Master Course Computer Networks IN2097

Prof. Dr.-Ing. Georg Carle,
Christian Grothoff, Ph.D.,
Dr. Nils Kammenhuber,
Dirk Haage

Chair for Network Architectures and Services
Department of Computer Science
Technische Universität München
http://www.net.in.tum.de

Chapter 7:
Network Measurements
Part 2

Chapter 7 Outline – Network Measurements

- Localization of nodes
  - Geoip
  - Network coordinates
- Cross-Layer considerations

Localization of nodes

Localization of nodes

- Provide location-based services
  - Local advertisements
  - Extend/reduce service for local/non-local users (e.g. IPTV often restricted to country boundaries)
- Choosing of servers
  - Load balancing between hosting location
  - Choose nearest instance of a service (anycast)
  - Locate nearest peers in P2P networks
  - Content delivery networks
  - Online games (gameserver)
  - Resource placement in distributed systems
  - TOR
- Find friends, coworkers, …
  - Google Latitude
- Optimization of application layer multicast trees
- …

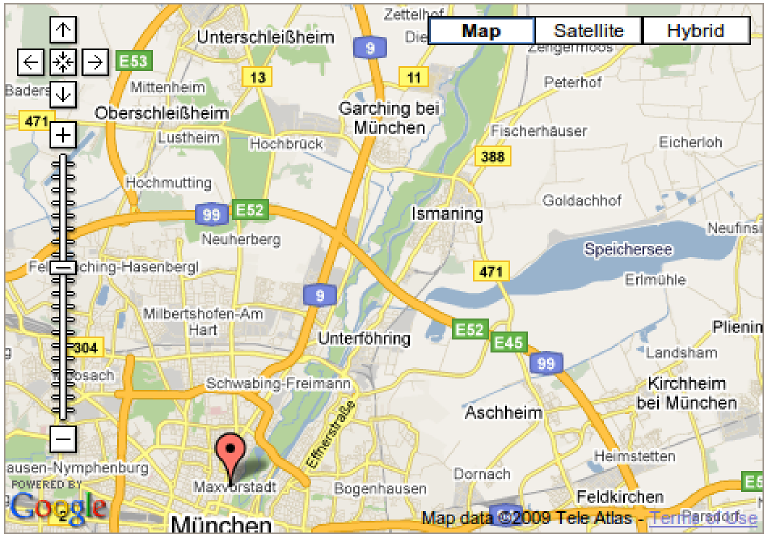
Localization of nodes (II)

- Mapping IP addresses to geo locations
- Determination of distance via latencies
- Triangulation, Trilateration (e.g. wireless networks)
- GPS, Cellular positioning/ Cell ID
- …

GeoIP

- Map IP to a location in the world
- Granularity levels
  - Country/ continent
  - City, maybe urban districts
  - Street/ exact location

Host Name:	131.159.20.45
IP Address:	131.159.20.45
Country:	Germany
Country code:	DE (DEU)
Region:	Bayern
City:	Munich
Postal code:	
Calling code:	+49
Longitude:	11.5833
Latitude:	48.15



GeoIP (II)

- Basic data sources
  - AS information
  - Whois/RIR information
  - Provider data

- Additional sources
  - User input
    - Update location manually
    - Accurate positioning devices
      - Smart phone with GPS
      - Verify/ update current position for used public IP
    - Track changes in IP of same user
      - Mitigate effect of changing IP connection

- Reduce bias by combining sources
  - Verify data, filter inaccurate data

GeoIP (III)

- Accuracy depends on location database
  - More accurate for static IPs (server, university, …)
  - Less accurate for home connections
    - Frequent changes
    - Change in IP often also changes the geo location of that IP
  - Not usable in private networks
    - E.g. cellular network (currently private networks + NAT)

- Provider cooperation is required
  - Detailed information for each and every IP
  - Disclose internal structure (subnets, connectivity)
    - Which subnet is used at which site (city, maybe even parts)
  - Update in case of changes

- Single point of failure
  - Excessive use slows down localization
  - Not usable for massive requests

- Many different implementations

Network coordinates

- Latencies between nodes as a metric for distance
  - Round trip time
    - Simplest measurement at all (ping)
    - Most accurate (only one clock involved)
    - Similar to real distance (propagation speed nearly constant)

