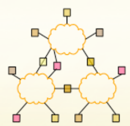
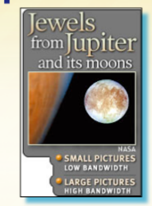
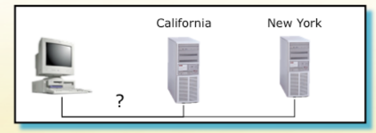
15-446 Distributed Systems
Spring 2009



L-24 Adaptive Applications

State of the Art – Manual Adaptation

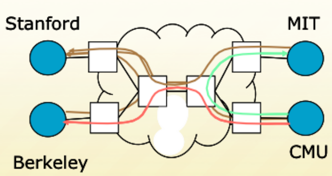


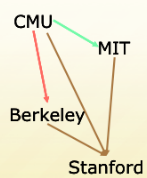


Objective: automating adaptation

Motivation

- Large-scale distributed services and applications
  - Napster, Gnutella, End System Multicast, etc
- Large number of configuration choices
- K participants ⇒ $O(K^2)$ e2e paths to consider





Why is Automated Adaptation Hard?

- Must infer Internet performance
  - Scalability
  - Accuracy
  - Tradeoff with timeliness
- Support for a variety of applications
  - Different performance metrics
  - API requirements
- Layered implementations hide information

Tools to Automate Adaptation

- Tools to facilitate the creation of adaptive networked applications
- Adapting on longer time scale (minutes)
  - Deciding *what* actions to perform
  - Deciding *where* to perform actions
  - Need to predict performance
- Adapting on short time scale (round-trip time)
  - Deciding *how* to perform action
  - Need to determine correct rate of transmission

5

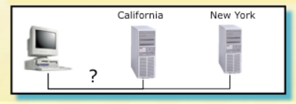
Adaptation on Different Time Scales

Long Time Scale

Short Time Scale

Content Negotiation

Server Selection

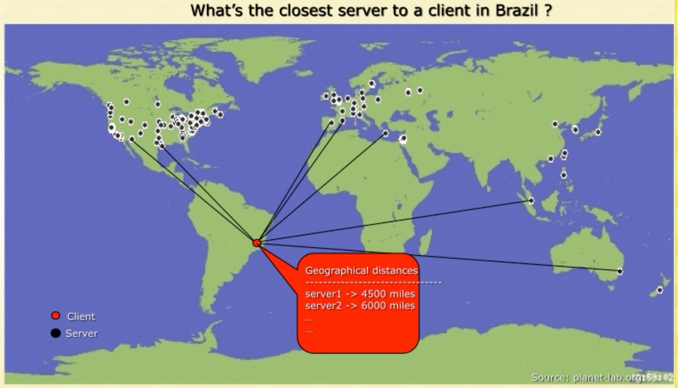


Adaptive Media

6

Motivation

What's the closest server to a client in Brazil ?



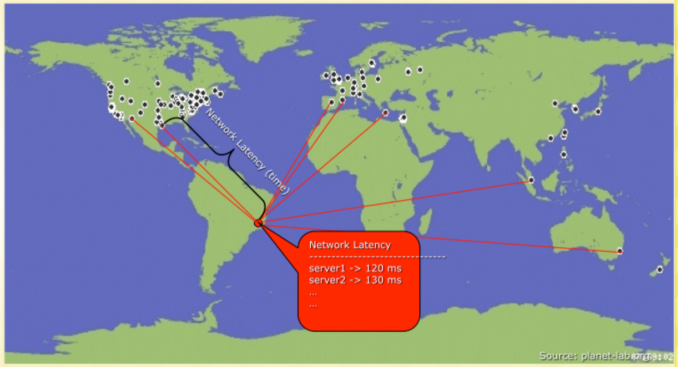
7

Motivation

- Difficulties:
  - Geographical distances ≠ network distances
    - Routing policies/Connectivity
    - GPS not available
  - Client needs 'N' distances to select the closest server

8

Motivation



9

Motivation

- Network latency = network distance
  - E.g. ping measurements
- Still have the issue of 'N' distances…
  - Need 'N' measurements (high overhead)
  - Update list of network distances
  - How do we solve this problem ?

10

Outline

- Active Measurements
- Passive Observation
- Network Coordinates

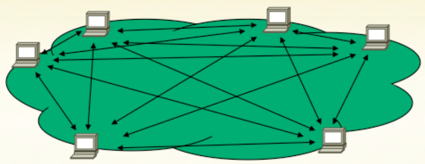
11

Network Distance

- Round-trip propagation and transmission delay
- Reflects Internet topology and routing
- A good first order performance optimization metric
  - Helps achieve low communication delay
  - A reasonable indicator of TCP throughput
    - Can weed out most bad choices
- But the $O(N^2)$ network distances are also hard to determine efficiently in Internet-scale systems

