	101	366	366
Glattbrugg	34	*1*	*64*	221	109	109	64	64
Gothenburg	328	*69*	*402*	951	1063	1015	396	396
Hamburg	16	*24*	*3*	475	586	586	3	3
Kiel	109	*453*	*381*	567	676	771	382	382
Prag	132	*12*	*2*	335	459	459	2	2
Stuttgart	322	*28*	*254*	11	64	64	6	254
Turin	23	*433*	*477*	411	483	483	485	485
Wien	0	*14*	*1*	478	481	481	1	1
Zürich	7	*24*	*2*	193	179	179	2	2
⌀ Abw.	101	*434*	*492*	378	394	395	471	487
⌀ Abw.*	104	*98*	*159*	371	408	411	143	160

P…Passive

A…Active

Abw…Average Deviation

Abw*…Average Deviation without Amsterdam

4. Improvements

- **Determination of the subnetwork paths**
  - Using precise network infrastructure and routing information
  - Looking Glass

  →More exact length of the travel distance of the signal

- **Locate nodes on the path**

- **Detection of Proxy, VPN, …?!**

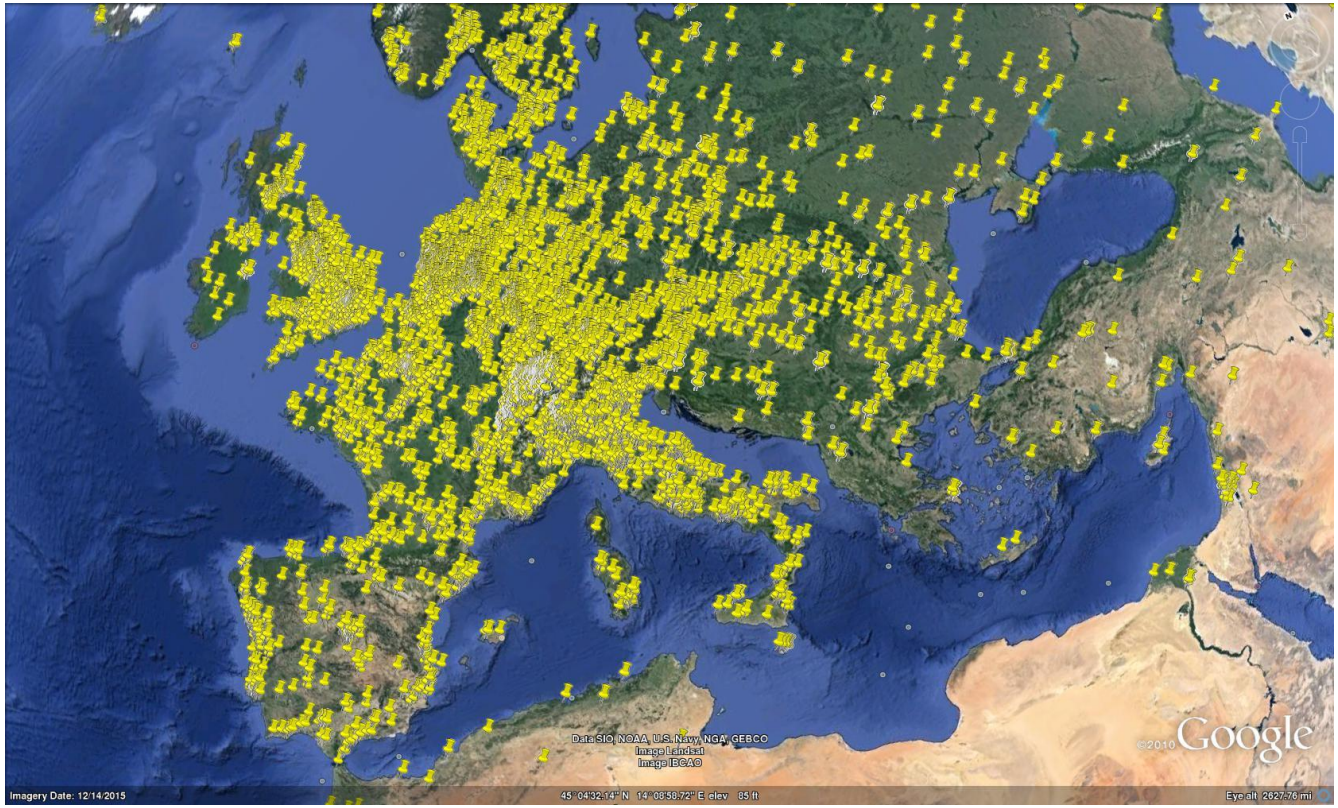
- **Filtering of strange measurement results**

