ArgP2P: An argumentative approach for intelligent query routing in P2P networks

Ana L. Nicolini
aln@cs.uns.edu.ar

Ana G. Maguitman
agm@cs.uns.edu.ar

Carlos I. Chesñevar
cic@cs.uns.edu.ar

Content

- Searching efficiently in P2P networks: Problem context
- Thematic Search
- Closed communities problem
- ArgP2P framework
- Ilustrative example
- Conclusions

Searching efficiently in P2P networks: Problem context

- P2P network systems are becoming an increasingly important development in Web search technology.

- P2P application performance:
  - **degree of fault tolerance**
  - **scalability**
  - **quality of service**
  - **and network traffic**

Thematic Search

- T**hematic search** improves the performance of a P2P search application.
- To achieve good results for a query the system must try to predict which peers are best suited to answer the query.
- the best candidates are those peers whose content is semantically closest to the query topic.

Thematic Search

- All the participants (nodes) in the P2P network have a profile based on their interest.
- When a node sends a query, it can predict which are the candidate nodes by analyzing their profile.
- The knowledge of the profiles is accumulated query by query (past experience).

Thematic Search

1

How to route q?

Node 1 knowledge

Chemistry#_22#_46#_99 Math#_4#_5 ⇦

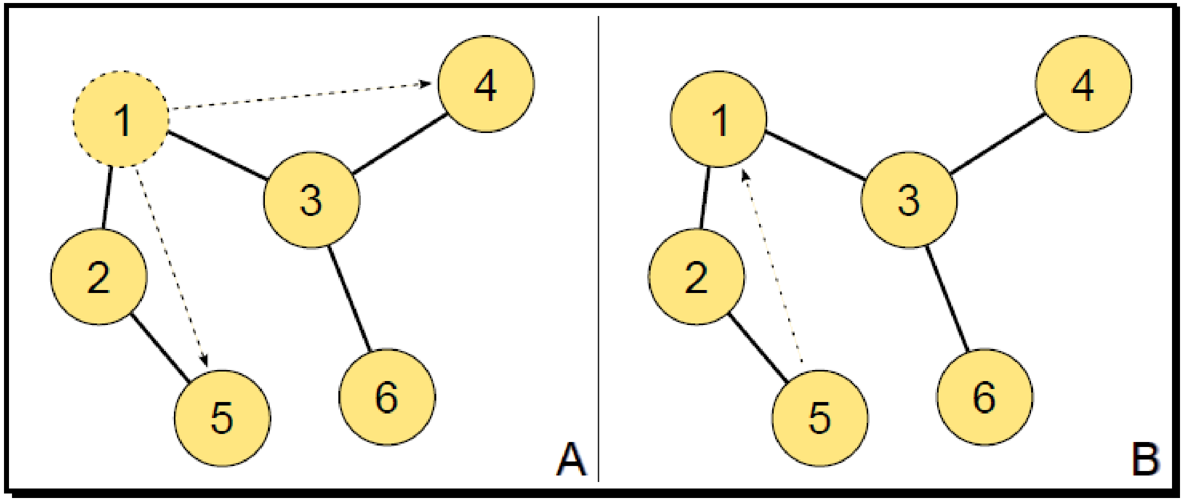
Node 1 has to route the following query *q*:

scalar product and vector product

Query topic

Math ⇦

Thematic Search



A) Node 1 sends query *q* to candidate nodes.

