Lecture 14

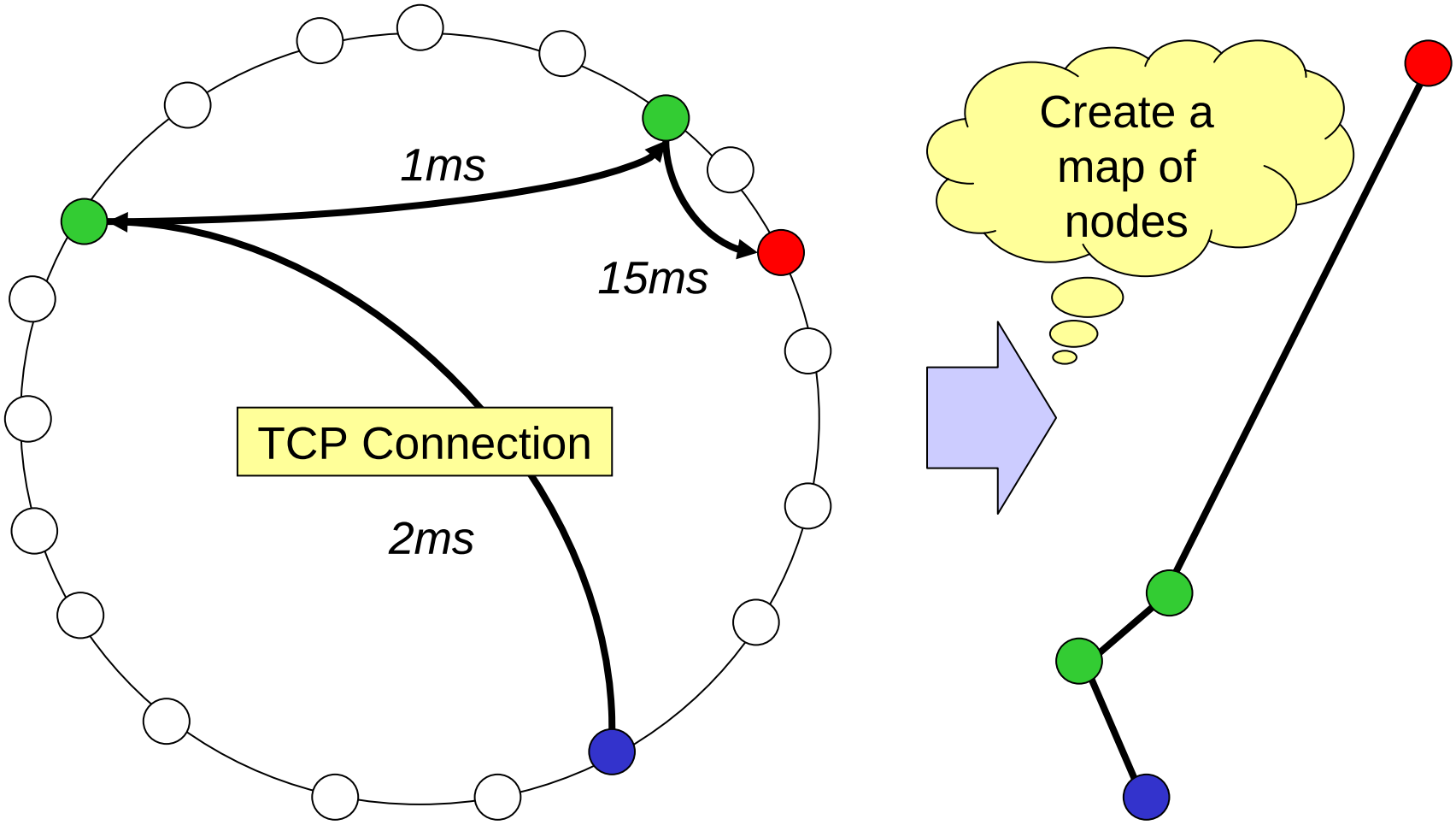
Internet Protokolle II

Proximity Awareness
& Network Coordinates

Thomas Fuhrmann

Network Architectures
Computer Science Department
Technical University Munich

Network Coordinates



Overview of Today's Lecture

Problem definition:

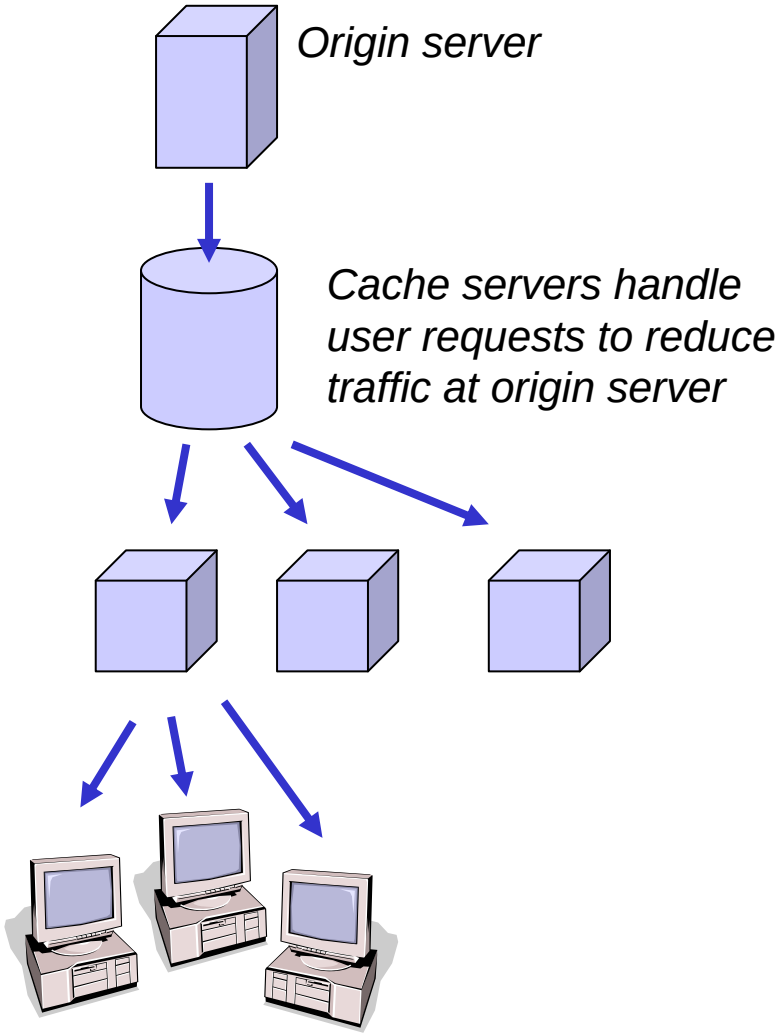
- Whom shall a peer make its direct neighbor?
- Which neighbor shall it ask for a piece of content?

Proposed solutions:

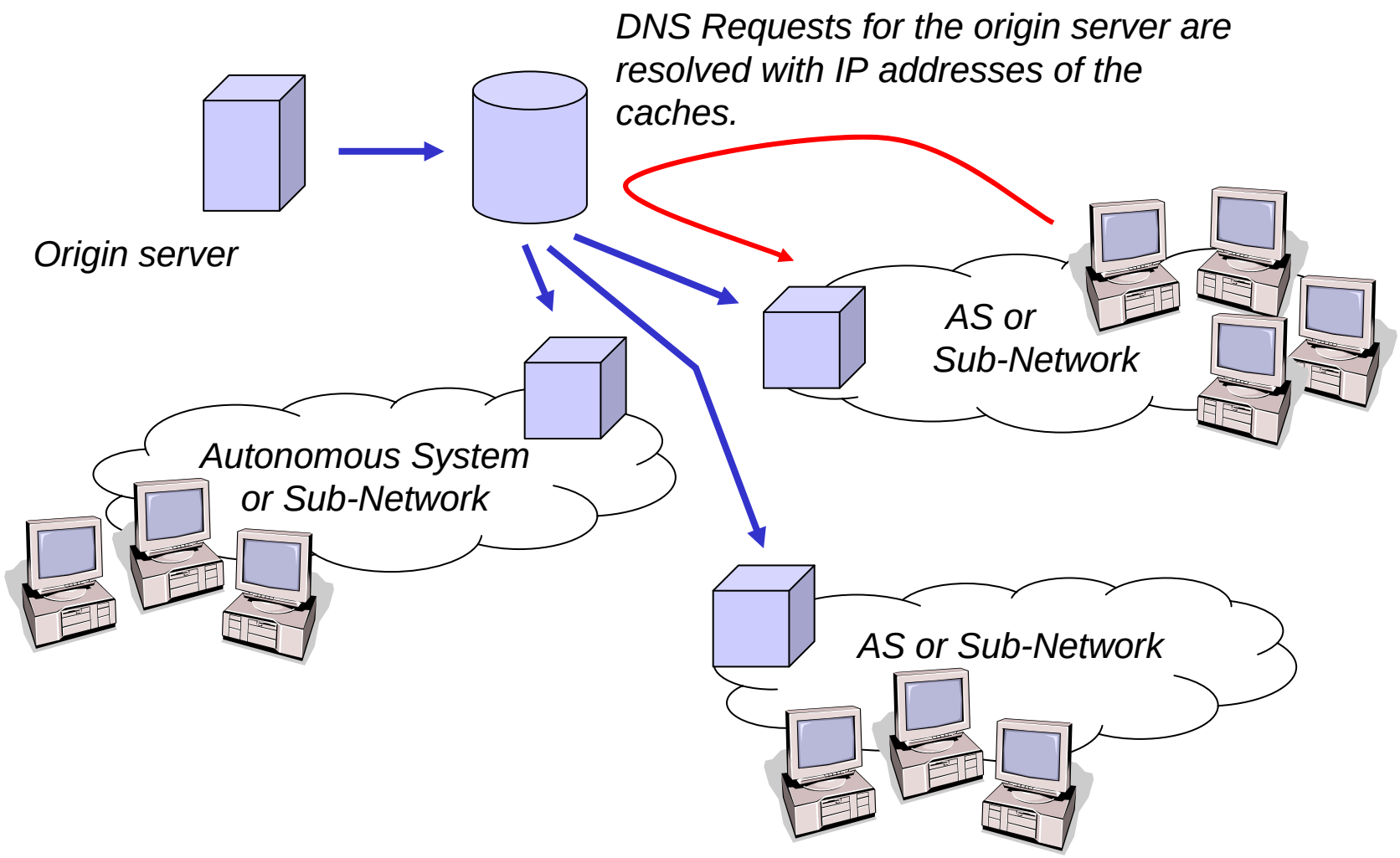
- Content Distribution Networks
- Network coordinates
  - GNP
  - Vivaldi
- Optimizations of Vivaldi's network coordinates
- Meridian
- Ono

Content Distribution Networks

- Zu viele Anfragen für ein Cluster
- Je nach Ort zu große Latenz
- Caches 'nahe' bei Benutzern aufstellen
- Transparent durch DNS Umleitungen
- Anbieter z.B.: Akamai, Limelight
- Kunden sind fast alle großen Inhaltsanbieter wie Yahoo, BBC, CNN, …
- Bestimmung der Position: Exzessives Messen *im* Cluster



