EWSDN
European Workshop on Software Defined Networks

1-Oct 2015, Bilbao, Spain

Towards Semantic Network Models via Graph Databases for SDN Applications

Talita de Paula Cypriano de Souza and Christian Esteve Rothenberg
University of Campinas (UNICAMP)
{cypriano,chesteve}@dca.fee.unicamp.br

Mateus Augusto Silva Santos
Ericsson Research
Indaiatuba, SP, Brazil
mateus.santos@ericsson.com

Luciano Bernardes de Paula
Federal Institute of Sao Paulo (IFSP)
Braganca Paulista, SP, Brazil
lbernardes@ifsp.edu.br

INTRIG
INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

FEEC
Faculdade de Engenharia Elétrica e de Computação
UNICAMP

Agenda

- Introduction
- Goals
- Related Work
- Proposal
- Experimental Evaluation and Results
- Conclusions and Future Work

Agenda

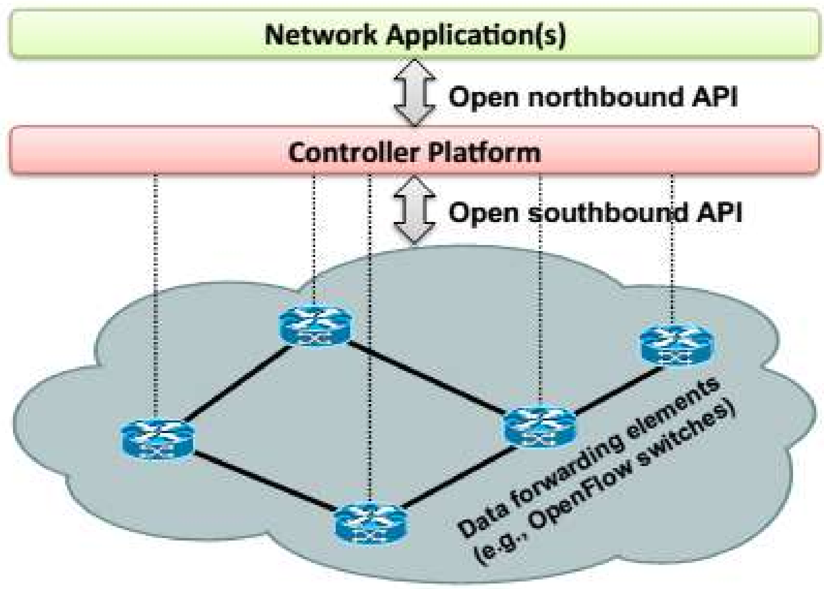
- Introduction
- Goals
- Related Work
- Proposal
- Experimental Evaluation and Results
- Conclusions and Future Work

Introduction & Motivation

- ✓ Rise of Software Defined Networking (SDN);
- ✓ Information models and data structures for topology (and more) needed in *any* network management / control technology;
  - ✓ Lots of related standardization efforts (IETF, YANG, ONF CIM, etc.)
- ✓ Evolution and Maturity of the Semantic Web (e.g., ontologies, RDF)
  - ✓ ... and success stories in niche application domains;
  - ✓ Networking? NML
- ✓ Popularity of NoSQL DB, e.g., graph databases such as Neo4j;
  - ✓ Scalabilty properties & Natural approach to graph/network problems

Software Defined Networking

- ✓*Software Defined Networking* (SDN):
  - ✓Clean / Programmatic Separation of control and data planes;
- ✓New abstractions in controlling and network forwarding;

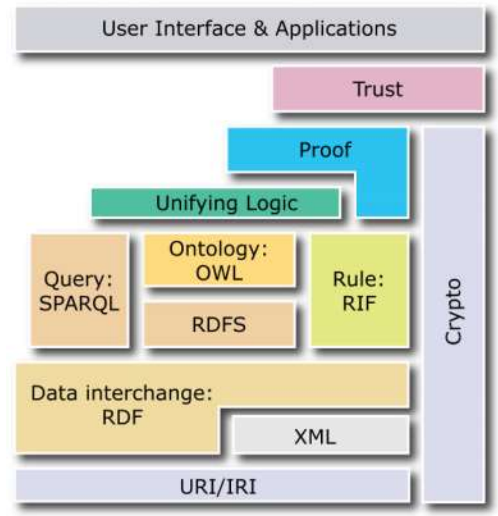


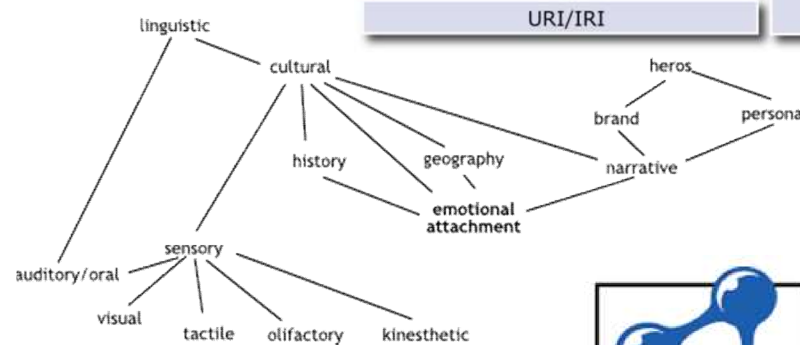
KREUTZ, D. et al., 2015

- ✓API for packet flow abstraction
  - ✓OpenFlow Protocol;
- ✓Network Topology Abstraction;
  - ✓?

