A Framework for the Evaluation and Management of Network Centrality

Vatche Ishakian, Dora Erdos, Evimaria Terzi, Azer Bestavros

Department of Computer Science

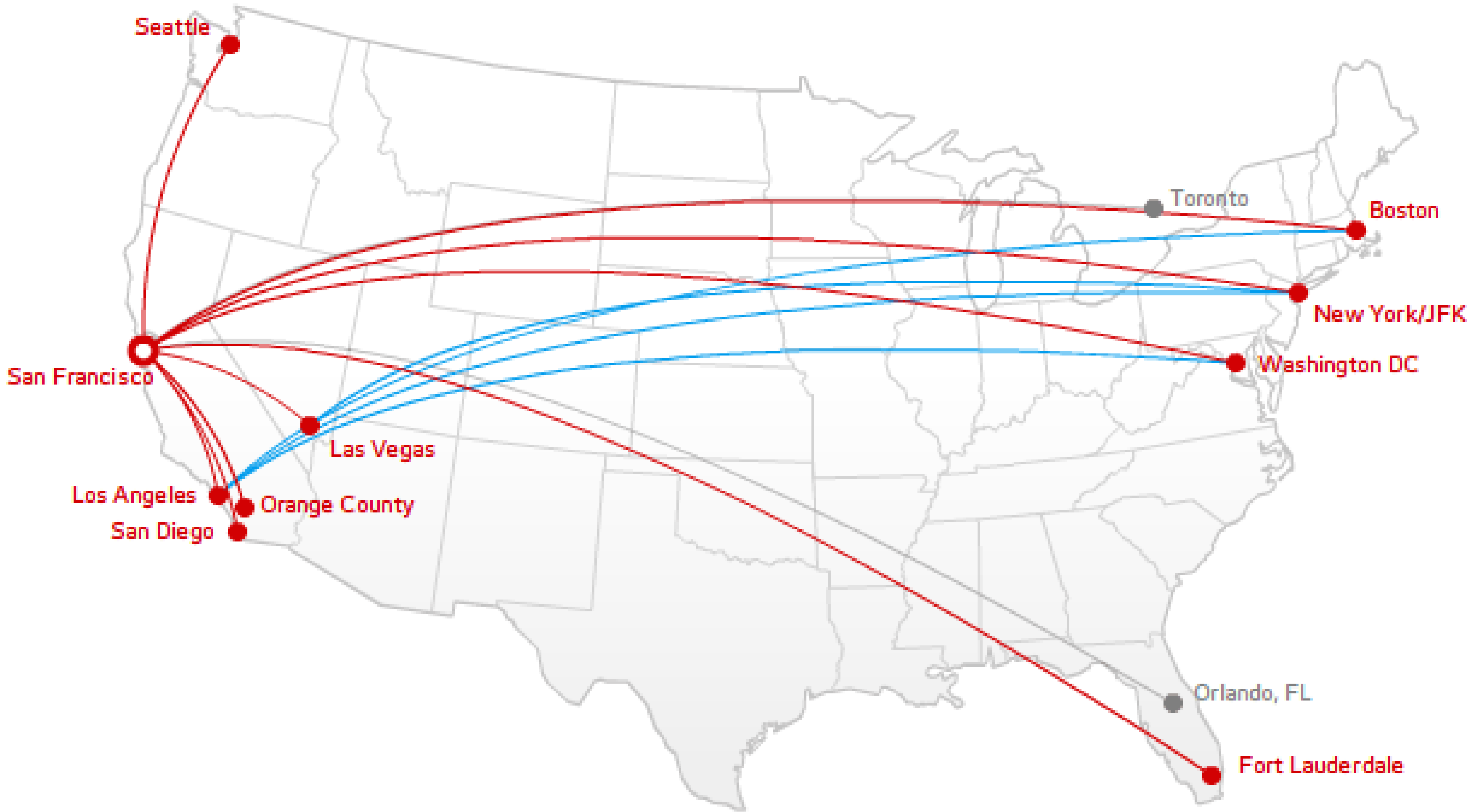
Boston University

What is the busiest stop?

What is the busiest stop?


CANADA
MEXICO
Atlantic Ocean
Pacific Ocean
Gulf of Mexico
BAHAMAS
ALASKA
HAWAII
Mountain Time 5:00
Central Time 6:00
Eastern Time 7:00
Hawaii-Aleutian Time 2:00
Edmonton
Vancouver
Calgary
Saskatoon
Charlottetown
Halifax
Minneapolis/St. Paul
Salt Lake City
Detroit
Atlanta
Memphis
New York
Boston
Miami
Key West
Honolulu
San Diego
Los Angeles
San Antonio
Austin
New Orleans
McAllen/Reynosa

DELTA



Where to add new flights so that the number of travellers is maximized?

Outline

Motivation

General framework for computing centrality

Centrality of nodes

Centrality of Groups

Graph modifications to change centrality values

Experiments

Outline

Motivation

General framework for computing centrality

Centrality of nodes

Centrality of Groups

Graph modifications to change centrality values

Experiments

Outline

Motivation

General framework for computing centrality

- Centrality of nodes
- Centrality of Groups

Graph modifications to change centrality values

Experiments

Centrality

- G(V,E) directed acyclic graph
- $S \subseteq V$ set of source nodes
- $T \subseteq V$ set of target nodes
- P set of *special paths* connecting nodes in S with T

$P_v(s,t)$: set of special paths between source s and target t covered by node v.

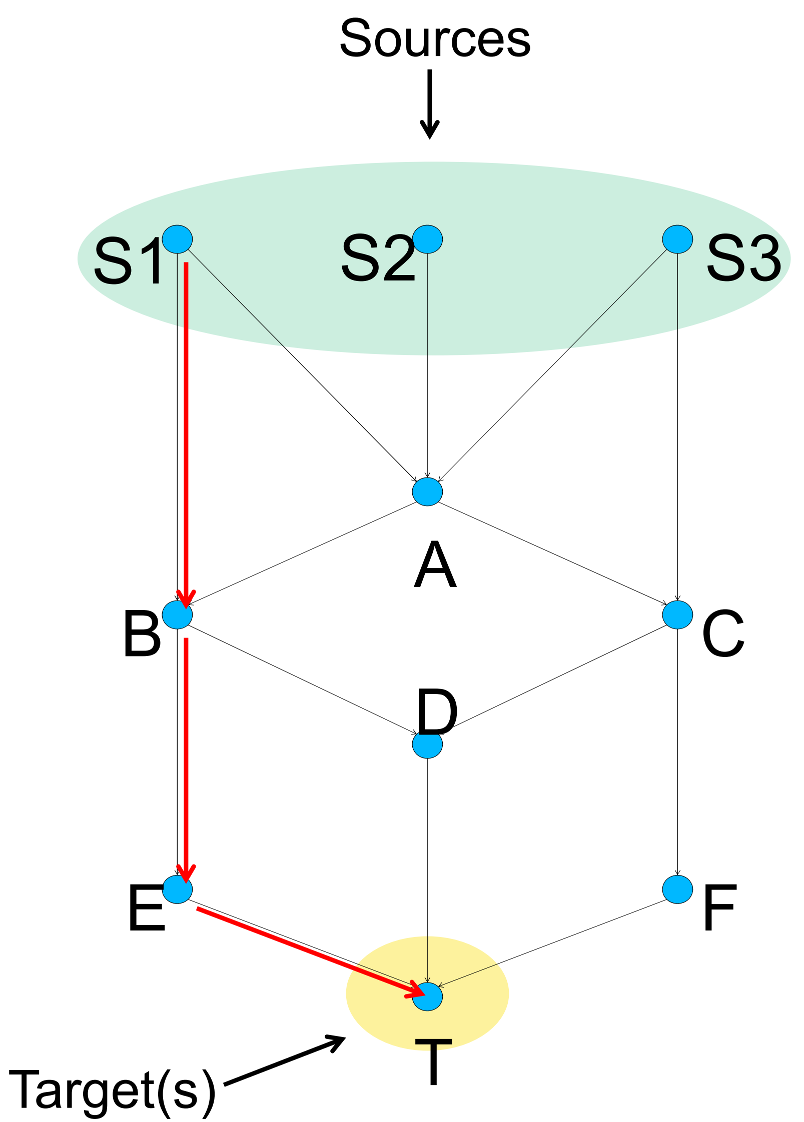
Centrality of node v is a function of the paths in P that v covers:

$$C(v) = \sum_{s,t} F(P_v(s,t))$$

Centrality – Stress centrality - NumPaths

$$C(v)=\sum_{s,v} F(P_v(s,t))$$
$$C(v)=\sum_{s,t} |P_v(s,t)| = |\sum_{s,t} |P_v(s,t)|$$

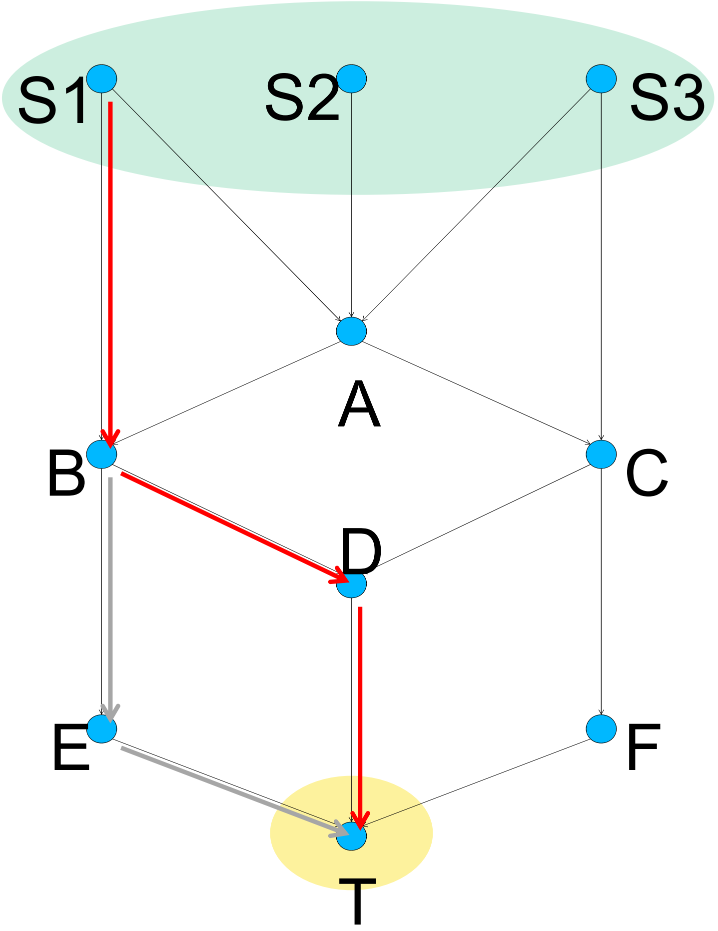
Compute C(B)