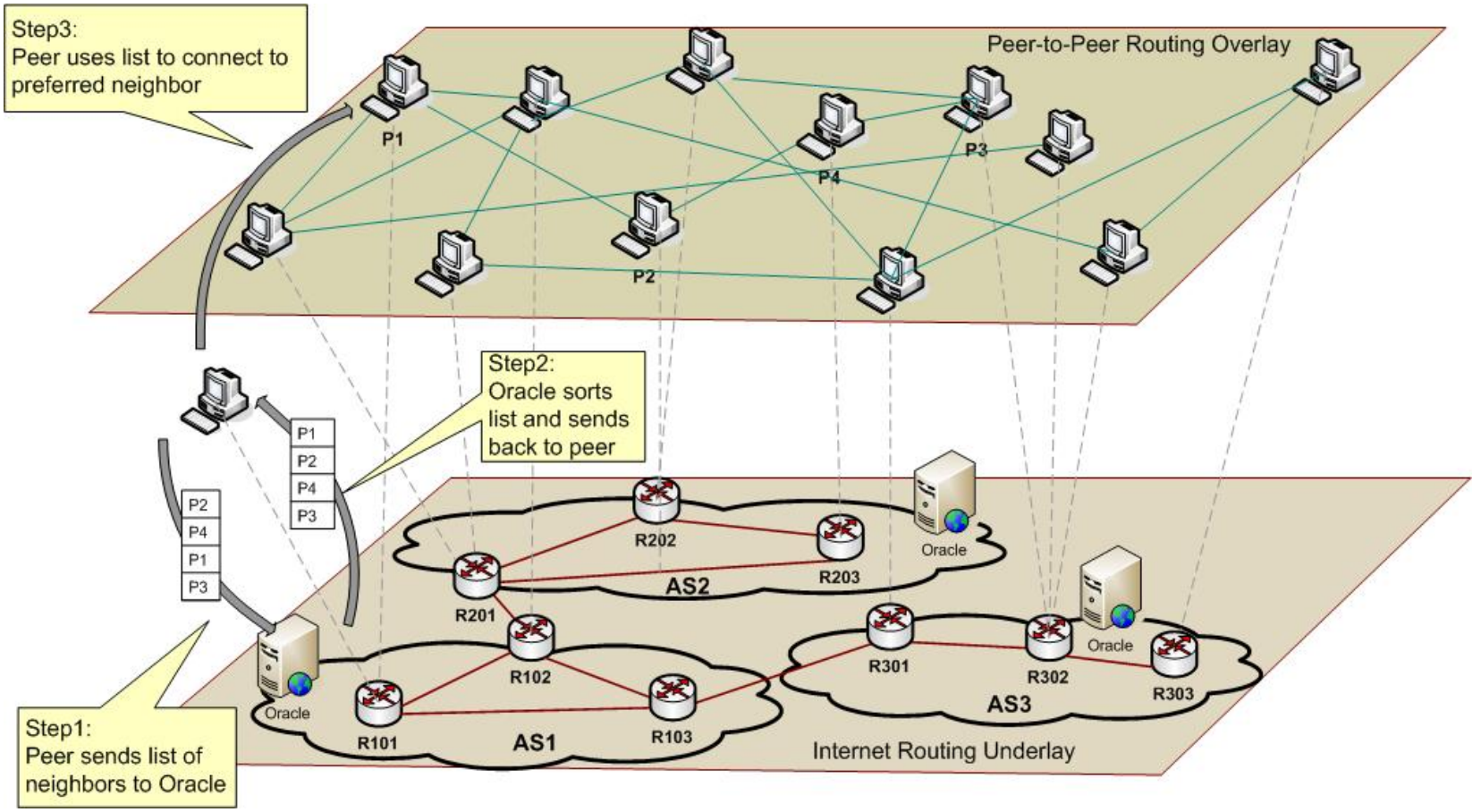
Content Distribution Networks



Oracle Service (TU Berlin)

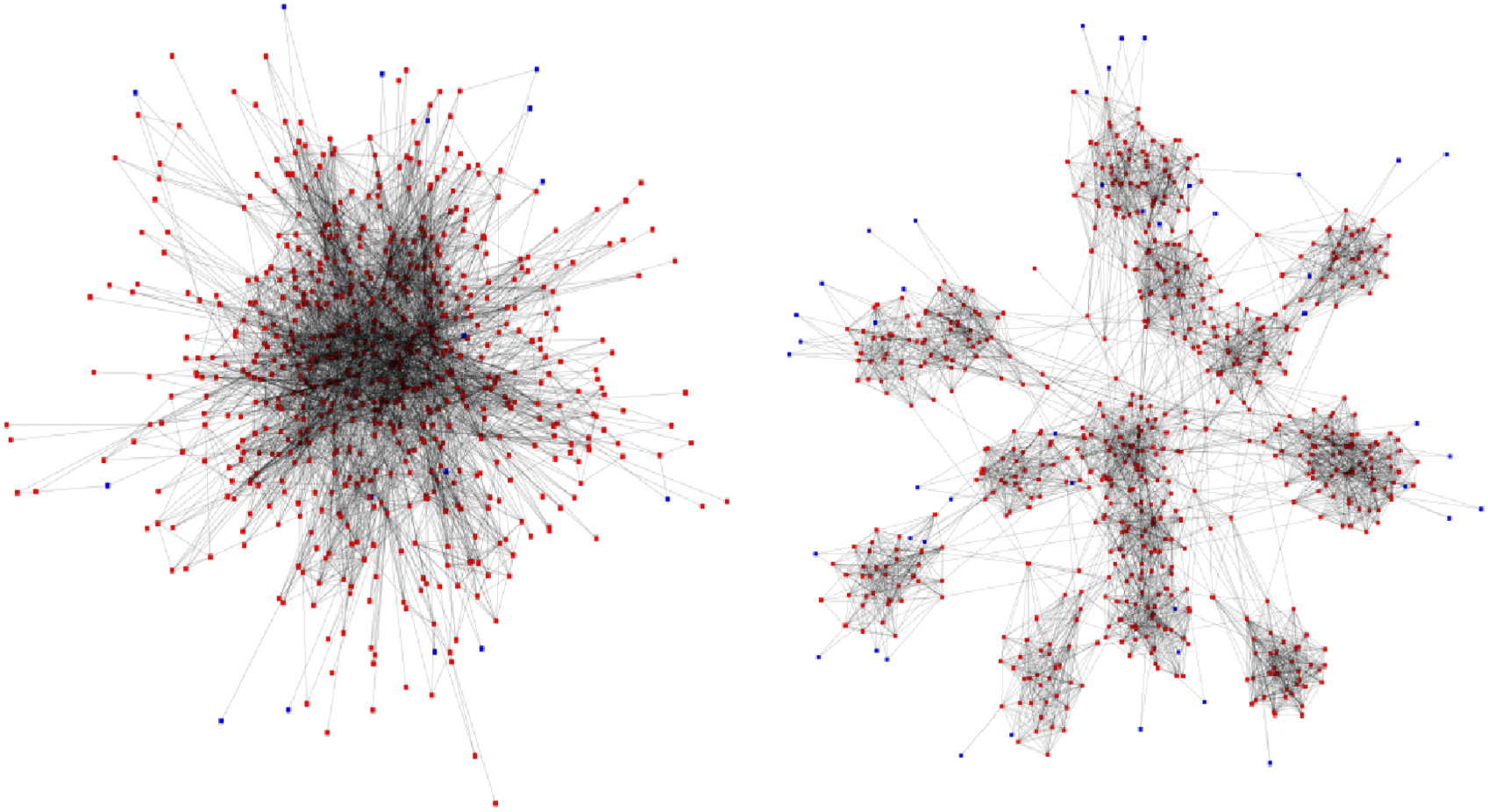
- - Idee: ISPs kennen ihr Netz
  - Bandbreite, QoS Parameter jeder Node
  - Routing innerhalb Ihres Netzes ist ISP Geheimnis
- Idealer Partner um ein P2P Netz “richtig” zu bauen
- Oracle Service:
  - ISP stellt den Dienst zur Verfügung
  - P2P Applikation nutzt diesen:
    - Eingabe: Liste von möglichen Peers
    - Ausgabe: Gerankte Liste von Peers
- Ranking ist beliebig wähl- und erweiterbar

Oracle Service II

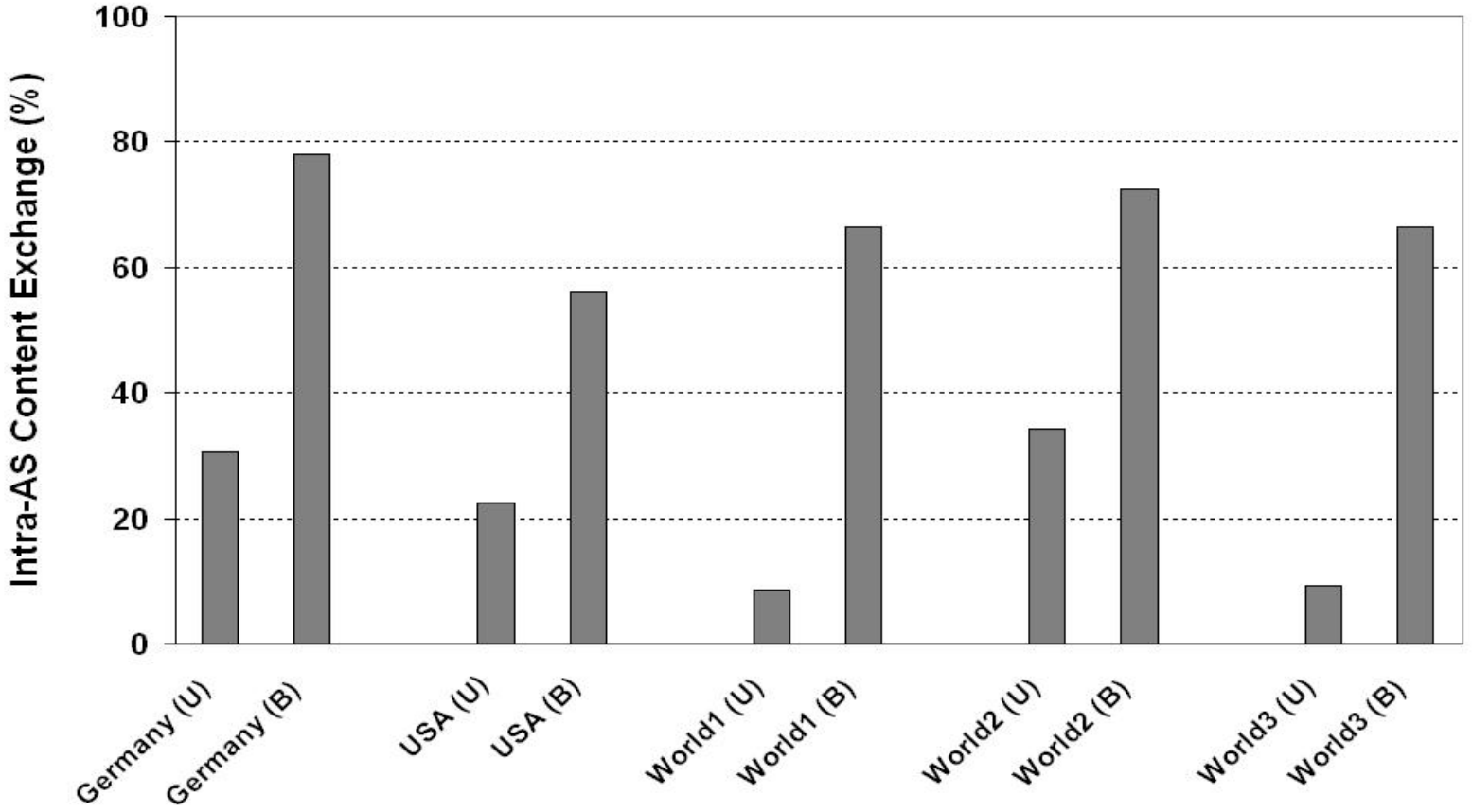


Vinay Aggarwald, Anja Feldmann, ISP-Aided Neighbor Selection for P2P Systems, IETF 2008

