Complex networks

Navigation on networks

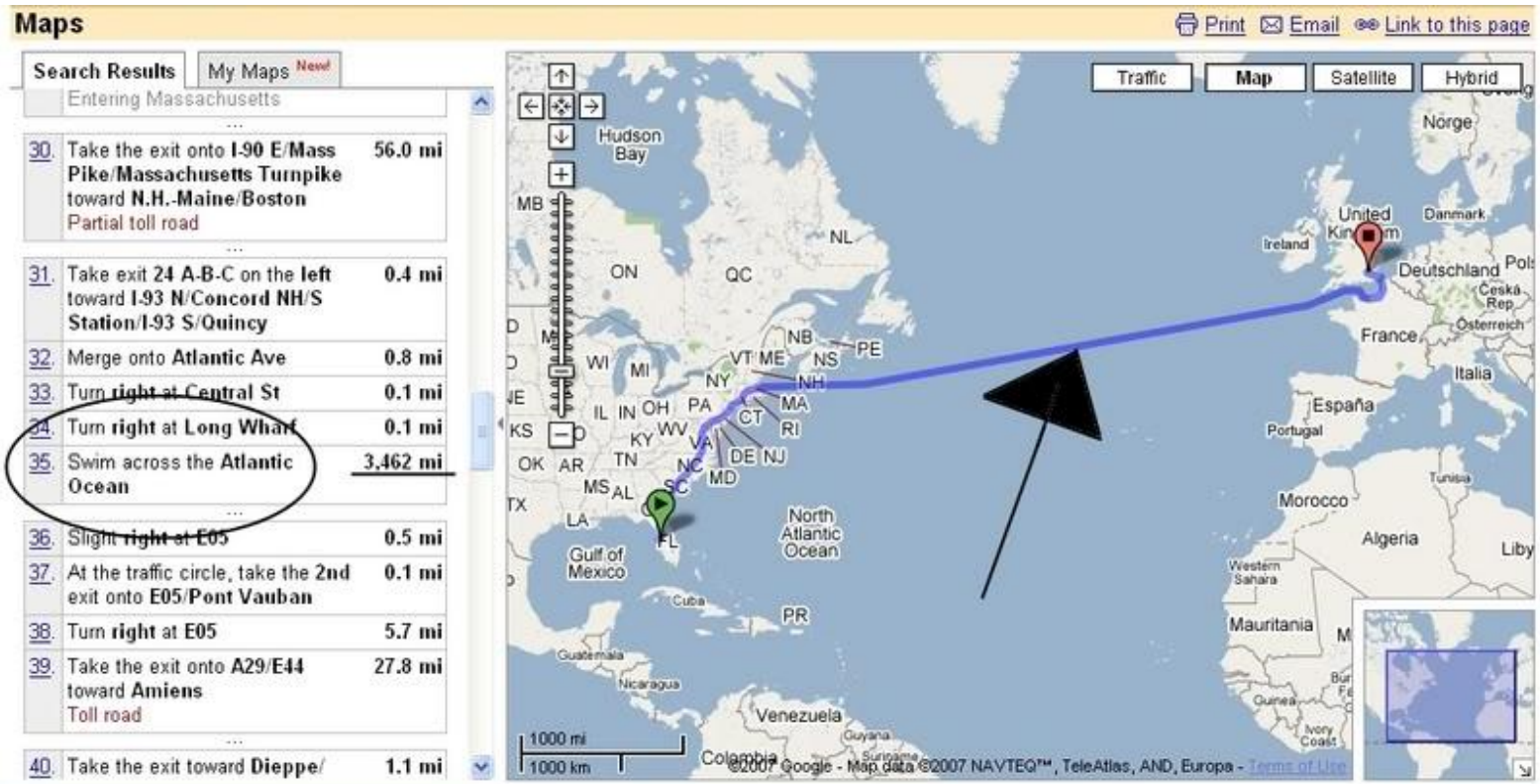
János Török

Department of Theoretical Physics

May 22, 2020

Search on graphs

- Shortest path algorithm
  - Many applications: e.g. Route planning
  - Calculation of *betweenness centrality*
  - Global information needed