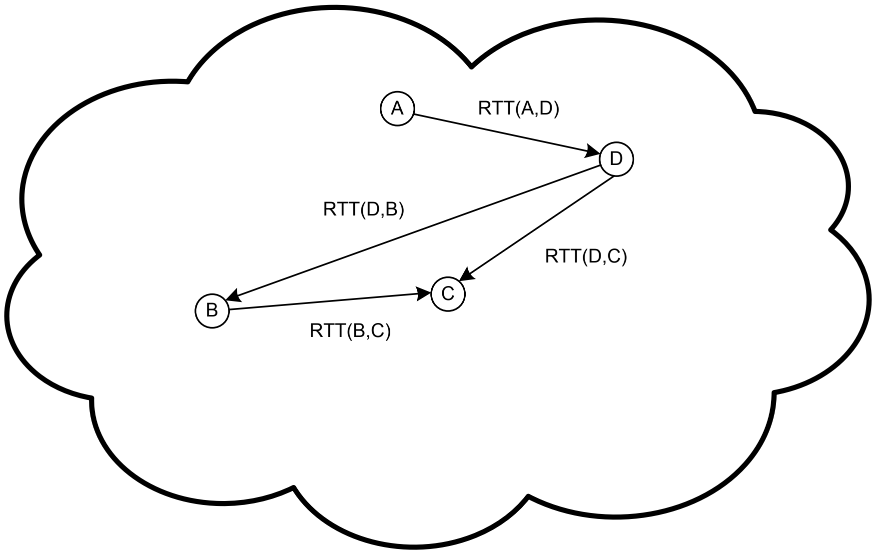
- How to get?
- Simple approach:
  Measurements between all pairs of nodes
  - $O(n^2)$
  - Does not scale (cannot be used for large networks)
  - Rely on actual traffic → hybrid measurement
  - Normally no traffic to all nodes available
    - Active measurements (even worse scaling)

Network coordinates (II)

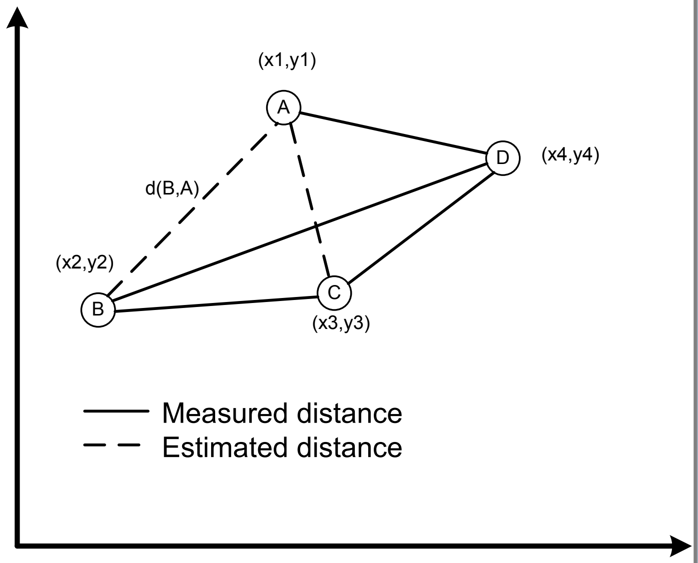
- Measure the distances to some neighbors
  - Neighbors might be known hosts, not near hosts
- Calculate a artificial coordinate in a metric space
  - Metric space = distance between nodes can be calculated
  - E.g. Euclidean n-space
- Approximate the latency
  - Distance between nodes in the coordinate system is approximation to the latency
- Abstract definition:
  - Embed network graph into a metric space
  - Metric embedding/ graph embedding

Example

Internet



Euclidean space (2D)



$$d(B,A) = |(x_2, y_2),(x_1, y_1)| = (|x_1 - x_2|,|y_1 - y_2|)$$

Network coordinates (III)

- Advantages
  - Small overhead
    - Only requires small number of measurements
    - No additional traffic
      (application traffic = measurement traffic)
    - Piggy-back the coordinate information
  - Each host can calculate the distance to every other host
    - Only requires the coordinates

- Design goals
  - Accuracy: small error for RTT estimations
  - Scalability: large-scale networks, small overhead, no bottlenecks
  - Flexibility: adapt coordinates to network changes
  - Stability: no drift, oscillation of coordinates
  - Robustness: small impact of error by malicious nodes, nodes with high errors