Oracle Random vs Oracle Gnutella



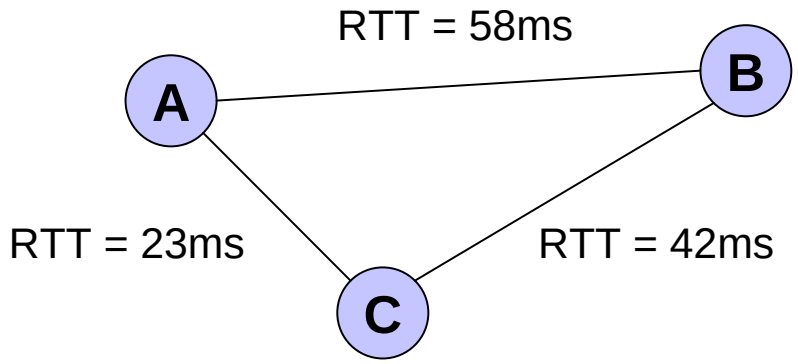
Oracle Service Results



Bis zu 80% content wird im AS gehalten

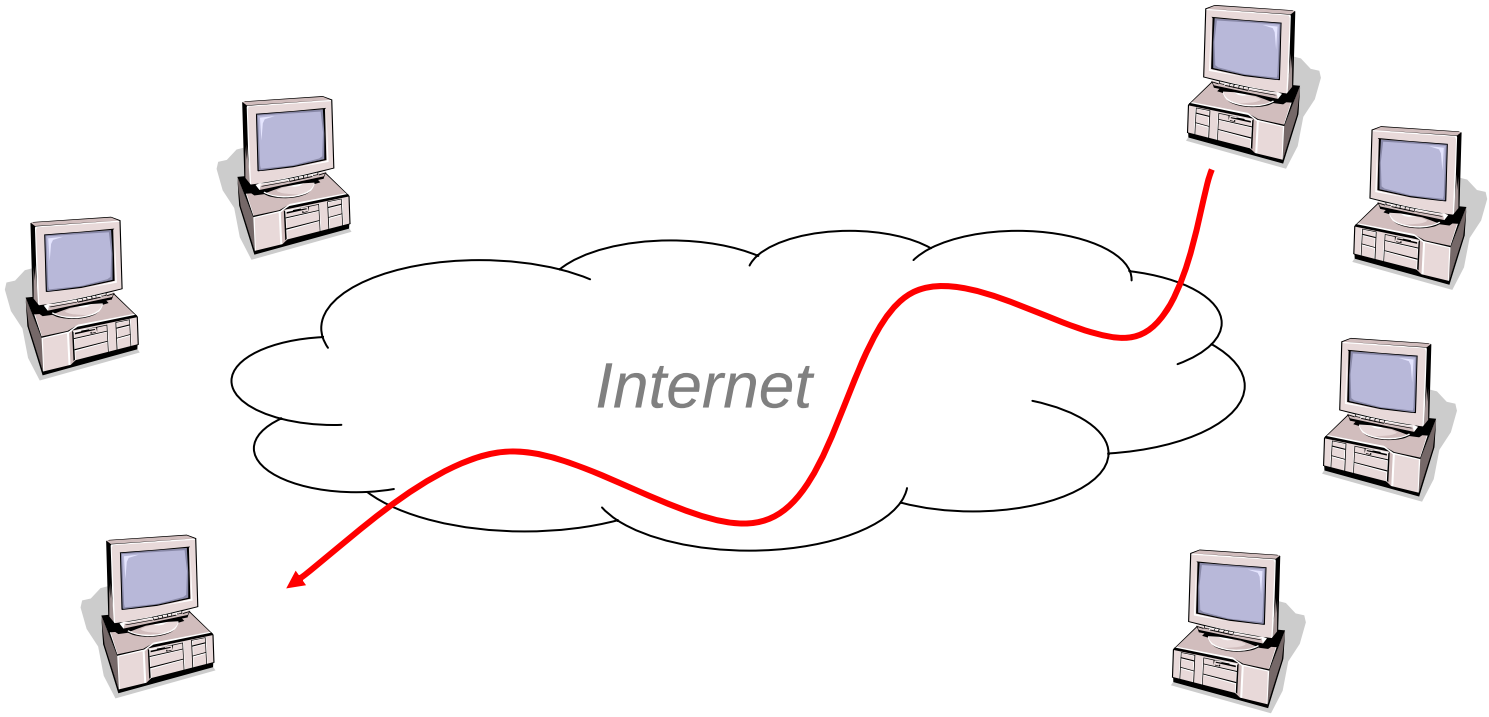
Einbettungen des Internets

- Frage: Wie finde ich physisch nahe Peers? Muss ist z.B. im Overlay fluten (vgl. Gnutella), oder gibt es eine Karte, auf der ich nachsehen kann?
- Idee: Finde eine Abbildung der Rechner auf einen metrischen Raum, so dass der Abstand dort dem gemessen Abstand im Internet entspricht.
- Beispiel:



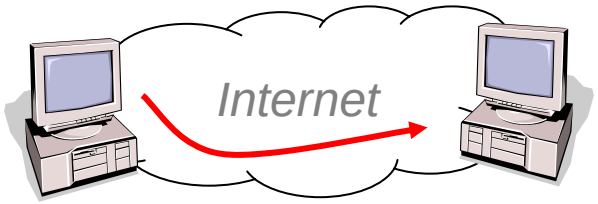
- Achtung: Eine solche Einbettung kann i.a. nur eine Näherung sein!
- (Gegen-) Beispiel: Durchsatzmessung an einer DSL-Leitung ergäbe Metrik die Symmetrie verletzt!

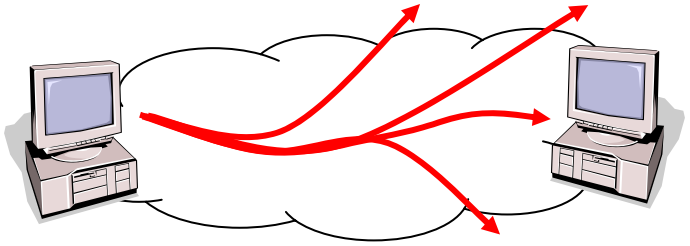
Network Tomography



In overlays, we only have end-to-end measurements.
What can we nevertheless learn about the Internet‘s topology?

Network Tomography



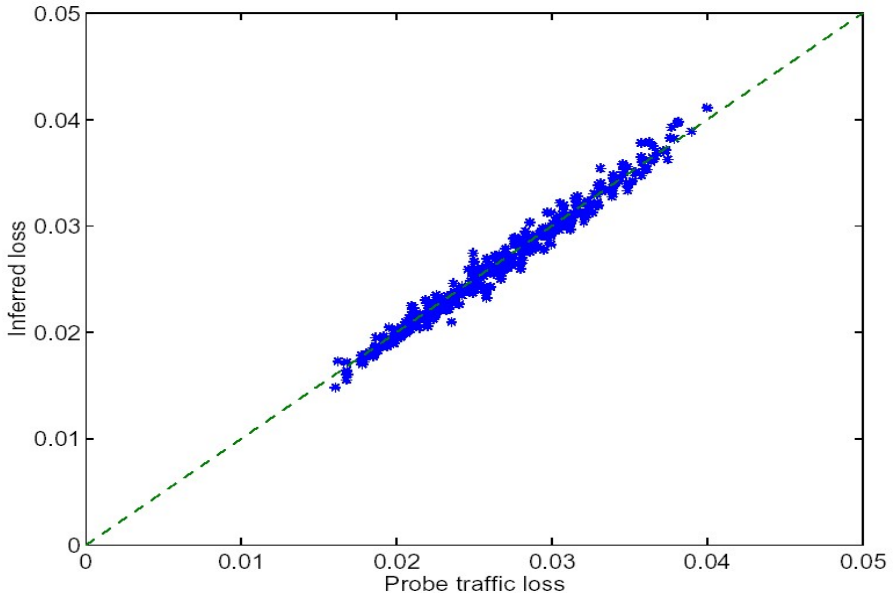


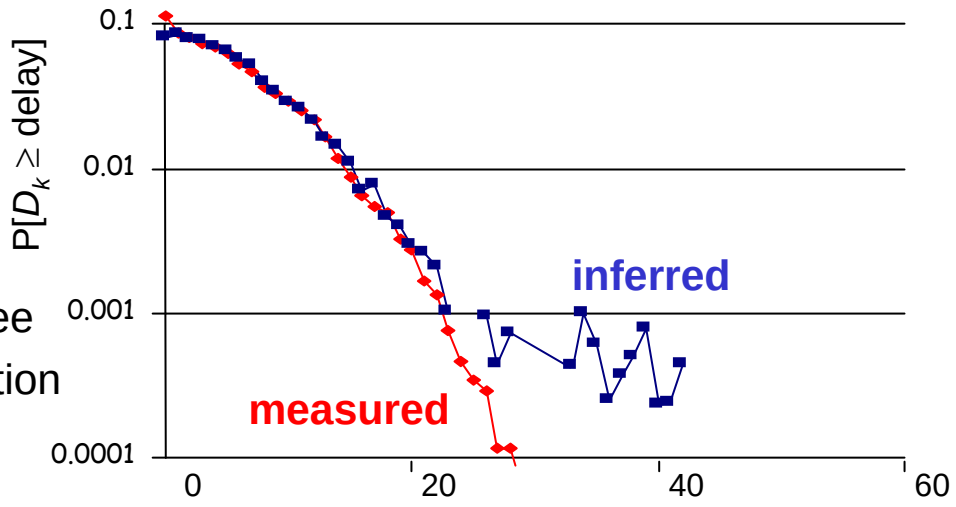
- Unicast probes
  - Infer path characteristics, such as loss rate, delay, available bandwidth.
- Pair-wise probing:
  - Send back-to-back packets sent to different receivers.
  - Correlate the probes among all the pairs.
- Needs many pairs to cover network.
- Correlation may erroneously indicate overlapping paths where just the traffic patterns have similarities.
- Improve quality by using “stripes”, i.e. long sequences of back-to-back packets.

- Multicast probes
  - Reduce required amount of traffic
  - Improve correlation: Joint path segments have been probed by the very same packet!
  - Thereby increase network coverage, and ..
  - … the number of links that can be identified.
- Infer network topology from several independent multicast probes:
  - Correlate loss and delay within each multicast tree
  - Infer the characteristics for the segments between the branching points.
  - Create a logical tree for each multicast group.
  - Correlate these trees to obtain the network topology.
- Assume …
  - Independent loss and delay to allow spatial correlation
  - Stationary behavior to allow temporal correlation

Network Tomography

- Maximum Likelihood
  - Calculate L(X;a) the likelihood of the receivers' observations X
  - Estimate a = argmax_a L(X;a)
- Examples:
  - Loss probability with closed form expressions
  - Delay distribution with EM algorithm
  - Topology with exhaustive search, Markov Chain, or Monte Carlo
- MLE Properties
  - asymptotic consistency
  - asymptotic normality
  - asymptotically efficient
- Bayesian, e.g. for multicast tree
- Greedy Heuristics, e.g. for multicast tree
- Method of Moments, e.g. delay correlation

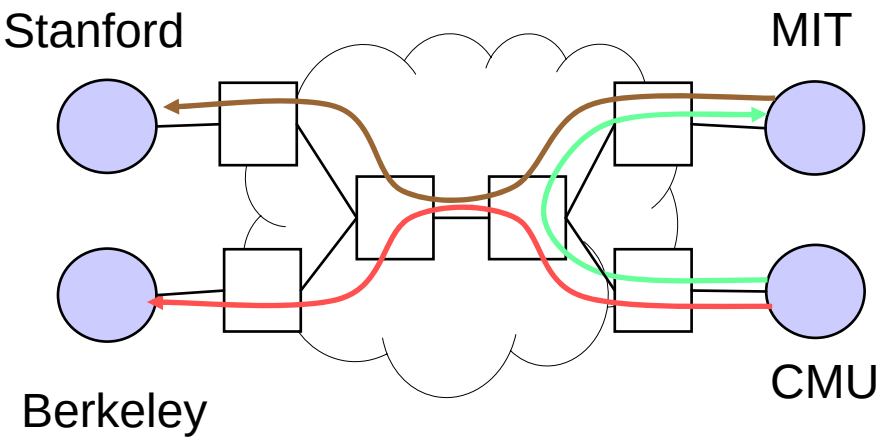


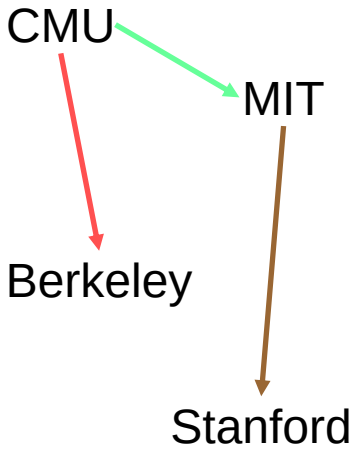


Source: Francesco Lo Presti

Network Tomography & Peer-to-Peer

- Large-scale distributed services and applications such as Napster, Gnutella, End System Multicast, etc
- Large number of configuration choices
- On-demand network measurement can be highly accurate, but
  - Not scalable: K participants $\Rightarrow$ $O(K^2)$ e2e paths to consider
  - Slow: Need to route a request w/o probing the potential alternatives first.
- Network distance
  - Round-trip propagation and transmission delay
  - Relatively stable
- Network distance can be predicted accurately without on-demand measurement
  - Fast and scalable first-order performance optimization
  - Refine as needed



