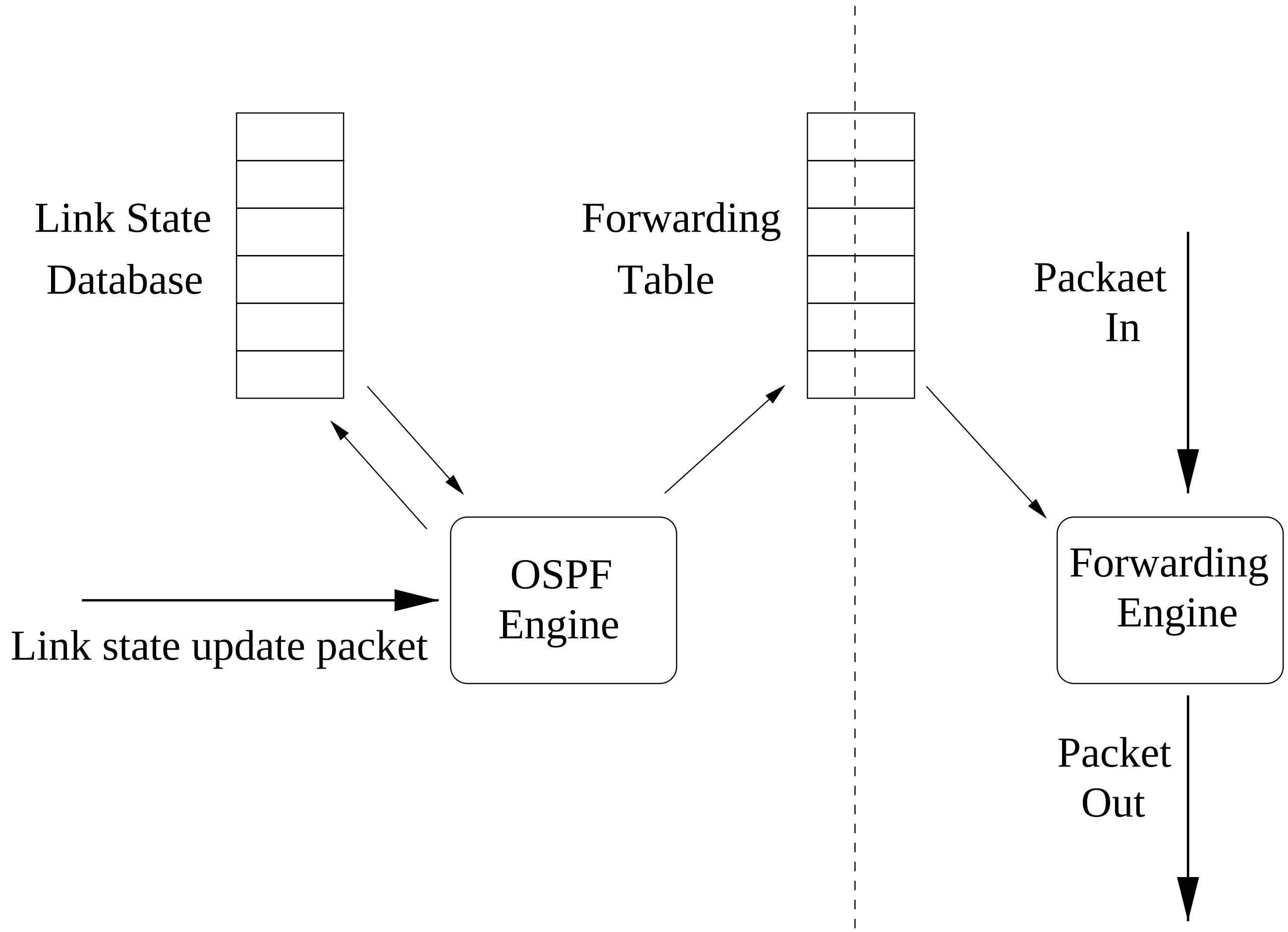
Traffic Engineering in IP Networks

Prof. Kai-Yeung (Sunny) Siu
Massachusetts Institute of Technology

Traffic Engineering in IP Networks

- **Key Ideas - adaptive routing, increased robustness, faster convergence time.**
- **A fast algorithm to recompute a shortest path tree.**
  - Uses information on the previous shortest path tree.
- **An efficient data structure to compute multiple viable paths.**
  - Metric based approach which ranks the desirability of each path.
  - Modify traffic behavior on a small time scale.
  - Allows efficient load balancing and avoids route flapping.
- **A robust protocol to reroute traffic with minimum communication overhead.**
  - Keeps information on the network conditions in the local neighborhood.