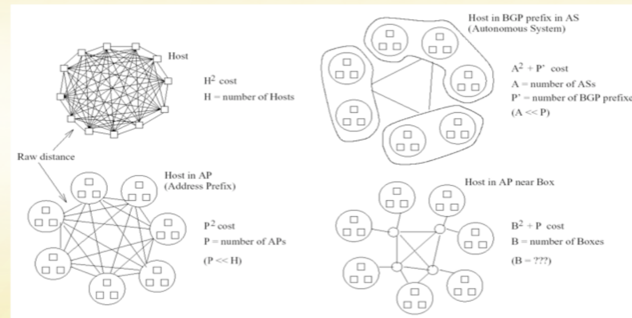
12

Active Measurements



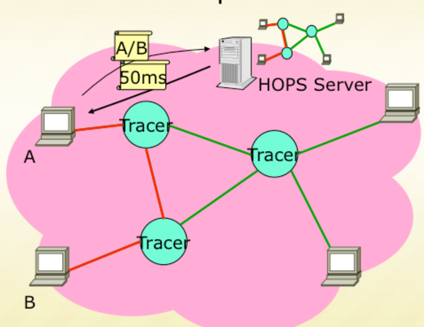
- Network distance can be measured with ping-pong messages
- But active measurement does not scale

Scaling Alternatives



State of the Art: IDMaps [Francis et al '99]

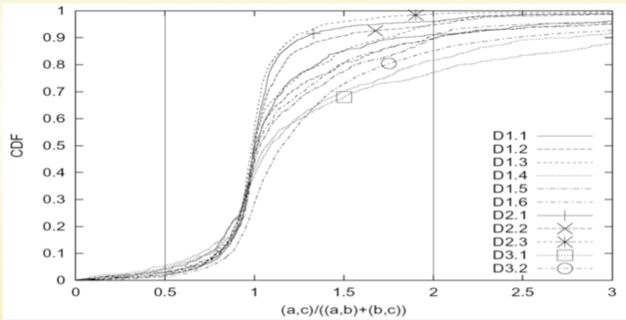
- A network distance prediction service



Assumptions

- Probe nodes approximate direct path
  - May require large number
  - Careful placement may help
- Requires that distance between end-points is approximated by sum
  - Triangle inequality must hold (i.e., (a,c) > (a,b) + (b,c)

Triangle Inequality in the Internet



17

A More Detailed Internet Map

- How do we …
  - build a structured atlas of the Internet?
  - predict routing between arbitrary end-hosts?
  - measure properties of links in the core?
  - measure links at the edge?

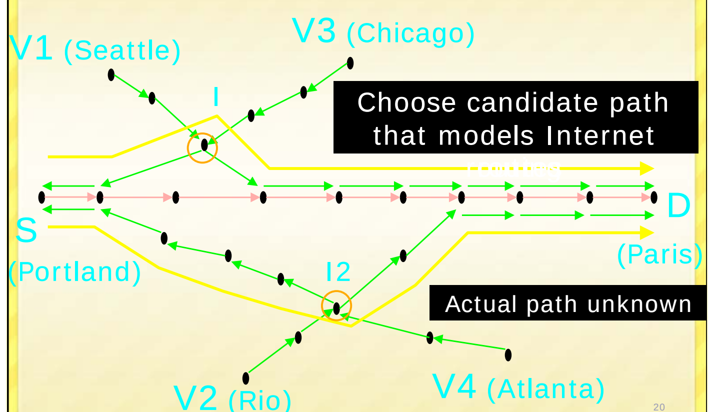
18

Build a Structural Atlas of the Internet

- Use PlanetLab + public traceroute servers
  - Over 700 geographically distributed vantage points
- Build an atlas of Internet routes
  - Perform traceroutes to a random sample of BGP prefixes
  - Cluster interfaces into PoPs
  - Repeat daily from vantage points

19

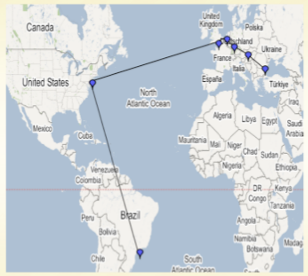
Model for Path Prediction



20

Example of Path Prediction

Actual path: RTT 298ms



Predicted path: RTT 310ms

Predicting Path Properties

- To estimate end-to-end path properties between arbitrary *S* and *D*
  - Use measured atlas to predict route
  - Combine properties of
    - Links in the core along predicted route
    - Access links at either end

Latency	Sum of link latencies
Loss-rate	Product of link loss-rates
Bandwidth	Minimum of link bandwidths

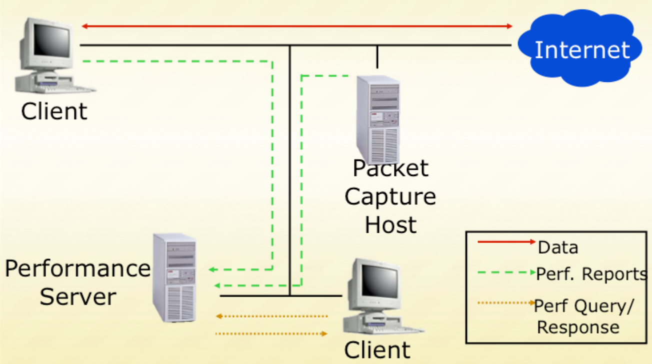
Outline

- Active Measurements
- Passive Observation
- Network Coordinates

SPAND Design Choices

- Measurements are *shared*
  - Hosts share performance information by placing it in a per-domain repository
- Measurements are *passive*
  - Application-to-application traffic is used to measure network performance
- Measurements are *application-specific*
  - When possible, measure application response time, not bandwidth, latency, hop count, etc.