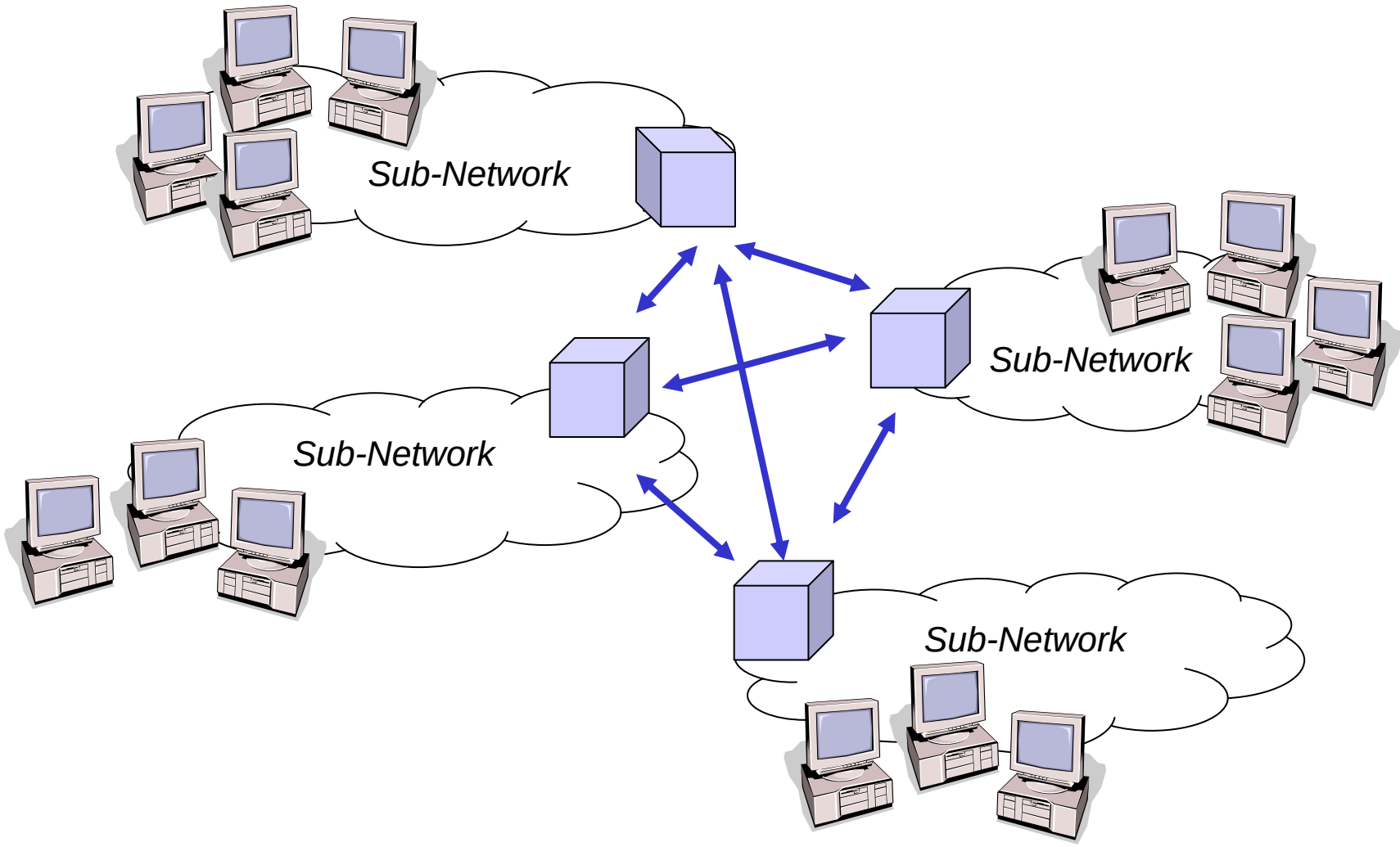


Source: T. S. Eugene Ng and Hui Zhang

RTT Prediction with IDMaps

- IDMaps [Francis et al. 2001]
  - Pioneer work about RTT prediction in the Internet
  - A global service for RTT estimation
  - Estimation of the RTT using triangulation
  - Proactive measurements
- Ingredients:
  - Address Prefix (e.g. sub-network): Consecutive address range of IP addresses within which all hosts with assigned addresses are equidistant (with some tolerance) to the rest of the Internet.
  - Tracer: A host deployed in the access network. The tracer measures the network distance to all other tracers in the Internet.
  - Virtual Link: A raw distance between two tracers (Tracer-Tracer VL) and between a tracer and an AP (Tracer-AP VL).
- Deployment needs infrastructure support: One tracer must be deployed to each access sub-network.
- Poor scalability:
  - Each tracer measures and stores RTT to all other tracers.
  - Thus the complexity of storage and measurement traffic generation grows quadratic with the number of tracers deployed in the Internet - $O(n^2)$.

RTT Prediction with IDMaps

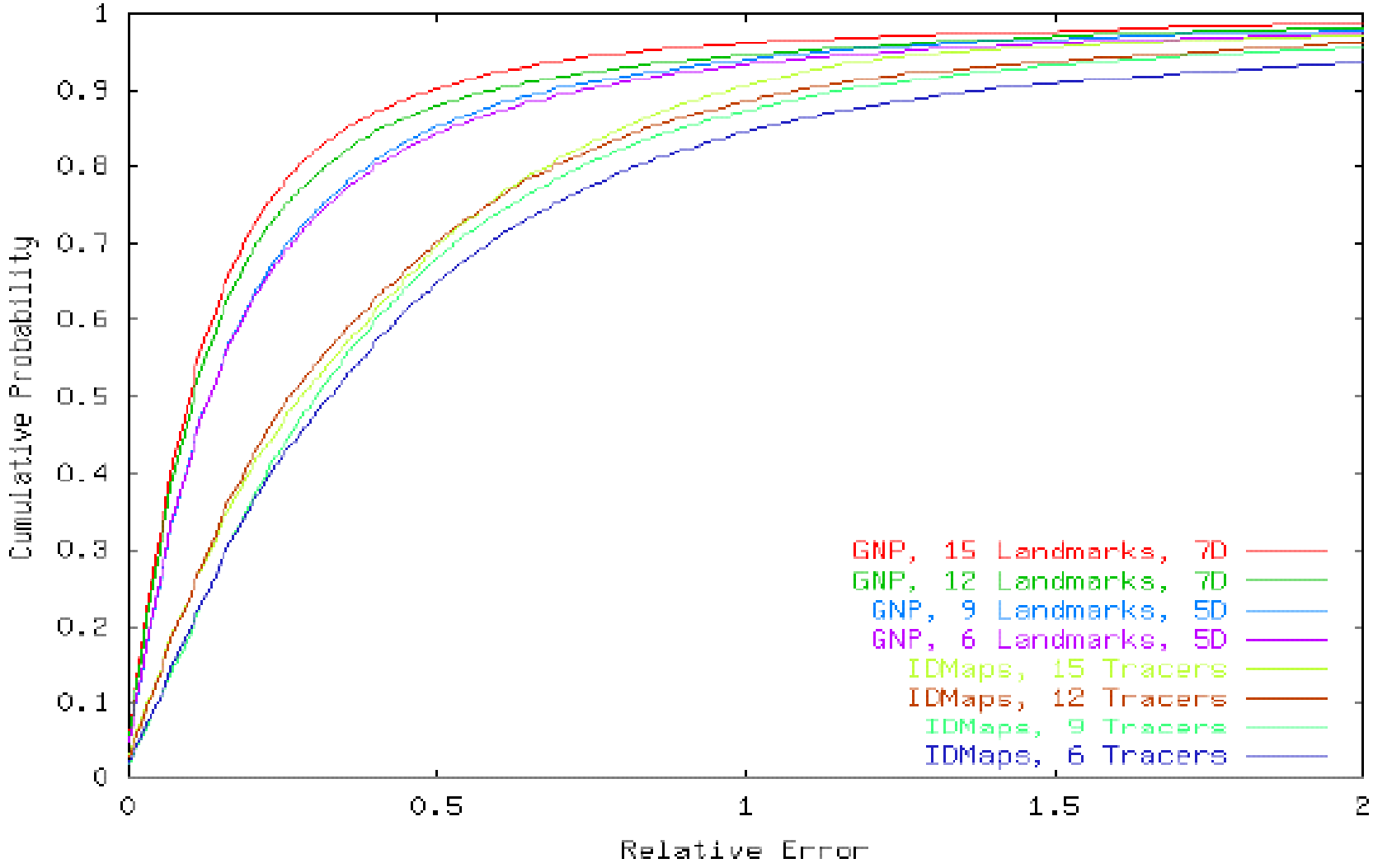


Global Network Positioning

- Einige wenige Server (=Landmarken) werden im Internet verteilt und weisen sich Koordinaten zu.
  - Die Landmarken verwenden ihre geographische Position.
  - Die Landmarken messen ihre RTT untereinander und berechnen daraus näherungsweise Koordinaten (Global Network Positioning).
- Clients, die ihre Position im Netz ermitteln wollen …
  … erhalten den vollständigen Satz der Landmarken inkl. deren Koordinaten: (L_1: x_1,y_1; L_2:x_2, y_2;…L_n: x_n,y_n), und …
  … pingen diese Landmarken um so ihre eigene Koordinate zu berechnen.
- Landmark Server und Clients müssen also ihre eigene Koordinate einem Optimum annähern: *Global minimization problem*

T. S. Eugene Ng and Hui Zhang. Towards global network positioning. SIGCOMM Workshop on Internet Measurement , 2001

GNP Anzahl der Landmark Server



Vivaldi

- Vivaldi bettet das Internet anhand der paarweise gemessenen RTT in einen *n* dimensionalen euklidischen Raum ein.
- Ziel ist es durch Messungen einer **Teilmenge** aller Hosts die Umlaufzeiten zu den Restlichen vorhersagen zu können.
- Grundlage für den Algorithmus ist ein Feder-Modell:
  - Zwei Knoten tauschen ihre Koordinaten aus und messen dabei die RTT
  - Aus der Koordinatendifferenz und der gemessenen RTT folgt die wechselseitige Koordinatenabweichung
  - Diese Abweichung wird teilweise korrigiert
  - Ziel des Algorithmus ist es den Fehler folgender Gleichung zu minimieren

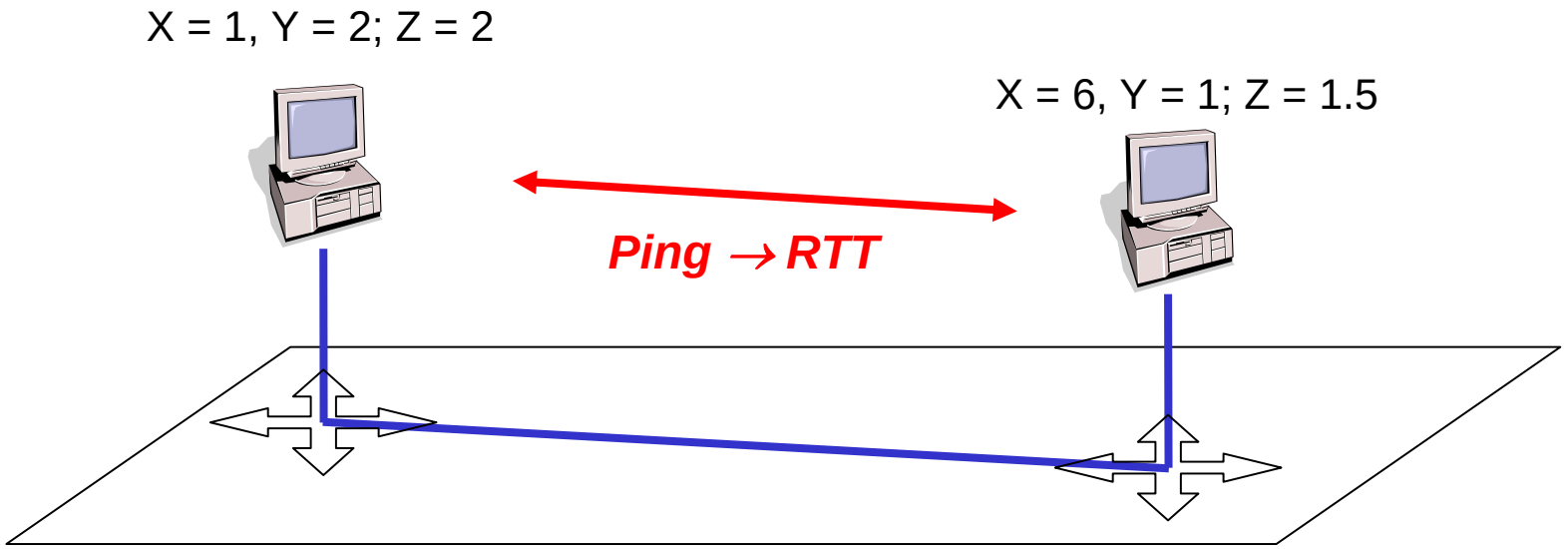
$$E = \sum_i \sum_j \left(L_{ij} - \| x_i - x_j \| \right)^2$$

Vivaldi – RTT Vorhersage mittels Einbettung

- Die genaue Dynamik ist schwierig einzustellen
  → Fehler minimieren, aber Oszillationen vermeiden
- Vivaldi erreicht Fehler von ca. 11%
- Grundlage der Einbettung ist eine
  2D Ebene mit Höhe. Diese modelliert
  lokale Verzögerungen, z.B. an einem
  Switch an dem die Rechner sternförmig angebunden sind.
- Komplexere metrische Räume (z.B. Kugel) ergeben keine wesentlichen Verbesserungen.