Centrality – Stress centrality - NumPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

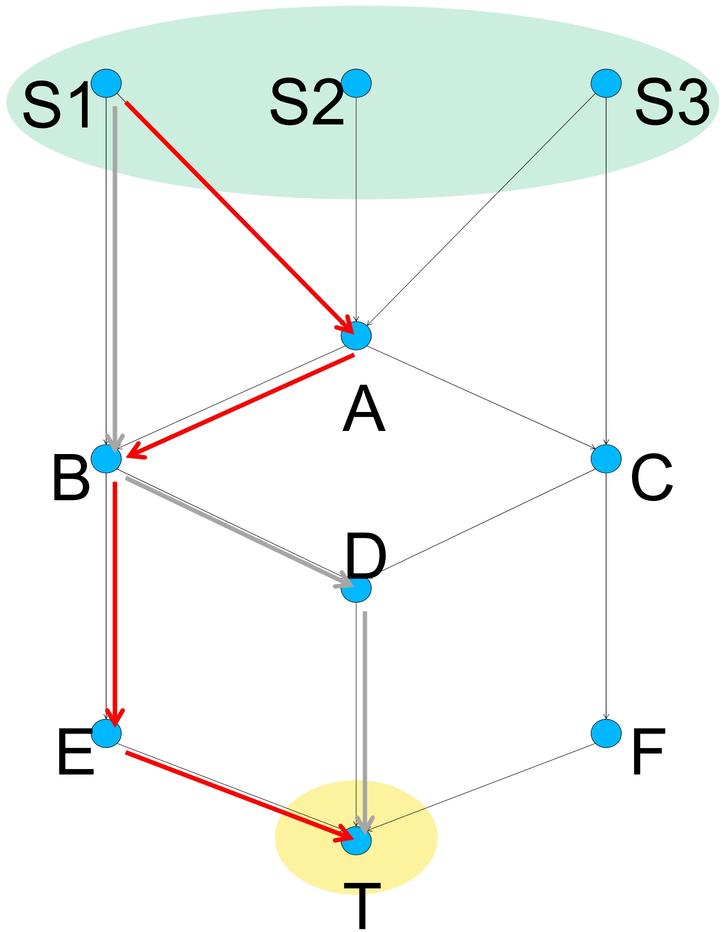
Compute C(B)



Centrality – Stress centrality - NumPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

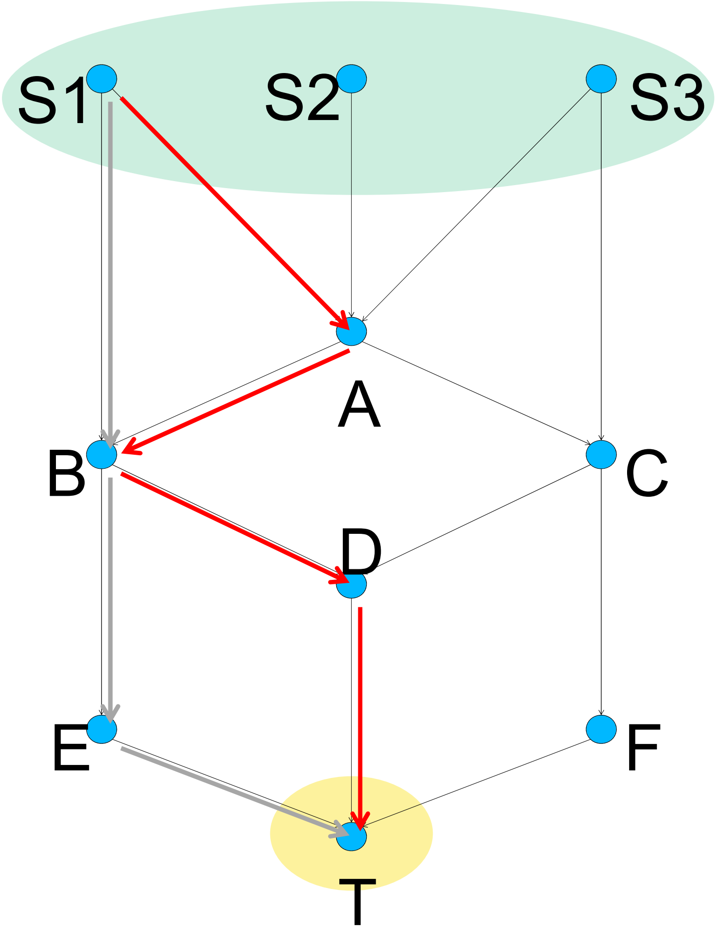
Compute C(B)



Centrality – Stress centrality - NumPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

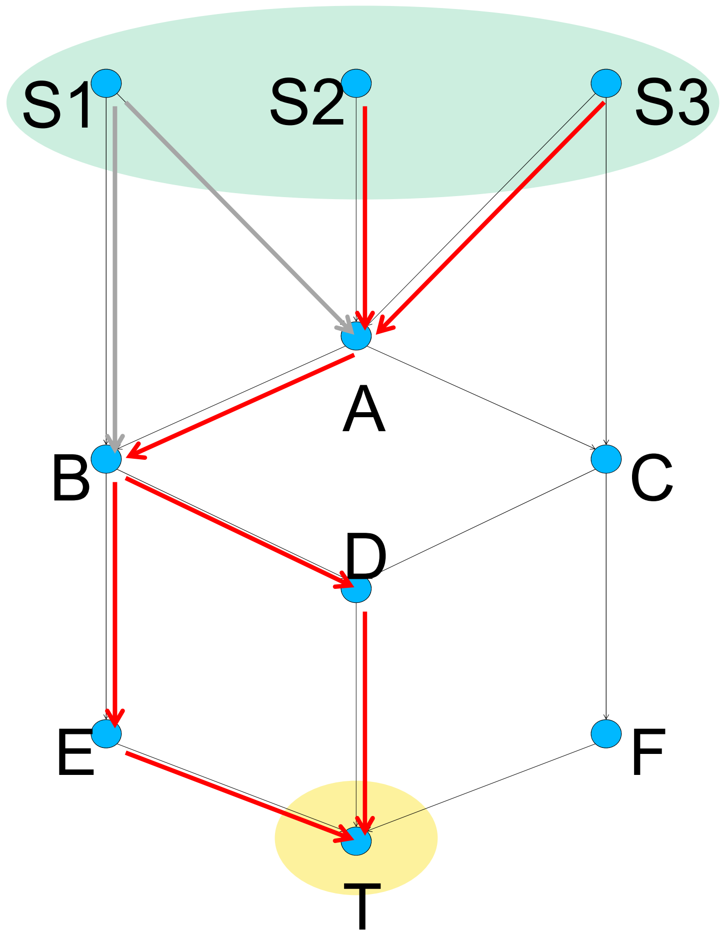
Compute C(B)



Centrality – Stress centrality - NumPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

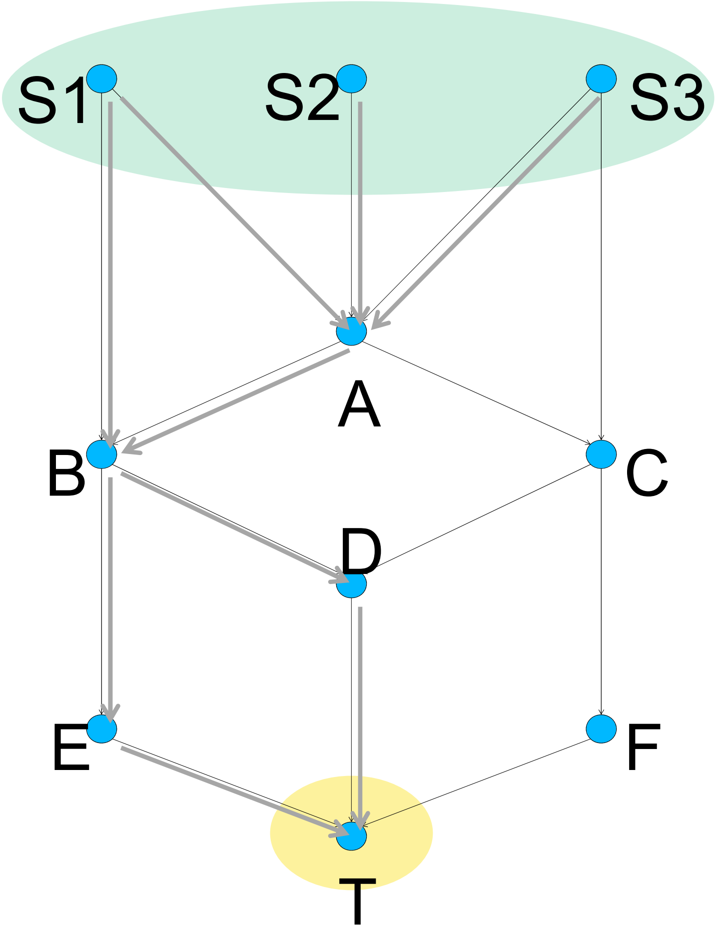
Compute C(B)



Centrality – Stress centrality - NumPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

Compute C(B) = 8

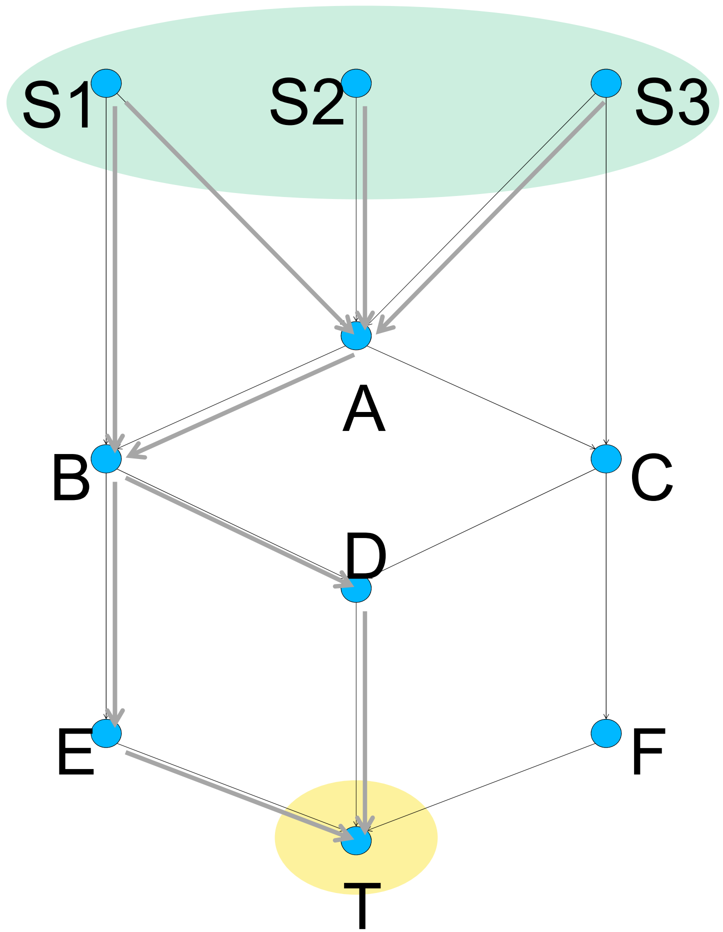


Centrality – Stress centrality - NumPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

