Efficient algorithms for distributed shortest paths on power-law networks

13th Italian Conference on Theoretical Computer Science September 19-21, 2012 – Villa Toeplitz, Varese (Italy)

Gianlorenzo D'Angelo [1] **Mattia D'Emidio**[2]
Daniele Frigioni [2] Daniele Romano [2]

[1] MASCOTTE Project INRIA/I3S (CNRS/UNSA), Sophia Antipolis (France)
[2]Dept of Information Engineering, Computer Science and Mathematics, University of L'Aquila (Italy)

WEB: `informatica.ing.univaq.it/demidio`
MAIL: `mattia.demidio@univaq.it`

Outline

Outline

The problem

Distributed maintainance of minimum cost paths in dynamic asynchronous scenarios

- Crucial problem in today's practical applications (networking)
- Most important solving approach:
  - model the network as a graph
  - solve the **dynamic asynchronous distributed all-pairs shortest paths problem** by using a message passing communication model
- Widely studied in the literature, many solutions have been developed over the years

Known solutions

Link-State

- Each node stores the whole graph topology and computes locally the SPs
- Each topology change needs to be notified to all the nodes
- High space/computational complexity
- Loop-free by definition

Distance-Vector

- Each node computes the SPs by interacting with neighbors
- Broadcast a topology change only to affected nodes
- High message complexity but store very little information
- Require strategies to guarantee loop-freedom (**SNC**)

Known solutions

Link-State

- Each node stores the whole graph topology and computes locally the SPs
- Each topology change needs to be notified to all the nodes
- High space/computational complexity
- Loop-free by definition

Distance-Vector

- Each node computes the SPs by interacting with neighbors
- Broadcast a topology change only to affected nodes
- High message complexity but store very little information
- Require strategies to guarantee loop-freedom (**SNC**)

Recently

Renewed interest mainly in two areas

- Devising new efficient **light-weight** distributed and dynamic shortest path solutions for **Large Scale Ethernet Networks**:
  - highly important class in practice
  - power-law node degree distribution
  - includes the Internet, WWW, some Social Networks and WSN
- Engineering and improving existing solutions in real world practically interesting scenarios (like, e.g. power-law networks)

Where the focus has been placed

First area: Distance-Vector algorithms

An attractive alternative to link-state solutions when:

- *scalability* and *reliability* are key issues
- routing devices have limited storage/computation capabilities

Second area: algorithm engineering heuristics for DV algorithms

Interesting approach when theoretical bounds are not good enough

- Typical flow-chart:
  - Identify instances of interest
  - Study the structural properties
  - Design the technique in order to improve the existing DV solutions in such instances
  - Experimentally prove the effectiveness

N.B. improving Distance-Vector algorithms = reducing message complexity while preserving low space complexity/loop-freedom

Where the focus has been placed

First area: Distance-Vector algorithms

An attractive alternative to link-state solutions when:

- *scalability* and *reliability* are key issues
- routing devices have limited storage/computation capabilities

Second area: algorithm engineering heuristics for DV algorithms

Interesting approach when theoretical bounds are not good enough

- Typical flow-chart:
  - Identify instances of interest
  - Study the structural properties
  - Design the technique in order to improve the existing DV solutions in such instances
  - Experimentally prove the effectiveness

N.B. improving Distance-Vector algorithms = reducing message complexity while preserving low space complexity/loop-freedom

State-of-the-art

Distance-Vector algorithms: **DUAL**

- Part of CISCOs widely used EIGRP protocol
- Fully dynamic and loop-free (by **SNC**)
- Uses a single phase distributed computation and a FSM
- Good message complexity $O(\Phi \cdot \Delta)$ but high space complexity per node $\Theta(n \cdot \Delta)$ where:
  - Φ: total number of nodes *affected* by a set of updates
  - n and Δ: number of nodes and maximum node degree

Algorithm-engineering-based techniques: **DLP**

- allows to reduce the message complexity of Distance-Vector algorithms in power-law networks at the price of a little overhead in the space occupancy per node