Semantic Models

- ✓Principles of the Semantic Web:
  - ✓Allows reuse of information;
  - ✓Data integration among organizations;
  - ✓Enhanced (rich) Web search;
  - ✓Guaranteed accessibility
- ✓Ontology;
  - ✓ "specification of a conceptualization"
- ✓*Web Ontology Language* (OWL);
- ✓*Resource Description Framework* (RDF):
  - ✓<subject, predicate, object>;





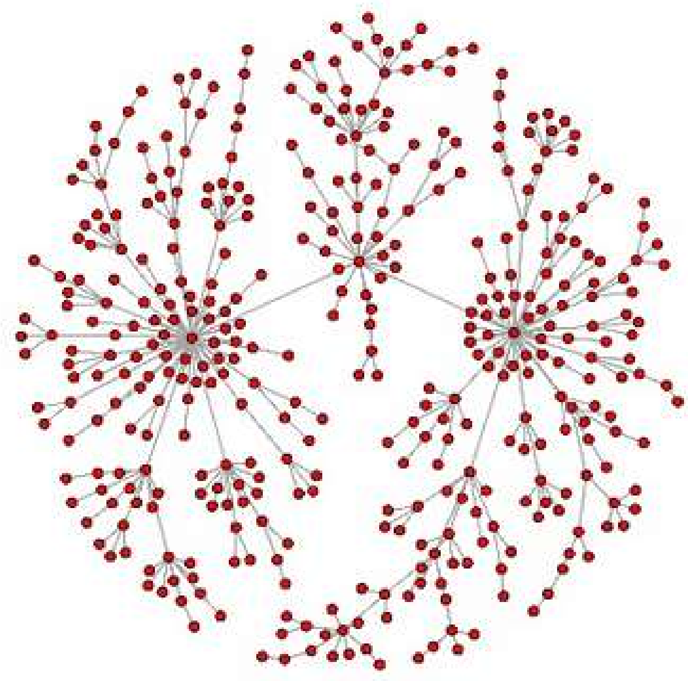
Databases

- ✓ **Relational Database Model**
  - ✓ **Consolidated; Well Documented;**
  - ✓ **ACID Transactions (Atomicity, Consistency, Isolation and Durability)**
  - ✓ **Limitations:**
    - ✓ **Queries of highly interconnected data;**
    - ✓ **Data Modelling is Adapted**
- ✓ **NOSQL (http://nosql-database.org/)**
  - ✓ **Schema Free;**
  - ✓ **Scalability;**
  - ✓ **Disponibility;**
  - ✓ **Response time ;**
  - ✓ **Horizontal Scalling;**

Not Only SQL

Graph Databases (GDB)

- ✓Graph:
  - ✓Nodes;
  - ✓Edges;
- ✓Topology;
- ✓Interconnected Data; (Relationship-centered);
- ✓Natural modeling of problems, e.g:
  - ✓Semantic Web;
  - ✓Computer Networks;
  - ✓Recommendation Engines, etc;



Agenda

- Introduction
- Goals
- Related Work
- Proposal
- Experimental Evaluation and Results
- Conclusions and Future Work

Goals

- Apply a semantic model to describe network topologies (and complete network+compute infrastructures) in the context of SDN controllers leveraging graph databases;
- Map SDN primitives (in the literature) as graph database queries;
- Identify limitations of the chosen semantic language in support of SDN application primitives;
- PoC system profiling: Evaluate the performance of a prototype;

Agenda

- Introduction
- Goals
- Related Work
- Proposal
- Experimental Evaluation and Results
- Conclusions and Future Work

SDN Controllers

- ONIX Controller (Koponen et al., 2010):
  - Pioneer distributed control plane implementation of SDN;
  - Graph model to define the partitioning and state distribution/maintenance among different controllers;
- Graphs in SDN (Pantuza et al., 2014):
  - Support of dynamic network representation;
  - Mininum Spanning tree of network graph in a real time;
- NetGraph Library (Raghavendra et al., 2012):
  - Periodic updates of network state;
  - Queries results to SDN controller;

Do not use semantic notations
(n)or store network graphs in a persistent way.