Compute C(B) = 8

Compute C(B) in a smarter way



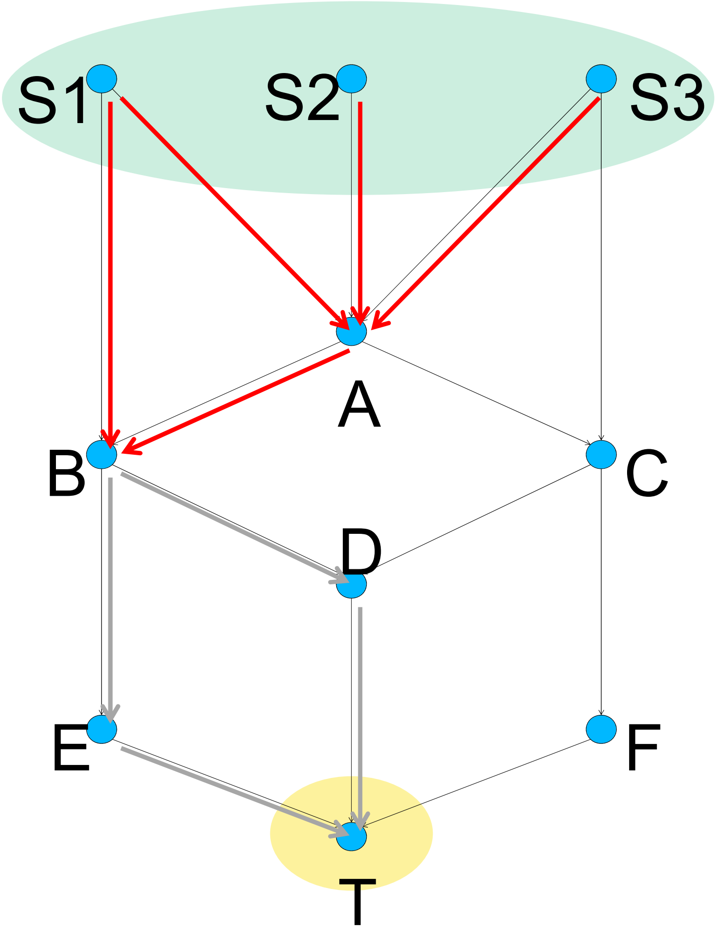
Centrality – Stress centrality - NumPaths

$$C(v) = \sum_{s,v} | P_v(s,t) |$$

Compute C(B) = 8

Compute C(B) in a smarter way

Prefix(B) = # of paths from S to B = 4



Centrality – Stress centrality - #P

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

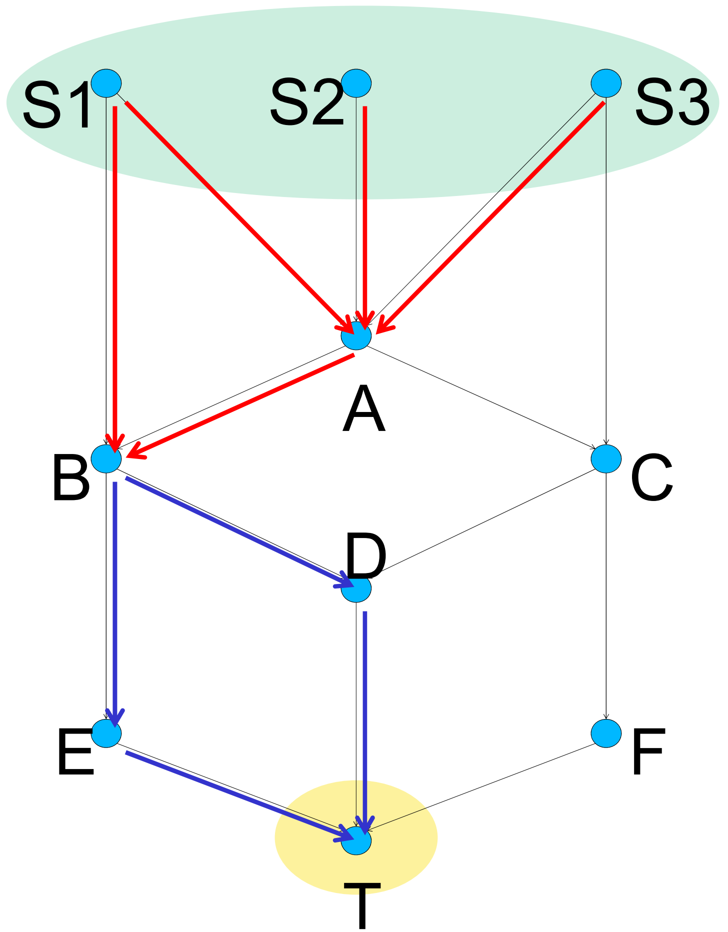
Compute C(B) = 8

Compute C(B) in a smarter way

Prefix(B) = # of paths from S to B = 4

Suffix(B) = # of paths from B to T = 2

C(B) = Prefix(B) x Suffix(B)

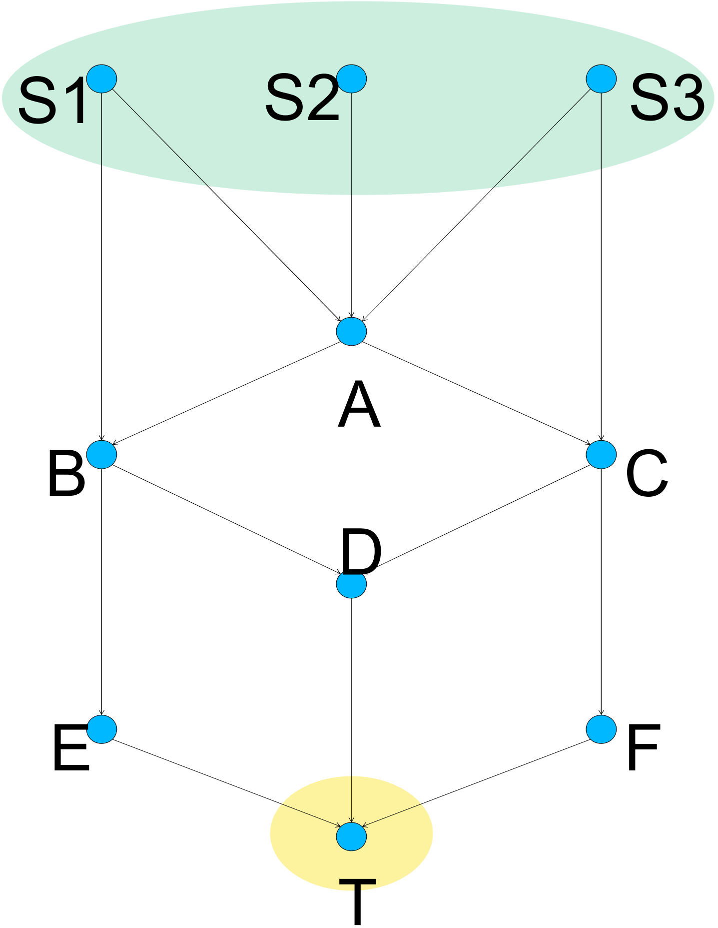


Centrality – Stress centrality - NumPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

C(v) = Prefix(v) x Suffix(v)

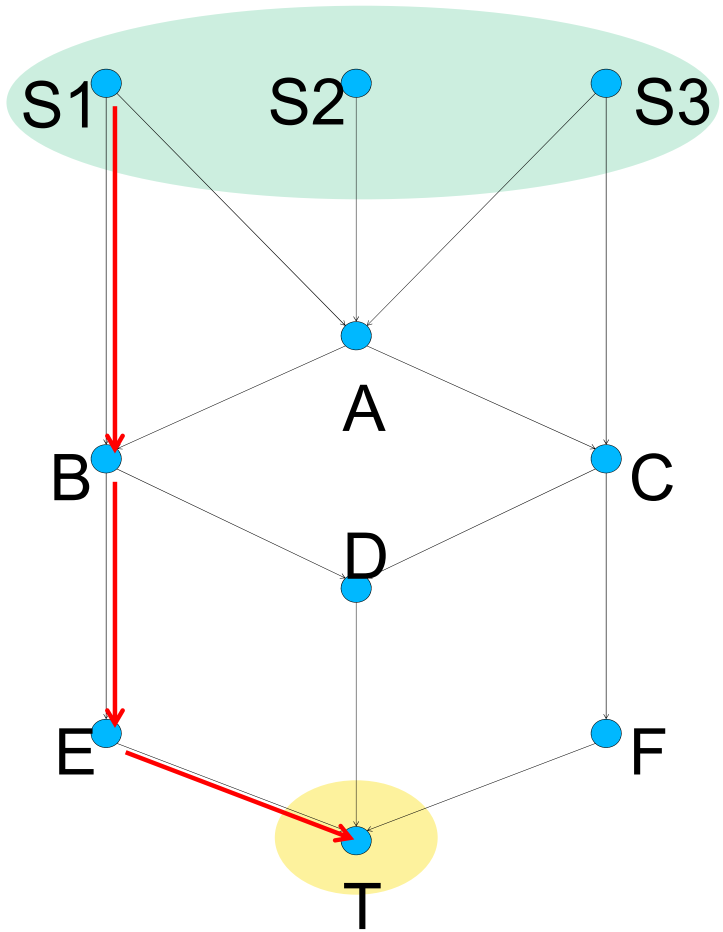
	Prefix	Suffix	C
A	3	4	12
B	4	2	8
C	4	2	8
D	8	1	8
E	4	1	4
F	4	1	4



Centrality – Betweenness centrality - NumShortestPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

Compute C(B)