Dijkstra's algorithm

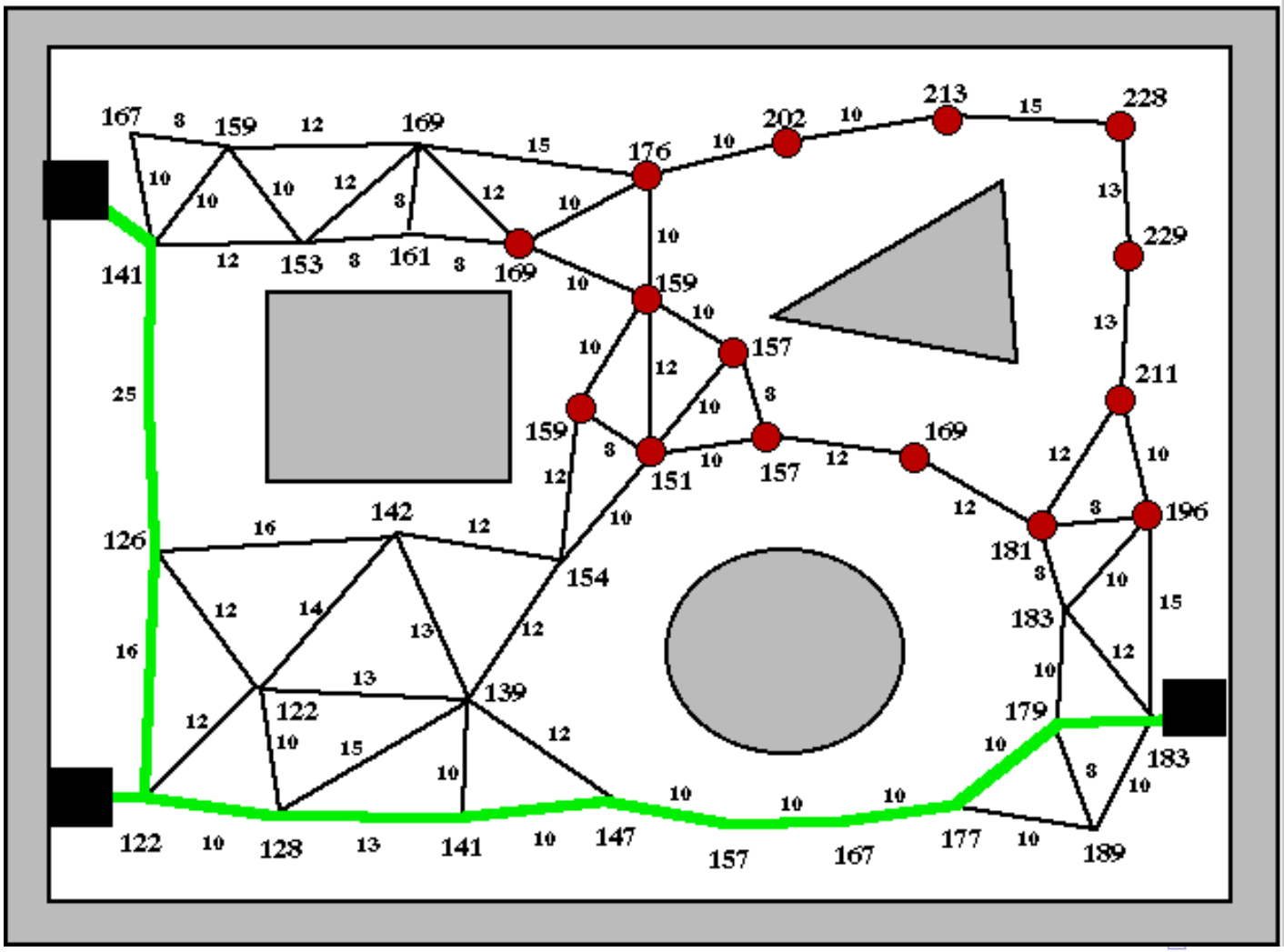
- Find the shortest path from a source
- Known: links, link weights (node distances)
- Store: distance to that point, link to previous element in shortest path
- List of unvisited path sorted by distance to origin (set to infinity if unknown)
- Algorithm:
  1. Choose the unvisited node with the smallest distance to the origin
  2. Visit all its unvisited neighbors: if distance is smaller than the current distance to that point, store it and set link to previous element to the current active node
  3. Mark node as finished
  4. If list of unvisited nodes is not empty, go to 1.

Related problems

- Finding out of a labyrinth
- Search path with local knowledge
  - Very important!
  - Global optimization can be too expensive
  - Global structure may net be known, or varies fast
- Recommender systems
- File sharing

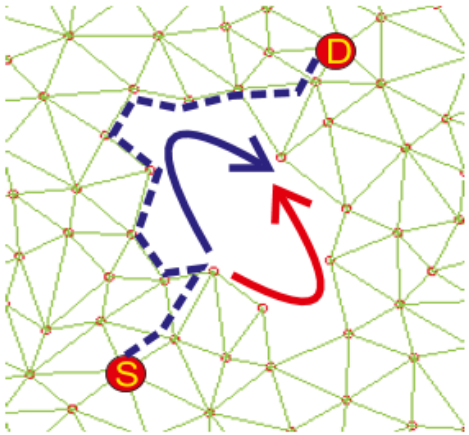
Path search

- Milgram experiment
- There are short paths (topology)
- They can be found using local info (search)



Greedy routing

- Agents have only local information
- They know how far their neighbors are from the target
- They forward the packet to the neighbor with the smallest distance to the target
- May lead to dead end
- Navigability:
  - Fully: The network is navigable if there exists a greedy path between all pairs of nodes
  - Fractional p_s: the fraction of node pairs with greedy route between them



Navigability of scale free networks

- Scale free network (configuration model) $P(k) \sim k^{\gamma}$
- Metric space is needed, here nodes are randomly placed on a ring
- Probability of connection:

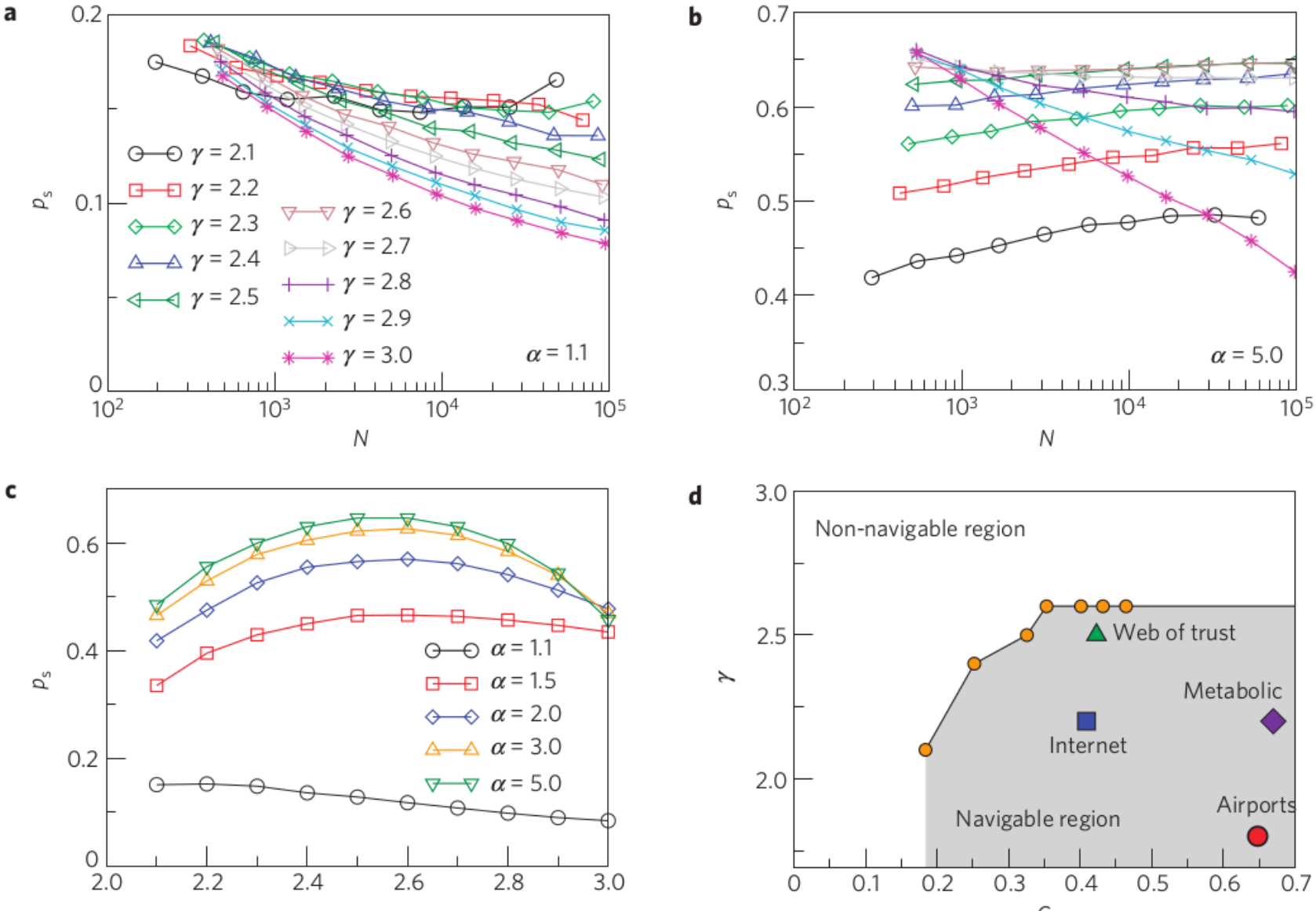
$$r(g; k, k') = \left(1 + \frac{d}{kk'}\right)^{-\alpha}$$