SPAND Architecture



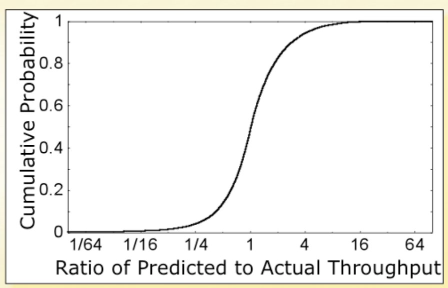
25

SPAND Assumptions

- *Geographic Stability:* Performance observed by nearby clients is similar → works within a domain
- *Amount of Sharing:* Multiple clients within domain access same destinations within reasonable time period → strong locality exists
- *Temporal Stability:* Recent measurements are indicative of future performance → true for 10's of minutes

26

Prediction Accuracy



- Packet capture trace of IBM Watson traffic
- Compare predictions to actual throughputs

27

Outline

- Active Measurements
- Passive Observation
- Network Coordinates

28

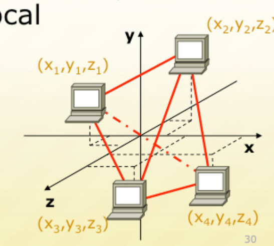
First Key Insight

- With millions of hosts, "What are the $O(N^2)$ network distances?" may be the wrong question
- Instead, could we ask: "Where are the hosts in the Internet?"
  - What does it mean to ask "Where are the hosts in the Internet?" Do we need a complete topology map?
  - Can we build an extremely simple geometric model of the Internet?

29

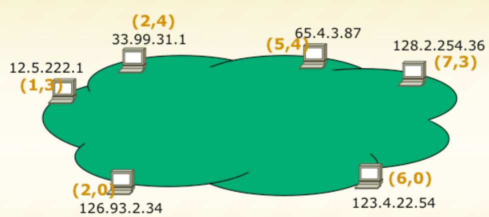
New Fundamental Concept: "Internet Position"

- Using GNP, every host can have an "Internet position"
  - O(N) positions, as opposed to $O(N^2)$ distances
- Accurate network distance estimates can be rapidly computed from "Internet positions"
- "Internet position" is a local property that can be determined **before** applications need it
- Can be an interface for independent systems to interact



30

Vision: Internet Positioning Service

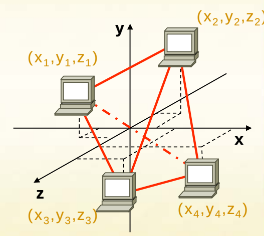


- Enable every host to independently determine its Internet position
- Internet position should be as fundamental as IP address
  - "Where" as well as "Who"

31

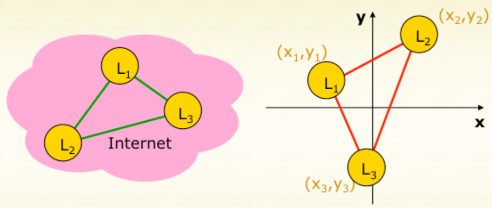
Global Network Positioning (GNP) Coordinates

- Model the Internet as a geometric space (e.g. 3-D Euclidean)
- Characterize the position of any end host with geometric coordinates
- Use geometric distances to predict network distances



32

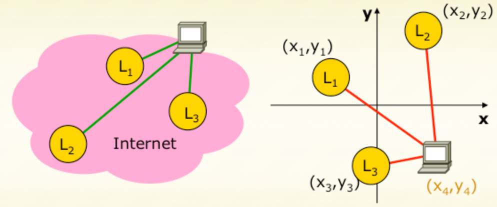
Landmark Operations (Basic Design)



- Measure inter-Landmark distances
  - Use minimum of several round-trip time (RTT) samples
- Compute coordinates by minimizing the discrepancy between measured distances and geometric distances
  - Cast as a generic multi-dimensional minimization problem, solved by a central node

33

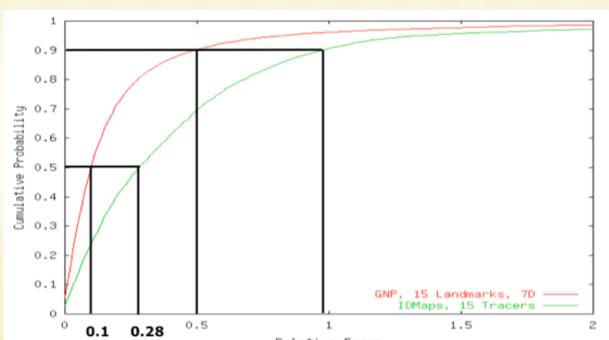
Ordinary Host Operations (Basic Design)



- Each host measures its distances to all the Landmarks
- Compute coordinates by minimizing the discrepancy between measured distances and geometric distances
  - Cast as a generic multi-dimensional minimization problem, solved by each host

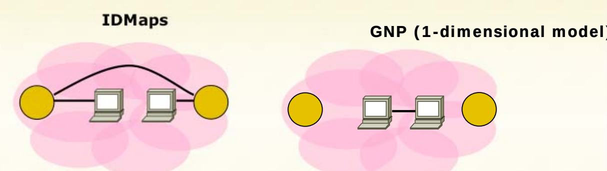
34

Overall Accuracy



35

Why the Difference?