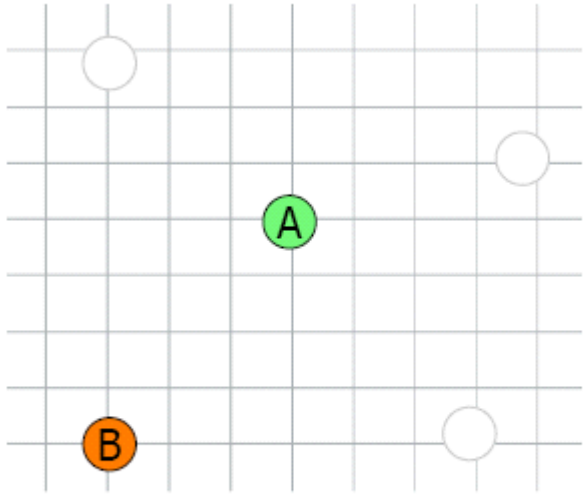
Frank Dabek, Russ Cox, Frans Kaashoek, Robert Morris. Vivaldi: A Decentralized Network Coordinate System, SIGCOMM 2004

Vivaldi – Übersicht



Vivaldi Vorgehen I

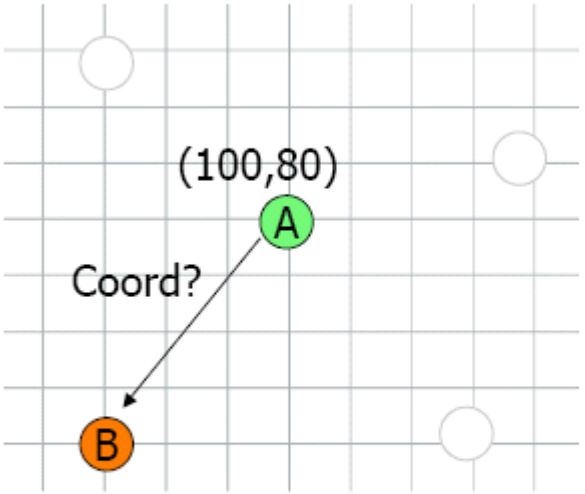
* Vivaldi [Cox '03,Dabek '04]



Quelle: Jonathan Ledlie - Harvard University - NSDI - April '07

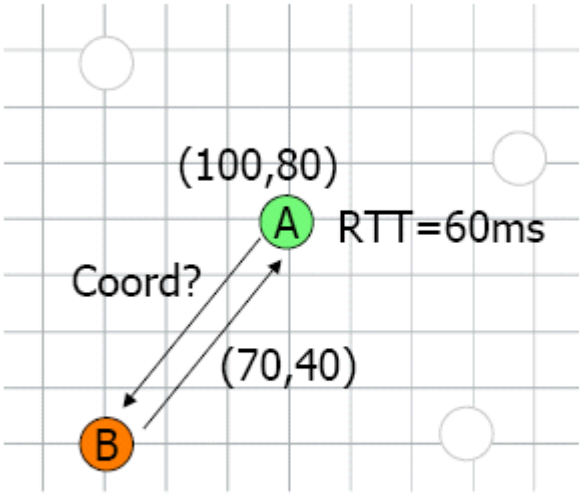
Vivaldi Vorgehen II

1. A measures latency to B.



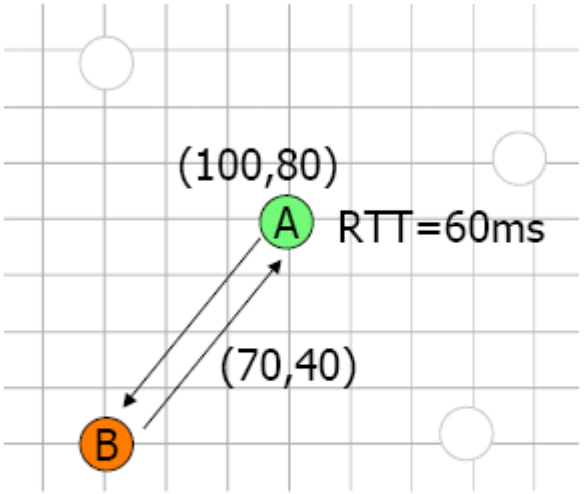
Vivaldi Vorgehen III

1. A measures latency to B.
2. B replies with its coord. A deduces RTT.



Vivaldi Vorgehen IV

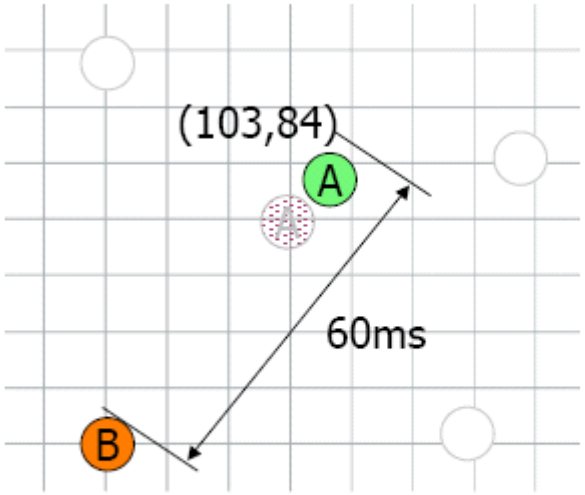
1. A measures latency to B.
2. B replies with its coord. A deduces RTT.
3. A computes estimate and error.



Estimate = |(100,80)-(70-40)|=50ms
Error = (60 - Estimate) = 10ms

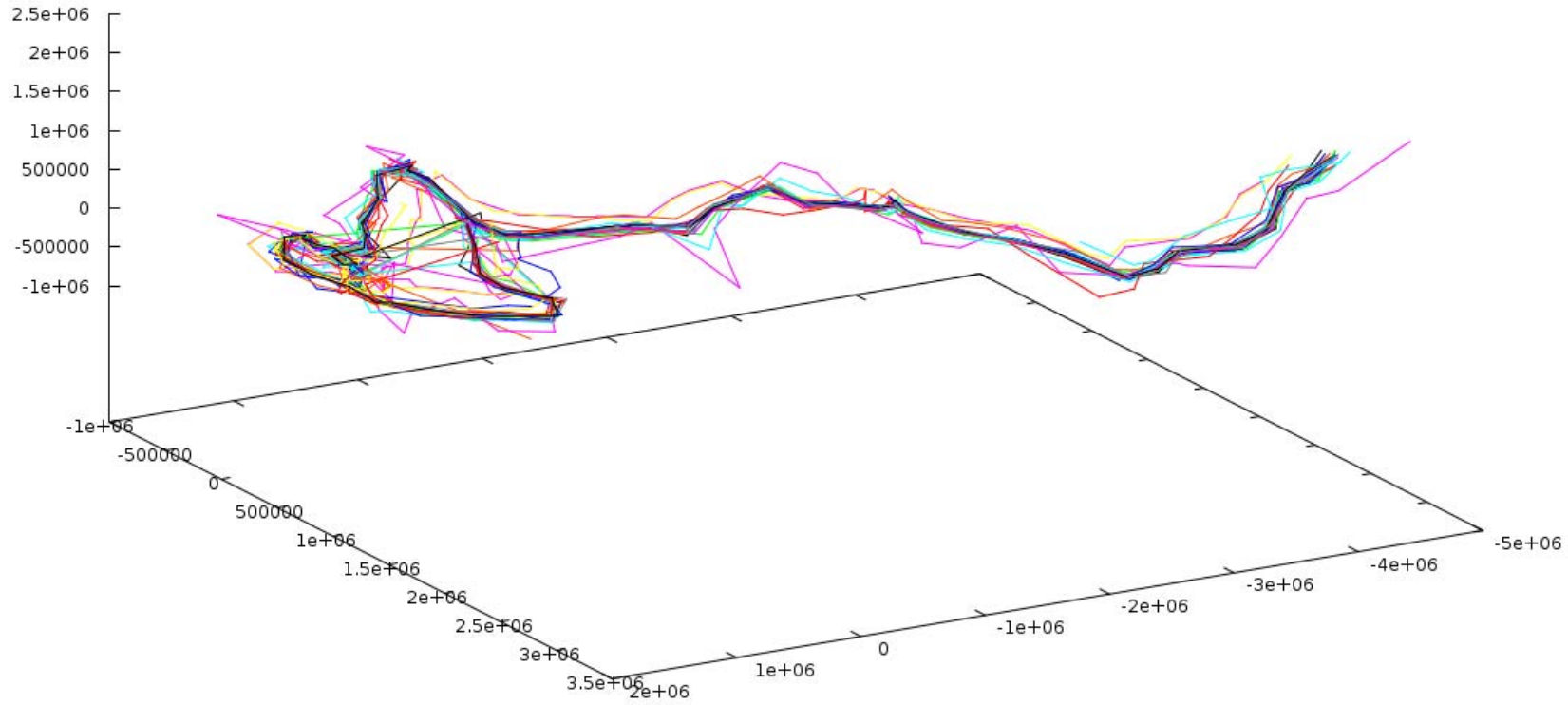
Vivaldi Vorgehen V

1. A measures latency to B.
2. B replies with its coord. A deduces RTT.
3. A computes estimate and error.
4. A moves toward ideal coord, relative to B.



Estimate = |(100,80)-(70-40)|=50ms
Error = (60 - Estimate) = 10ms

Vivaldi – Und es bewegt sich doch



Entwicklung von Vivaldi Koordinaten auf 20 Hosts

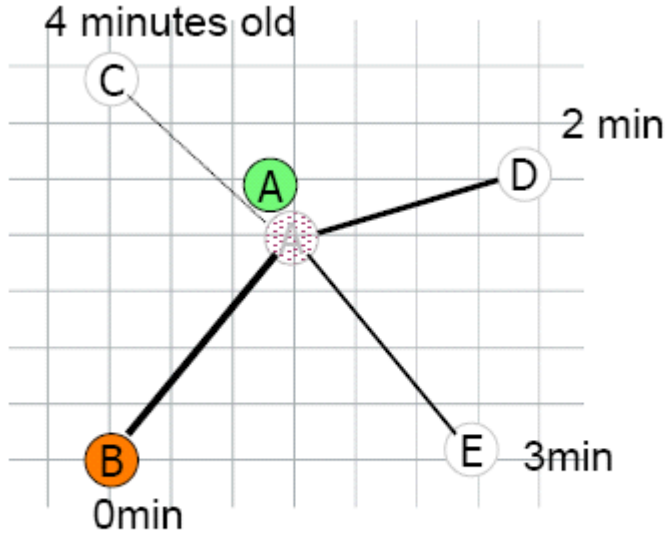
Vivaldi: Optimierungen

- Problem: Vivaldi Koordinaten ändern sich zu häufig.