- the probability of link connection between two nodes is decreases with the distance as $\sim d^{-\alpha}$
- Increases with their degrees as $\sim (kk')^{\alpha}$
- Measure: Greedy navigation success rate p_s

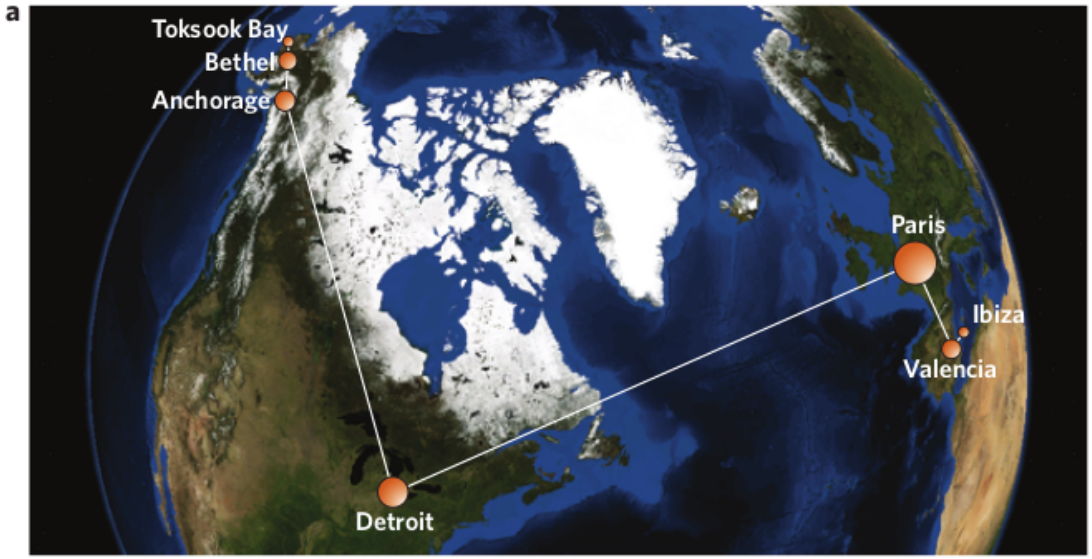
Boguñá et al. Navigability of complex networks, (2009)

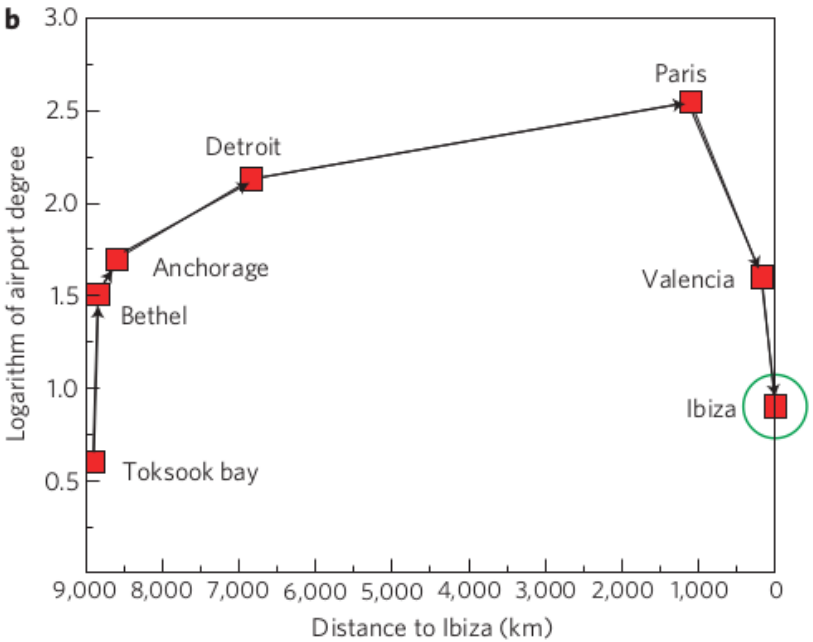
Navigability of scale free networks

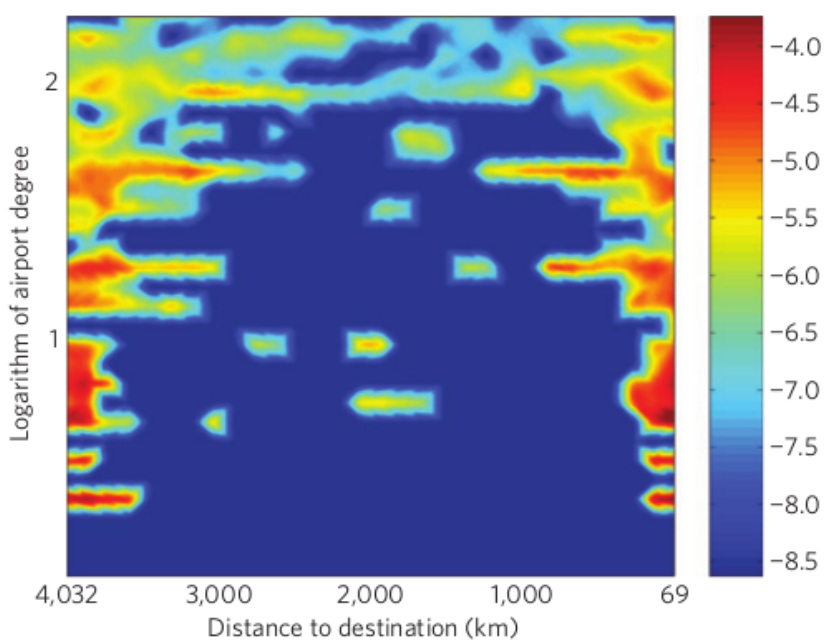
- Navigable if $p_s(N)$ increases with N
- C is clustering for given α



