Adapting to Change
Routing in a Future Internet

Kevin Fall
kfall@acm.org
Routing Problems as Graph Problems

• Euler’s Seven Bridges of Königsberg (1736)
Some Basics on Graphs \((G=(V,E), \ n=|V|, \ m=|E|)\)

- Connected simple graphs \((m = O(n^2))\) – single edges, connected
  - Might be edge-weighted, might be directed or not, or acyclic
- Single Source Shortest path (Dijkstra) on weighted (+) graph: \(O(m + n \log n)\)
  - Bellman-Ford also allows for negative edge weights [not cycles] \(O(nm)\); Also see Yen [1971]
- All-pairs shortest path (Floyd): \(O(n^3)\) [negative edge weights ok]
- MST (Prim): \(O(m + n \log n)\) or \(O(m \log n)\) // MST (Kruskal): \(O(m \log n)\)
  - Chazelle (1991) \(O(m \alpha(m,n))\) \([\alpha\text{ is inverse Ackermann function } \sim \text{ constant } < 4]\)
- BFS and DFS: \(O(m+n)\) [list] or \(O(n^2)\) [adjacency matrix]
- Max flow: \(O(mn)\) [lots of others]; Disjoint SPs (Suurballe) \(O(m + n \log n)\)
- HMM most likely path (Viterbi): \(O(n^2T)\) [T observations, n states]
- NP-complete: Hamilton Circuit, TSP, capacitated MST, longest path, Steiner tree, degree-constrained Steiner tree
Graphs and Routing

• Finding a routing is an assignment \( R \) on \( G(V,E) \) that provides paths \( \pi(s,d) = \{ e_1, ..., e_n \} \) between vertices \( s \) and \( d \); usually with some associated cost \( C(s,d) \) [which is often a sum: \( C(e_1,e_2)+C(e_2,e_3)+... \)]
  • See definition for \( R \) in Brady/Cowen for additional formality

• We’re primarily concerned with the computational cost and possibly memory required to compute paths

• Commonly we compute ‘shortest’ paths from all \( s \) to \( d \) that minimize the costs. This is the All Pairs Shortest Path (APSP) problem.
  • Often a distributed solution... think OSPF or distance-vector
  • This is a form of self-adaptation that operates well given certain limitations
Incrementally Adapting to Change

• Given a collection of shortest path(s) on a graph, what’s the complexity to compute new one(s) if the graph changes?
  • **Fully dynamic** – allows edges to be deleted or added to graph
    • Versus *incremental* or *decremental* (add or delete edges only) which have other algorithms
  • What complexity to answer the questions 1> d(u,v)? and 2> perform an update?
  • Obviously, can always just re-compute as new static graph

• Demetrescu and Italiano (2003) : amortized $O(n^2 \log^3 n)$ – fully dynamic APSP algorithm for digraphs with non-negative edge real edge weights
  • Also: dynamic SPSP at least as hard as static APSP
  • $O(1)$ query time

• Thorup (2005) : $O(n^{2+3/4})$ update complexity (deterministic algorithm)

• Abraham, Chechik, Krinninger (2016) : $O(cn^{2+2/3} \log^{4/3} n)$ w/ prob 1-1/n^c; c>1
Routing in the Graph - Approximations

- Almost-shortest paths can be rather useful as well. A tradeoff:
  - For optimal (shortest), $O(n \log n)$ switch memory required
    - One output ‘port’ (neighbor edge) for every possible destination
  - With smaller memory, must sacrifice something (e.g., stretch/correctness)
  - **Stretch** of $R$ is $\max(C_R(s,d)/C_{opt}(s,d))$ for all costs $C$ on path $(s,d)$ with routing $R$ vs opt [1 is ‘best case’ = optimal]
Compact Routing – (sublinear switch memory with polylog headers)

• Fact: no stretch < 3 universal CR schemes with o(n) at each node
  • Universal – for any graph topology
• Thorup-Zwick (TZ) scheme (2001) for static graphs
  • Delivers stretch-3 max for switch memory $O(n^{1/2})$ [sub-linear...a ha!]
  • More generally, $O(n^{1/k})$ with stretch $4k-5$ ($k>1$)
• Chechik (2013) for weighted undirected static graphs
  • $O(n^{1/k})$ with stretch $ck$ (for $c < 4$) [so better than TZ for $k>=4$]
• Abraham (2004) – Name-Independent Compact Routing
  • Achieves $O(n^{1/k})$ w/stretch $O(k)$
Special Graphs