- Ideen:
- Einzelmessungen zwischen Nodes könnten Aussreisser sein → Speichere die letzten *n* RTT Messungen zu diesem Knoten ab und verwende den Median
- Passe nicht die Koordinate isoliert an. Im derzeitigen Vivaldi wird die Node nur in Richtung des derzeitigen Messungspartners verschoben. Die Kräfte der anderen Nodes werden ignoriert → Berechne eine Gesammtkraft, zu der die neue Messung beiträgt.

Vivaldi: Optimierungen II

```
Force F = 0^d
for (n : neighbors) {
    Compute force F_n
    Norm F_n by age_n
    F += F_n
}
A += F
```



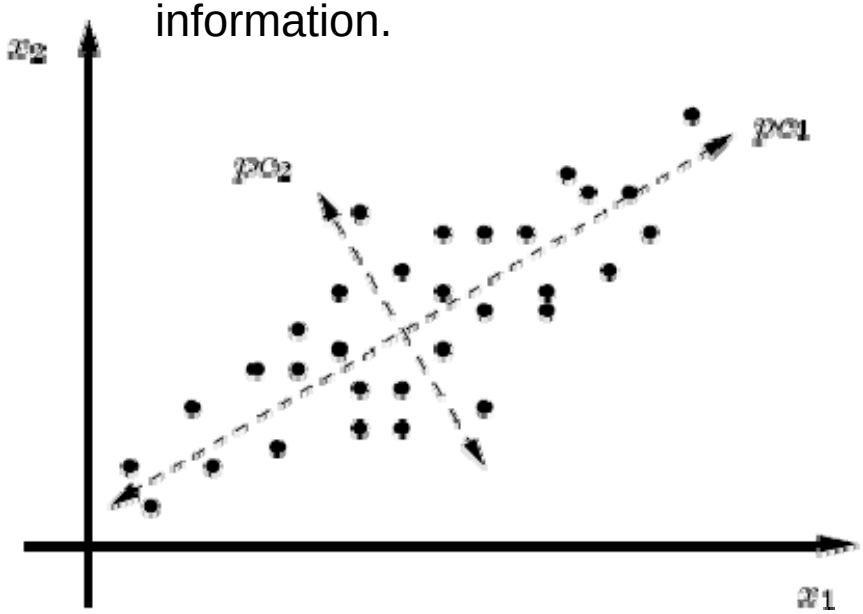
Older information gets less weight

Internet Coordinate System

- Internet Coordinate System (ICS) [Lim et al. 2003]
  - Uses the Principal Component Analysis (PCA) to determine landmark and host coordinates.
  - Lower computational overhead than GNP (only basic matrix transformations and eigenvalue decomposition).
  - Positions of landmarks in the virtual space can be uniquely determined.
  - The sufficient number of dimensions needed to represent the whole system as a virtual space can be computed!

Principal Component Analysis (PCA)

- Linear transformation of the sample data using eigenvalues and eigenvectors.
- Transform the data so that the variance of the data decreases for each (further) dimension.
- Thereby, we can reduce the number of dimensions with minimal loss of information.



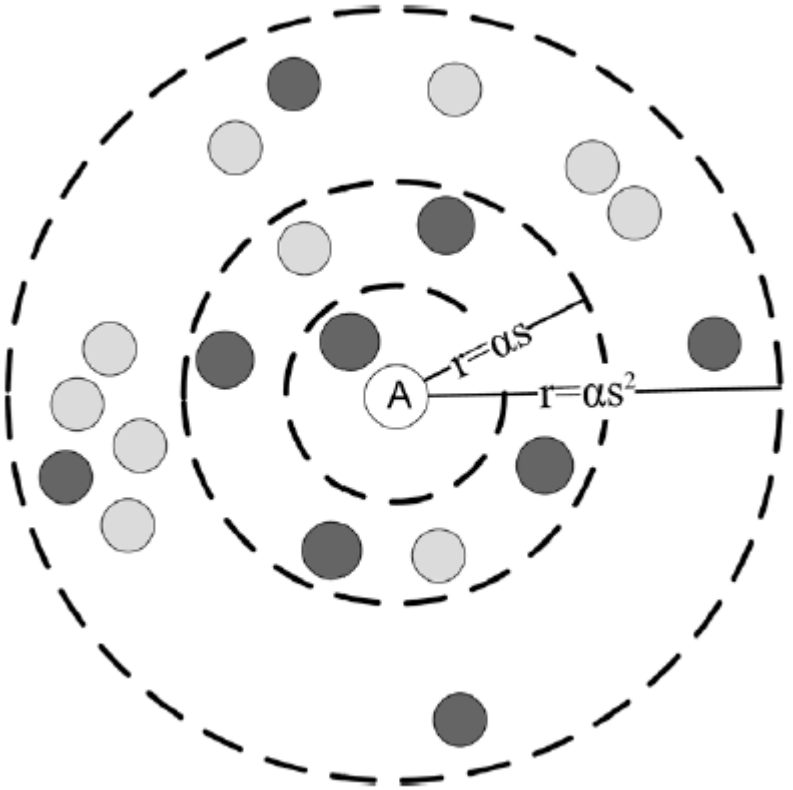
Source: Dragan Milic

Internet Coordinate System

- Landmarks:
  - Determine the RTT between all landmarks.
  - Represent it as a n*n matrix (for n landmarks).
  - Perform PCA of the RTTs. The result is the PCA transformation matrix.
  - Determine the number of dimensions that are sufficient to represent most of the measured data.
  - Scale the calculated transformation by using an least square estimator to achieve the preservation of distances between the landmarks in the transformed space.
- Hosts:
  - Measure distance to all (or a subset of) landmarks.
  - Represent the measurements as a n-dimensional vector.
  - Retrieve the scaled transformation matrix and calculate the host position by multiplying the distance matrix with the received transformation matrix.

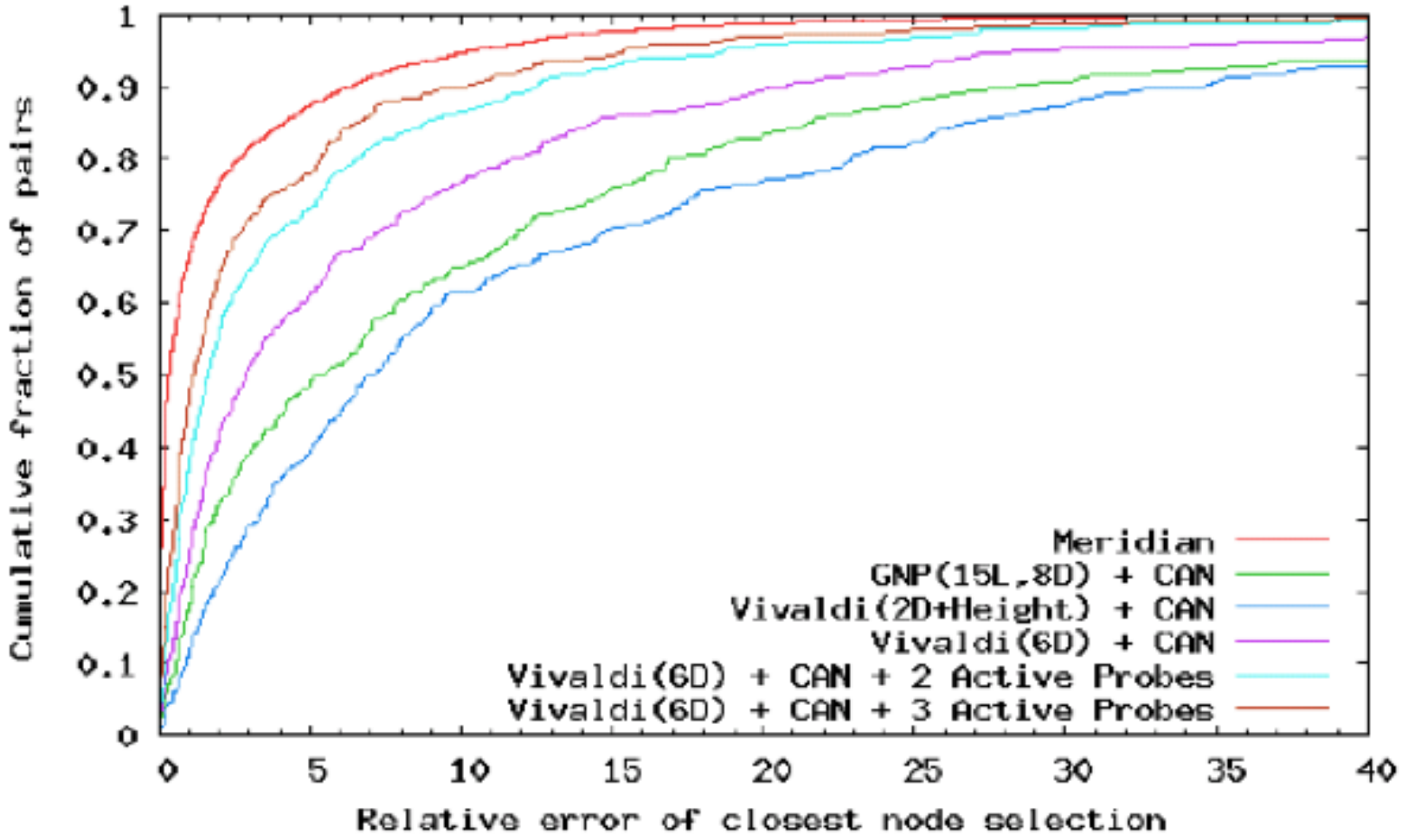
Meridian – RTT Vorhersage ohne Einbettung

- Jeder Knoten unterhält einen Cache mit anderen Knoten, nach exponentiell steigendem Abstandskörben sortiert.
- Fragt ein anderer Knoten nach einem für sich nahen Knoten, …
  - Wird die RTT bestimmt und
  - einige Knoten aus dem entsprechenden Korb ausgewählt.
- Diese werden dann iterativ weiter befragt.
- Meridian hat einen kleineren Kommunikationsaufwand als Vivaldi (on demand) und erzielt deutlich kleinere RTT-Fehler.



Bernard Wong, Aleksandrs Slivkins, Emin Gün Sirer, Meridian: A Lightweight Network Location Service without Virtual Coordinates, SIGCOMM 2005