Navigability of scale free networks: Airport example

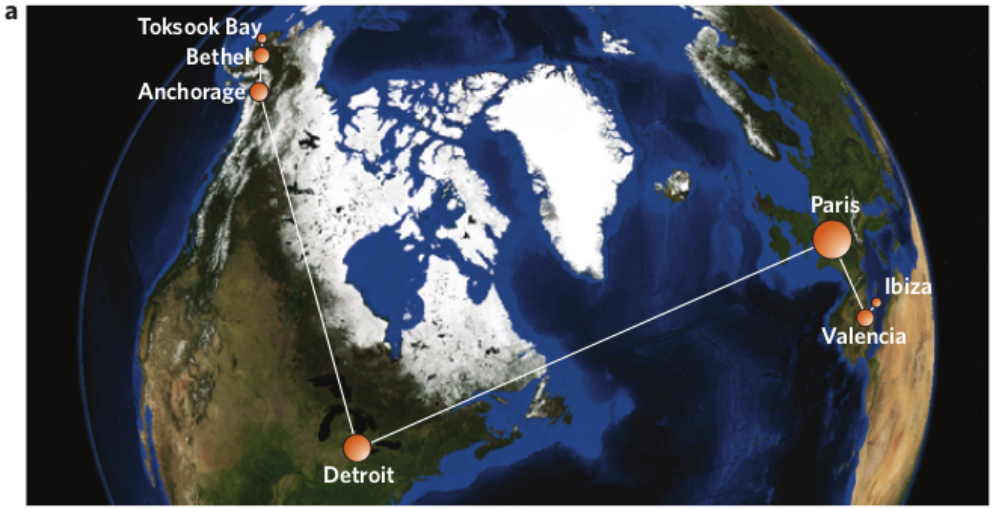


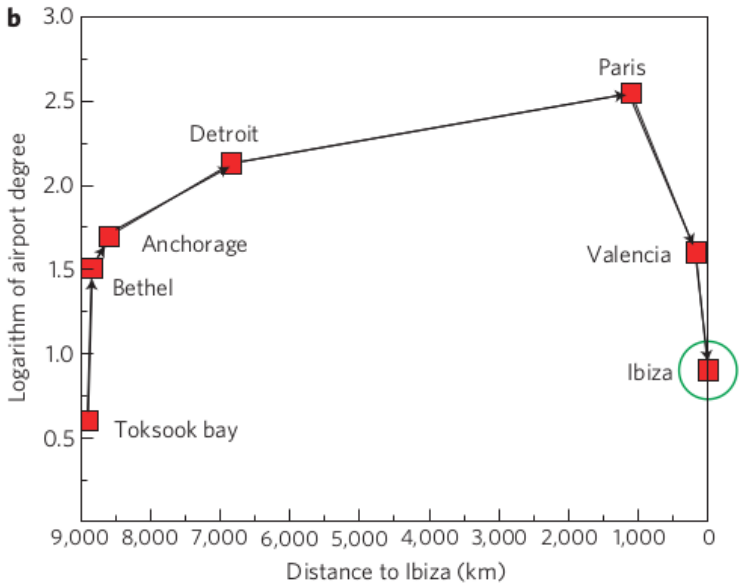


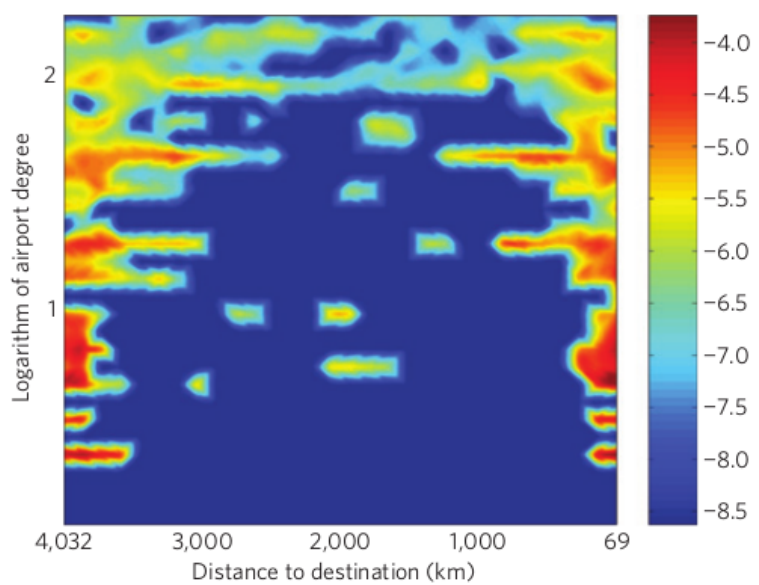


Navigability of scale free networks: Airport example

- General greedy routes: generally go though large degree nodes

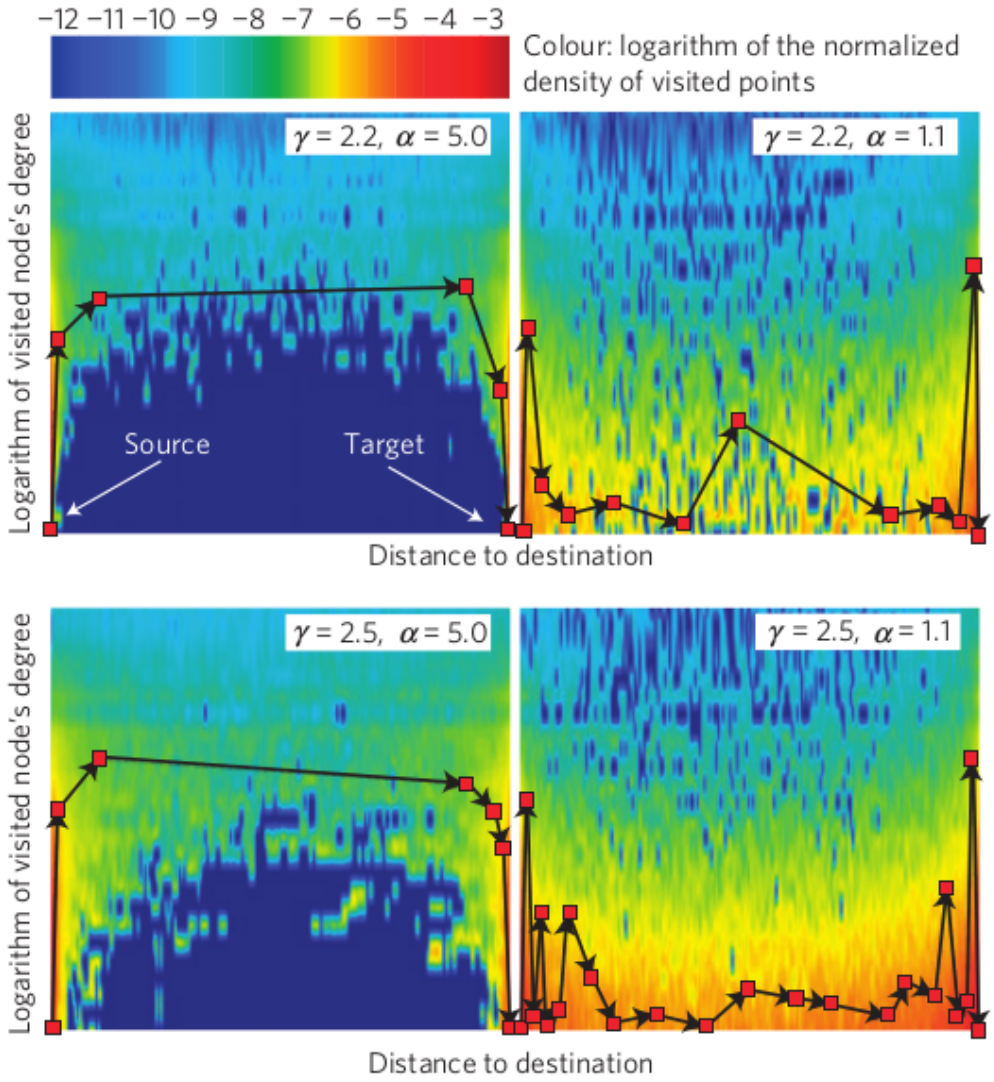






Navigability of scale free networks: Airport example