- IDMaps overpredicts

36

Alternate Motivation

- Select nodes based on a set of system properties
- Real-world problems
  - Locate closest game server
  - Distribute web-crawling to nearby hosts
  - Perform efficient application level multicast
  - Satisfy a Service Level Agreement
  - Provide inter-node latency bounds for clusters

37

Underlying Abstract Problems

I. Finding closest node to target
II. Finding the closest node to the center of a set of targets
III. Finding a node that is $< r_i$ ms from target t_i for all targets

38

Meridian Approach

- Solve node selection directly without computing coordinates
  - Combine query routing with active measurements
- 3 Design Goals
  - Accurate: Find satisfying nodes with high probability
  - General: Users can express their network location requirements
  - Scalable: O(log N) state per node
- Design Tradeoffs
  - Active measurements incur higher query latencies
  - Overhead more dependent on query load

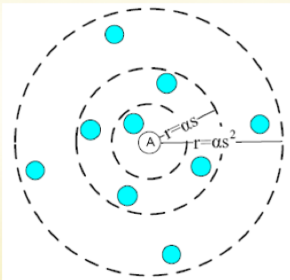
39

Multi-resolution Rings

- Organize peers into small fixed number of concentric rings
- Radii of rings grow outwards exponentially
- Logarithmic number of peers per ring
- Retains a sufficient number of pointers to remote regions

40

Multi-resolution Ring structure



For the i^{th} ring:
Inner Radius $r_i = \alpha s^{i-1}$
Outer Radius $R_i = \alpha s^i$
α is a constant
s is multiplicative increase factor
$r_0 = 0$, $R_0 = \alpha$
Each node keeps track of finite rings

41

Ring Membership Management

- Number of nodes per ring represents tradeoff between accuracy and overhead
- Geographical diversity maintained within each ring
- Ring membership management run in background

42

Gossip Based Node Discovery

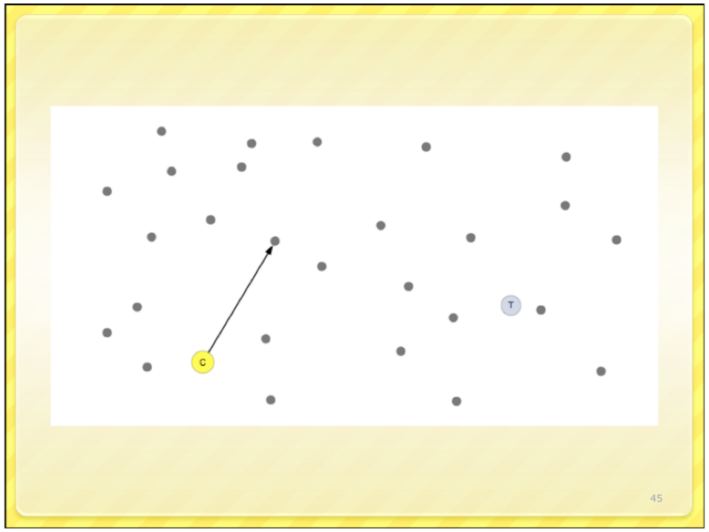
- Aimed to assist each node to maintain a few pointers to a diverse set of nodes
- Protocol
  1. Each node A randomly picks a node B from each of its rings and sends a gossip packet to B containing a randomly chosen node from each of its rings
  2. On receiving the packet, node B determines through direct probes its latency to A and to each of the nodes contained in the gossip packet from A
  3. After sending a gossip packet to a node in each of its rings, node A waits until the start of its next gossip period and then begins again from step 1

