The Bellman-Ford Algorithm

Algorithms: Design and Analysis, Part II

Internet Routing

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")

Handling Asynchrony

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

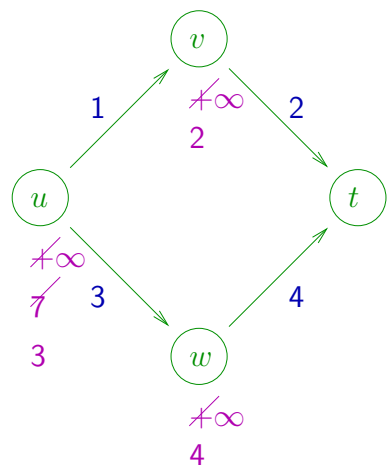
Fix: Switch from "pull-based" to "push-based": As soon as $A[i, v] < A[i-1, v]$, v notifies all of its neighbors.

Fact: Algorithm guaranteed to converge eventually. (Assuming no negative cycles)

[Reason: Updates strictly decrease sum of shortest-path estimates]

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

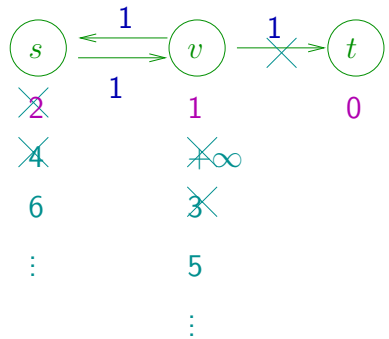
Example:



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).