- Results of the model
- C increases with α



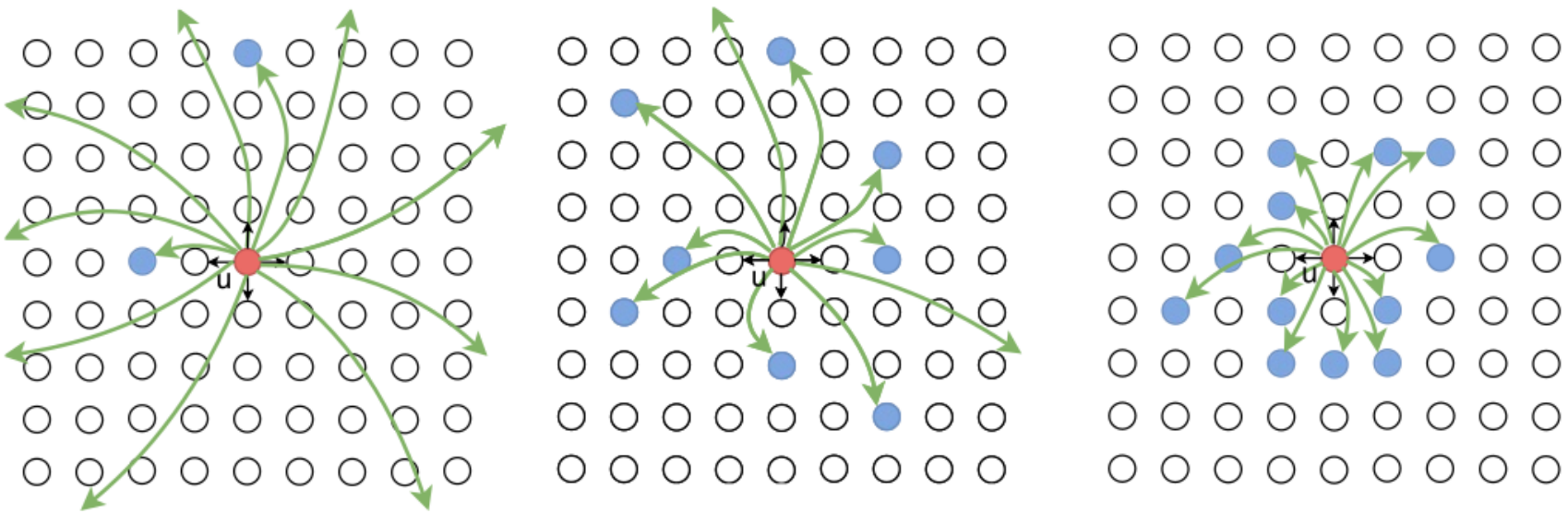
d 3.0

Kleinberg model

- Square lattice, with next nearest neighbor links
- Distance is defined in lattice (Manhattan) distance
- One long range link to a randomly selected node with probability proportional to $r^{-\alpha}$ (here also r is measured in Manhattan distance)
- Expected behaviour