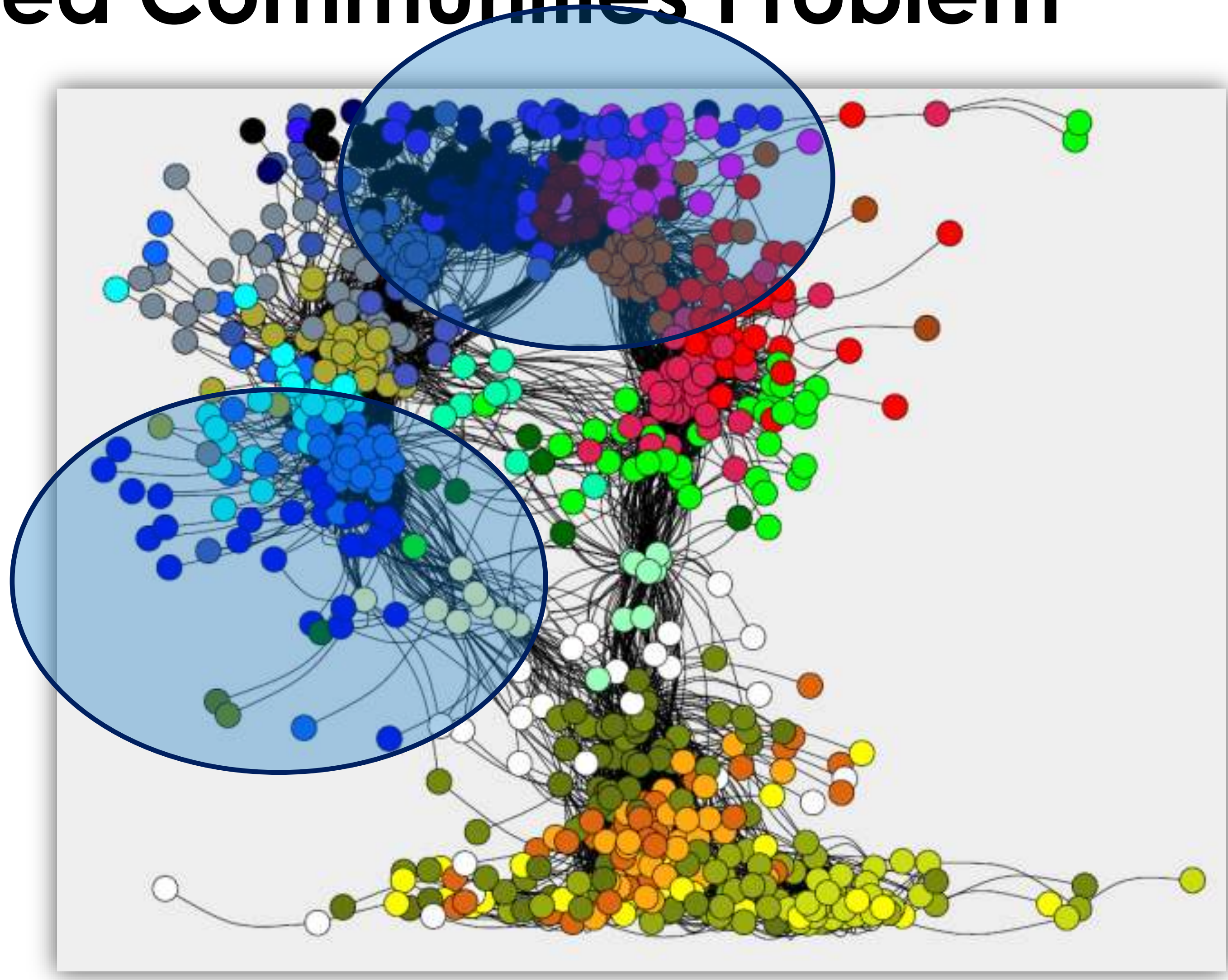
B) Node 5 answers query *q*.

Closed Communities Problem

- A no minor inconvenient that presented these algorithms is that when learning achieves a high degree, each node only communicates with those who know;

- A possible solution is to "explore" the network beyond their community to see what offers the rest of the peers.

Closed Communities Problem



Logical network obtained from a thematic search algorithm

ArgP2P framework

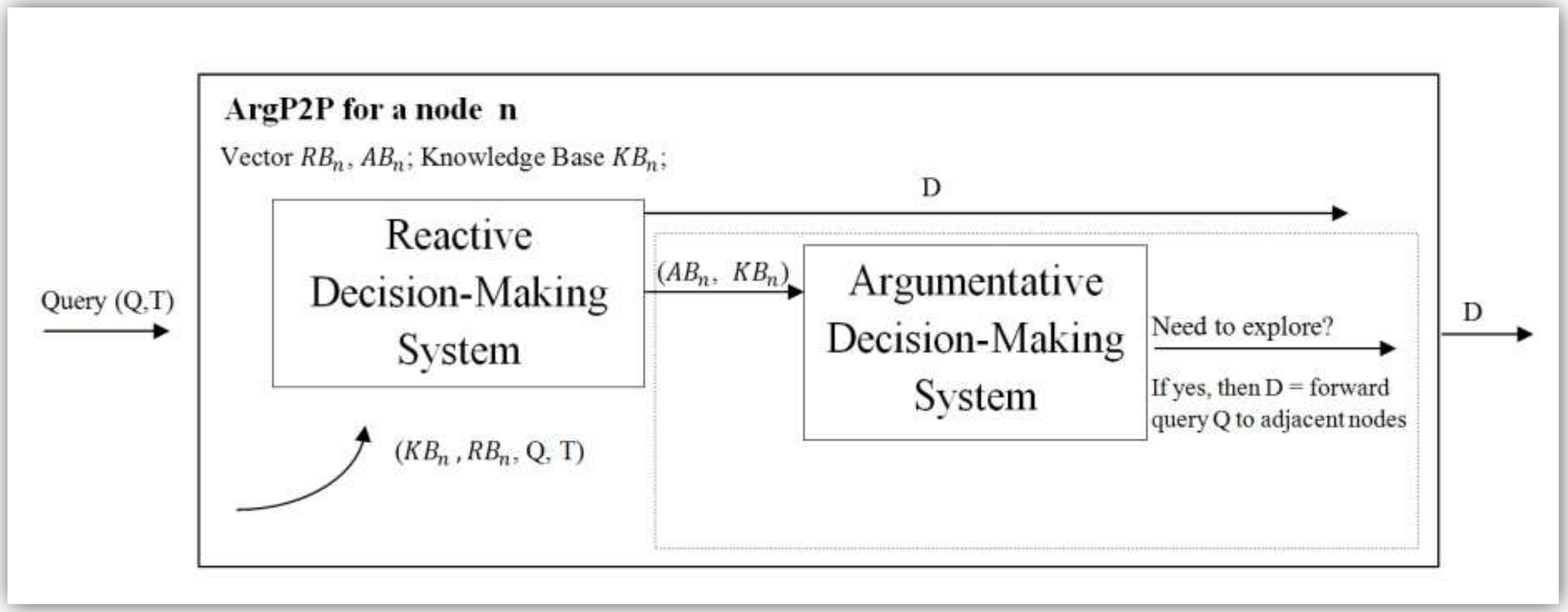
- The new proposal is to attack the "Closed communities problem" by adding a second degree of reasoning to each node, so they can decide when it is time **to explore** the network.