Semantic Model for Networking

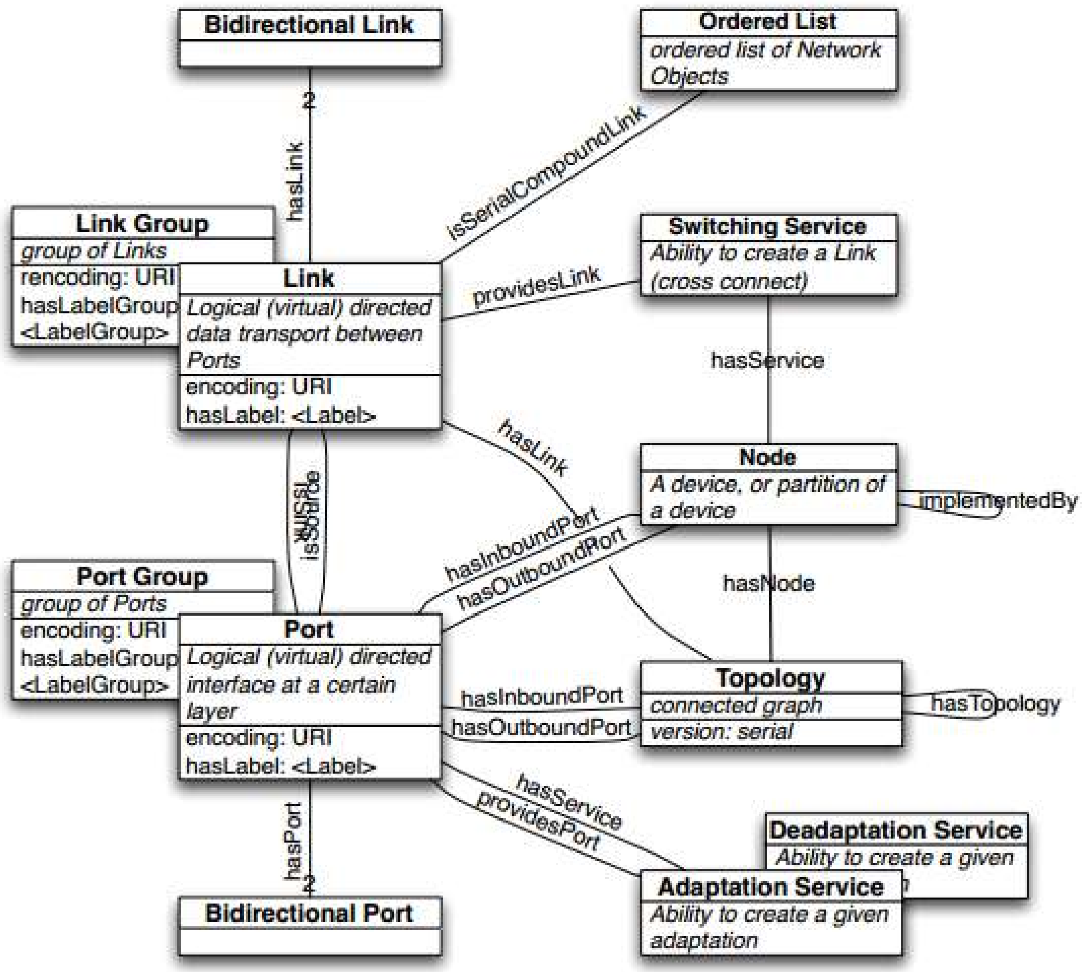
Network Markup Language **– NML (van der Ham et al., 2013)**

- Single network description standard under guidance of the Open Grid Forum (OGF)

  "complex multi-layer path finding, with a technology independent algorithm"

- Supports description of multi-layer and multi-domain networks:
  - Virtualized networks;
  - Heterogeneous network technologies;
- Model extensibility as neccessary, e.g.,
  - NOVI – Future Internet Platform:
    - http://www.fp7-novi.eu/
  - GEYSERS – Virtualizing Optical Networks:
    - http://www.i2cat.net/en/projects/geysers
  - CINEGRID – Distribution of Digital Media:
    - http://www.cinegrid.org

Semantic Models
Main NML Classes and Properties



GHIJSEN, M. et al., 2013

Graph Databases

- Benchmark of GDBs (Jouili e Vansteenberghe, 2013):
  - Neo4j, OrientDB, Titan, DEX;
  - Neo4j obtained the best query time results;
- Auditing Cloud Architectures (Soundararajan and Kakaraddi, 2014):
  - Neo4j e *Cypher*:
    - Risk Analysis;
    - Simple Reporting;
    - Inventory Comparison;