$$T \sim L^x$$

- In small word we expect $x \to 0$ and thus $T \sim \log^y L$



(a) $r << 2$ (b) $r \sim 2$ (c) $r >> 2$

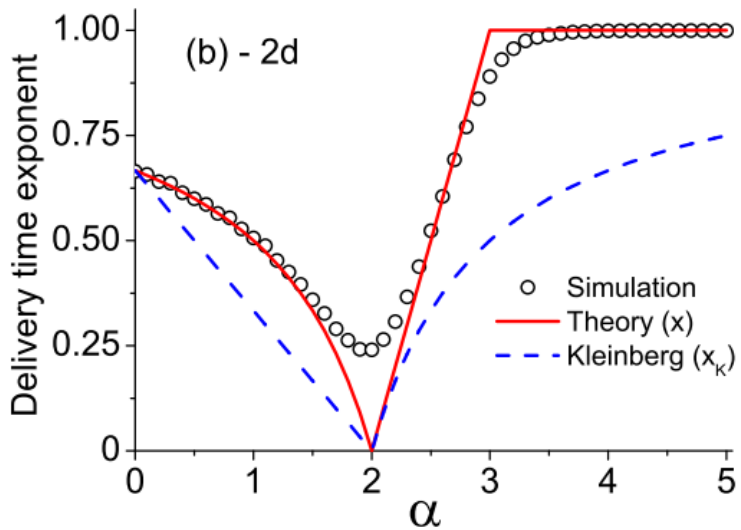
Kleinberg model

- Expected behaviour $T \sim L^x$
- Kleinberg lower bounds (2d):

$$x = \begin{cases} (2-\alpha)/2 & 0 \leq \alpha < 2 \\ (\alpha-2)/(\alpha-1) & \alpha > 2 \end{cases}$$

- Master equation

$$x = \begin{cases} \frac{d-\alpha}{d+1-\alpha} & 0 \leq \alpha < d \\ \alpha - d & d \leq \alpha < d+1 \\ 1 & \alpha > d+1 \end{cases}$$



Search strategies: Network

- What if there is no underlying metric?
- The position of the target is unknown
- Networks
  - Power law degree distribution with exponent between 2 and 3
  - Random weights on the links: smaller weights correspond to shorter paths
  - No global information: each node has information about its neighbors (or second neighbors)
  - structure may change in time

Thadakamalla et al. Search in weighted complex networks (2006)

Search strategies

- Strategies:
  - Random walk
  - (Semi) Self avoiding random walk (do not send the package back to the one from which it was received)
  - Self avoiding random walk, do not send back to nodes where packed already has been. (can lead to dead ends!)
  - Pass through the link with the smallest weight (at least it is not expensive)
  - Choose the best connected neighbor (we saw in the metric version that it is not a bad idea)
  - Choose the neighbor with the smallest average link weight (it is close to many)
  - Choose neighbor with the highest link betweenness centrality (use all available information)

Search strategies: Results

- Random graph:

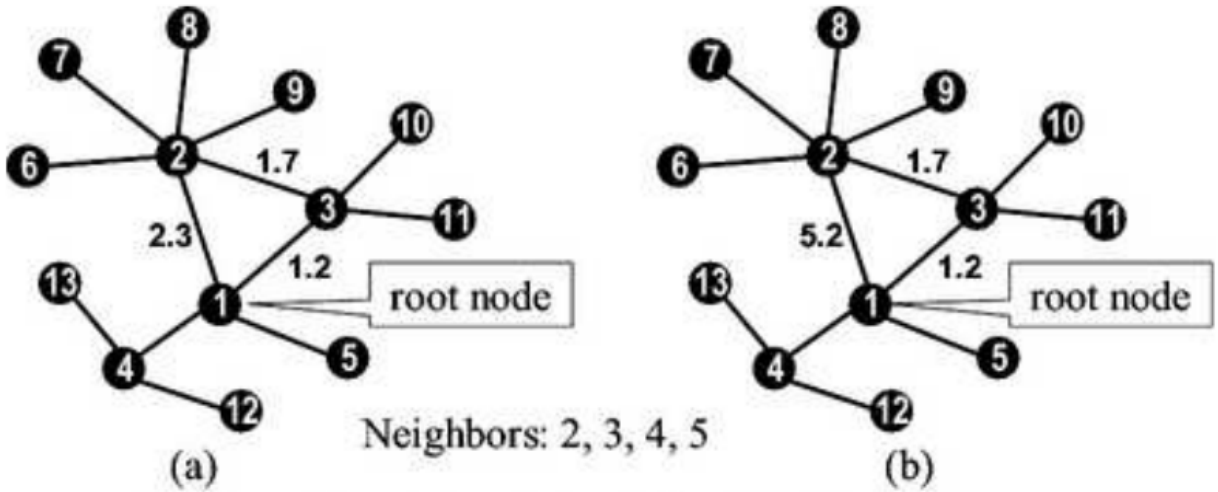
Search strategy	Beta $\sigma^2 = 2.3$	Uniform $\sigma^2 = 8.3$	Exp. $\sigma^2 = 25$	Power-law $\sigma^2 = 4653.8$
Random walk	1271.91	1284.9	1253.68	1479.32
Minimum edge weight	1017.74	767.405	577.83	562.39
Highest degree	994.64	1014.05	961.5	1182.18
Minimum average node weight	1124.48	954.295	826.325	732.93
Highest LBC	980.65	968.775	900.365	908.48

- Scale free network:

Search strategy	Beta $\sigma^2 = 2.3$	Uniform $\sigma^2 = 8.3$	Exp. $\sigma^2 = 25$	Power-law $\sigma^2 = 4653.8$
Random walk	1107.71 (202%)	1097.72 (241%)	1108.70 (272%)	1011.21 (344%)
Minimum edge weight	704.47 (92%)	414.71 (29%)	318.95 (7%)	358.54 (44%)
Highest degree	379.98 (4%)	368.43 (14%)	375.83 (26%)	394.99 (59%)
Minimum average node weight	1228.68 (235%)	788.15 (145%)	605.41 (103%)	466.18 (88%)
Highest LBC	**366.26**	**322.30**	**298.06**	**247.77**