State-of-the-art

Distance-Vector algorithms: **DUAL**

- Part of CISCOs widely used EIGRP protocol
- Fully dynamic and loop-free (by **SNC**)
- Uses a single phase distributed computation and a FSM
- Good message complexity $O(\Phi \cdot \Delta)$ but high space complexity per node $\Theta(n \cdot \Delta)$ where:
  - Φ: total number of nodes *affected* by a set of updates
  - n and Δ: number of nodes and maximum node degree

Algorithm-engineering-based techniques: **DLP**

- allows to reduce the message complexity of Distance-Vector algorithms in power-law networks at the price of a little overhead in the space occupancy per node

Outline

Our contributions (1)

Loop Free Routing

We developed **LFR**, a new distributed asynchronous distance vector algorithm which:

- is fully dynamic and loop-free (by **SNC**)
- combines a **two** phases distributed computation with a FIFO queue
- has the same message complexity of **DUAL**
- requires less memory: $O(n + \phi \cdot \Delta)$, where ϕ is the maximum number of destinations for which a node is *affected* (usually $\phi << n$)

LFR basics

Data structures

- All the data structures stored by **LFR** require $O(n)$ space per node
- Except for a temporary data structure, allocated only when weight increases occur $\Rightarrow O(\phi \cdot \Delta)$

Behaviour

- Nodes do not permanently store neighborhood data and interact with neighbors when such information are needed
- Interaction can be confined to neighbors (Local-Computation) or, if needed, can be propagated to the whole network, by a coordinated distributed computation phase (Global-Computation) $\Rightarrow O(\Phi \cdot \Delta)$
- The aim: avoid the formation of loops in the SPs tree

Our contributions (2)

Distributed Pruning

We developed a new technique, named **DP**, which:

- can be combined with every distance-vector routing algorithm based on shortest paths
- allows to reduce the total number of messages sent by that algorithm and its space occupancy per node in power-law networks

Power-law networks

- Low *avgdeg* and high number of nodes with small degree
- In many topological situations these nodes do not provide useful information for the distributed computation of SP_s
- Their SP_s depend, with few degrees of freedom, on higher nodes' SP_s

Our contributions (2)

Distributed Pruning

We developed a new technique, named **DP**, which:

- can be combined with every distance-vector routing algorithm based on shortest paths
- allows to reduce the total number of messages sent by that algorithm and its space occupancy per node in power-law networks

Power-law networks

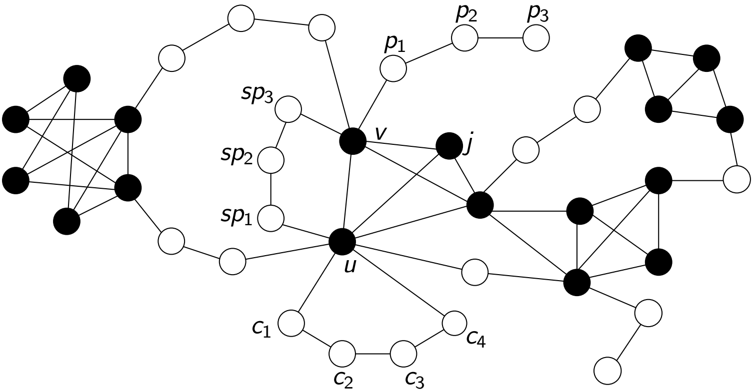
- Low *avgdeg* and high number of nodes with small degree
- In many topological situations these nodes do not provide useful information for the distributed computation of SP_s
- Their SP_s depend, with few degrees of freedom, on higher nodes' SP_s

How it works

How **DP** takes advantage of these structural properties

- Nodes are classified wrt to their degree
- Edges are classified wrt to their adjacent nodes
- Paths are classified wrt to nodes and edges classification
- Distributed computations are forced to be carried out only by nodes with $deg \geq 3$ (central nodes)
- Non-central nodes do not perform nor are involved in any kind of distributed computation and execute only few operations in order to update the routing information
- Some routing information can be avoided to be stored as it can be inferred by using the classification (see the paper for details)
- Then, the larger is the set of non-central nodes, the bigger is the improvement in the performances

Example of classification



Additional data structure

CHP

- In order to implement **DP**, nodes need to store few additional information about the classification of nodes and edges/paths
- Each node mantain a data structure, called *CHain Path* (CHP)
- Overhead in the space occupancy per node is negligible ($O(n)$)