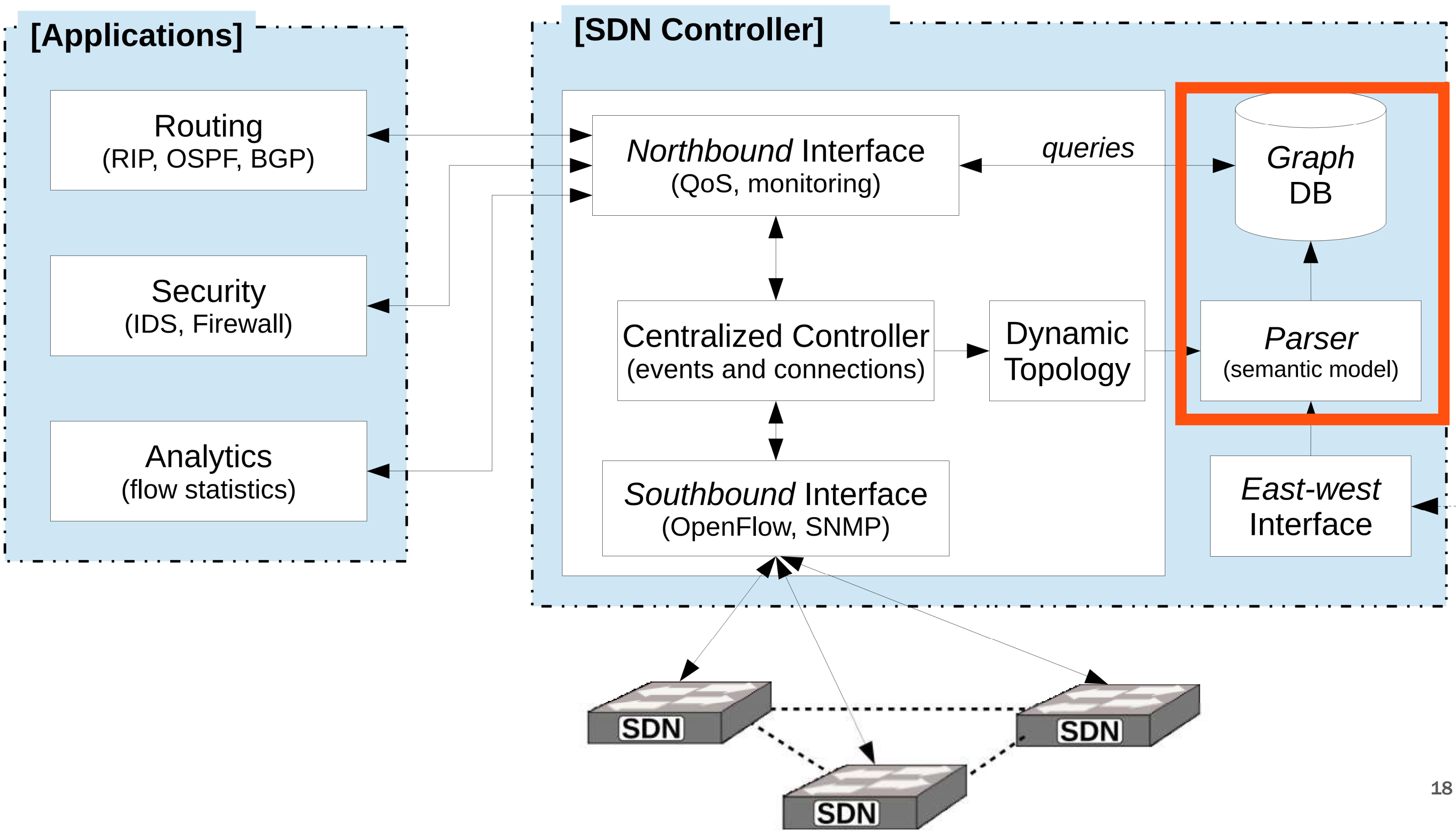
Graph Databases
Neo4j

- Native Storage and Processing of Graph;
- Open Source (Community Version);
- Property Model:
  - Nodes and relationships have properties;
- Query Language:
  - Cypher;
  - Gremlin (TinkerPop);

Agenda

- Introduction
- Goals
- Related Work
- **Proposal**
- Experimental Evaluation and Results
- Conclusions and Future Work

Strawman Architecture



Primitives Analysis
Compatibility of Primitives [NetGraph]

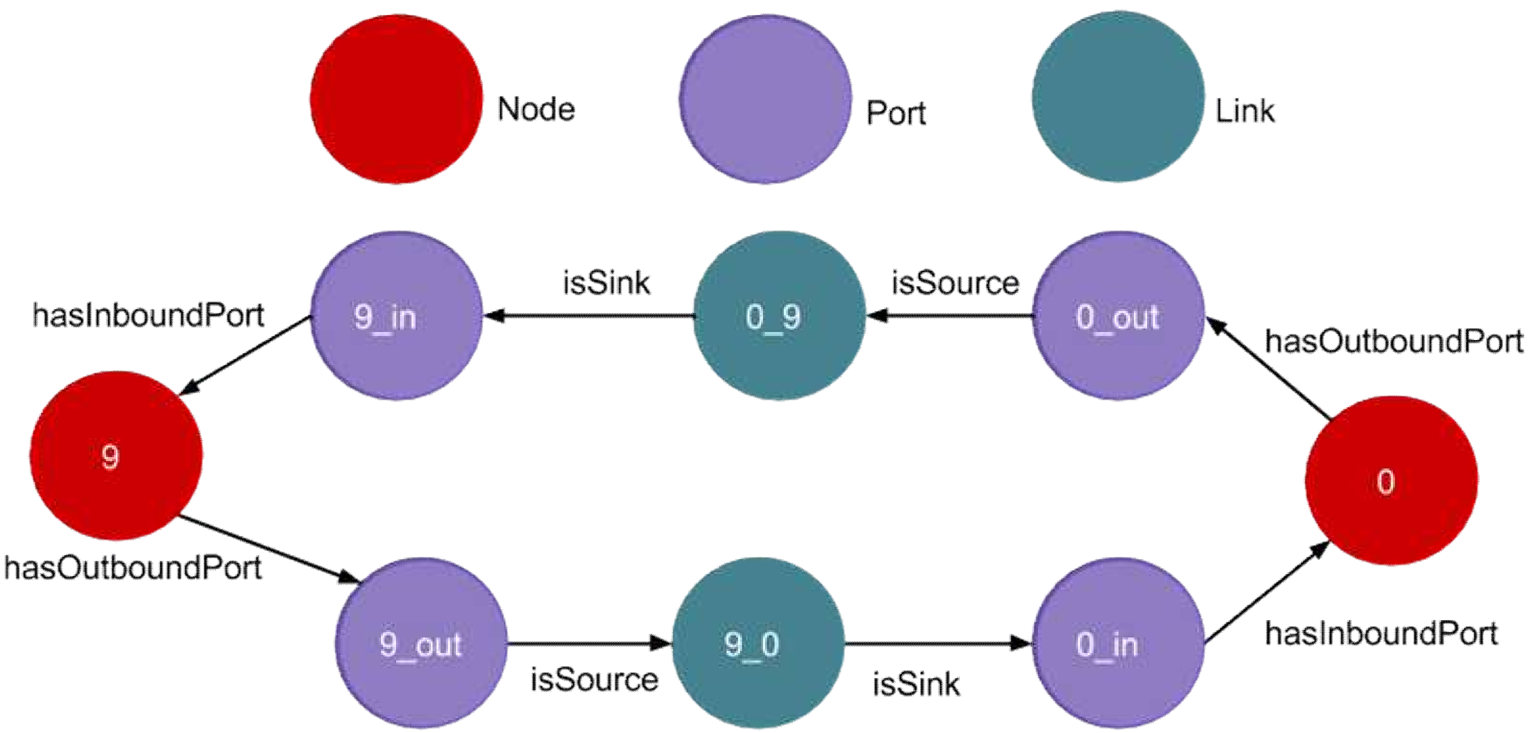
Primitive	Semantic Model	GBD	Read/Write
setEdgeWeight	No	Yes	W
getEdgeWeight	No	Yes	R
countInDegree	Yes	Yes	R
countOutDegree	Yes	Yes	R
countNeighbors	Yes	Yes	R
computeMST	Yes	Yes	R
computeAPSP	Yes	Yes	R
computeSSSP	Yes	Yes	R
doesRouteExist	Yes	Yes	R
computeKSSSP	Yes	Yes	R
delete	Yes	Yes	W
insert	Yes	Yes	W