- The Arg P2P framework for a node *n* has the following components:
  - A Dynamic Knowledge base **KB***n* where the node knowledge is stored;
  - A vector **RB***n* that given a query Q associated with a topic *T* determines the following Boolean properties for node *n* : availability, relevance, awareness.
  - A vector **AB***n* that takes **KB***n* and determines the following Boolean properties: interest, reliability, congestion and need to explore.

ArgP2P framework

- Combining *reactive* and *rational* behavior:
  - The **Reactive Decision-Making System** for a node *n* is a black box system that implements the thematic search described above.
  - The **Argumentative Decision-Making** is an **ABA** (assumption-based argumentation) program for a node *n*, (based on KB*n*, AB*n*) used to determine whether the decision "need to explore" is warranted under grounded skeptical semantics.

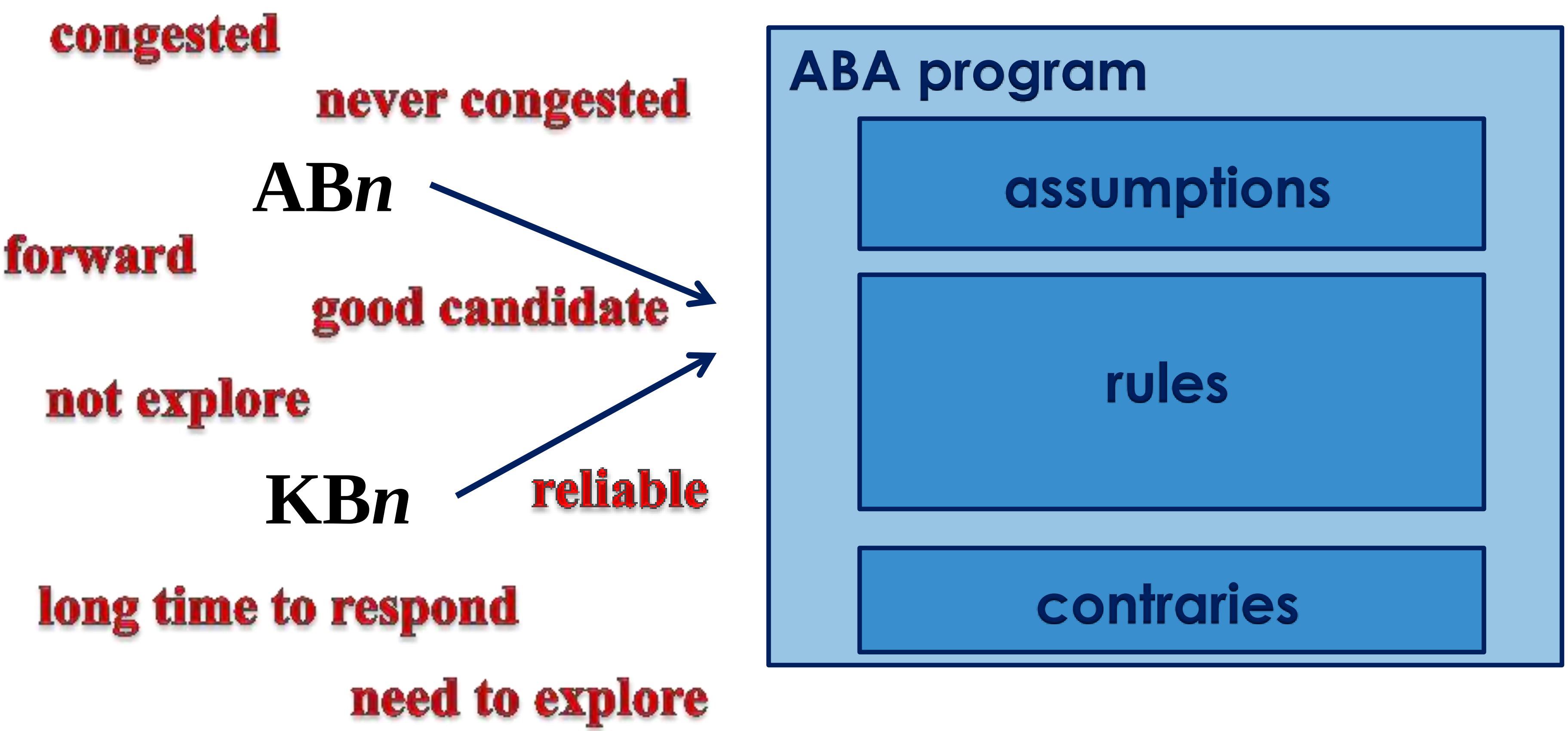
ArgP2P framework



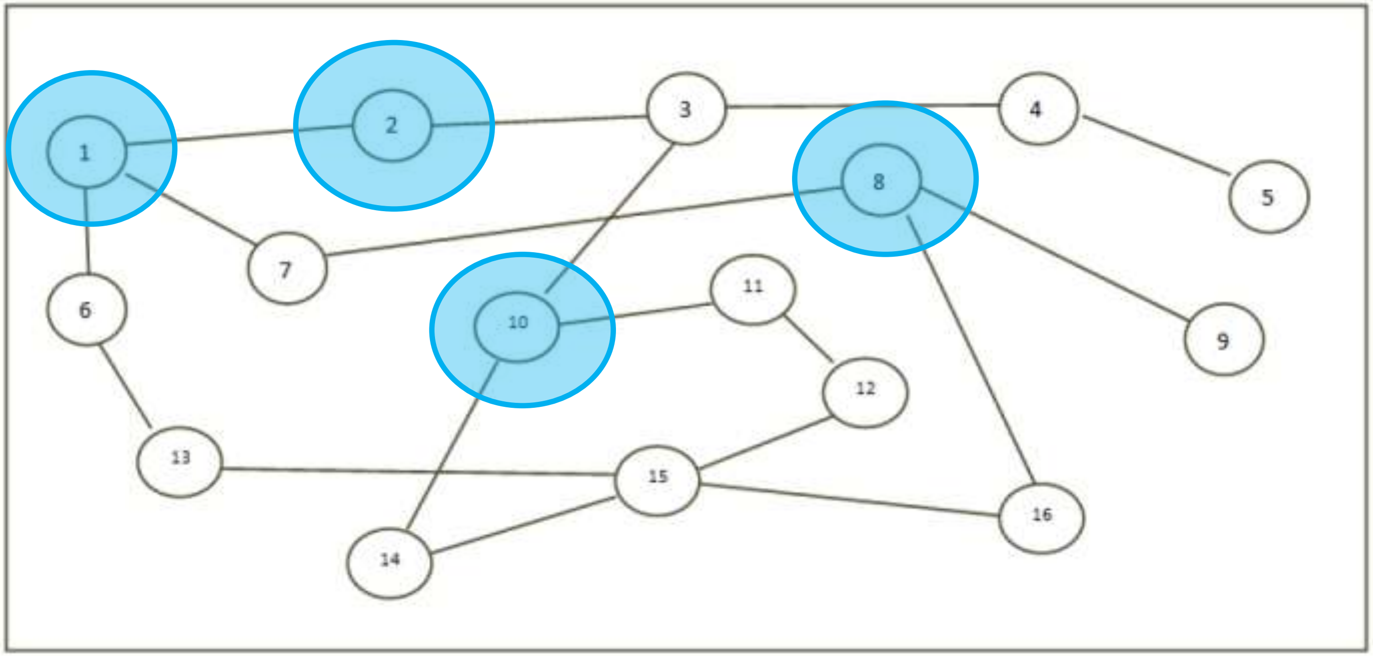
High-level description of the components in every node *n* of an ArgP2P framework.

ArgP2P framework

congested(1) ← longTimeToRespond(1), alwaysCongested(1);



Illustrative example - 1



- Suppose that nodes 1, 2, 8 and 10 are interested in the **math** topic;
- node 1 generates a query ***q*** also associated with the **math** topic;
- only Node 15 can respond to this query.