Outline

Experimental setting

Simulation environment

- Extensive experimental evaluation within OmNet++, a well-known distributed systems simulation environment
- We measured the total number of messages sent and the space requirements

Implemented algorithms

- We combined **DP** with **LFR** and **DUAL**
- We implemented, in C++, **LFR** and **DUAL**, **LFR-DP** and **DUAL-DP**
- We used the implementations of **LFR-DLP** and **DUAL-DLP** from a previous work to make comparisons

Experimental setting

Simulation environment

- Extensive experimental evaluation within OmNet++, a well-known distributed systems simulation environment
- We measured the total number of messages sent and the space requirements

Implemented algorithms

- We combined **DP** with **LFR** and **DUAL**
- We implemented, in C++, **LFR** and **DUAL**, **LFR-DP** and **DUAL-DP**
- We used the implementations of **LFR-DLP** and **DUAL-DLP** from a previous work to make comparisons

Input data

Graphs

- Real-world graphs (G_{IP}), provided by the Cooperative Association for Internet Data Analysis (CAIDA)
- Artificial random graphs (G_{ER}), generated by the *Erdòs-Renyi* algorithm
- Artificial power-law random graphs (G_{BA}), generated by the *Barabási-Albert* algorithm

Edge weight updates sequences

- Randomly generated sequences of k edge weight changes
- Each weight change consists of multiplying the original weight by a percentage value randomly chosen in $[50\%, 150\%]$

Input data

Graphs

- Real-world graphs (G_{IP}), provided by the Cooperative Association for Internet Data Analysis (CAIDA)
- Artificial random graphs (G_{ER}), generated by the *Erdòs-Renyi* algorithm
- Artificial power-law random graphs (G_{BA}), generated by the *Barabási-Albert* algorithm

Edge weight updates sequences

- Randomly generated sequences of k edge weight changes
- Each weight change consists of multiplying the original weight by a percentage value randomly chosen in $[50\%, 150\%]$

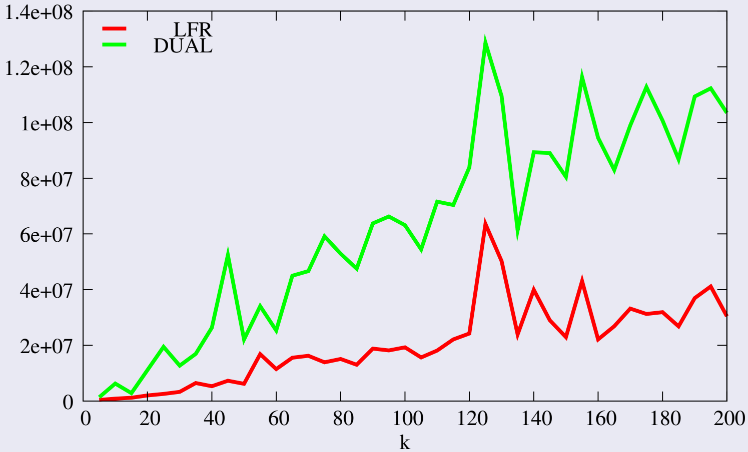
Executed tests

Executed tests

- We ran simulations in different dynamic **realistic** settings, where the weight of an edge represents the delay needed to a packet to traverse it
- Due to **DUAL**'s high memory requirements we used instances with max number of nodes equal to 8000 (G_{IP}, G_{BA}) and 2000 (G_{ER})
- We show the results for:
  - a G_{IP} graph with $n = 8000$, $m = 11141$ ($m/n \approx 1.4$) and $k \in \{5, 10, ..., 200\}$
  - a G_{ER} graph with $n = 2000$, *density* ranging from 0.01 to 0.61 and $k = 200$
- Other instances give similar results