• Particular graphs have special routings
  • Some are regular special cases like grids/lattices, trees
  • Or Erdős–Rényi random graphs, etc... but some aren’t

• Of particular interest are ‘complex networks’ or graphs
  • The related ‘small world’ phenomena was studied rigorously through the 60s
    • Recall the 1969 Milgram experiment (many letters traveled on shortest paths)
    • US population ~ small world graph -> “three degrees of separation”
  • Heavy-tailed degree distribution, high clustering coefficient
  • (dis)/Assortativity among vertices, community and hierarchical structures
  • In technical networks, mostly dis-assortative

• Feb 2018 -> WikiPedia pages avg separation degree is 3.019

Really? (see sixdegreesofwikipedia.com)

Pope ↔ Alternator
Routing on Special Graphs

• The Internet’s inter-AS topology graph “appears” to be scale free: with power-law degree and clustering coefficient distributions
  • Intuitively: relatively common to have very high-degree vertices
  • Low-degree nodes belong to very dense subgraphs which are connected to each other by ‘hubs’ (high degree vertices)
  • Arguably responsible for the ‘small world’ phenomenon
  • Robust to random vertex failure; fragile to targeted vertex deletion
  • Diameter is $O(\log \log n)$ – very nearly constant

• Scale-freeness is controversial, but that is somewhat an aside here...
Compact Routing on Power-Law Graphs

• Krioukov, Fall, Yang (2004) – CR looks to be good on “Internet” graph
  • TZ scheme shows most paths are stretch one (average about 1.1)
  • Simulation and mathematical result (but no bounds proven)

• Brady and Cowen (2006) – additive stretch
  • $O(e^2 \log n)$ with additive stretch $d$ ($d, e$ are small params of the topology)
    • But with $O(e^2 \log n)$ message addresses too
  • Using exact distance labelings

• Chen, Sommer, Teng, Wang (2009) – CR on power-law graphs
  • Expected size $O(n^g \log n)$ sufficient memory for stretch 3 and $g = (t-2)/(2t-3)$
    where $t$ is the power law exponent of the graph (typ $2 < t < 3$)
  • Requires initial stretch-5 (max) handshake setup
Can We Get a Smaller Distributed Algorithm?

• We can get to O(log n) if we flood – but doesn’t scale well
  • The log n then is essentially our own label

• What if we just greedily “go closer” using some coordinates
  • In the simple geo location case, this is also called Geographic routing
  • Each node need only store locations of neighbors and choose closer one
  • Follows triangle inequality: \(d(a,c) \leq d(a,b) + d(b,c)\)

• ‘Dead Ends’ become a problem – requires backtracking

• If we have a greedy embedding, we can avoid the backtracking
  • That is, a mapping from the topology graph to coordinate assignments such that greedy forwarding ‘just works’ without backtracking
Kleinberg and related results

• There exist planar graphs that do not admit a Euclidean greedy embedding

• Greedy embedding for all graphs in a hyperbolic space (Kleinberg 07)
  • Problem: the labels in doing this directly are large...O(nlogn)... so large the scheme doesn’t really win inherently over non-greedy

• Eppstein and Goodrich (2008) – a succinct greedy embedding
  • Use ‘autocratic (balanced) binary tree’ to assign positions in dyadic tree metric space
  • Effectively ‘discretizes’ (to a grid) in the hyperbolic plane while preserving the overall coarse distance relationships (but not the exact points)
Yes, but...

- Practicalities include management, $ costs, etc
  - OSPF includes: hello/flood protocols, areas, authentication, virtual links, designated routers/backups, non-broadcast support, summarization

- Also, if I purchase a link, I want to *use* it...
  - Traffic engineering and policy routing: overriding your routing protocol
  - TE largely for modifying utilization (e.g., load balancing) and policy
  - Match network resources to the traffic (minutes or longer)
    - Stuff like: OSPF/ISIS weights, capacity planning, BGP import policy

- SR-TE (Segment Routing / Traffic Engineering)
  - A generalized mechanism to help evolve from RSVP-TE (which uses RSVP to provision MPLS LSPs); see RFC 8402 [also see RFC8277 - prefix/label bindings]
Changing the Problem