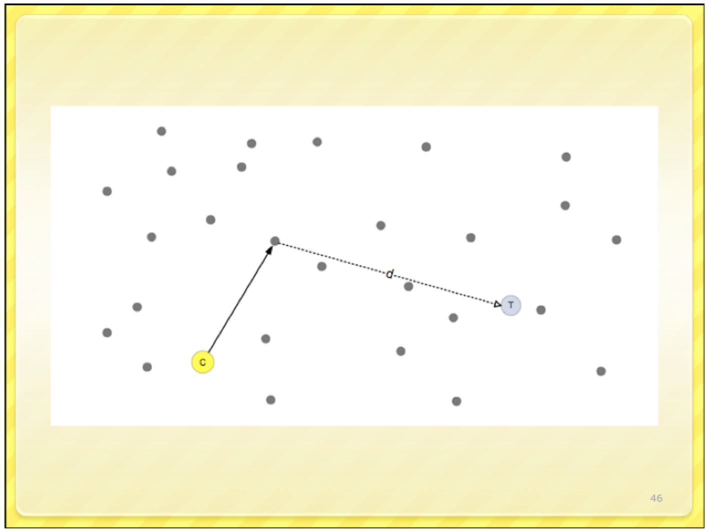
43

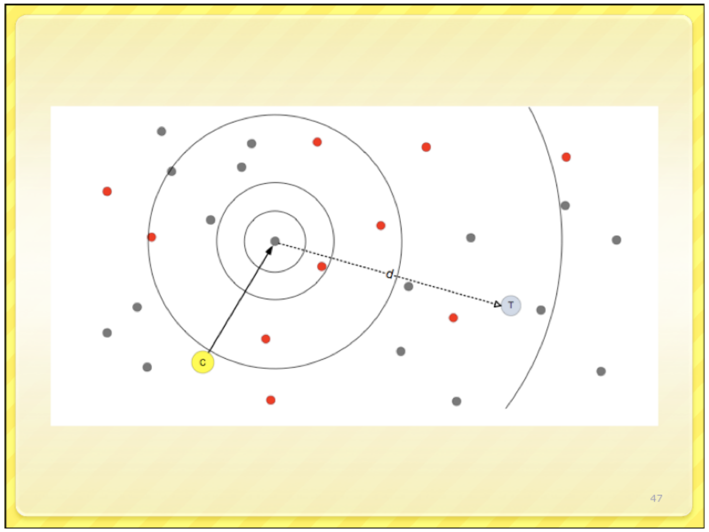
Closest Node Discovery

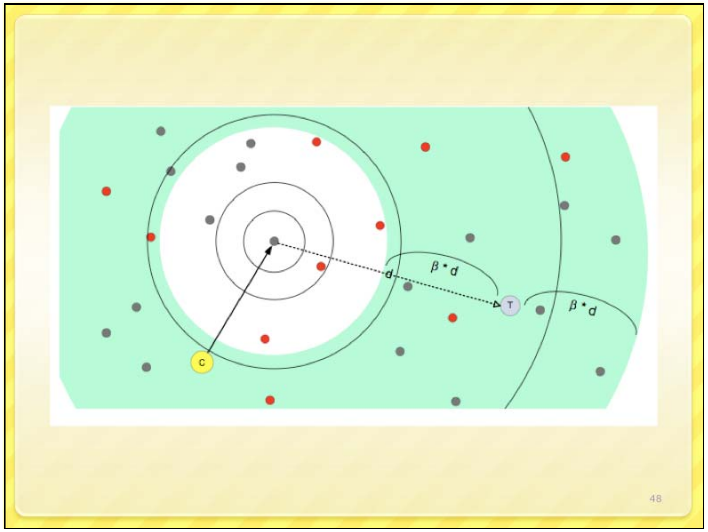
- Client sends closest node discovery request for target T to Meridian node A
- Node A determines latency to T, say d
- Node A probes its ring members within distance $(1-\beta)\cdot d$ to $(1+\beta)\cdot d$, where β is the acceptance threshold between 0 and 1
- The request is then forwarded to closest node discovered that is closer than β times the distance d to T
- Process continues until no node that is β times closer can be found