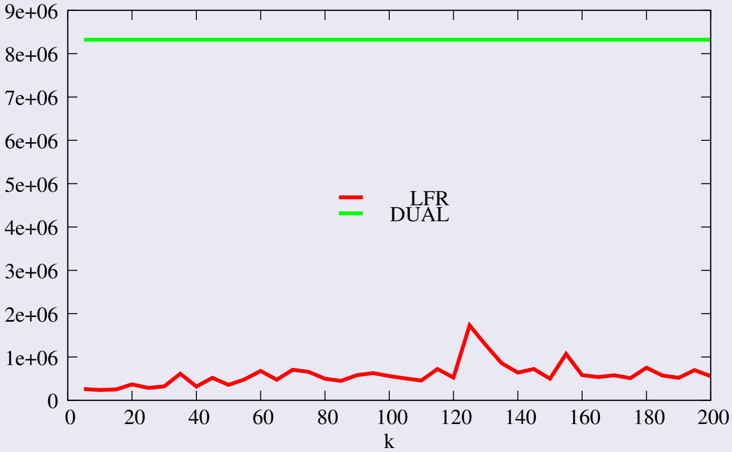
LFR vs **DUAL** – Message complexity in real-world instances

Number of messages sent – $G_{IP-8000}$



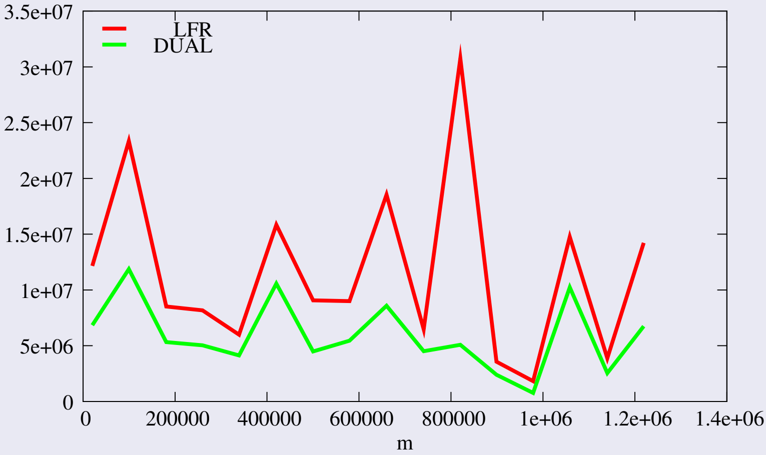
LFR vs **DUAL** – Space complexity in real-world instances

Maximum space occupancy – $G_{IP-8000}$



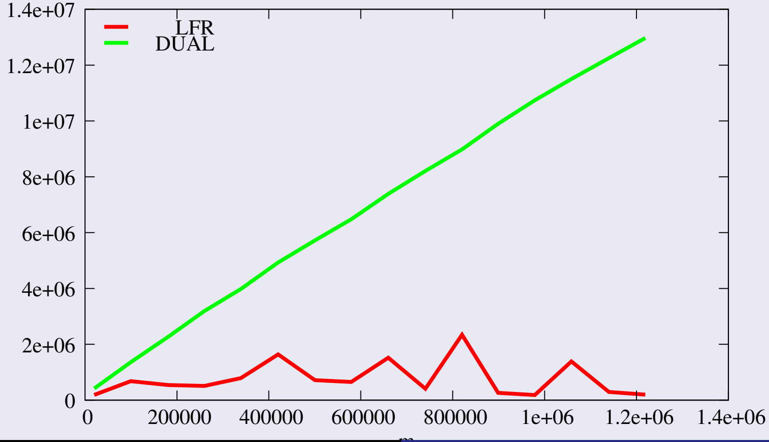
LFR vs **DUAL** – Message complexity in dense artificial instances