Triangle inequality

- Intuition:
  direct latency between 2 nodes should be smaller than any indirection

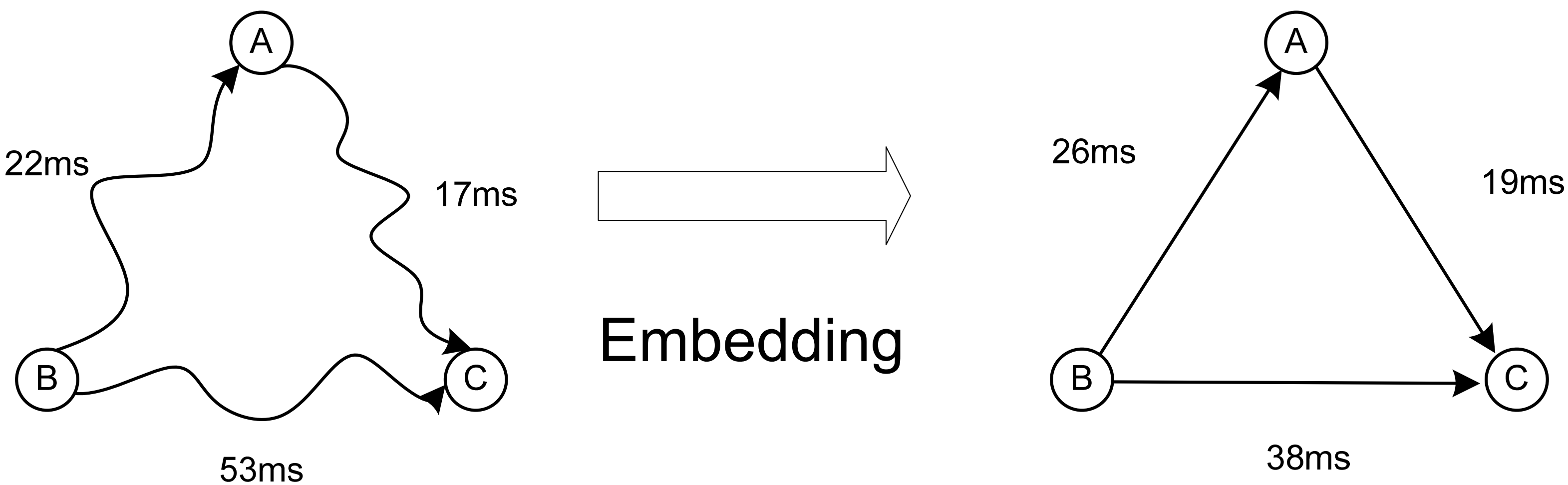
$$d(a,b) + d(b,c) \geq d(a,c)$$

- Triangle inequality violations (TIV) inherent to Internet routing structure
  - Selective/ private peering
  - Hot potato routing
  - Link metric ≠ latency
  - Asymmetric links (e.g. DSL, UMTS)
- TIVs are common
  - >85% of all host pairs part of a TIV
  - For 20-35% exists a path that is at least 20% shorter

(Traces: King, Azureus)

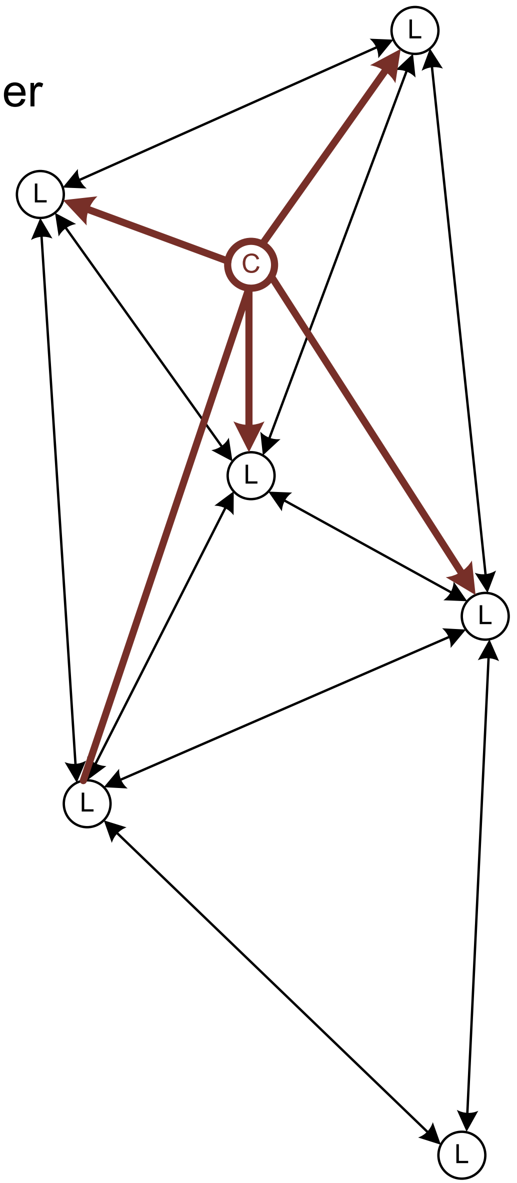
Triangle inequality (II)

