The Bellman-Ford Algorithm

Internet Routing

Algorithms: Design and Analysis, Part II

From Bellman-Ford to Internet Routing

Note: the Bellman-Ford algorithm is intuitively "distributed".

Toward a routing protocol:

① switch from source-driven to destination-driven
[just reverse all directions in the Bellman-Ford algorithm]

- every vertex v stores shortest-path distance from v to destination t and the first hop of a shortest path
[for all relevant destinations t]

("distance vector protocols")

Tim Roughgarden

Handling Asynchrony

② Can't assume all A[i,v]'s get computed before all A[i-1,v]'s

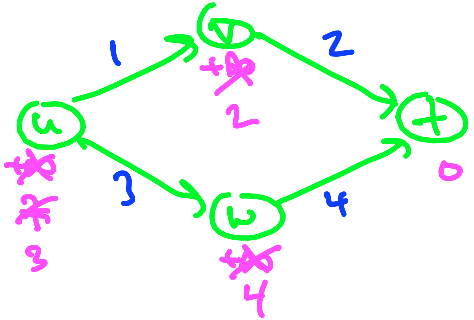
fix: switch from "pull-based" to "push-based": as soon as A[i,v] < A[i-1,v], v notifies all of its neighbors.

fact: algorithm guaranteed to converge eventually. (assuming no negative cycles)

[reason: updates strictly decrease sum of shortest-path estimates]

⇒ RIP, RIP2 Internet routing protocols very close to this algorithm [see RFC 1058]

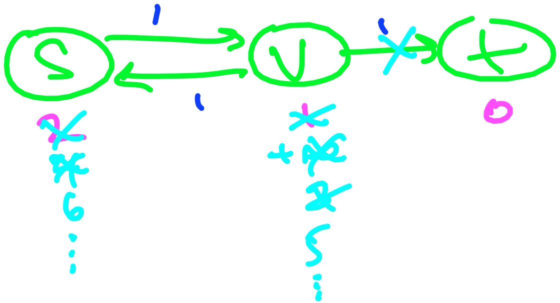
Example



Handling Failures

Problem: convergence guaranteed only for static networks (not true in practice).

Counting to Infinity:



Fix: each v maintains entire shortest path to t, not just the next hop.

"path vector protocol"

"Border Gateway Protocol (BGP)"

Con: more space required.

Pro#1: more robust to failures.

Pro#2: permits more sophisticated route selection (e.g., if you care about intermediate stops).

Tim Roughgarden