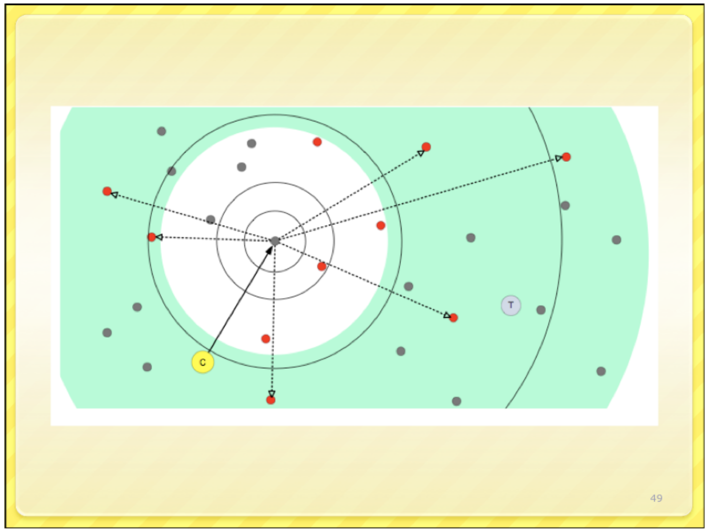
44

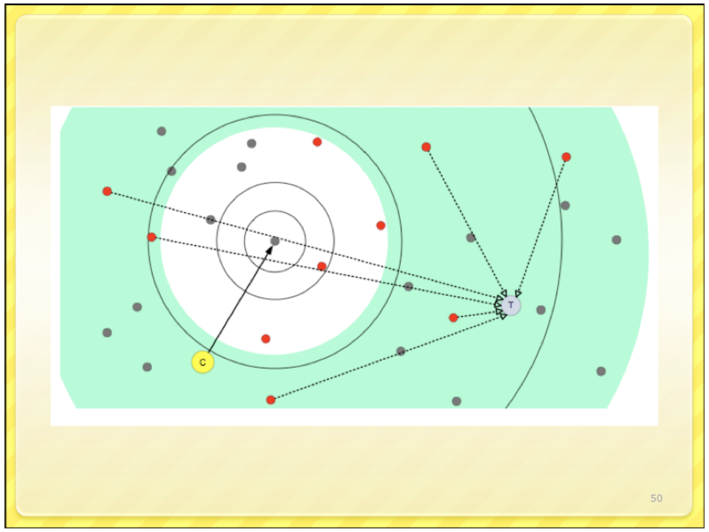

C
T
45

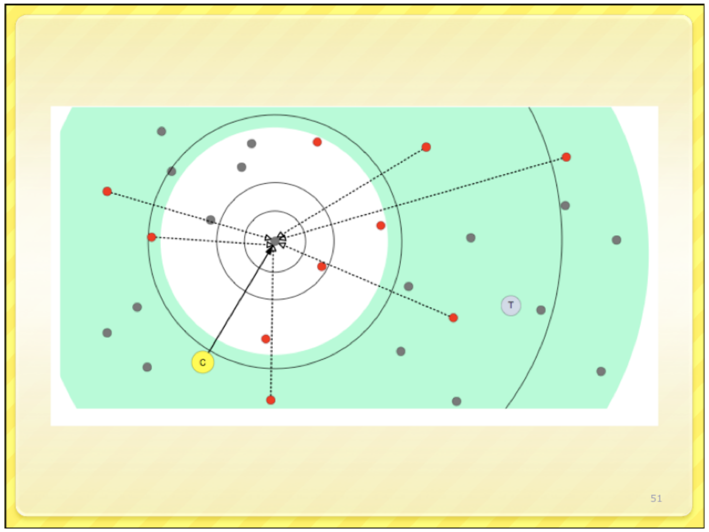

C
d
T
46

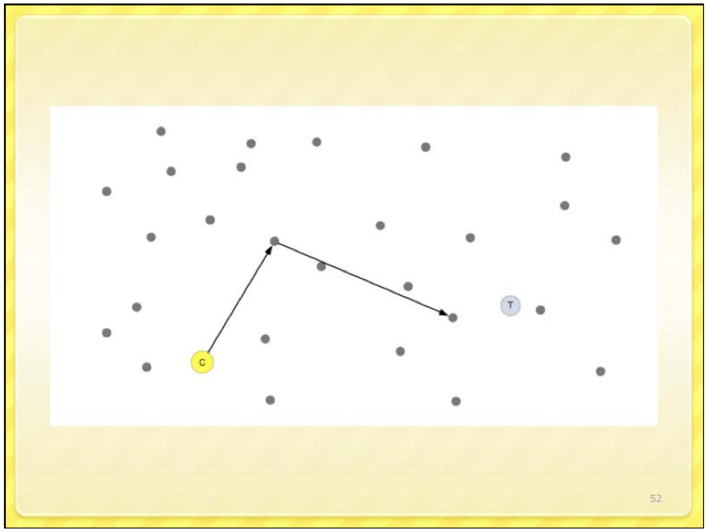

C
d
T
47

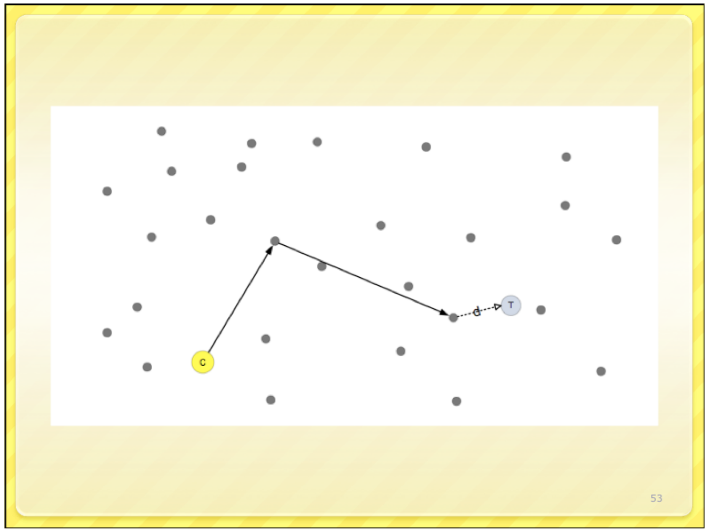

C
d
β * d
T
β * d
48

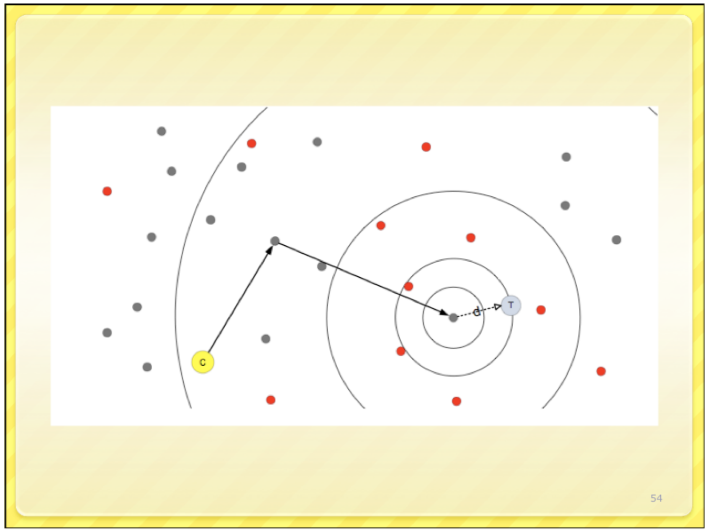

C
T
49

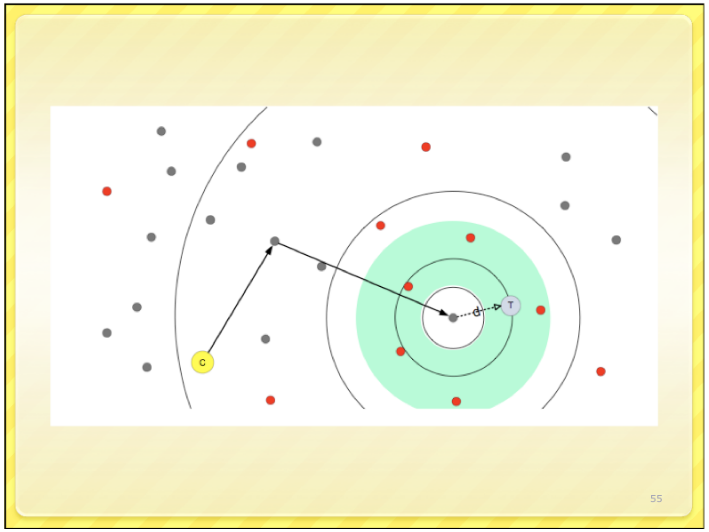

C
T