- Possible spaces for embedding are metric
  - Distance function satisfies triangle inequality
- Embedding can not be exact
  - Number and weight of TIVs limits embedding quality



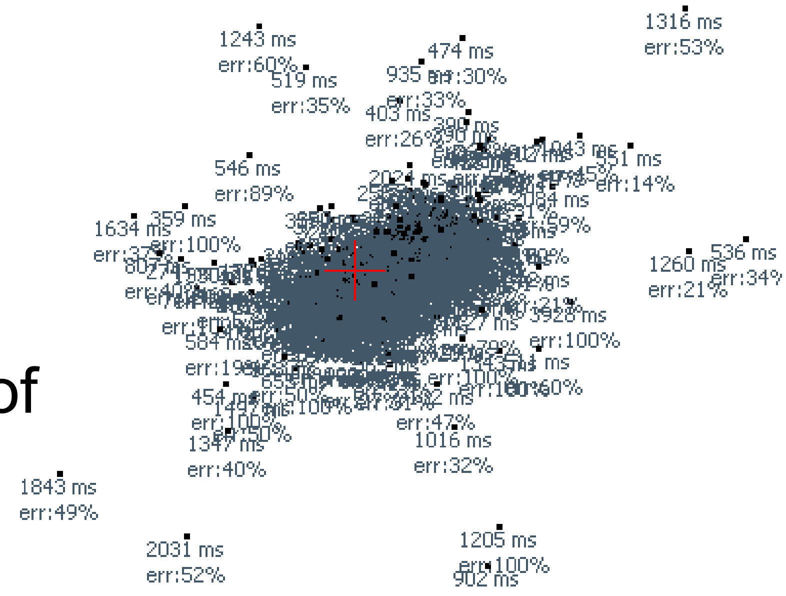
History

- Global Network Positioning (Ng, Zhang, 2002)
  - Landmark nodes measure distance between eachother
  - New nodes measure distance to landmarks
  - Coordinates relative to landmarks
  - Embedding via Downhill-Simplex in 3D space
  - Problems:
    - Scalability
    - Placement of landmarks
    - Single point of failure
- Lighthouse (Pias et al., 2003)
  - Several groups of landmarks
- PIC (Costa, Castro, Rowstron, Key, 2004)
  - Generalization of GNP
  - All nodes with known coordinates can be landmarks
- Big-Bang-Simulation (2004)
  - Analogy to physics: nodes as particles in a force field



Vivaldi (Dabek, Cox, Kaashoek, Morris, 2004)

- Fully distributed
  - No infrastructure, no specialized nodes
- Continuous upgrade of coordinates with new latency values
- Based on application traffic
- Small number of communication partners required for meaningful results
- Can be used with various types of spaces
- State of the art
- Actively used (e.g. bittorrent, azureus)



Vivaldi Algorithm

1. Choose random (obviously wrong) position
2. Initiate communication with some nodes
3. Measure latency
4. Nodes provide coordinates and error estimation
5. Revise coordinates (relative to other nodes)

Optimization

- due to TIVs and measurement errors
  - No exact embedding in low-dimensional spaces
  - Requires at most n-1 dimensions

- Optimization problem
  - Minimize error
    (= difference between real and estimated latency)

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

$\|x_i - x_j\|$: distance between coordinates i, j

L_{ij} : measured latency

  - Distance depends on space

Optimization (II)