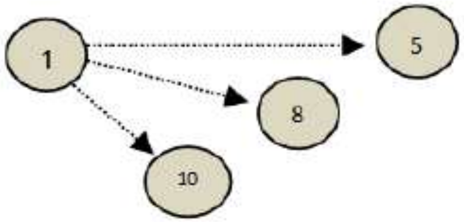
Illustrative example - 2

- Each node have the following knowledge:

KB_n	Nodes associated with the topic "math"		
KB_1	5	8	10
KB_5	1	8	10
KB_8	1	5	10
KB_{10}	1	5	8

- Formal representation of the network at time *t=1*

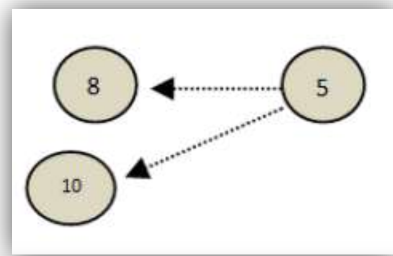
$$S_1(n) = \text{idle}\ \forall\ n \in \mathcal{N} - \{1\}$$
$$S_1(1) = (consult, \{5, 8, 10\})$$

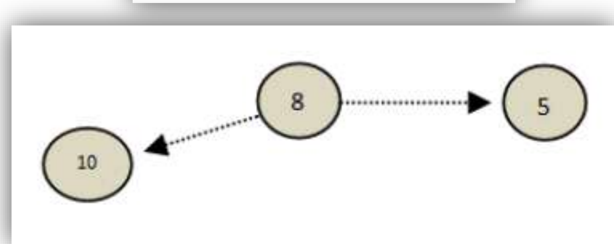


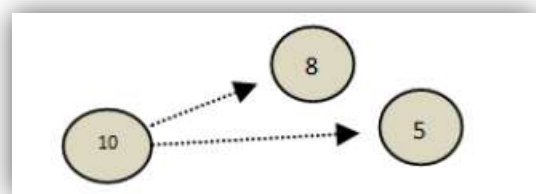
Illustrative example - 3

- Then nodes 5, 8 and 10 receive the forwarded message and they cannot answer the query so they forward the message to candidate nodes at time *t*=2.

$$
\begin{aligned}
S_2(n) &= \text{idle}\ \forall\, n \in \mathcal{N} - \{5, 8, 10\} \\
S_2(5) &= (forward, \{8, 10\}) \\
S_2(8) &= (forward, \{5, 10\}) \\
S_2(10) &= (forward, \{5, 8\})
\end{aligned}
$$







Illustrative example - 4

- After forwarding the query (time *t*=3):
  - node 5 receives a message from node 8 that was generated by node 1 and a message from node 10 that was also generated by node 1.
  - node 10 receives a message from node 5 generated by node 1 and another from node 8 also originated in node 1.