• In the DTN (and ICN) worlds, looked at some different ideas
  • DTN: storage and path selection; routing over time; controlled replication
  • ICN: targets are data objects which reside on topology vertices

• DTN Examples
  • Epidemic, ProPHET, MaxProp, RAPID, Spray & Wait, Bubble Rap, DTLSR, CGR/SABR

• ICN and related examples
  • TRIAD, DONA (crypto addresses), PURSUIT/PSIRP, NetInf (flat), SAIL, CCN/NDN (hierarchical routing)
  • NLSR, DABBER (wireless)
Quantum Communication

- Quantum communication may be useful for several applications
  - Confidential communications physically difficult to intercept/alter
  - Communicating quantum information between quantum computers

- Goal is to distribute ‘as much entanglement’ as possible to users
  - For supporting as high a rate of ‘quantum flow’ as possible
  - For supporting multi-party entanglement (~ quantum multicast)

- Basics: superposition & entanglement

\[
|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad |\alpha|^2 + |\beta|^2 = 1
\]

\[
|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\
|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\
|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\
|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)
\]

Bell states of orthogonal entanglement

And now for something [not quite] completely different…
Quantum Communication Environment

• Qubit – a quantum bit

• Setting: transport entangled qubits in (optical) network through quantum switches (that need to preserve coherence & store qubits)
  • Qubits encoded using polarized photons, trapped ions or superconductors
    • Superconductors are usually “cold,” but on Feb 21 2019 USPTO made public a US Navy patent application for a room-temperature superconductor – we shall see!

• Challenge: the fidelity of the quantum state can erode in the environment
  • When transmitted or when stored
Quantum Network Link

• Can send a qubit along a (photonic) network path: quantum link

• Distance limitation comes from several sources
  • Degradation of fidelity as a function of distance (loss of coherence)
  • Inability to ‘just copy’ as in classical memories due to no-cloning theorem
    • So simple forms of classical error correction/detection do not readily apply
  • Challenge in converting qubit encoding from photon to matter and back

• Distance limitation can be addressed with multiple constituent links
  • This would require a form of quantum repeater or router/switch
  • They can only be placed ~100s or less of kms apart from each other
  • Nodes contain: quantum memories, sources and processors
Entanglement Swapping

• Two sources emit entangled qubits (A,B) and (C,D)
• Take a joint measurement (BSM) of one from each source; say (B,C)
  • This will cause the others A,D to fall into an entangled state
• Achieved with distance between sources of 2km using telecom-style fiber optic cables

Riedmatten et al, Physical Review 2005
EPR = Einstein-Podolsky-Rosen
Quantum Error Correction & Fault Tolerance

• One option for quantum communication in noisy channels
• Qubits may suffer from continuous errors (not just bit flips)
  • In particular, moving closer to one basis or changing sign
  • Operations may also introduce errors (limiting this = ‘fault tolerance’)
• Basically, expand into a larger dimensional Hilbert space
  • Measurements collapse superposition w/out affecting quantum information
  • Error discretization can allow a finite syndrome to perform a correction
• Threshold theorem – ‘good enough’ gates are effectively error free
  • The basis for fault tolerance
Quantum Routing

• Basic approach: compute shortest paths, use entanglement swapping to extend links to all necessary (s,d) pairs, employ QEC and/or teleportation and purification

• Better result: multi-path routing has better rate-vs-distance scaling

• Parameters of interest: G (topology graph), p (Pr{quantum link established in a time step}), q (Pr{successful Bell measurement}), S (# parallel links in edge), T (# of time slots before stored decoherence)
Multi-Path Quantum Routing

- $X,Y =$ locations of Alice, Bob [on a grid]
- $R_g(p,q) =$ global knowledge
- $R_{loc} =$ local knowledge
- $R_{lin} =$ linear cascade of repeaters
- $p =$ Pr{link establishment}
- $q =$ Pr{successful measurement}

Pant et al, arXiv:1708.07142v2, 9/2017
Conclusions

• Incremental computation for dynamic graphs updates can be a significant win but progress has taken considerable time/effort

• Giving up strict optimality admits many more efficient options that might be very close to optimal nonetheless (compact routing)

• Mapping the topology path problem to another form (greedy embedding) may allow for even more efficient approaches

• Adaptation in a quantum world (of non-local effects) opens up some new ways of thinking and opportunities and significant challenges
Thanks

kfall@acm.org