Agenda

- Introduction
- Goals
- Related Work
- Proposal
- **Experimental Evaluation and Results**
- Conclusions and Future Work

Data Modeling

✓ Example of Relationship Modeling between Nodes "9" and "0"



Topologies

- Topology Generator: BRITE (Boston University);

Topology	Nodes (BRITE)	Resultant Graph
Tiny	10	76 nodes (160 relationships)
Small	100	640 nodes (1.760 relationships)
Medium	1.000	4.978 nodes (11.912 relationships)
Large	10.000	109.932 nodes (359.728 relationships)

Queries

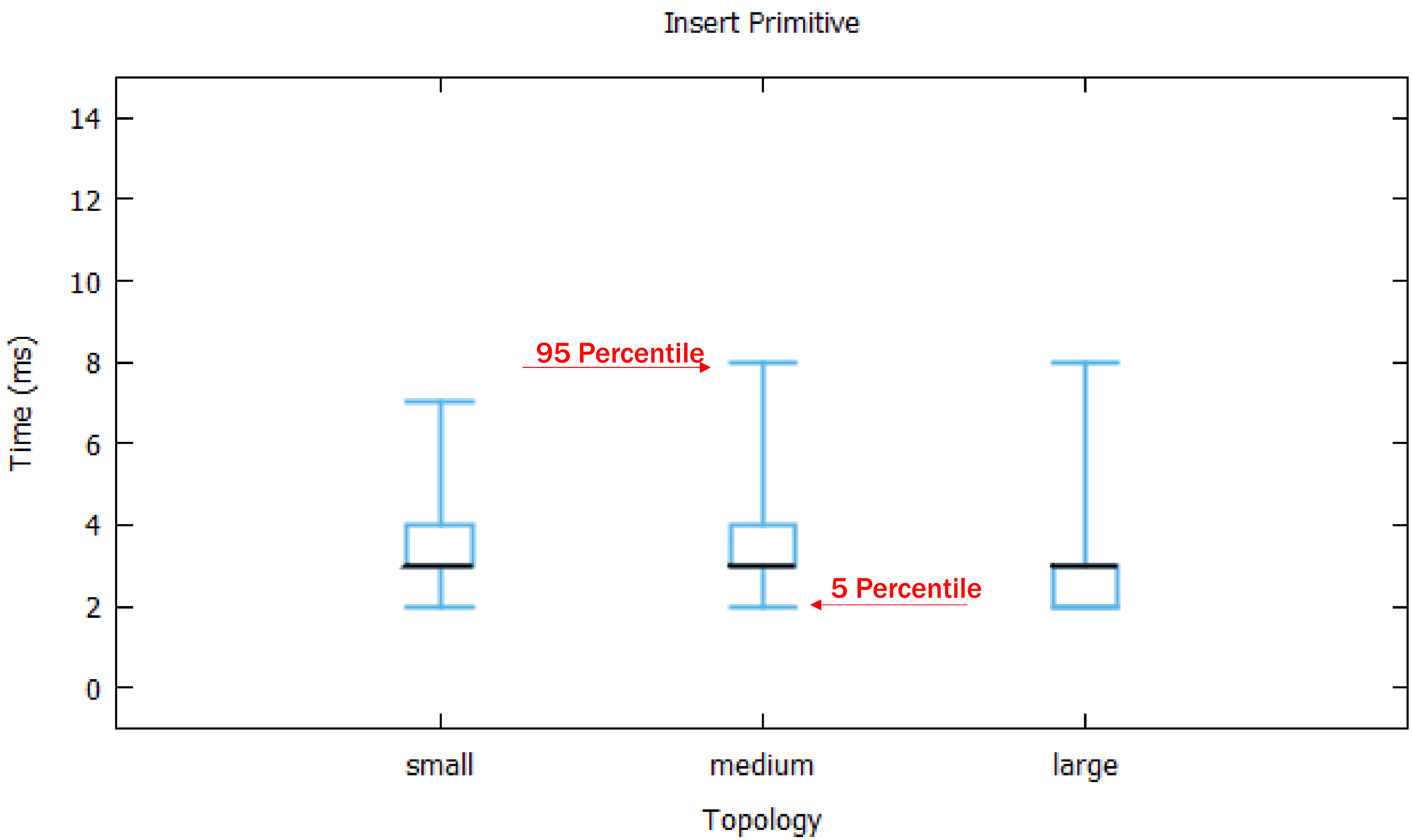
- **Fixed Internet-like topologies (BRITE) of diferent sizes;**
- **Random atributes;**
- **Each primitive executed 1.000 times in each topology:**
- **Cypher Language:**
  - **E.g.:**

```
MATCH (n:Node)-[:hasOutboundPort]->(p:Port)-[isSource]->(l:Link)
WHERE n.name="A"
RETURN COUNT(l) AS CountOutDegree
```