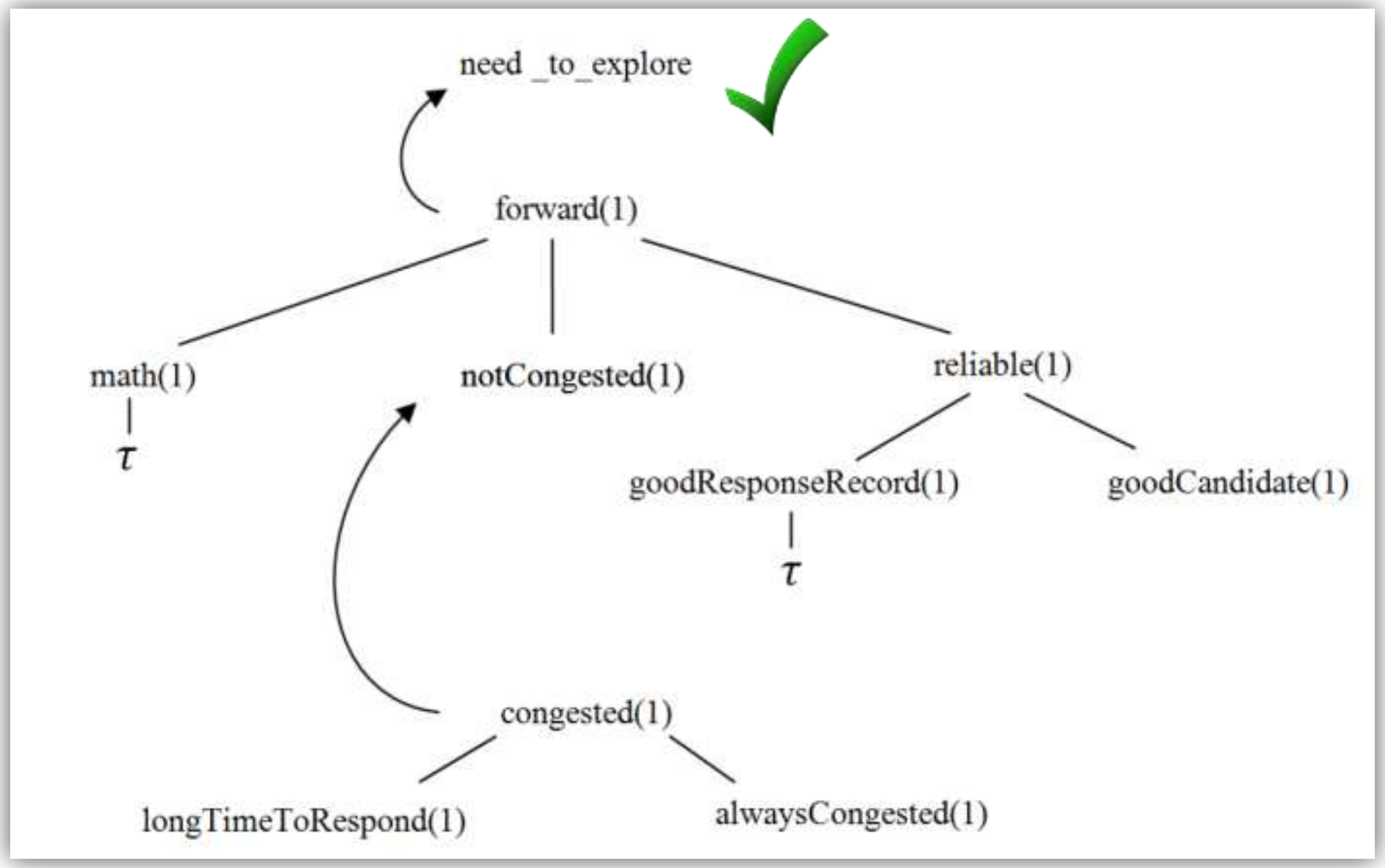
ALL THESE MESSAGES ARE DISCARDED

$$S_3(n) = \text{idle} \; \forall \; n \in \mathcal{N} - \{5, 8, 10\}$$
$$S_3(5) = \text{discard}$$
$$S_3(8) = \text{discard}$$
$$S_3(10) = \text{discard}$$

Illustrative example - 5

- Assume that in order to solve this problem, node 8 executes the **Argumentative Decision-Making System**.
- From the execution an ABA program is built from the AB_8 vector and KB_8 knowledge base.
- This program allows node 8 to determine if it is time to explore the network.
- If the node decides to explore, then it discards the decision taken by the **Reactive Decision-Making System** and explores the network.

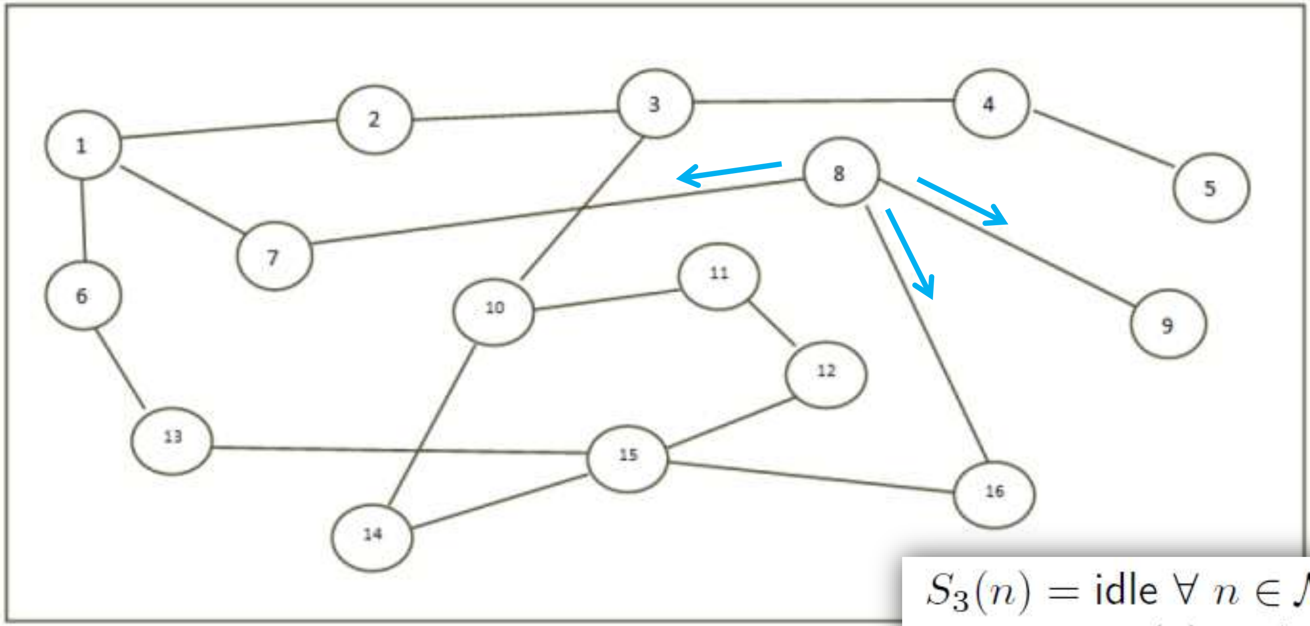
Illustrative example - 6



Arguments and attacks involved in a derivation tree supporting the decision “need to explore“ .