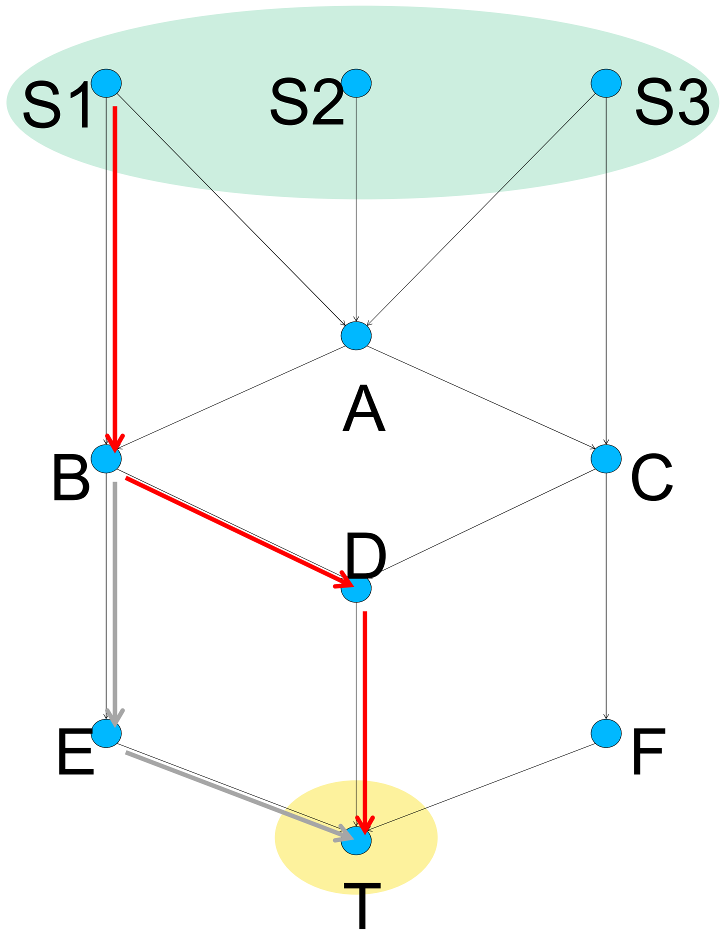


[Brandes 2001, 2008]

Centrality – Betweenness centrality - NumShortestPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

Compute C(B)

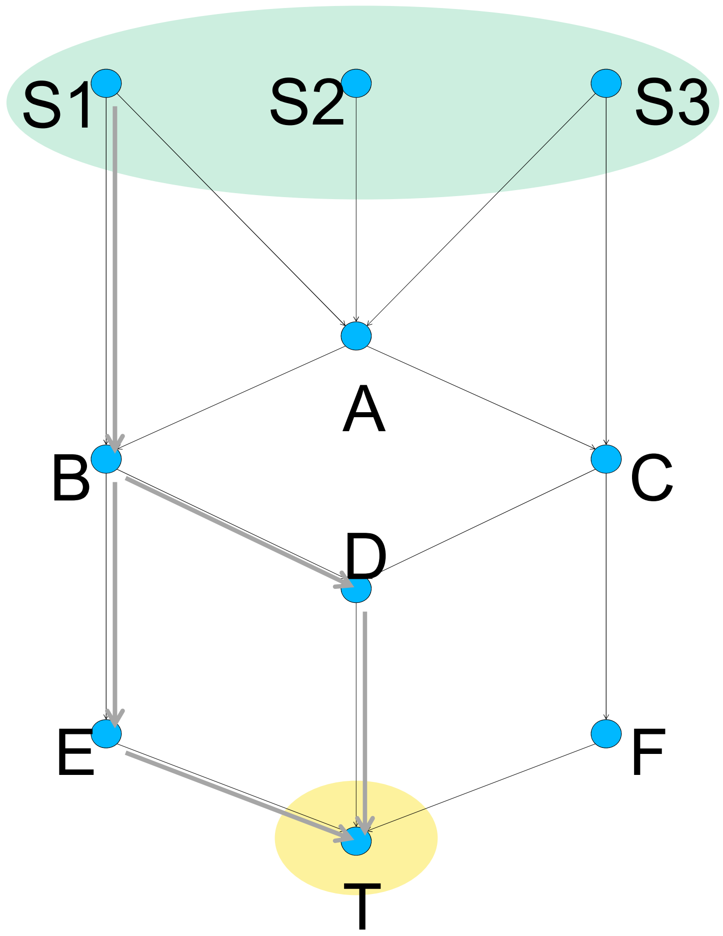


[Brandes 2001, 2008]

Centrality – Betweenness centrality - NumShortestPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

Compute C(B)

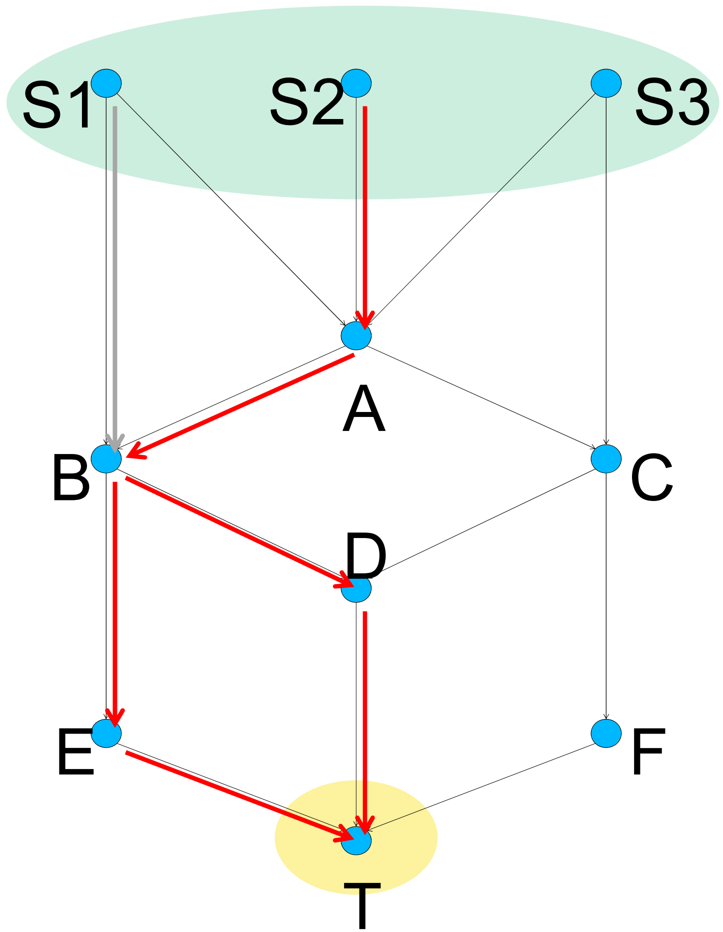


[Brandes 2001, 2008]

Centrality – Betweenness centrality - NumShortestPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

Compute C(B)

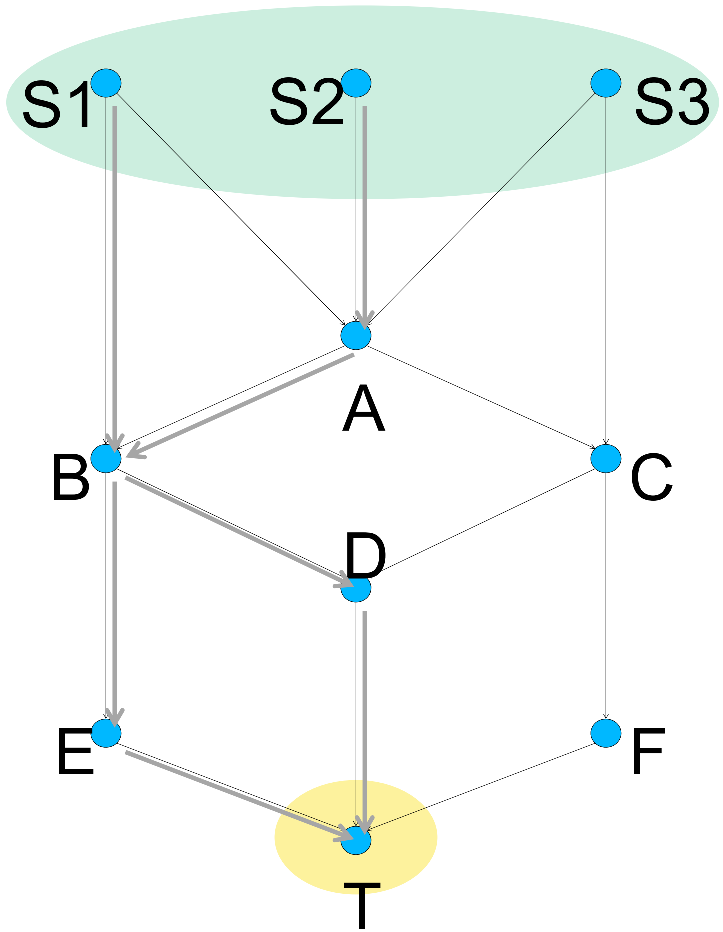


[Brandes 2001, 2008]

Centrality – Betweenness centrality - NumShortestPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

Compute C(B) = 4



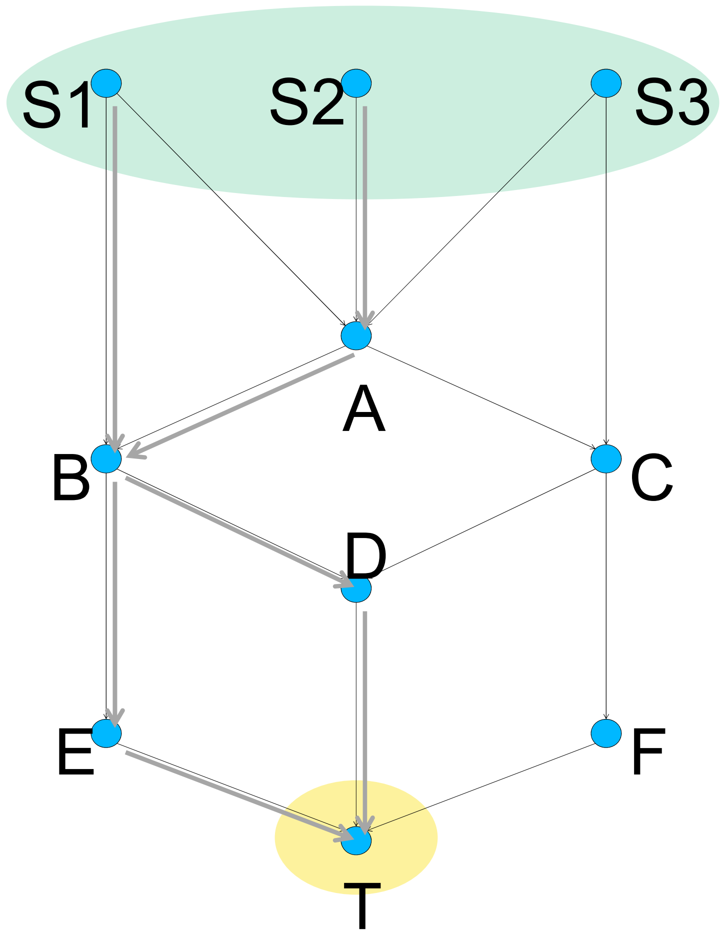
[Brandes 2001, 2008]