Search strategies

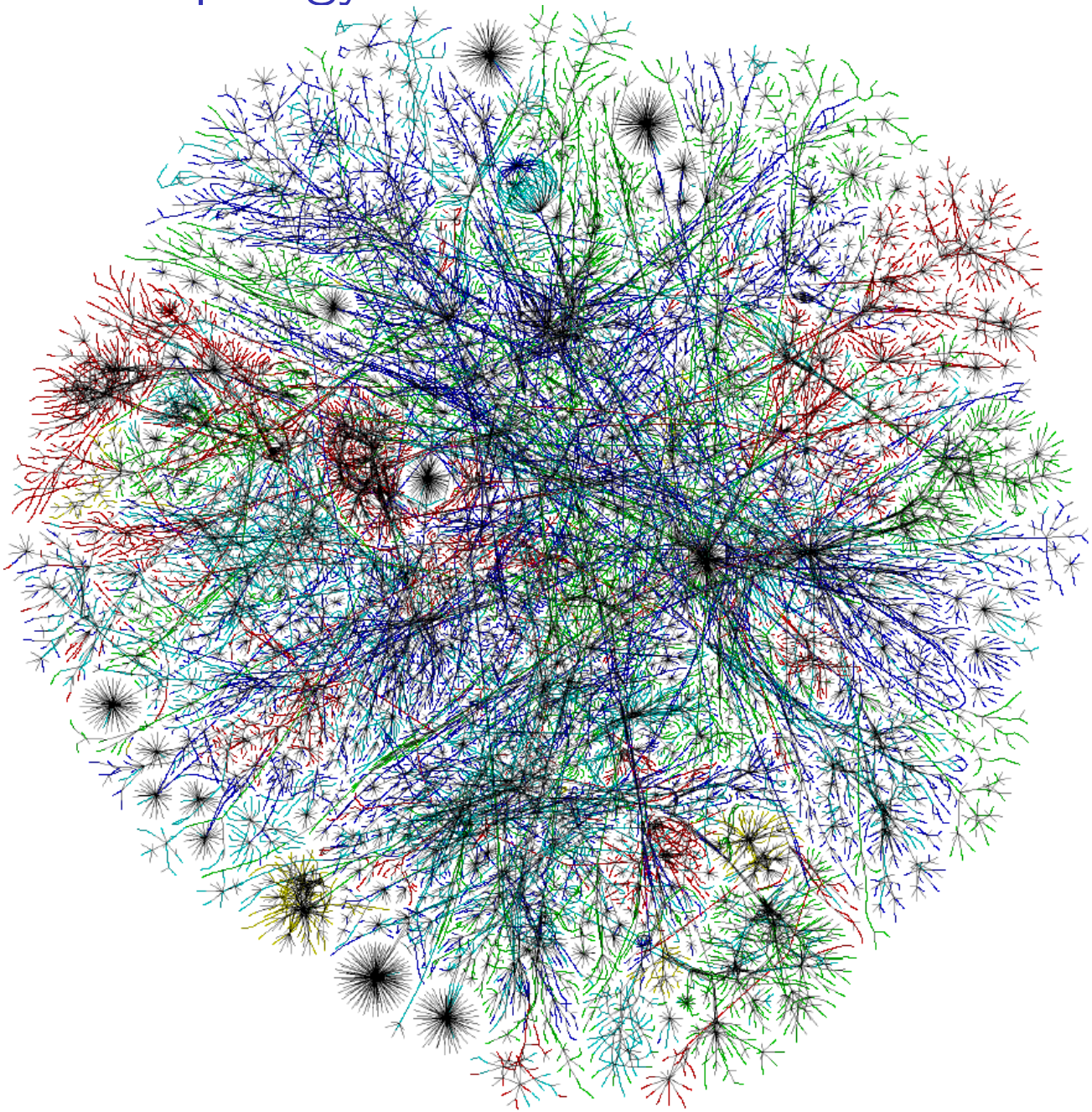
- Strategies:
  - If heterogeneity is small the best performing method is the minimum weight search, which outperforms methods using more information
  - If link weights get homogeneous ($\sigma \sim 1$) then minimum edge weight becomes random walk, highest LBS becomes highest degree and the latter performs better
  - In scale free networks: highest LBS performs best as it incorporates both degree and weight information
- Edge weights not shown is 1



(a) L(2) = **76.0**, L(3) = 42.0, L(4) = 42.0, L(5) = 0.0

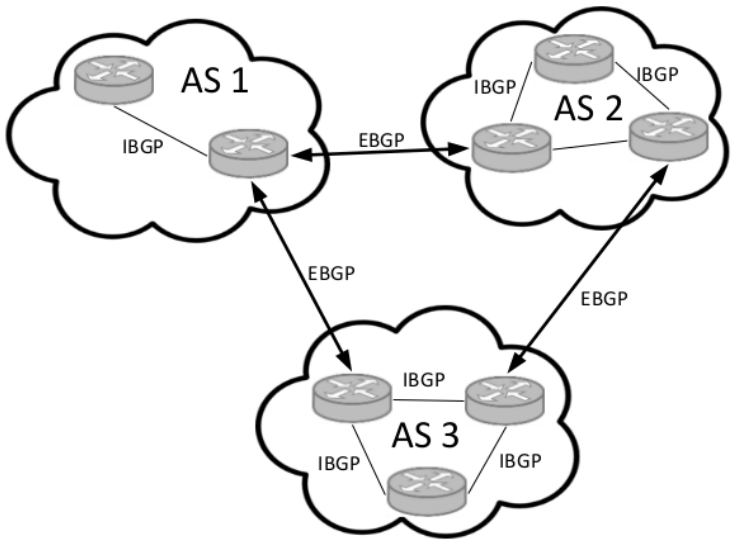
(b) L(2) = 76.0, L(3) = **92.0**, L(4) = 42.0, L(5) = 0.0

Internet topology



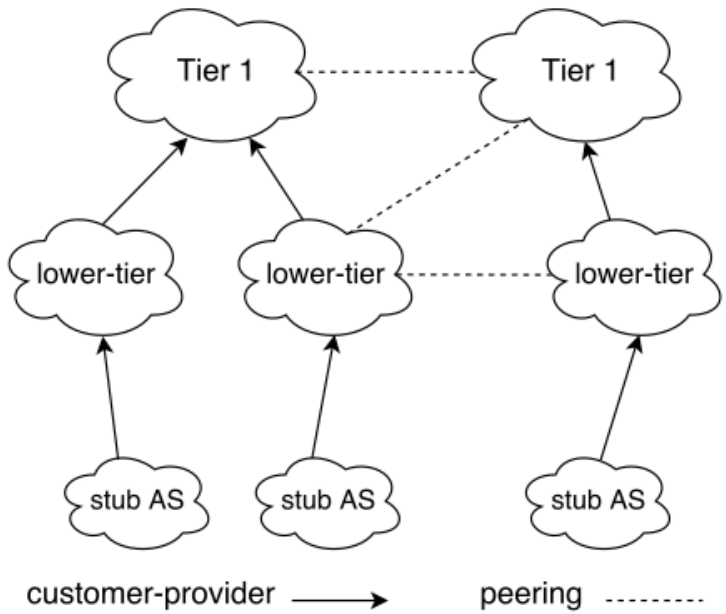
Internet topology

- Autonomous systems (AS) of the Internet
- Routing between AS
- Must be *fully* navigable
- Impossible to know the full structure $\rightarrow$ local routing