- Spring Embedder
  - Physical analogy: network of springs
  - Between each pair (i,j) of hosts exists a spring
    - Length in equilibrium position: L_{ij}
    - Current length: $||x_i-x_j||$
    - Potential energy proportional to expansion squared: $(L_{ij}-||x_i-x_j||)^2$
      - Energy of the spring = error
      - Minimal energy in the system = minimal global error
  - Force between i and j (Hooks law)

$$F_{ij} = (L_{ij} - \left\|x_i - x_j\right\|) \times u(x_i - x_j)$$

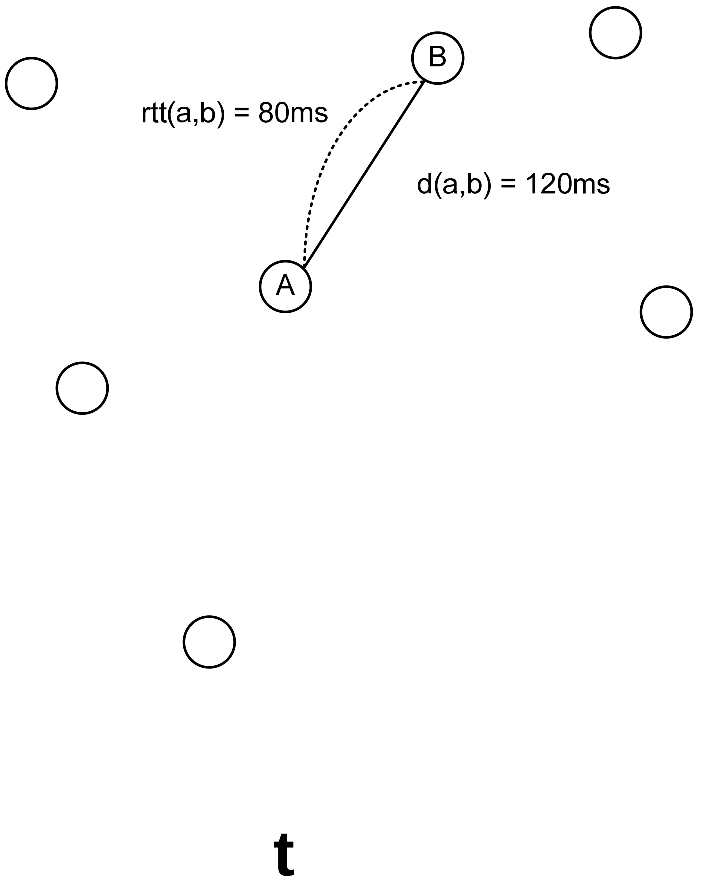
  - Move node to minimize its energy

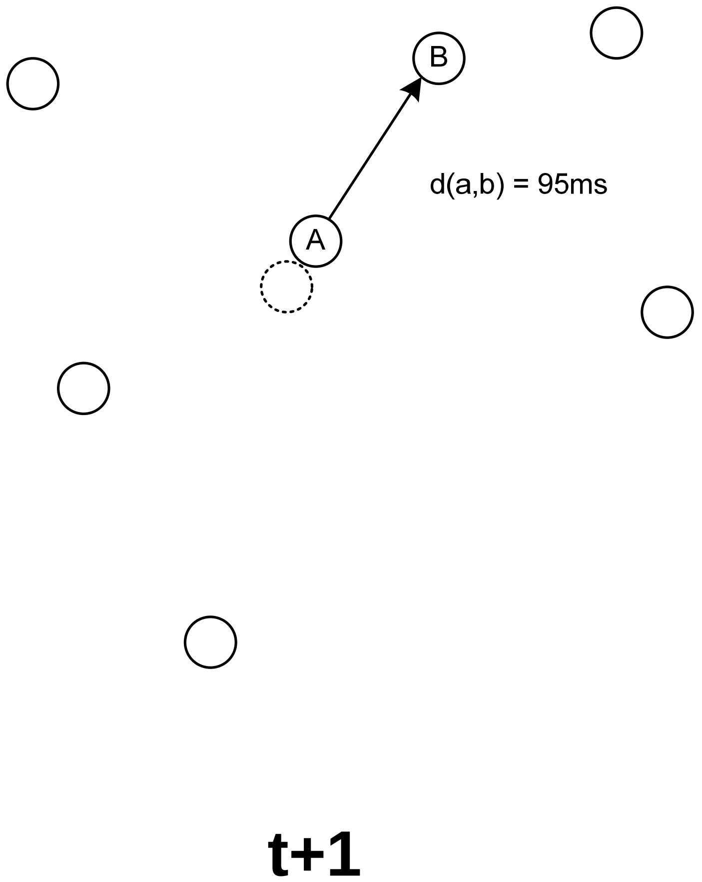
Optimization (III)