Centrality – Betweenness centrality - NumShortestPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

C(v) = Prefix(v) x Suffix(v)^T

	Prefix	Suffix	C
A	(1,1,1)	(0,4,0)	4
B	(1,1,0)	(2,2,0)	4
C	(0,1,1)	(0,2,2)	4
D	(1,2,1)	(1,1,1)	4
E	(1,1,0)	(2,2,0)	4
F	(0,1,1)	(0,2,2)	4

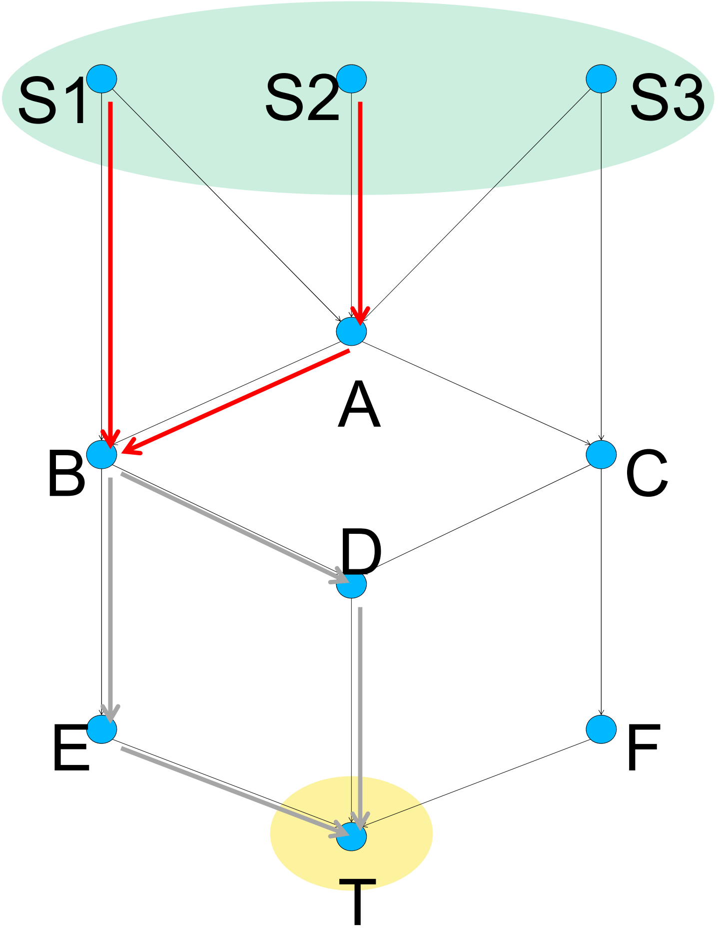


Centrality – Betweenness centrality - NumShortestPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

C(v) = Prefix(v) x Suffix(v)^T

	Prefix	Suffix	C
A	(1,1,1)	(0,4,0)	4
B	(1,1,0)	(2,2,0)	4
C	(0,1,1)	(0,2,2)	4
D	(1,2,1)	(1,1,1)	4
E	(1,1,0)	(2,2,0)	4
F	(0,1,1)	(0,2,2)	4

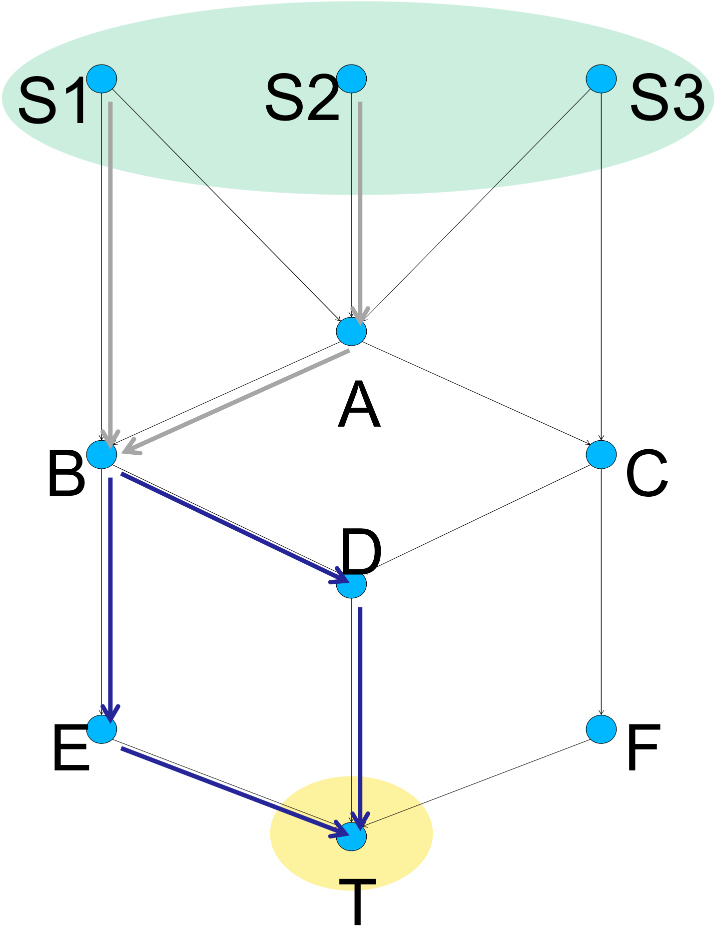


Centrality – Betweenness centrality - NumShortestPaths

$$C(v) = \sum_{s,v} |P_v(s,t)|$$

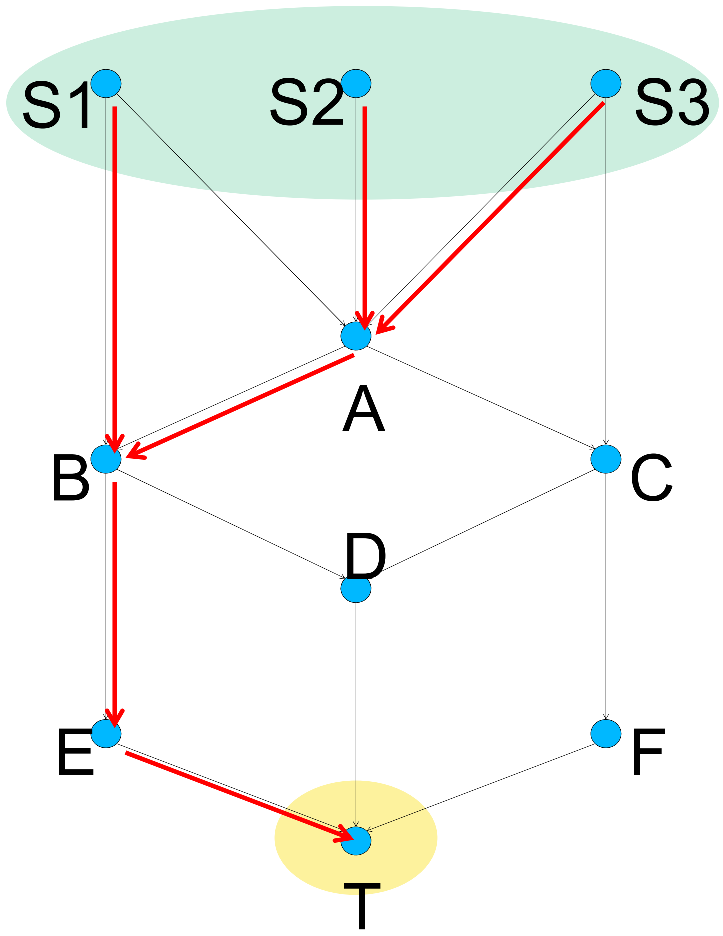
C(v) = Prefix(v) x Suffix(v)^T

	Prefix	Suffix	C
A	(1,1,1)	(0,4,0)	4
B	(1,1,0)	(2,2,0)	4
C	(0,1,1)	(0,2,2)	4
D	(1,2,1)	(1,1,1)	4
E	(1,1,0)	(2,2,0)	4
F	(0,1,1)	(0,2,2)	4