How OSPF Works



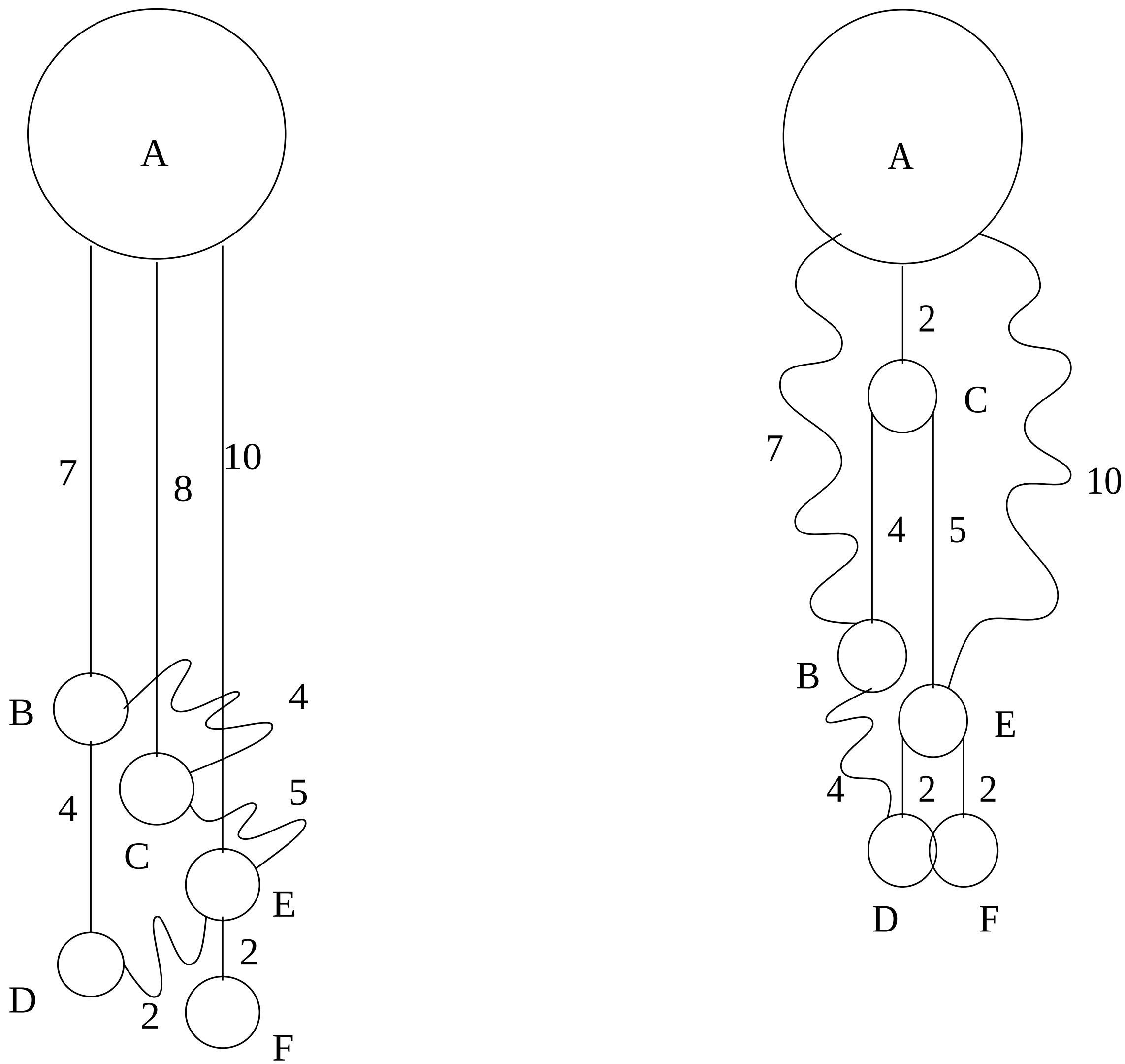
SPT Recomputation

- **Open Shortest Path Tree (OSPF)**
  - Every link state update is broadcast to the entire network.
  - Every router has an entire link state database and computes the shortest path tree using Dijkstra's algorithm.
- **Advantages**
  - Fast Convergence
  - No Loops
- **Deficiencies**
  - Complexity: In the existing implementations, the shortest path tree has to be computed from scratch after each link state change.
  - The size of an OSPF area is limited by the computational complexity.
  - OSPF might cause unnecessary changes in the routing tables.

Dynamic SPT Algorithm

- **Recomputation:**
  - Algorithm to recompute the shortest path tree after changes in the link state.
- **Locality:**
  - Algorithm only visits nodes that are affected by the new changes in the link state database.
- **Speed:**
  - Algorithm visits fewer nodes.
- **Stability:**
  - Algorithm makes fewer changes in the routing table.