Illustrative example - 7

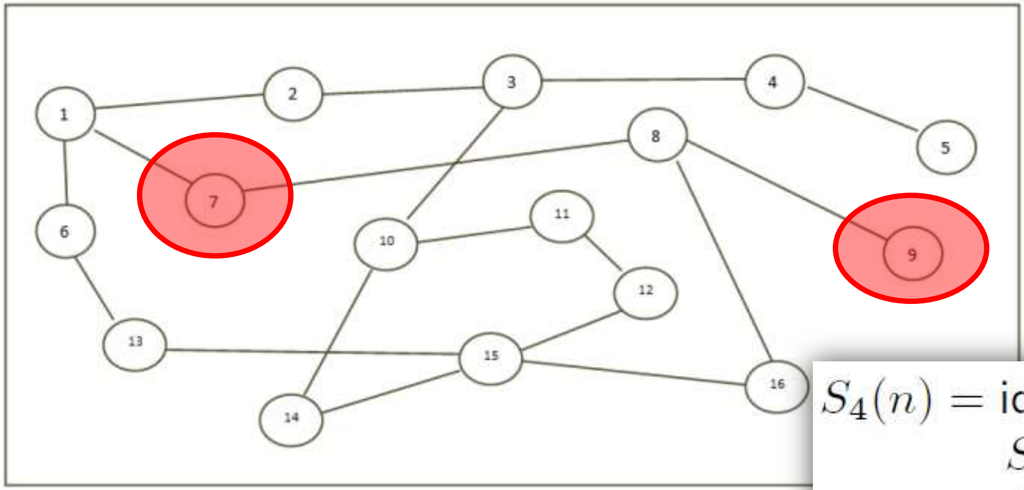
- So node 8 decides to explore the network:
  - it will send messages to nodes 7, 9 and 16 at *t* = 3.



$$S_3(n) = \text{idle} \ \forall\, n \in \mathcal{N} - \{5, 8, 10\}$$
$$S_3(5) = \text{discard}$$
$$S_3(8) = (forward, \{7, 9, 16\})$$
$$S_3(10) = \text{discard}$$

Illustrative example - 8

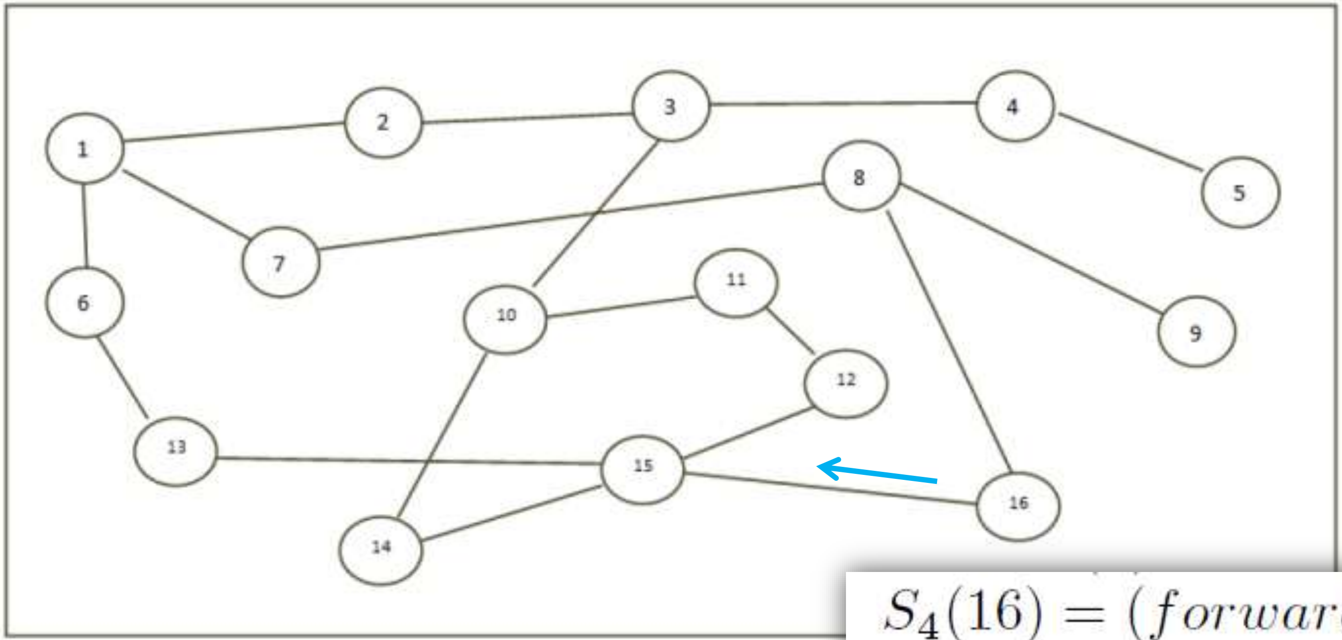
- Node 7 only has node 1 as an adjacent node, which is the same node that generated the query and therefore it is not considered. (*t=4*)
- Node 9 has no potentially useful adjacent nodes to forward the message (the message arrived from node 8, so it is not considered). (*t=4*)