Centrality – Paths

$$C(v) = \sum_{s,v} \delta(|P_v(s,t)| > 0)$$

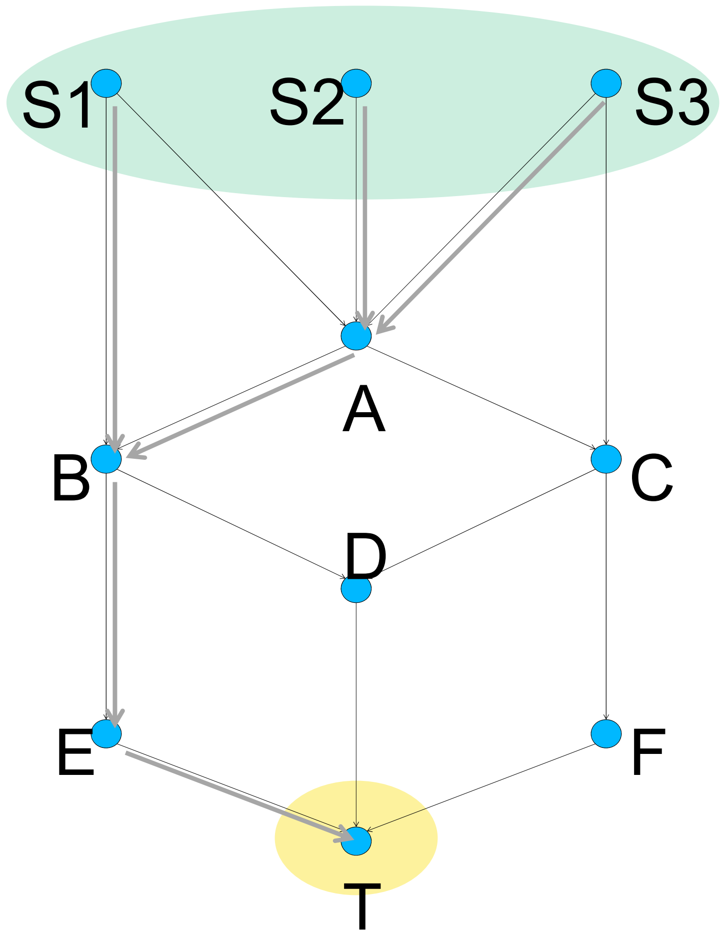


Centrality – Paths

$$C(v) = \sum_{s,v} \delta(|P_v(s,t)| > 0)$$

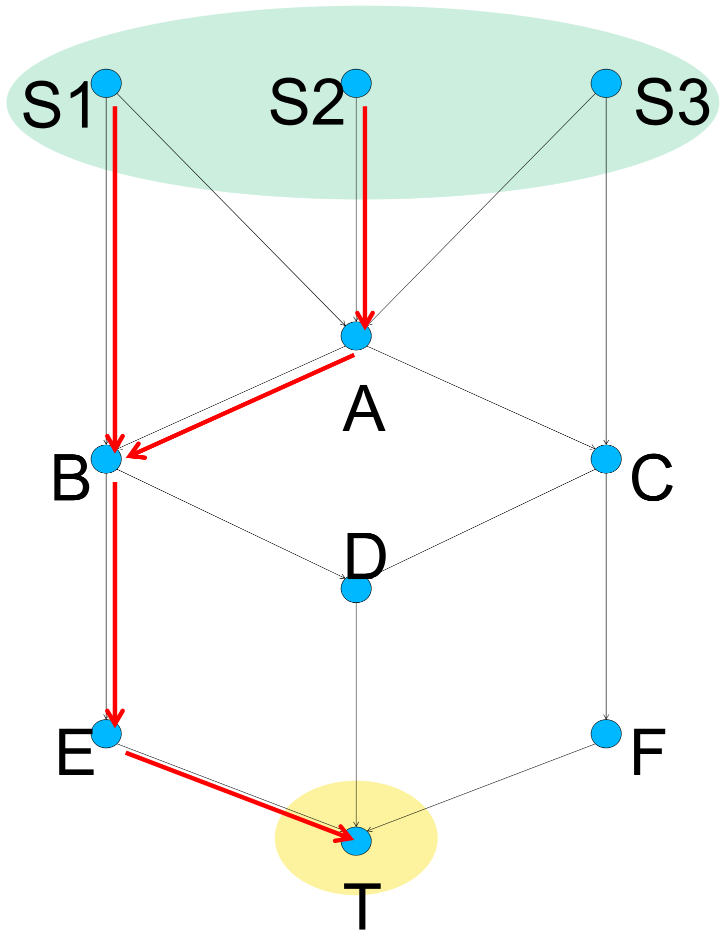
C(v) = Prefix(v) x Suffix(v)^T

	Prefix	Suffix	C
A	(1,1,1)	(1,1,1)	3
B	(1,1,1)	(1,1,1)	3
C	(1,1,1)	(1,1,1)	3
D	(1,1,1)	(1,1,1)	3
E	(1,1,1)	(1,1,1)	3
F	(1,1,1)	(1,1,1)	3



Centrality – ShortestPaths

$$C(v) = \sum_{s,v} \delta(|P_v(s,t)| > 0)$$

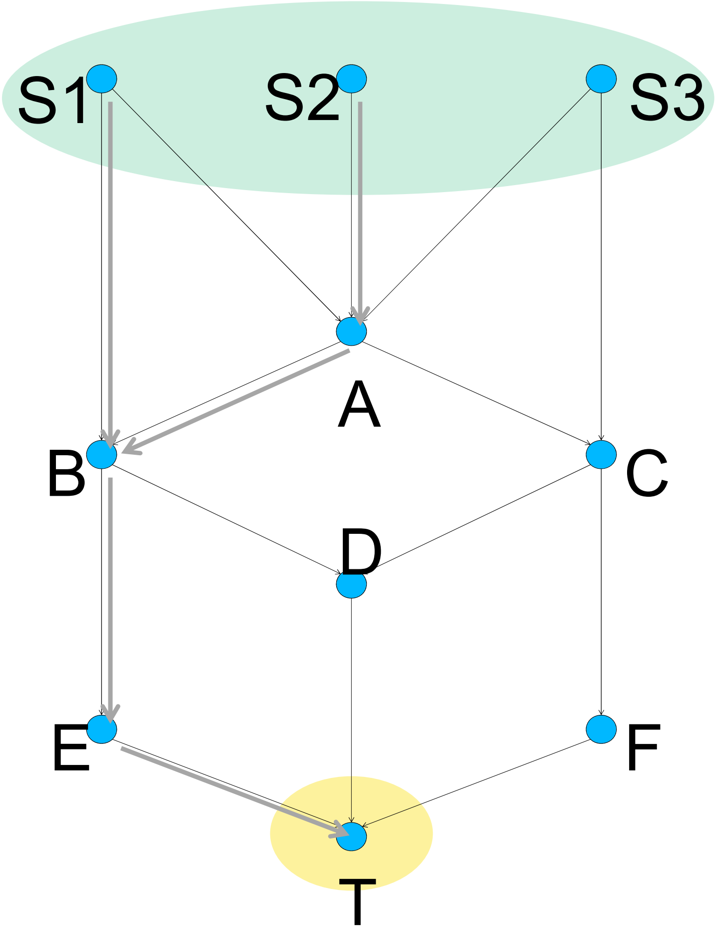


Centrality – ShortestPaths

$$C(v) = \sum_{s,v} \delta(|P_v(s,t)| > 0)$$

$C(v) = Prefix(v) \times Suffix(v)^T$

	Prefix	Suffix	C
A	(1,1,1)	(0,1,0)	1
B	(1,1,0)	(1,1,0)	2
C	(0,1,1)	(0,1,1)	2
D	(1,1,1)	(1,1,1)	3
E	(1,1,0)	(1,1,0)	2
F	(0,1,1)	(0,1,1)	2



Computing node centrality

Prefix(v) = # of paths from nodes in S to v

Suffix(v) = # of paths from v to nodes in T

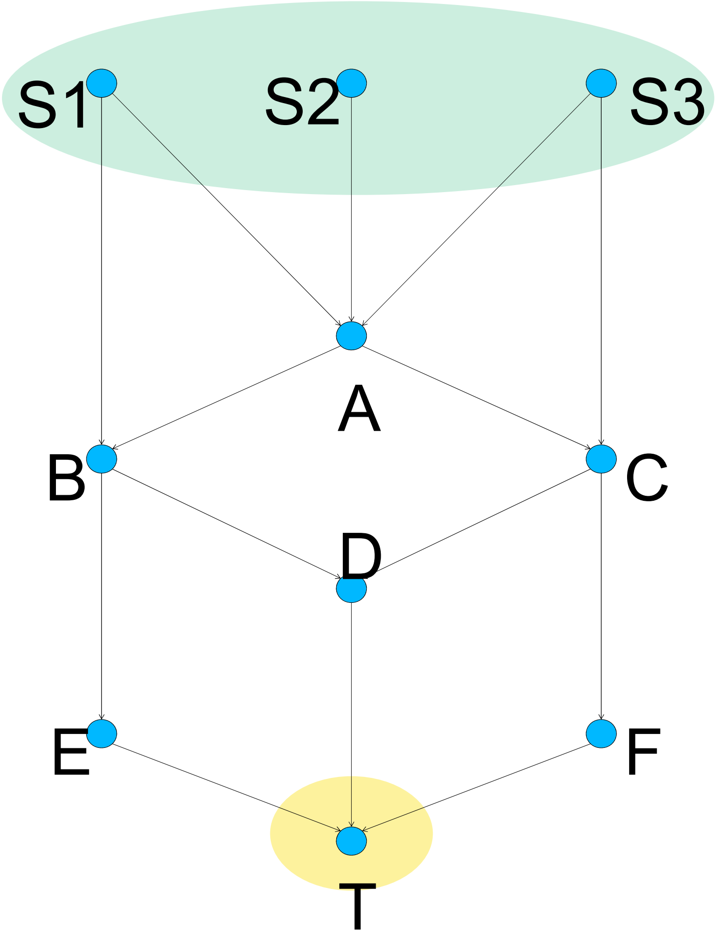
$$C(v) = \text{Prefix}(v) \times \text{Suffix}(v)^T$$

Computing node centrality – Computing Prefix( )

Prefix(v) = # of paths from nodes in S to v

Prefix(v) can be computed as the sum of the Prefix in v's parents Π_v :

$$\mathrm{Prefix}(v) = \sum_{u \in \Pi_v} \mathrm{Prefix}(u)$$



Computing node centrality – Computing Prefix( )

Prefix(v) = # of paths from nodes in S to v

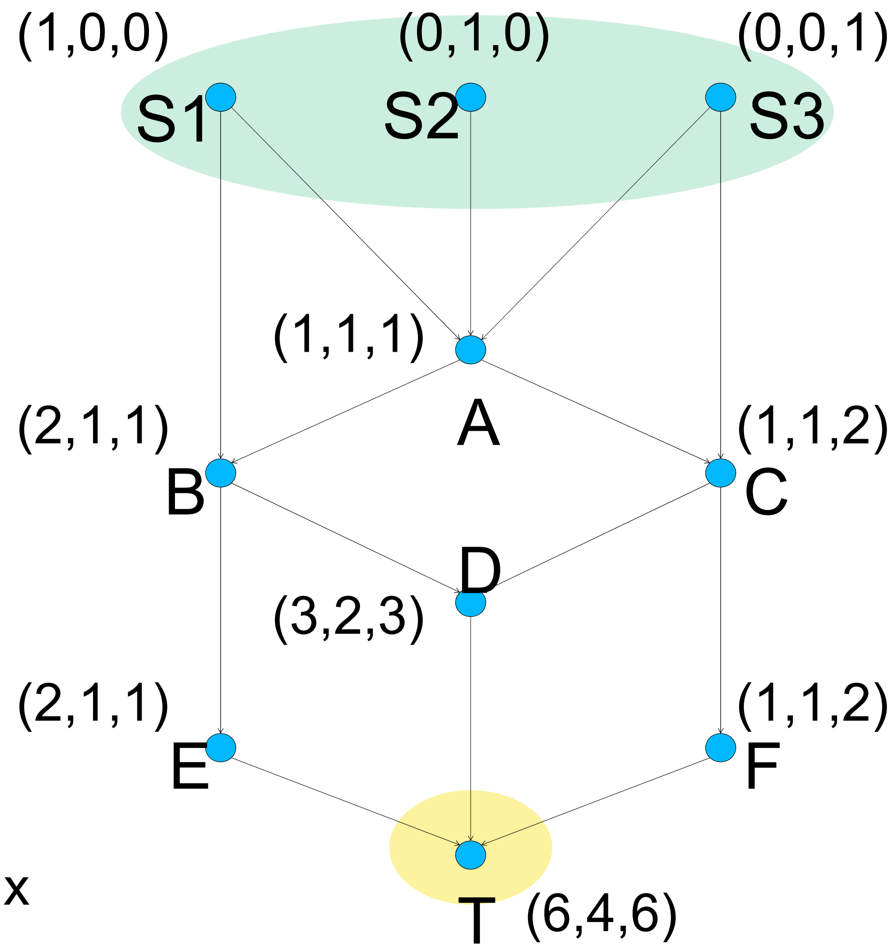
Prefix(v) can be computed as the sum of the Prefix in v's parents Π_v :

$$\mathrm{Prefix}(v) = \sum_{u \in \Pi_v} \mathrm{Prefix}(u)$$

1. Fix topological order σ of nodes.

σ = (S1 S2, S3, A, B, C, D, E, F, T)