Internet topology

- Traffic
  - local
  - transit
- Relationship
  - customer-provider
  - peering

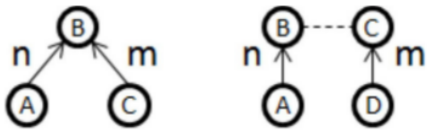


AS routing policy

1. Valley-free route
2. Highest local preference
3. Shortest AS path
4. etc.

AS routing policy

1. Valley-free route
   - Flow of traffic must coincide with the flow of cash
   - Data forward from AS A to B only if
     - incoming traffic if from a costumer of A
     - or B is a costumer of A
     - A valid path contains n customer-provider, at most 1 peer and m provider-customer link strictly in this order



Valley-free

not Valley-free

2. Highest local preference
3. Shortest AS path
4. etc.

AS routing model

- Number of players $\mathcal{P}$
- Edges: (p) directed *provider*, (r) undirected *peer*
- Valley-free routing: u can forward traffic coming from w to v only if
  1. w is a costumer of u (the relationship between u and v can be anything)
  2. v is costumer of u (the relationship between u and w can be anything)
- Payoff (of u):

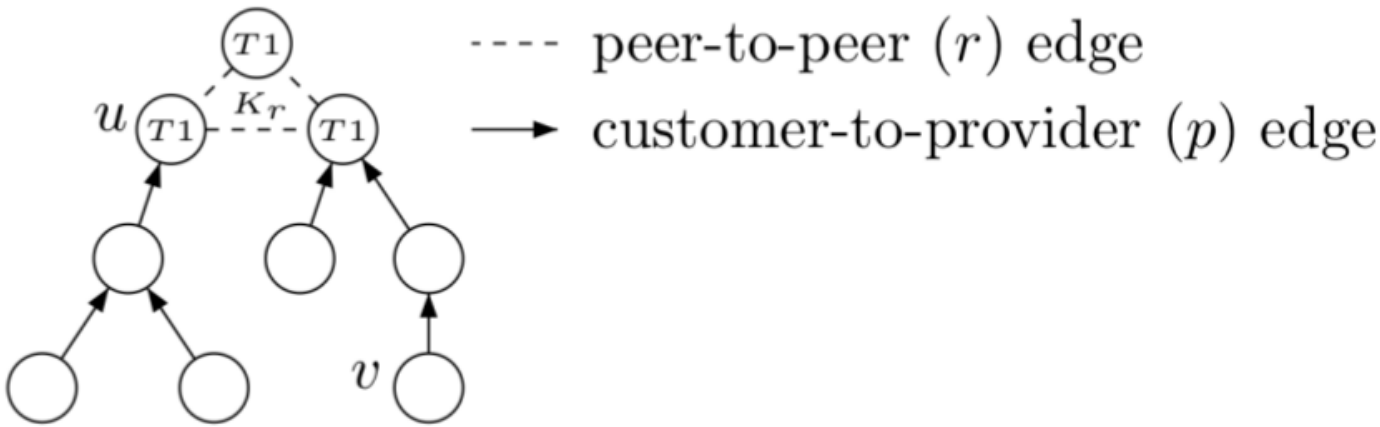
$$C_u = \sum_{v \neq u} d_{VF}(u, v) + \phi_p u_p + \phi_r u_r$$

where ϕ_x is the cost of an edge of type $x \in \{r, p\}$, u_x is he number edges of type x

$$d_{VF}(u, v) = \begin{cases} 0 & \text{There is a VF path between } u, v \\ \infty & \text{otherwise} \end{cases}$$

AS routing model

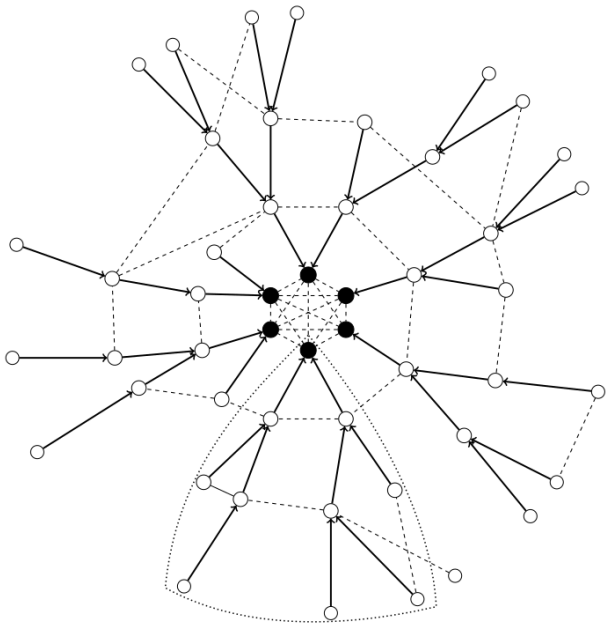
- Number of players $\mathcal{P}$
- Edges: (p) directed *provider*, (r) undirected *peer*
- Valley-free routing between all pairs
- Independent of the cost functions (provided they are positive)
- Resulting network
  - Has a clique core with only peer (r) links
  - Trees rooted at the core consisting exclusively from provider links (p), the provider is always closer to the clique than the consumer



AS routing model

- Include *Highest Local Preference* rule
- Player always picks from the available VF paths according to its local interest
- Players do not like customer-provider links
- Cost function

$$d_{VF}(u,v) = \begin{cases} 0 & \text{VF + first is peer or p} \rightarrow \text{c} \\ 1 & \text{VF + first is c} \rightarrow \text{p} \\ \infty & \text{otherwise} \end{cases}$$



AS routing model

- *Valley-free* rule
- *Highest Local Preference* rule