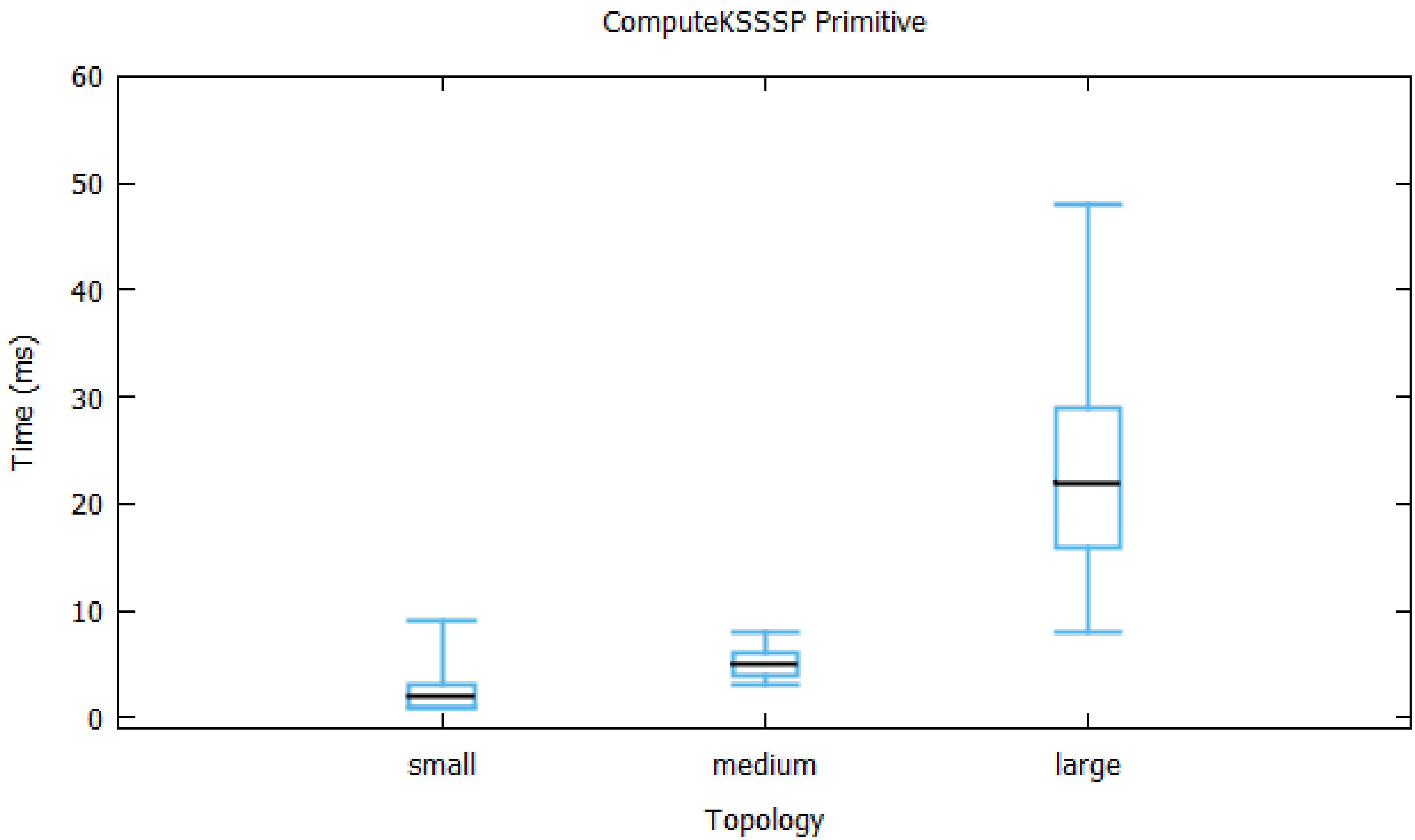
Results Analysis
Large Topology *(ms)*

Primitive	Average	Standard Deviation	99 Percentile
setEdgeWeight	162,33	9,46	205,01
getEdgeWeight	1,70	0,74	4,00
countInDegree	854,53	146,77	1.399,05
countOutDegree	425,17	68,36	699,02
countNeighbors	4,45	2,27	10,01
doesRouteExist	37,51	29,09	73,06
computeMST	1,44	1,25	3.02
computeSSSP	5,47	4,98	29,00
computeKSSSP	26,21	37,23	81,04
computeAPSP	1,04	0,68	3,01
delete	1053,89	162,55	1637,02
insert	3,57	3,21	16,01