2. Traverse nodes in order of σ to compute Prefix

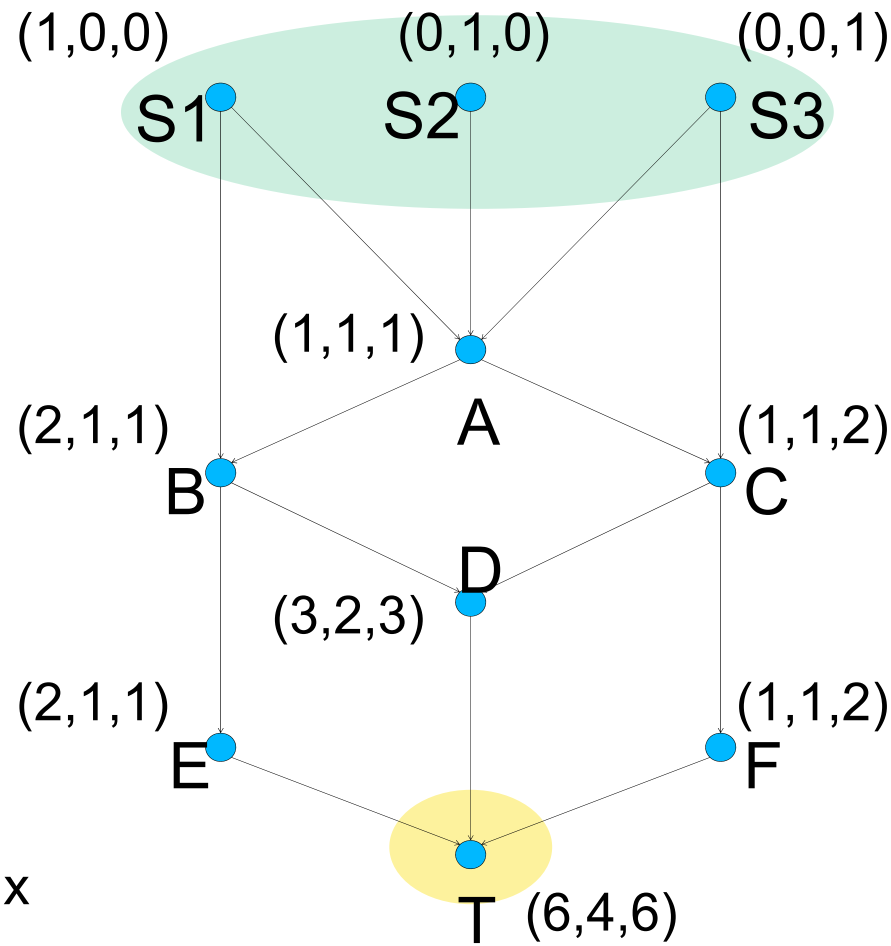


Computing node centrality – Computing Prefix( )

Prefix(v) = # of paths from nodes in S to v

Prefix(v) can be computed as the sum of the Prefix in v's parents Π_v :

$$\text{Prefix}(v) = \sum_{u \in \Pi_v} \text{Prefix}(u)$$



1. Fix topological order σ of nodes.

σ = (S1 S2, S3, A, B, C, D, E, F, T)

2. Traverse nodes in order of σ to compute Prefix

Computing node centrality

$C(v) = Prefix(v) \times Suffix(v)^T$

Small changes in the update of Prefix() and Suffix() taylor this general computation to the different instances of centrality.

NumShortestPaths : Only consider parents that are on a shortest path

Paths : Use boolean addition/union

ShortestPaths : Only consider parents on shortest paths and use boolean addition/ union

Outline

Motivation

General framework for computing centrality

Centrality of nodes

Centrality of Groups

Graph modifications to change centrality values

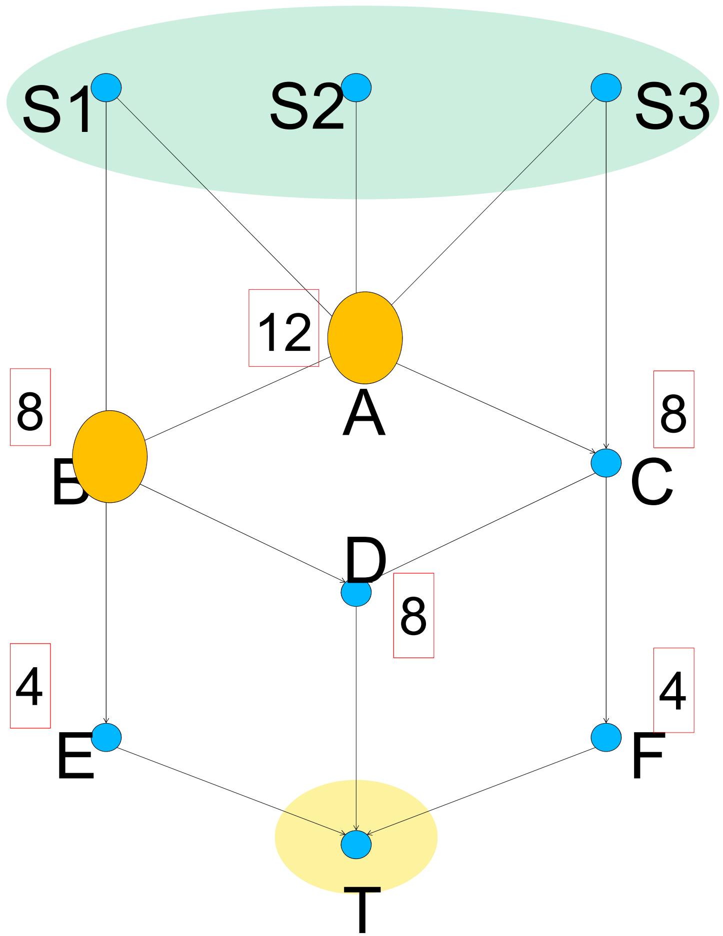
Experiments

Group centrality

Q: Which two nodes have the largest total NumPaths centrality?

Nodes A and B (or C or D) have the highest NumPaths-centrality

C (A,B) = 14



[Everett, Borgatti, '99]

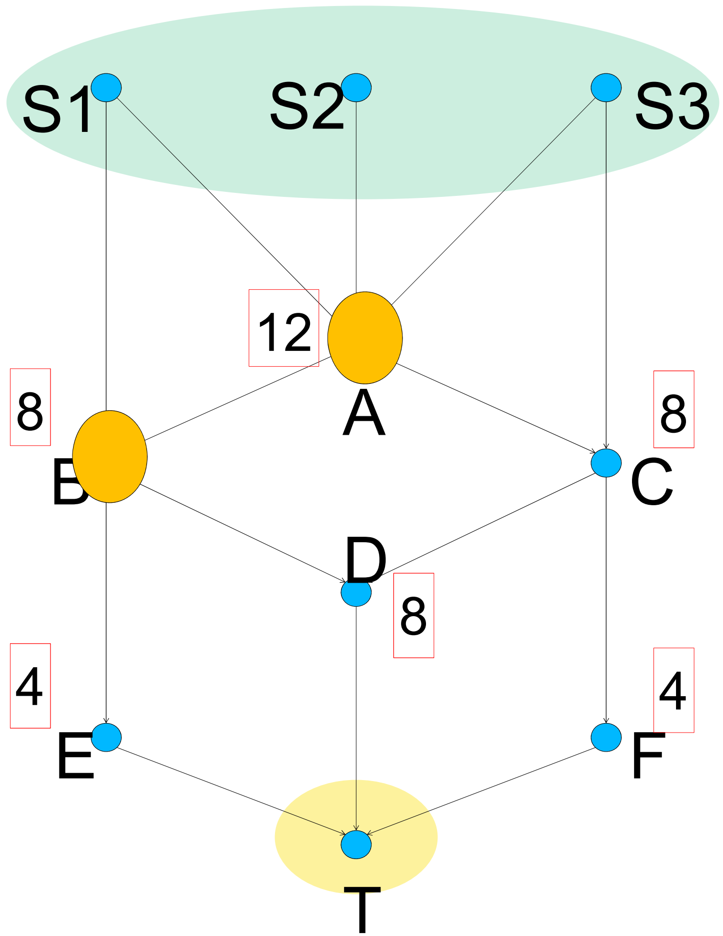
Group centrality

Q: Which two nodes have the largest total NumPaths centrality?

Nodes A and B (or C or D) have the highest NumPaths-centrality

C (A,B) = 14

Nodes B and C can cover all paths



[Everett, Borgatti, '99]

Group centrality

Q: Which two nodes have the largest total NumPaths centrality?

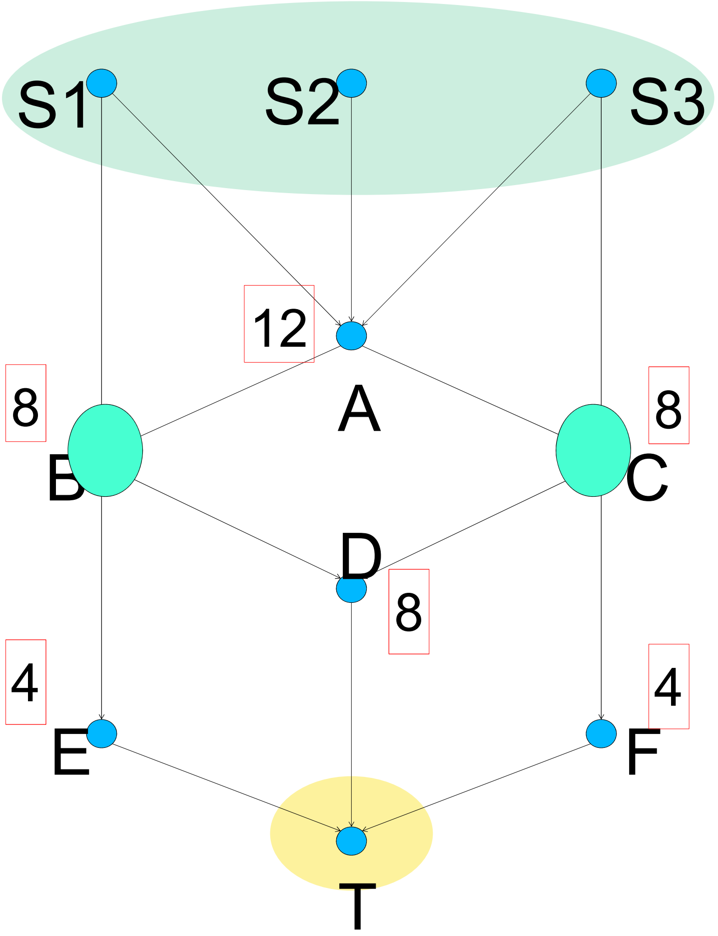
Nodes A and B (or C or D) have the highest NumPaths-centrality

C (A,B) = 14

Nodes B and C can cover all paths

C(B,C) = 16

Set{B,C} has the highest group centrality.