Meridian Ergebnisse



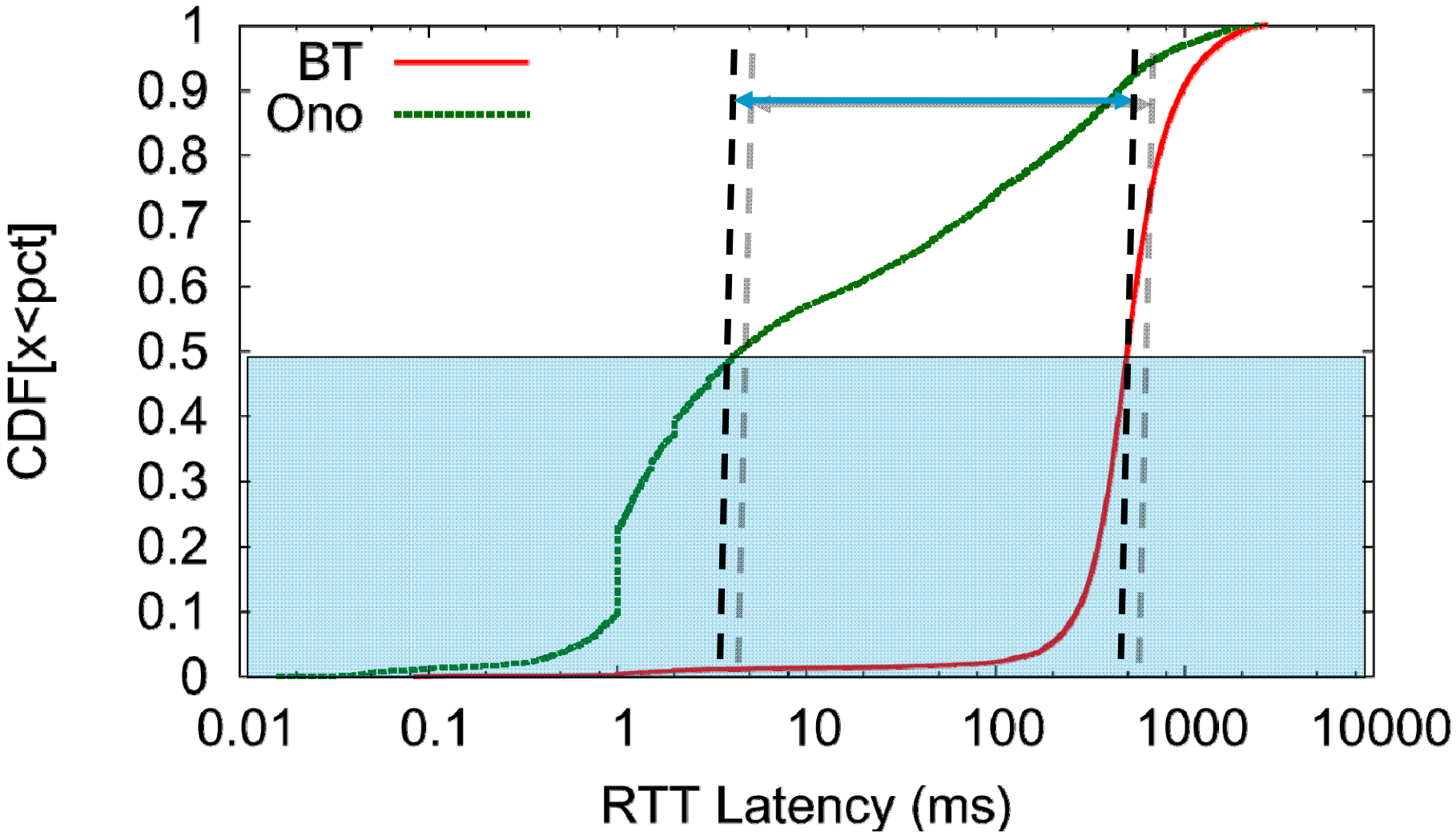
Ono – RTT Vorhersage mit CDNs

- CDNs haben bereits die gewünschte Information
- Warum also neu berechnen?
- Jede Anfrage gibt einem Client den *ihm* nahesten Server zurück, z.B.
  - www.yahoo.de → 141.12.178.1 (für DE)
  - www.yahoo.de → 211.13.17.43 (für AT)

David R. Choffnes, Fabián E. Bustamante: Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems. , SIGCOMM 2008

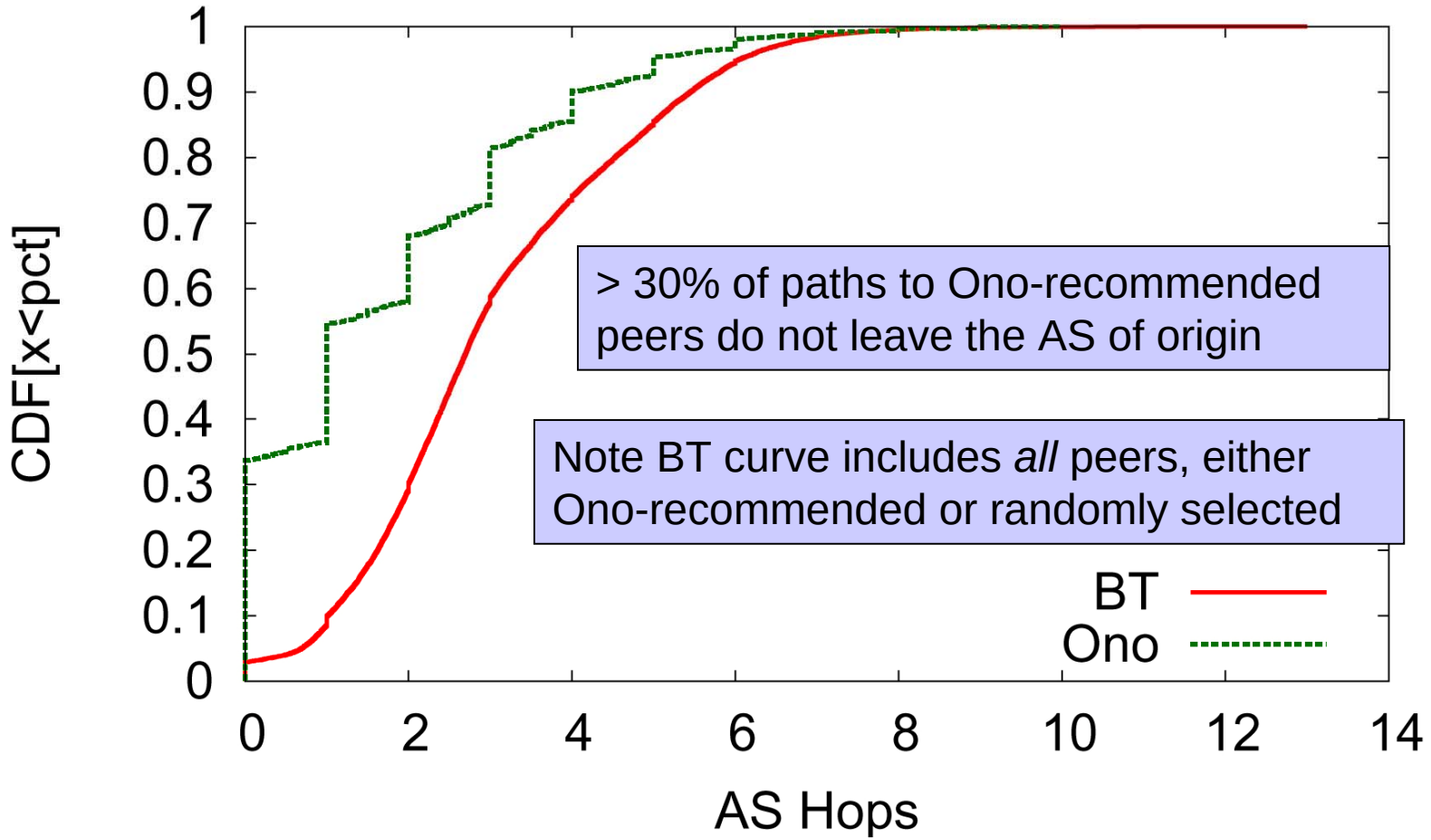
Idee von Ono:

Ono – Ergebnisse



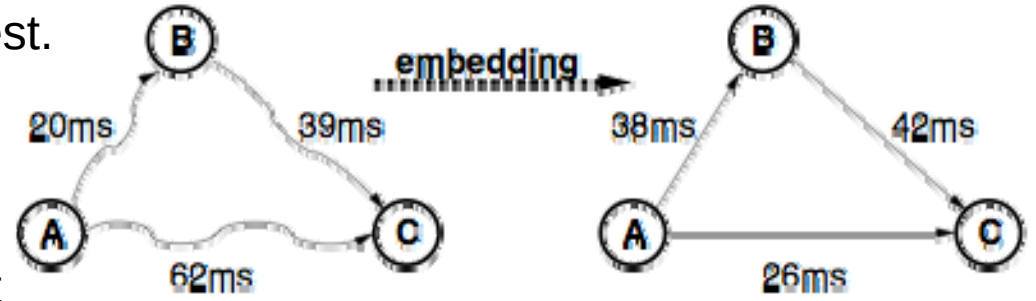
Ono – Ergebnisse

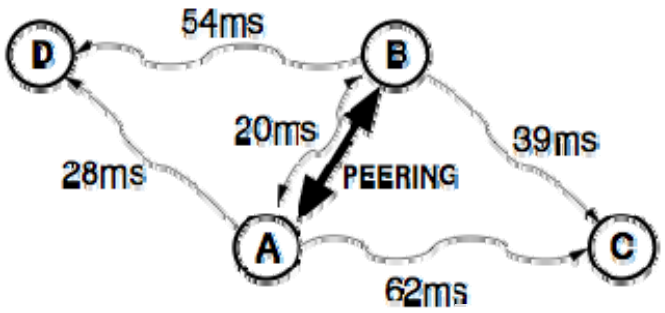
Average number of AS hops to reach Ono-recommended/random peers



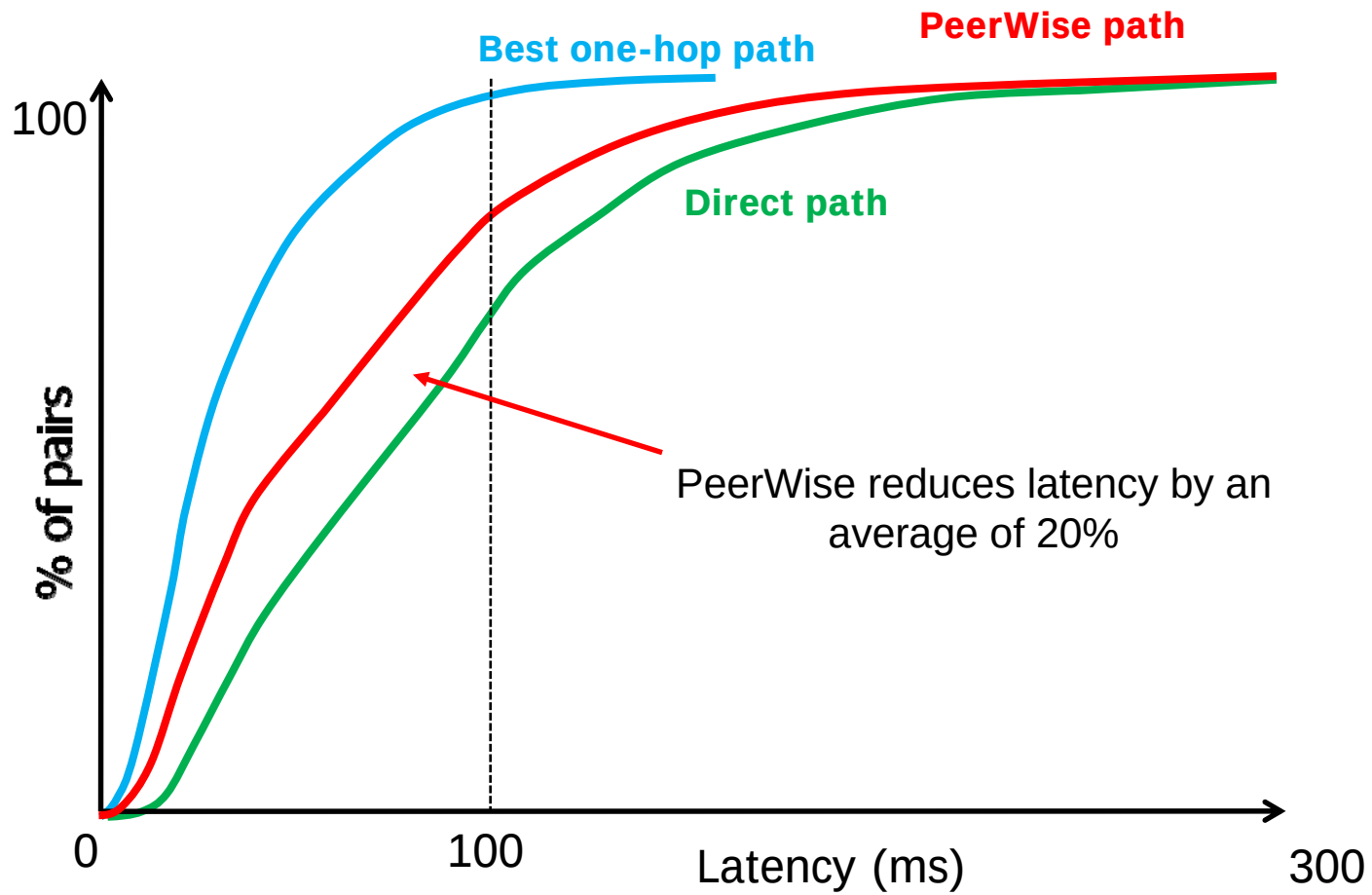
PeerWise

- Vivaldi and Ono are clever ideas, but seem to be less powerful in practice than the papers suggest.
- Triangle inequality violations (TIV) might be one source of the problems:
  - Measured latencies cannot be embedded into a metric space, beacuse a metric space does not allow TIVs.
  - Predicted latencies are less accurate if we force an embedding.
- PeerWise tries to improve over this shortcoming.