Number of messages sent – $G_{ER-2000}$



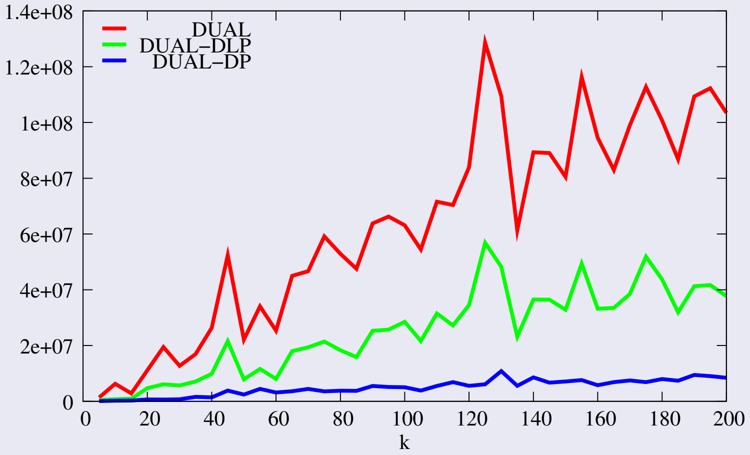
LFR vs **DUAL** – Space complexity in dense artificial instances

Maximum space occupancy – $G_{ER-2000}$



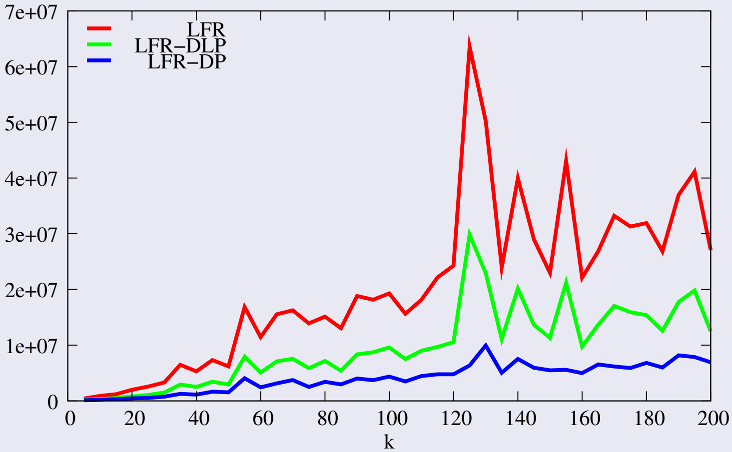
Effectiveness of **DP** in real-world instances – **DUAL**

Number of messages sent – $G_{IP-8000}$



Effectiveness of **DP** in real-world instances – **LFR**

Number of messages sent – $G_{IP-8000}$



Effectiveness of **DP** in real-world instances – Space complexity

Graph	Algorithm	MAX		AVG	
		Bytes	Ratio	Bytes	Ratio
$G_{IP-8000}$	**DUAL**	8 320 000	1	311 410	1
	DUAL-DLP	5 161 984	0.62	240 754	0.77
	DUAL-DP	2 517 680	0.30	252 625	0.81
$G_{IP-8000}$	**LFR**	549 170	1	192 871	1
	LFR-DLP	421 862	0.77	204 675	1.06
	LFR-DP	392 658	0.72	295 930	1.53

Outline

Conclusion

New algorithm

- We proposed **LFR**, a new fully dynamic loop-free shortest paths routing algorithm which:
  - has the same message complexity of **DUAL**
  - has a better space complexity
- We have experimentally proved its effectiveness
  - always the best choice in terms of space requirements
  - best choice in terms of messages sent in real-world networks

New technique

- We have proposed a new technique, named **DP**, which:
  - can be combined with every distance vector routing algorithm
  - allows to reduce the total number of messages sent and the space occupancy per node on power-law networks
- We have experimentally proved the effectiveness of this technique on real power-law networks

Conclusion

New algorithm

- We proposed **LFR**, a new fully dynamic loop-free shortest paths routing algorithm which:
  - has the same message complexity of **DUAL**
  - has a better space complexity
- We have experimentally proved its effectiveness
  - always the best choice in terms of space requirements
  - best choice in terms of messages sent in real-world networks

New technique

- We have proposed a new technique, named **DP**, which:
  - can be combined with every distance vector routing algorithm
  - allows to reduce the total number of messages sent and the space occupancy per node on power-law networks
- We have experimentally proved the effectiveness of this technique on real power-law networks

Future research

- Improve **LFR**, like e.g. by:
  - reducing the message/space complexity while preserving loop-freedom
- Design new algorithms-engineering based techniques, tailored to be effective in other real world practically interesting scenarios
- Extend the experimental evaluation to other real-world/artificial instances

Thank you for your attention - Q&A