[Everett, Borgatti, ‘99]

Group centrality

$U = \{u_1, u_2, \ldots, u_k\}$ set of k nodes.

$P_U(s,t)$: set of special paths between source s and target t covered by some node in U.

Group Centrality of set U is a function of the paths in P that any node in U covers:

$$C(U) = \sum_{s,t} F(P_U(s,t))$$

Group centrality – k-Group Centrality Maximization problem (k-GCM)

Optimization problem:

Given graph G(V,E) and integer k find the set of k nodes with highest group centrality.

- k-GCM is NP-hard for NumShortestPaths[1], NumPaths, ShortestPaths centrality
- Objective function is monotone submodular for these centralities
- Greedy-type heuristic yields (1-1/e)-approximation algorithm

[1][Dolev et al. 2009]

Outline

- Motivation
- General framework for computing centrality
  - Centrality of nodes
  - Centrality of Groups
- Graph modifications to change centrality values
- Experiments

Outline

Motivation

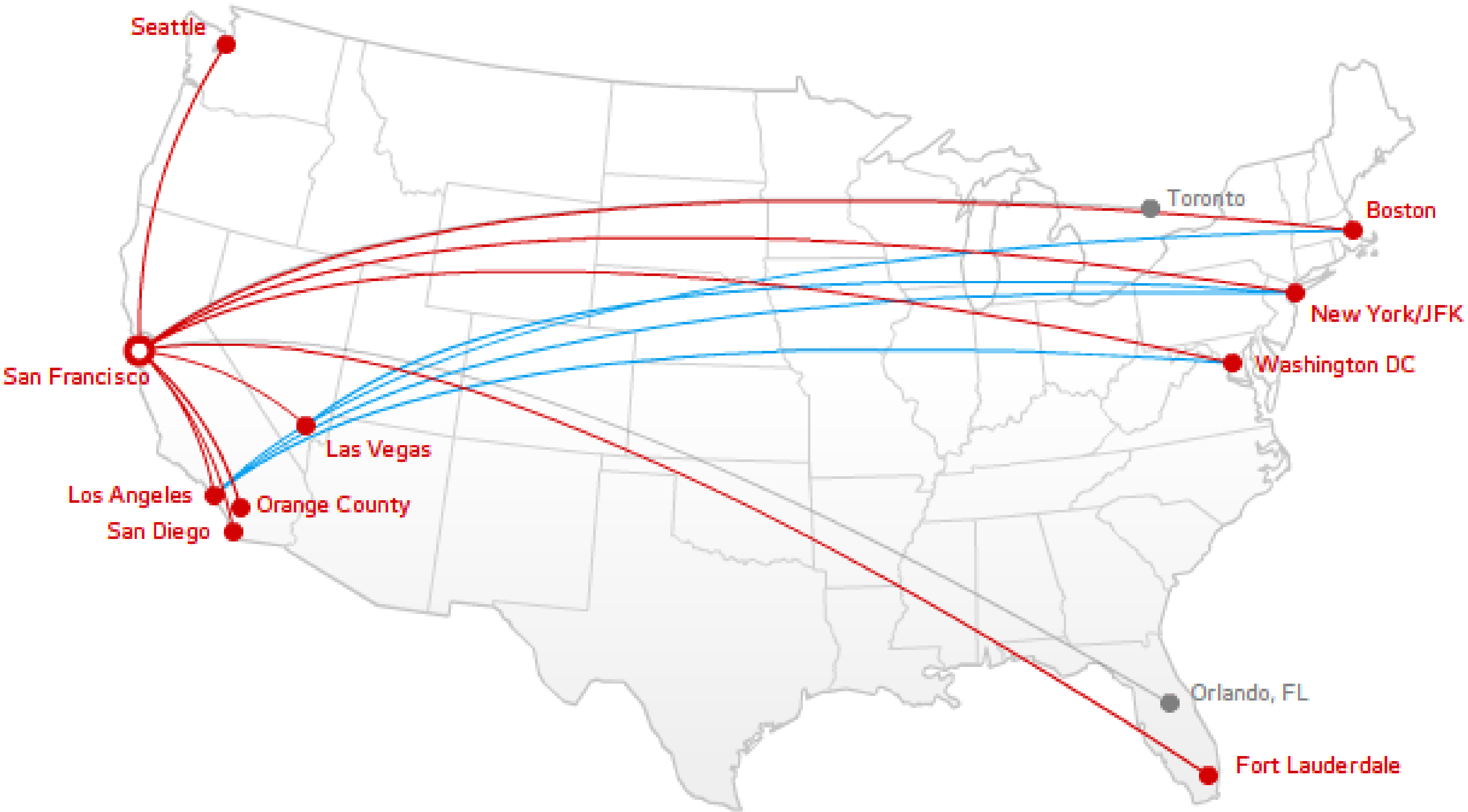
General framework for computing centrality

Centrality of nodes

Centrality of Groups

Graph modifications to change centrality values

Experiments



Where to add new flights so that the number of travellers is maximized?

Outline

Motivation

General framework for computing centrality

Centrality of nodes

Centrality of Groups

Graph modifications to change centrality values

Experiments

Experiments

MemeTracker[1] dataset – a network of online media sites, where edges correspond to hyperlinks. We choose a directed acyclic subgraph with 20K nodes and 80K edges.

[1][Leskovec et al. 2009]

Experiments – baseline algorithms

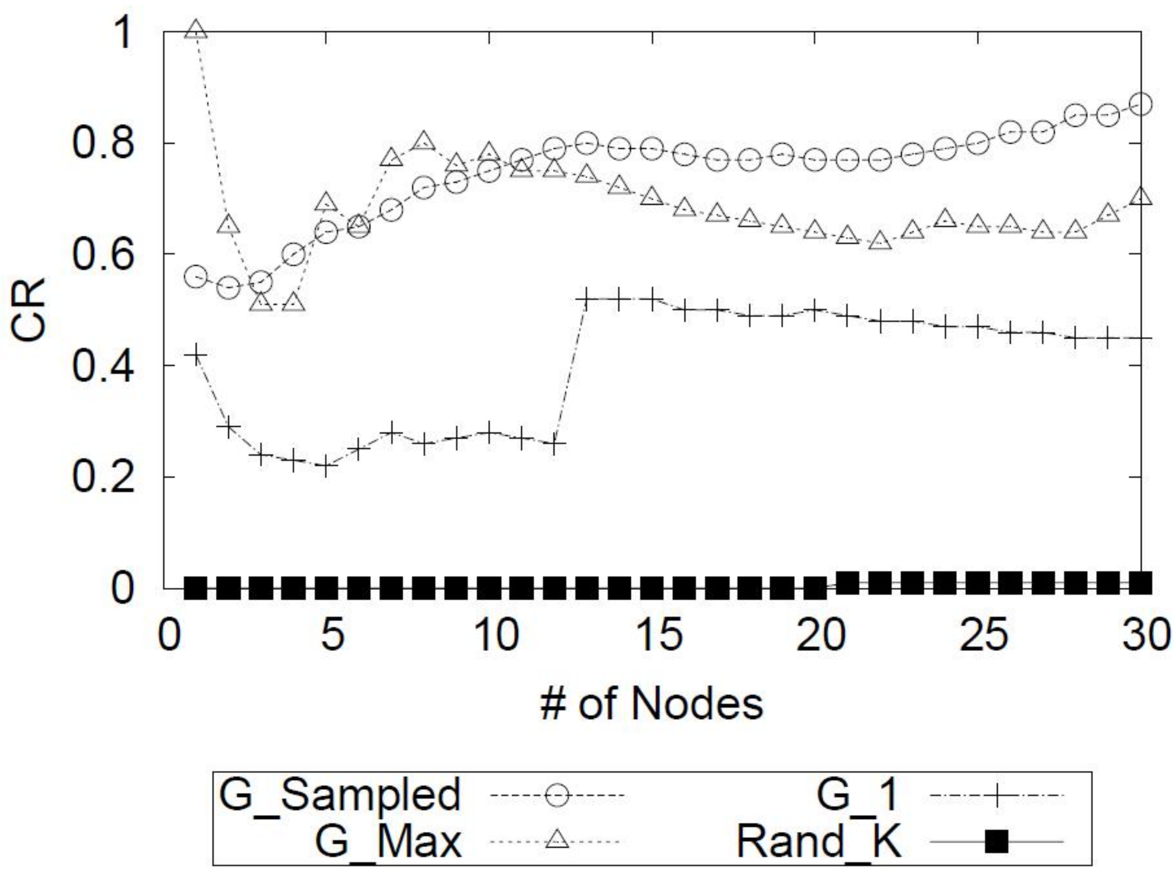
Greedy_Sampled: take a sample of the graph by removing edges at random. Then apply our Greedy algorithm to the sampled graph

Greedy_max: pick k nodes with largest individual centrality values

Greedy_1: pick k nodes with highest d_{in} x d_{out}

Random_k: pick k nodes uniformely at random

Experiments – NumShortestPaths - MemeTracker



Coverage Ratio – performance of baseline algorithm compared to our greedy algorithm

$$CR = \frac{C(A_{\text{Baseline}})}{C(A_{\text{Greedy}})}$$

References

- Brandes, *A faster algorithm for betweenness centrality*, J. of Math. Sociology, 2001.
- Brandes, *On variants of the shortest-path betweenness centrality and their generic computation*, Social Networks, 2008.
- Everett, Borgatti, *The centrality of groups and classes*, J. of Math. Sociology, 1999.
- Dolev, Elovici, Puzis, Zilberman, *Incremental deployment of networks based on group betweennes centrality*, Inf. Pdoccess. Letters 109, 2009.

Thank You!