$$S_4(n) = \text{idle} \ \forall \ n \in \mathcal{N} - \{7, 9, 16\}$$
$$S_4(7) = \text{discard}$$
$$S_4(9) = \text{discard}$$

Illustrative example - 9

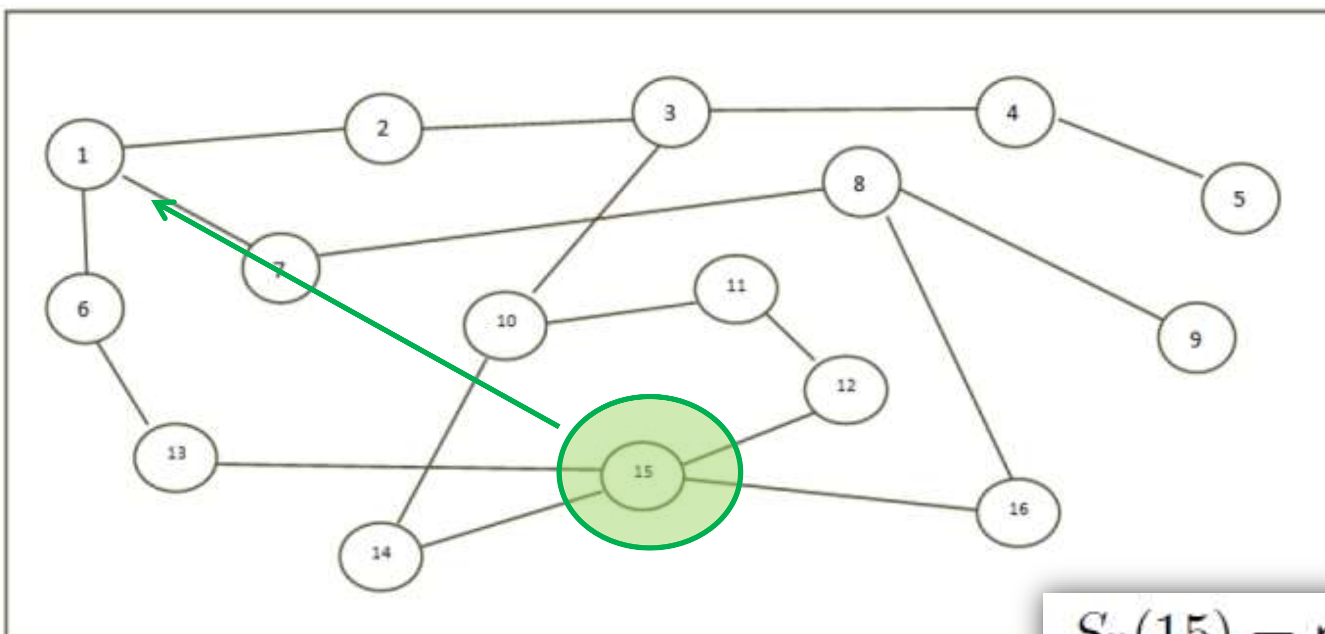
- Node 16 will forward the message to node 15 (at time t = 4).



$$S_4(16) = (forward, \{15\})$$

Illustrative example - 10

- Finally, at time $t = 5$, node 15 is able to answer the query originated by node 1.



$S_5(15) = \text{reply}$

Conclusions

- In this work we have outlined a model for thematic search in P2P networks, where every node in the network has the ability to combine both **reactive** and **rational** behavior.
- The argumentative inference engine is provided by an **ABA** program, which allows to enhance the decision making capabilities in every node, based on knowledge acquired by the node during its lifetime.
- Combining reactive and rational systems we offer a possible solution to the closed communities problem to perform **thematic search**.

Conclusions (cont)

- A configuration parameter allows to set the frequency by which argumentation is used during the lifetime of a node, so we can prioritize between efficiency and effectiveness.
- Part of our future work involves an empirical comparison between the ArgP2P framework and a purely reactive P2P framework.

Questions?

Ana L. Nicolini
aln@cs.uns.edu.ar

Ana G. Maguitman
agm@cs.uns.edu.ar

Carlos I. Chesñevar
cic@cs.uns.edu.ar