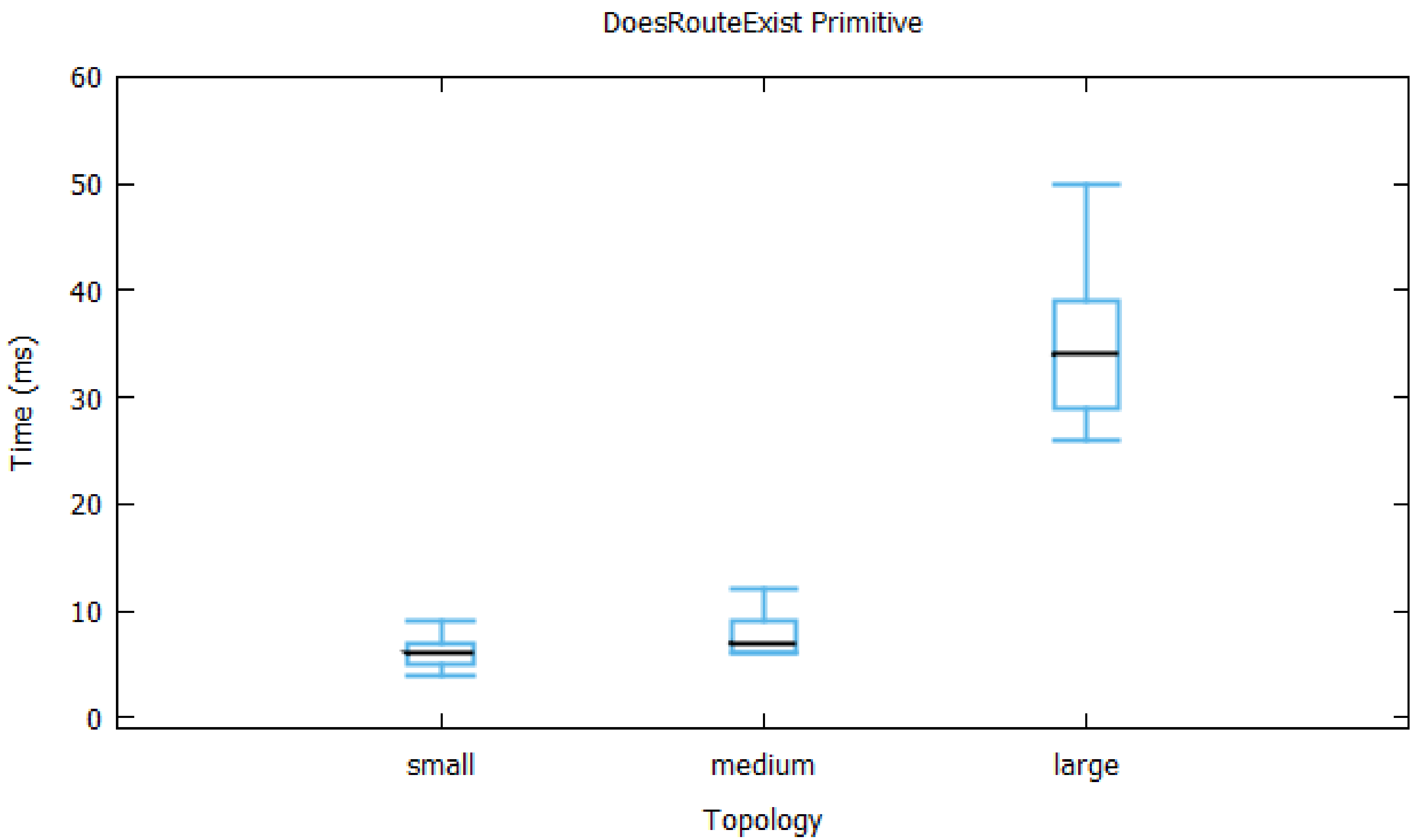
Results Analysis



Results Analysis



Results Analysis



Results Analysis
Primitives with large response time

- **Count In Degree and Count Out Degree:**
  - Number of hops (different relationships types):