50

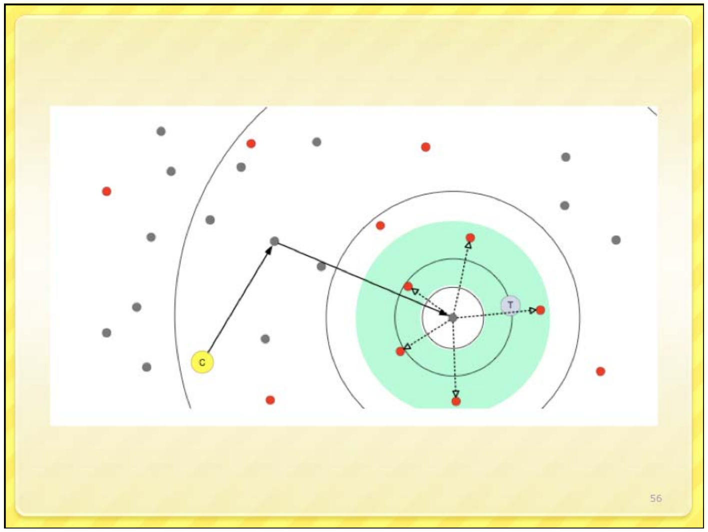

C
T
51

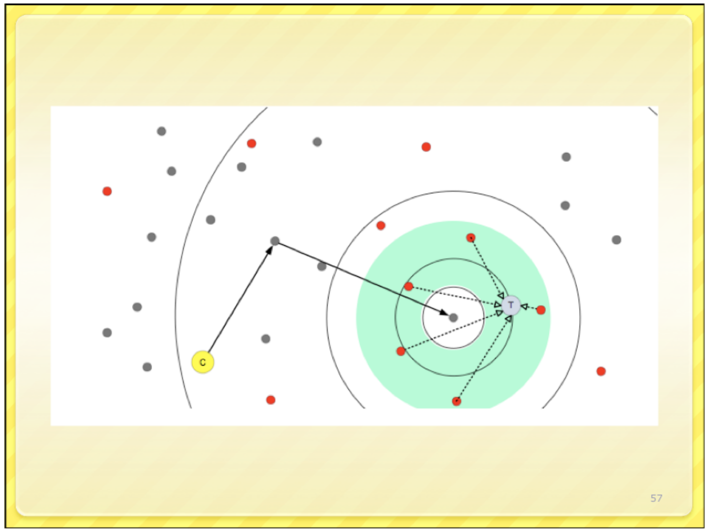

C
T
52

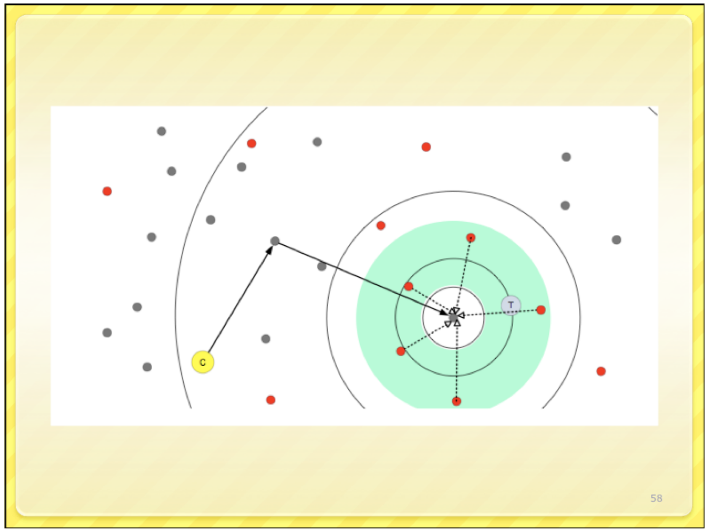

C
T
53

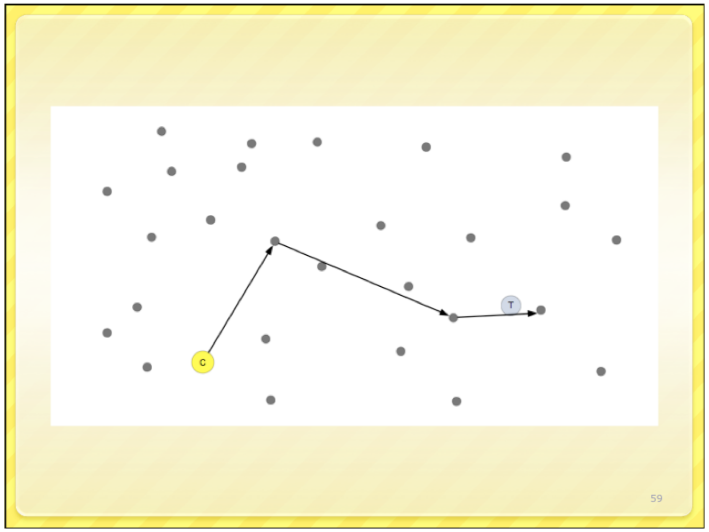

C
T
54

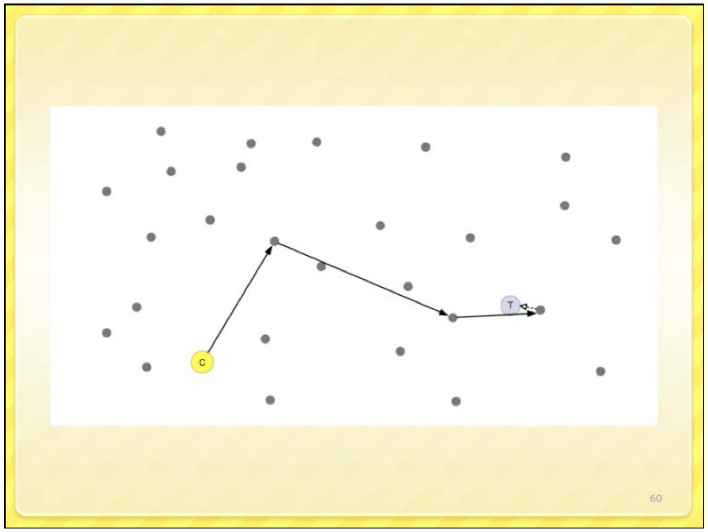

C
T
55

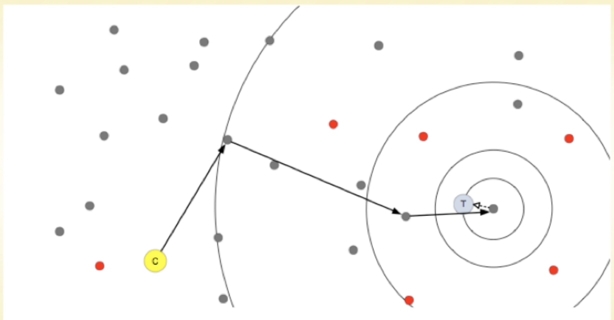

C
T
56


C
T
57

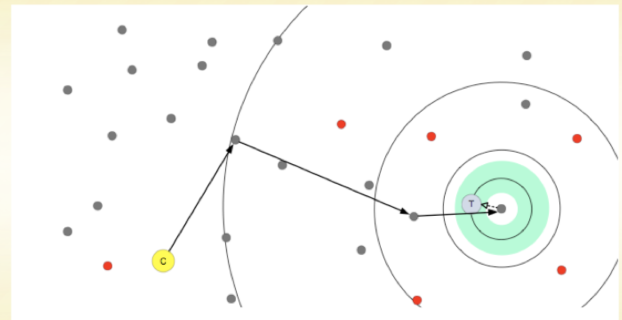

C
T
58


C
T
59

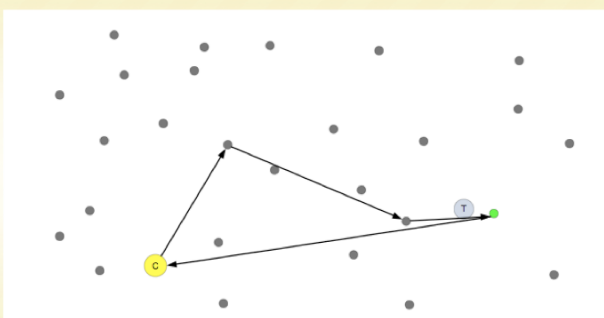

C
T
60



61



62



63

Revisit: Why is Automated Adaptation Hard?

- Must infer Internet performance
  - Scalability
  - Accuracy
  - Tradeoff with timeliness
- Support for a variety of applications
  - Different performance metrics
  - API requirements
- Layered implementations hide information

64