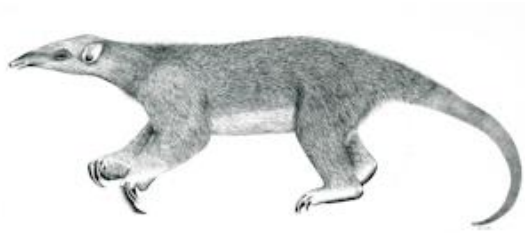
4. Use cases

- Heatmap
- Conent Delivery Networks
- Targetted advertisement
- Prosecution

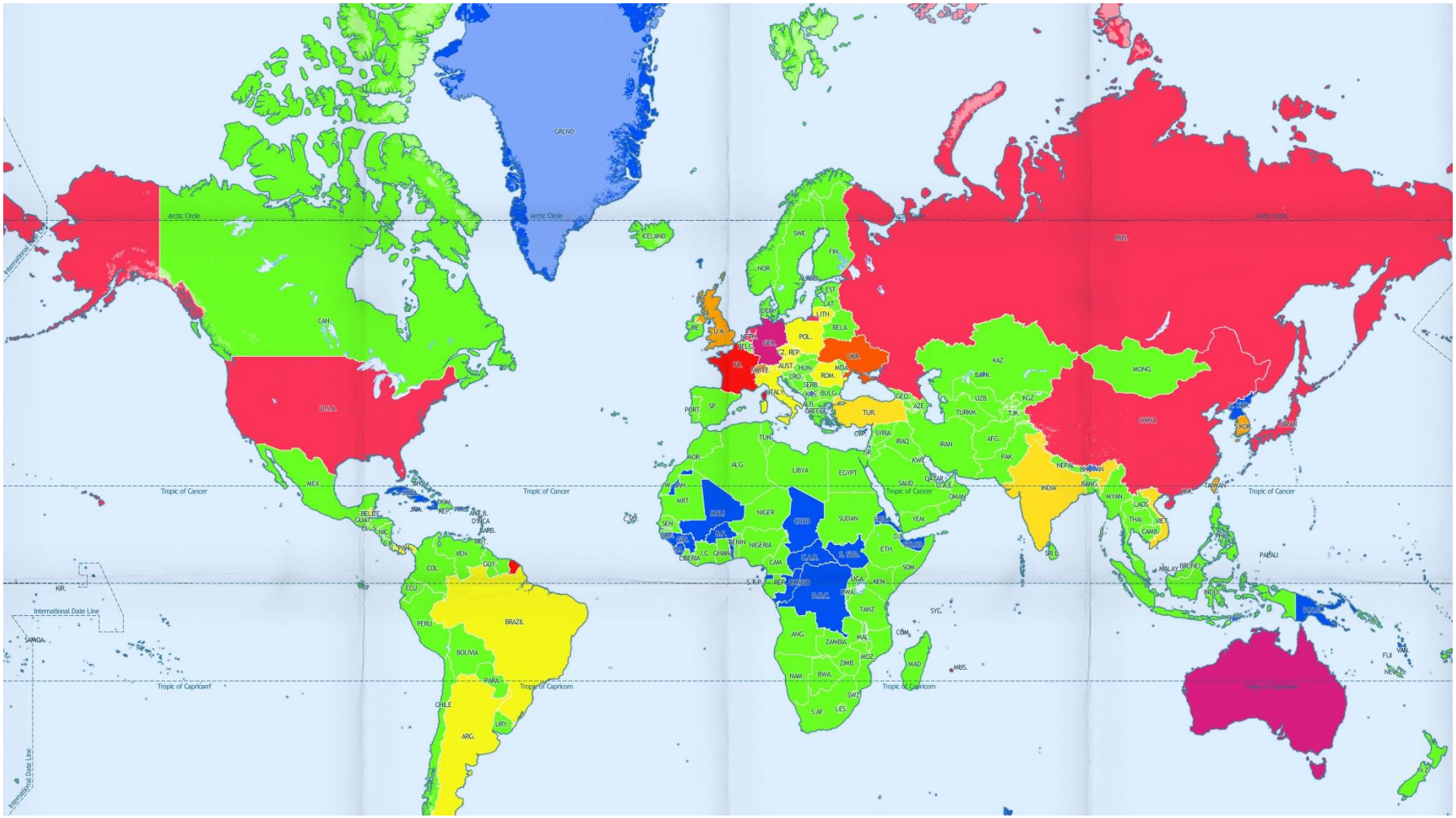
4. Tranalyzer – Traffic Labeling

der Bundeswehr Universität München





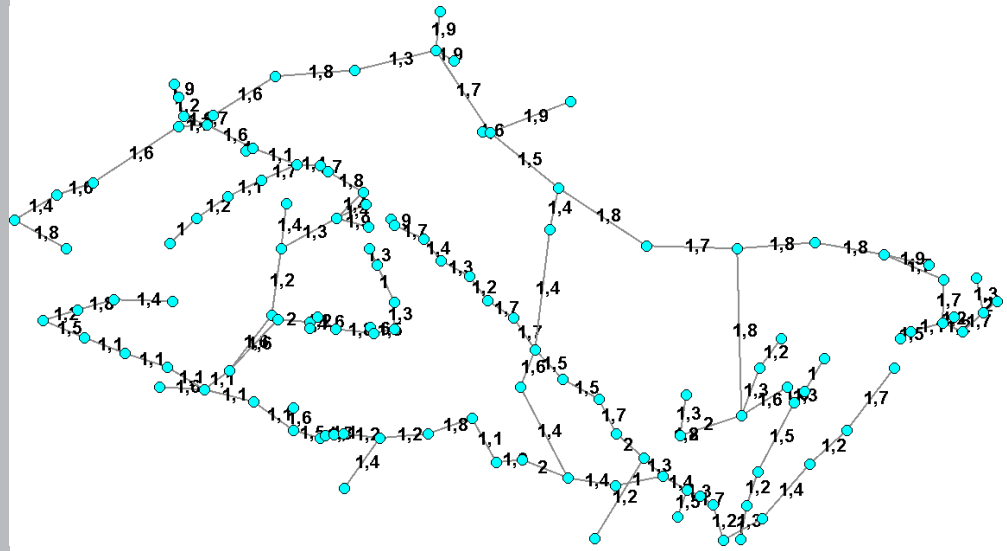
4. Heat-Map of Attacks

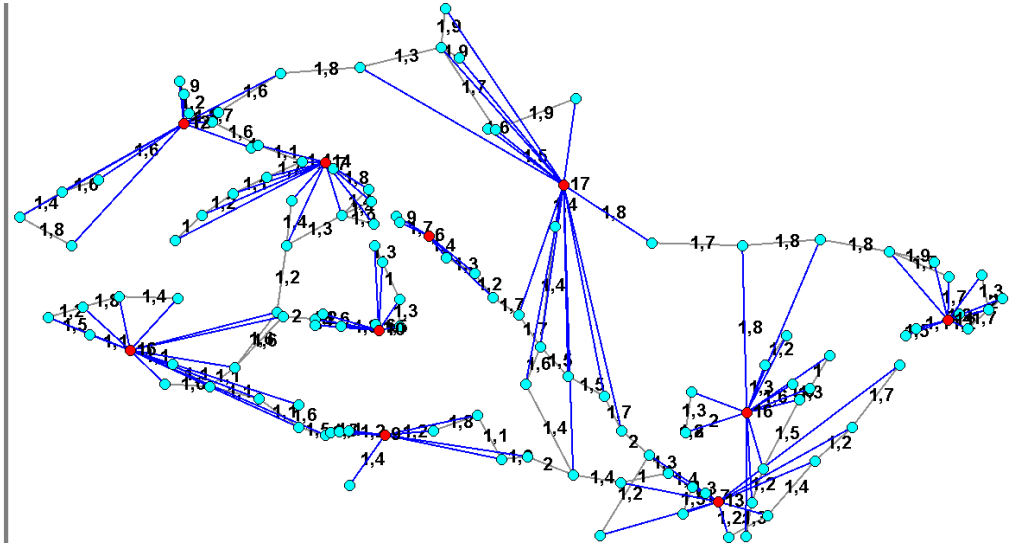


4. CDN – Load Balance

der Bundeswehr Universität München

Backbone Network of Dialtelecom





Summary

- **Show correlation of internet latency and distance**
- **Improved Landmark Selection → Algorithm Dragoon**
- **Precise Model for Geolocation**
  - Improved distance approximation and estimation
  - Road network to estimate Internet network distances
    → GoogleMaps instead of orthodromic distances
  - Transformation for Lateration
  - Self-optimization
- **Realisitic measurement results → Real-World environment**

Thank you for your interest.

Questions?