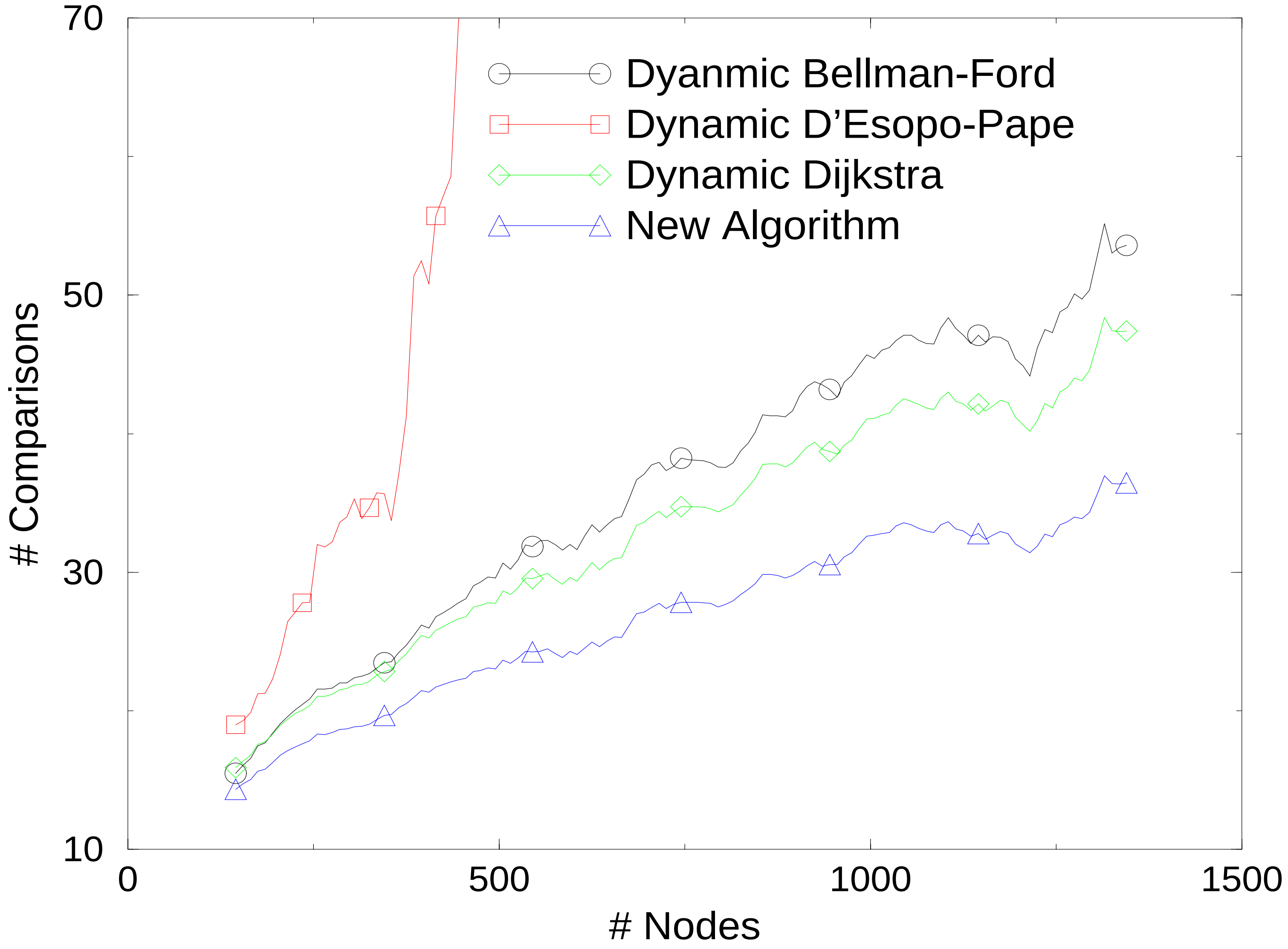
Ball and String Model



Results

- **Lowest Complexity**
  - The new algorithm visits the minimum number of nodes.
- **Route Stability**
  - The new algorithm makes the minimum number of changes to the structure of the SPT.

Complexity - Dynamic



Multiple Path Routing

- **Purpose:**
  - Better bandwidth utilization.
  - Faster recovery from link failures.

- **Objectives:**
  - No Loops.
  - Compatible with conventional OSPF routers.
  - No collaboration between routers.

Ideas

- **Principle**
  - Next hop must be closer to destination.

- **Search**
  1. Build alternative paths to destination through every port.
  2. When two paths for the same port collide, choose the shortest one.
  3. When the extra length of a path (inefficiency) is greater than the cost of the first hop, discard the path.

Results

- **Obtain best path for each output port to every destination.**
- **Classifies each path.**
  - ➪ Provides a quantitative measure which indicates how good or desirable is each path.
- **Linear Complexity.**
  - ➪ Most paths can be obtained in O(# ports x # edges).
- **Can be done dynamically.**
  - ➪ Can be incorporated into any dynamic or static algorithm that fits in the general framework.

Data Structure

- **At each node:**

Node X Optimal Distance	Port A	threshold	distance A	parent
	Port B	threshold	distance B	parent
	Port C	threshold	distance C	parent
	Port D	threshold	distance D	parent
	Port E	threshold	distance E	parent

Potential Problems with OSPF

- **Requires all routers in the same area to have consistent link information.**
- **Maintaining such consistency may require large communication overhead.**
- **During flooding, inconsistent states might lead to routing loops. Flooding a large area takes time.**
- **The above limits the scalability of OSPF to large routing areas, and limits the frequency of updates.**

What happens if a link fails

- A router attached to the failed link informs every router about the failure by flooding the entire area.
- Each router recomputes its shortest paths to every other router based on the new link states.

Our Main Results

- **A new routing algorithm that restores loop-free routing after link failures.**
- **Does not require flooding.**
- **Informs only the minimum number of routers after a link failure.**
- **Interoperable with OSPF.**

Our Algorithm

- **Compute a restoration path R and save it(1)**
- **Inform only routers on the restoration path R.**
- **These routers update their forwarding table using our new procedure (2).**
  - Key Idea: Use different levels of metric values. Use a vector metric rather than a scalar metric.

Virtual Node