- Local
  - Iteratively move each node I by $\delta \cdot F_i$ per step
    δ = attenuation
- Global/ distributed
  - Each node calculates its coordinates
  - Large attenuation: oscillation
  - Small attenuation: slow convergence
  - Small impact of coordinates with high error
    - Adaptive attenuation

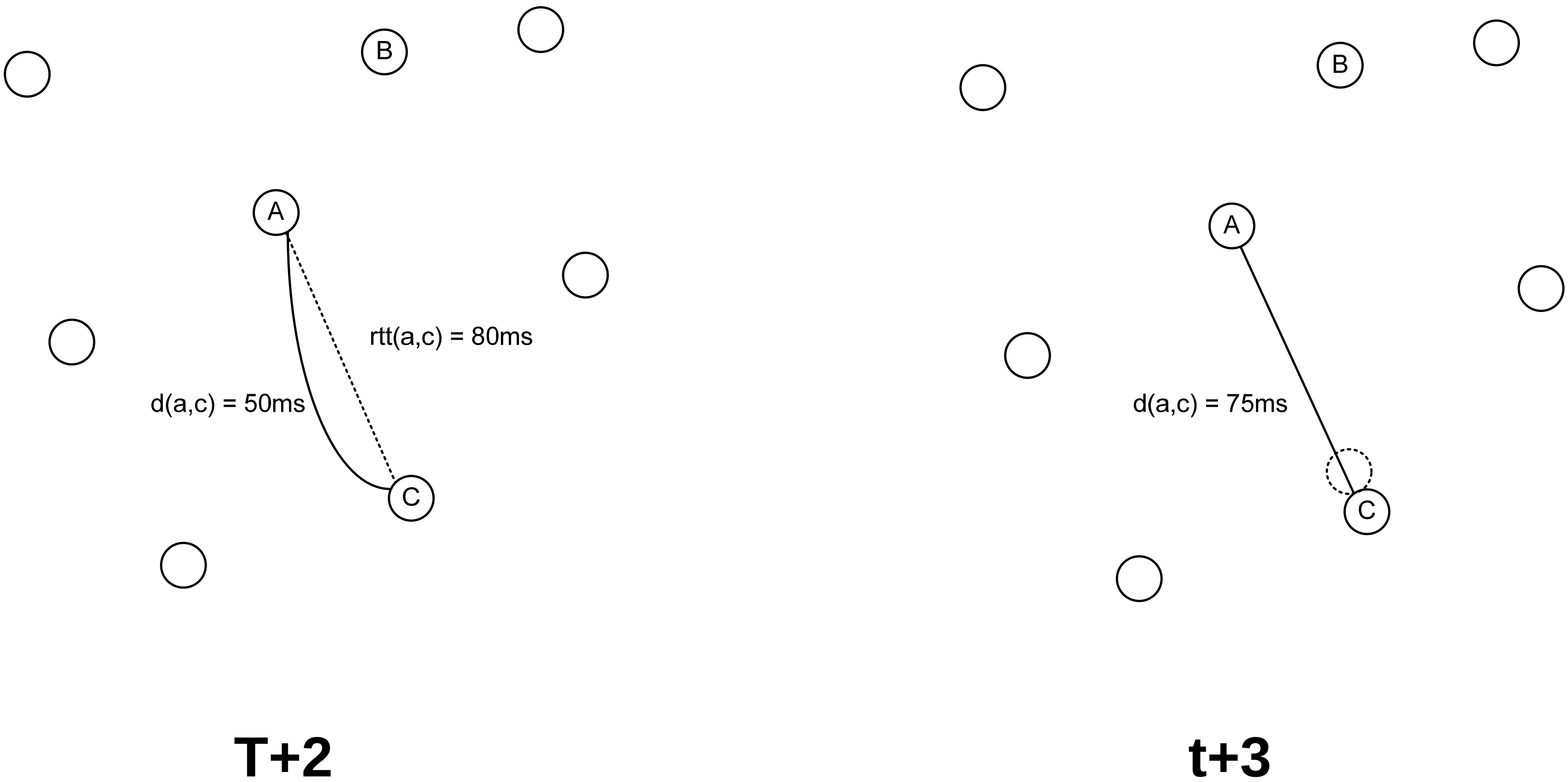
$$\delta = c_c \cdot \frac{e_i}{e_i + e_j}, c_c \text{constant}$$

Example





Example (II)



Which space to choose?