NodeA ← *hasInboundPort* ← *Port* ← *isSink* ← *Link*

- **Delete:**
  - Number of hops;
  - Depends on connectivity of deleted node

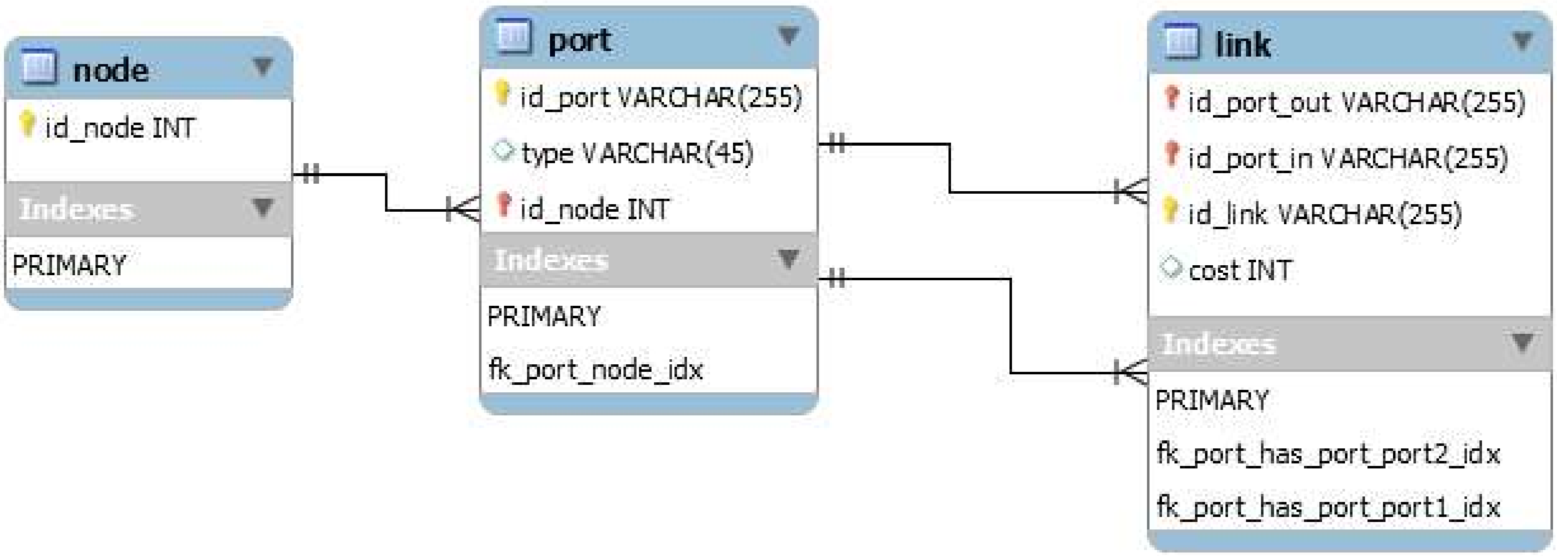
- **Set Edge Weight:**
  - Read-Write Operation;

Results Analysis
Shortest Path Primitives

- All Pairs of Shortest Path (compute APSP) is faster than K-Shortest Path (computeKSSSP) and Shortest Path (computeSSSP);

- The GDB optimizes All Pairs of Shortest Path Computing, because during the traversal it computes the shortest path among the intermediates nodes;

Relational Model
Enhanced Entity Relationship Model (EER)



Relational Model
Results (MySQL) –Large Topology

Primitive	Average	Standart Deviation	Percentile 99
countInDegree	1,39	4,57	22,02
computeSSSP	18,13	3,82	26,00
computeAPSP	2,11	1,39	7,00
Delete	162,86	79,93	405,00
Insert	137,36	43,80	300,00

Comparison

Relational Model

- Adapted modelling to tables;
- For computing shortest paths it was necessary to implement/adapt an algorithm
- Lower response time:
  - *CountInDegree*
  - *Delete*

Graph Model

- Natural modelling;
- Native functions to compute shortest paths;
- Lower response time:
  - *ComputeSSSP*
  - *Insert*

Agenda

- Introduction
- Goals
- Related Work
- Proposal
- Experimental Evaluation and Results
- **Conclusions and Future Work**

Conclusions

Feasibility of indexing a network topology following a semantic model (NML) in a graph database (Neo4j) in the context of SDN primitives;

- Integration Architecture (NML-to-Neo4j parser) proposed;
- Basic SDN control application primitives reproduced;
- Some limitations of the semantic model were identified;
- Neo4j graph DB technology choice compatible (property graphs) with the network modeling problem
  - Promising performance and scalability
  - Cypher language exhibited good flexibility;

Future Work

- Evaluate the performance with dynamic workloads and applications on OpenDaylight controller using REST APIs;
- Develop extensions of the Semantic Model (NML) to meet SDN (and NFV) appications needs;
  - Use cases under investigation include SDN eXchanges, east/west interfaces, controller platform for multiple applications
- Develop new graph-oriented primitives;
- Explore system optimizations to reduce latency and increase scalability;

Thanks! Obrigado! (More) Questions?

https://github.com/intrig-unicamp/NML-Neo4j

cypriano@dca.fee.unicamp.br

INTRIG INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

Backup

Results Analysis
Small Topology *(ms)*

Primitive	Average	Standard Deviation	99 Percentile
setEdgeWeight	8,78	3,23	23,02
getEdgeWeight	1,73	0,76	3,00
countInDegree	17,94	11,36	65,01
countOutDegree	8,35	3,46	23,00
countNeighbors	6,16	22,43	14,07
doesRouteExist	6,55	3,82	15,02
computeMST	1,12	0,66	2,00
computeSSSP	1,34	1,38	4,00
computeKSSSP	2,94	3,44	12,00
computeAPSP	1,04	0,84	4,01
delete	20,71	7,20	48,01
insert	3,66	3,26	15,02

Results Analysis