PeerWise - Performance

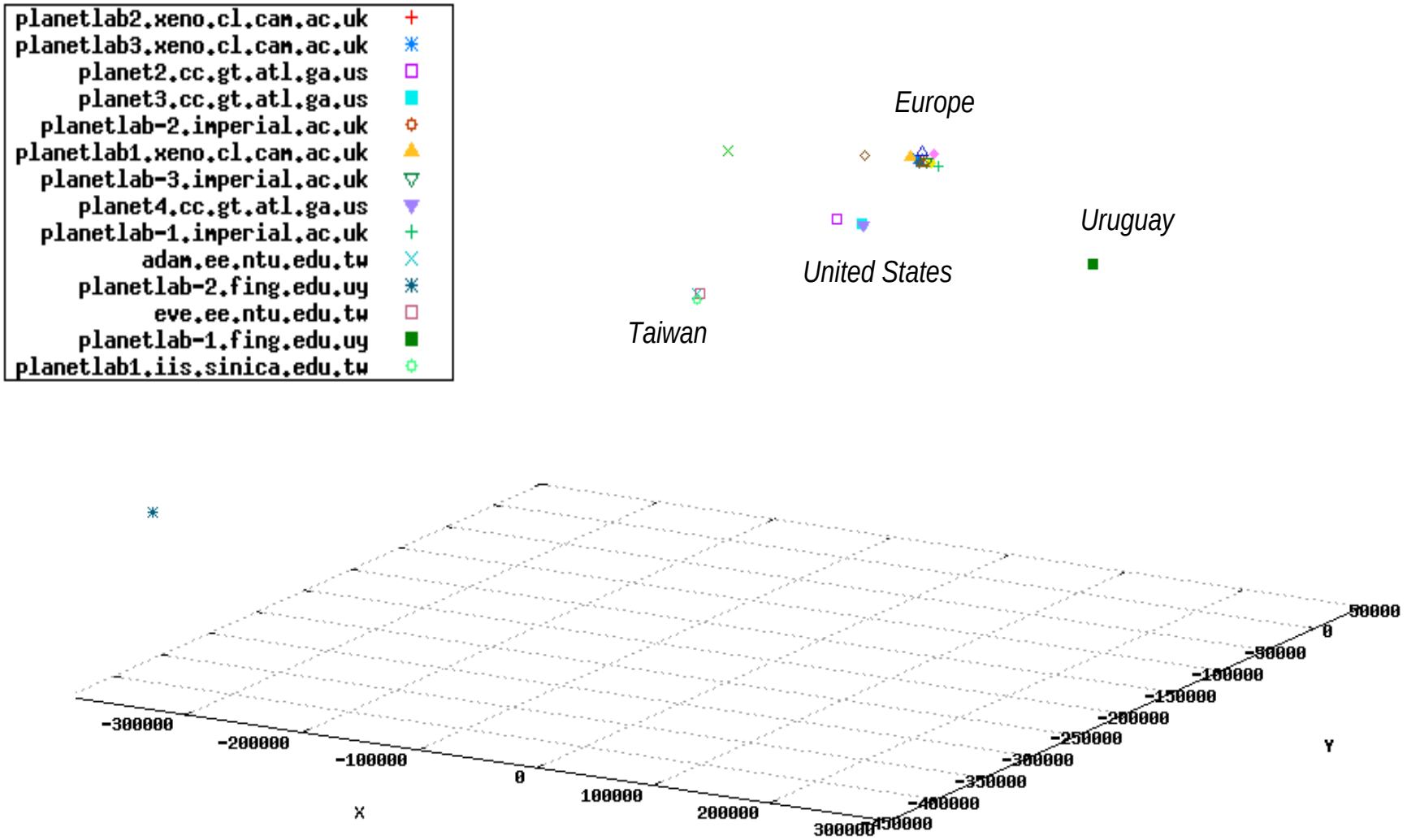


Source: Cristian Lumezanu. Dave Levin. Neil Spring. PeerWise Discovery and Negotiation of Faster Paths., HotNets 2007 presentation

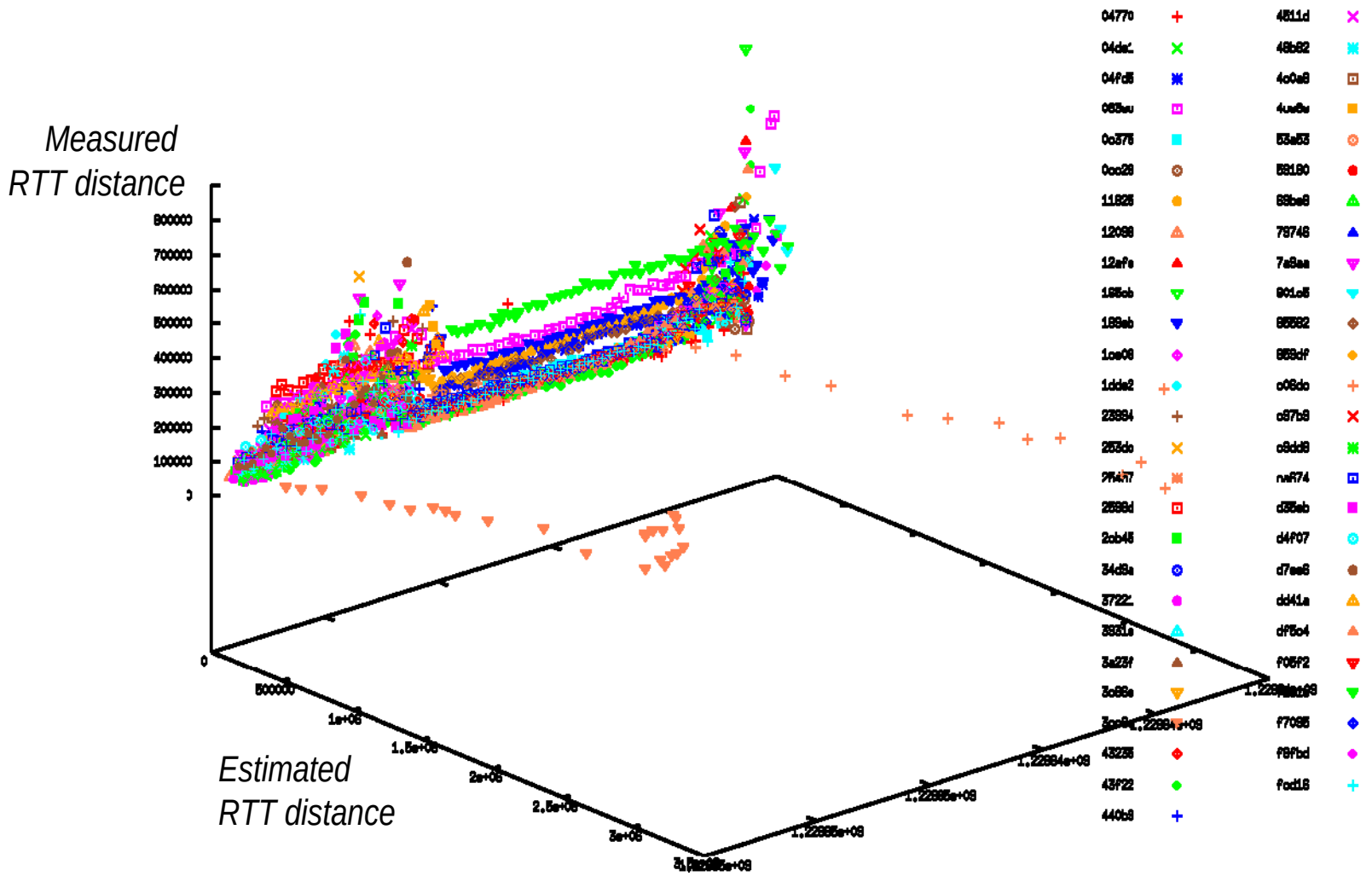
Summary

- Peer-to-peer systems, both structured and unstructured, need to predict which peer could serve them best.
- Content distribution networks improved their performance with a distributed, but centrally controlled infrastructure.
- Network tomography was used in the late 1990s to measure the Internet‘s topology.
- Exploit the ideas for network tomography to predict RTT in P2P systems:
  - IDMaps
  - Global Network Positioning (GNP)
  - Vivaldi
  - Internet Coordinate System (ICS)
  - Meridian
  - Ono
  - PeerWise

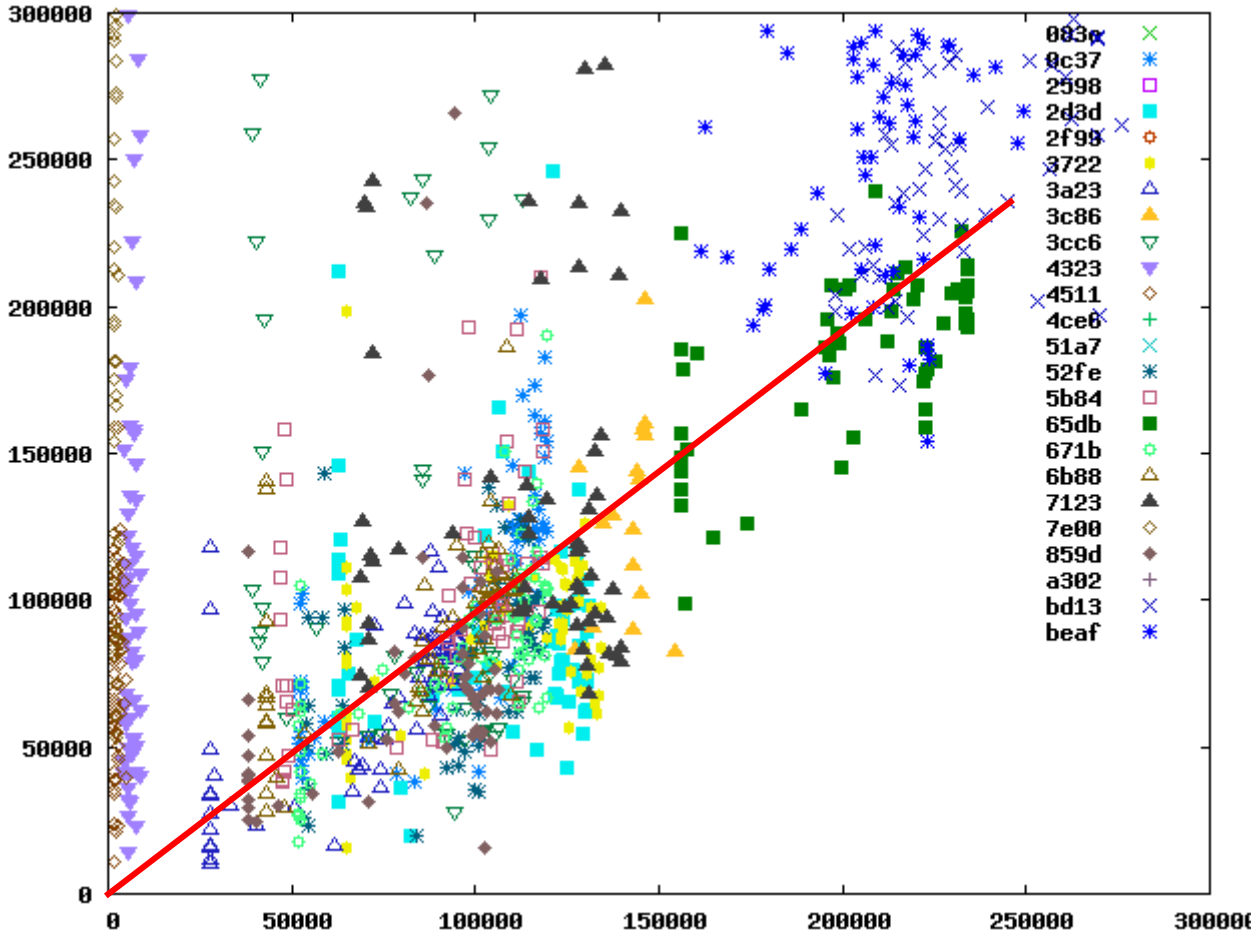
Measurement from IGOR (1)



Measurement from IGOR (2)



Measurement from IGOR (3)



Questions?

Thomas Fuhrmann

Department of Informatics
Self-Organizing Systems Group
c/o I8 Network Architectures and Services
Technical University Munich, Germany

fuhrmann@net.in.tum.de