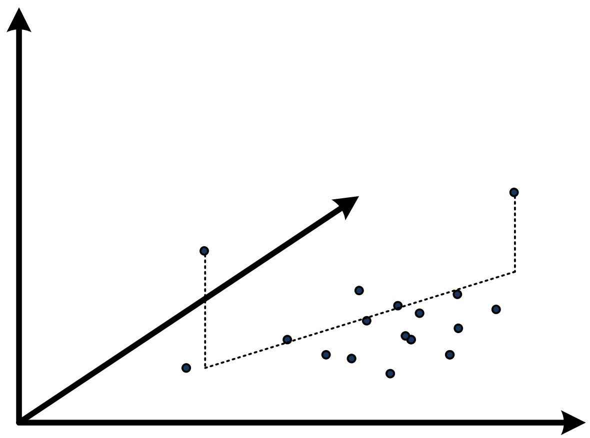
- Physics:
  - Anology uses 3D space
  - Any space with a definition of distance, difference between coordinates and scalar multiplication possible

- Question:
  Which space characterizes the Internet most?
  - 2D, 3D
  - Sphere, torus
  - Complex network → complex space?
  - From GNP: embedding in 3D, why?

- Result from tests and simulations:
  - 2-3 dimension sufficient
  - More dimensions require more computation without significant improvement

Handling TIVs

- Again:
  - TIVs occur for asymmetric routes, links, …
  - Occur quite often
  - Enlarge the error for the embedding
- Instead of using n dimensions, use n-1 + height
  - Euclidean n-space models the core network
    - High connectivity
    - Fast, symmetric links
  - Height models the slow access links
  - Packets are transmitted in the core, not above it
  - Slow hosts are pushed out of the plane



Some results

- Error below 20% for 80. percentile (2D+H)
- Spherical coordinates do not improve the result
- Adaptive attenuation improve result
- Neighbors
  - < 32: bad results
  - > 64: no improvements
  - Best results with a mixture of near and distant neighbors
- Lookup times in DHTs improved by 30% for 80% of the nodes

- Problems
  - Instability due to churn, latency fluctuation
    - Neighbor decay
    - Latency filter
    - Update filter
  - Drift

Security

- Attacks
  - Disorder
    - Maximize error in coordinates
    - Denial of service
  - Isolation/ Repulsion
    - Move target into “isolated space”
    - Convince target that another node is far away
    - Redirect target to malicious node, replica server
      - Man in the middle attack
- Mitigation based on statistics
  - Classify nodes into bad/good via their behaviour

Cross Layer Considerations

Crosslayer considerations

- Network stack
  - Encapsulation of functionality
  - No knowledge required in upper layers about how the network works

But

- Protocols and applications make assumptions on the underlying network
  - Network might change over time
  - Assumptions might not be correct for all parts of the network
- Diverse underlay

- Example: TCP
  - Loss = congestion
  - Increasing delay = upcoming congestion
  - Long delay = narrow bandwidth
  - Are these assumptions still true?
    - Wireless networks
    - Satellite links
    - …

- Include information from different layers in the network stack: Cross layer approach

Implications for measurements

- Upper layers are not fully isolated from the underlay
- Network types and condition might change the outcome

- Questions that should be answered:
  - Does the measurement change the network and how? Does the network condition changes the measurement?
    - UMTS RTT measurements
    - Number of nodes in a WLAN
  - Are the assumption made for the measurement evaluation correct for this specific network
    - Dependency between delay and bandwidth
    - The way the underlay acts on losses (WLAN vs. Ethernet)

Conclusion

- Localization of nodes
  - Required for many purposes
  - GeoIP
  - Network coordinates

- Cross layer considerations
  - Take all layers into account while measuring

Thanks for listening!

Questions?