Rethinking Network Policy Coordination
— A Database Perspective

Anduo Wang*  Seungwon Shin†  Eduard Dragut*
*Temple University  †KAIST
database usage in networking

great for managing network state
database usage in networking

great for managing network state — factual data
database usage in networking

data synchronization

transaction processing

SDN controller instance

SDN controller instance

host

host

host

host

host

host
database usage in networking

data query
database usage in networking

data query
database usage in networking

data query

distributed query processing
(declarative networking)
database usage in networking

focus on managing \textit{factual data}
database usage in networking
but database is also renowned for mediating semantic data
database usage in networking but database is also renowned for mediating semantic data — policies about what are the acceptable data
this talk

if and how can database systems help with managing network policies that can interact in complex ways
if and how can database systems help with managing network policies that can interact in complex ways
the policy interaction problem

disparate representations buried in the network
the policy interaction problem

disparate representations buried in the network
the policy interaction problem

disparate representations buried in the network
  - hinders rather than facilitates interaction
the policy interaction problem

disparate representations buried in the network

functions represent forwarding

graph represents service chain

FSM represents stateful middle-boxes

SDN controller
the policy interaction problem

disparate representations buried in the network

- hinders rather than facilitates interaction

functions represent forwarding
graph represents service chain
FSM represents stateful middle-boxes

SDN controller
the policy interaction problem

complementary policies in SDNs

- jointly satisfiable
the policy interaction problem

complementary policies in SDNs

- jointly satisfiable *but not independent*

network updates must be properly ordered
conflicting policies in inter-domain routing

- overlooked conflicts within an autonomous system (AS)

the policy interaction problem
the policy interaction problem

conflicting policies in inter-domain routing

- overlooked conflicts within an autonomous system (AS)
- AS3 and AS4 attempt to influence route selection of the middle AS

policies by multiple ASes unaware of each other conflict in a common neighbor
the policy interaction problem

disparate representations buried in the network

complementary policies in SDNs
- jointly satisfiable but not independent

conflicting policies in inter-domain routing
- overlooked conflicts within an AS
a database solution

disparate representations buried in the network

→ a unified knowledge representation

- policy as integrity constraints (ICs)

complementary policies in SDNs

- jointly satisfiable but not independent

conflicting policies in inter-domain routing

- overlooked conflicts within an AS
disparate representations buried in the network → a unified knowledge representation
- policy as integrity constraints (ICs)

complementary policies in SDNs → update orchestrator
- dependency analysis of policy ICs

conflicting policies in inter-domain routing
- overlooked conflicts within an AS
disparate representations buried in the network

complementary policies in SDNs
- jointly satisfiable but not independent

conflicting policies in inter-domain routing
- overlooked conflicts within an AS

a unified knowledge representation
- policy as integrity constraints (ICs)

update orchestrator
- dependency analysis of policy ICs

policy negotiator
- deriving and merging impacts of policy ICs
a unifying representation

network state — factual data — as relations

example schema

% intradomain tables
tp(sid,nid) % topology
rm(fid,sid,nid) % end-to-end reachability (matrix)
cf(fid,sid,nid) % configuration (forwarding table)
path(pv,cost,iid,eid) % internal path

% interdomain tables
aspath(did,rid,apv) % AS level path
a unifying representation

policies — semantic data — as integrity constraints (ICs)

- **denial form:** \( \left\lnot b_1, b_2, \ldots, b_n \right\lnot \) (\( \bot \leftarrow b_1 \land b_2 \land \ldots \land b_n \).)

- **meaning:** \( b_1, b_2, \ldots, b_n \) cannot be simultaneously true.

- **example**

```prolog
% routing policy
IC_1 :- \lnot rm(F,S,D), cf(F,X,Y).
IC_2 :- rm(F,S,D), \lnot cf(F,X,Y).

% security policy
IC_3 :- rm(F,S,D), blacklist(S,D).
```
relating network state & policy

- query policy \((x)\) violation
- network state
- update
- compute \(\Delta_x\)
- policy violation \(v_x\)
manage complementary policies in SDNs
- multiple disjoint policies oversee a single task
arrange network updates into a semantic layering

update orchestrator

x depends on y
semantic dependency

policy x depends on y if

- x update can violate y policy and trigger y action
- but y can not affect x
dependency analysis by satisfiability reasoning

check whether $\Delta_x$ can alter the result of $\nu_y$  
check whether $\Delta_y$ cannot alter $\nu_x$,  

\[
\begin{align*}
\Delta_x & \quad \Delta_y \\
\nu_x & \quad \neg \nu_y \\
x & \quad y
\end{align*}
\]
dependency analysis by satisfiability reasoning

\[(\Delta_x \text{ condition}) \land (v_y \text{ condition}) \text{ is SAT}\]

\[(\Delta_y \text{ condition}) \land (v_x \text{ condition}) \text{ is UNSAT}\]
policy negotiator

manage conflicting policies within an AS

- under the influences of multiple neighbors unaware of each other
- derive and merge policy impacts
derive and merge policy impacts

an AS under the influences of $p_1$ and $p_2$
derive and merge policy impacts

an AS under the influences of \( p_1 \) and \( p_2 \)
derive and merge policy impacts

an AS under the influences of \( p_1 \) and \( p_2 \)

- \( P_1 \) and \( P_2 \) are independent
  - \( P_1, P_2 \)

- \( P_1 \) is stronger?
  - \( P_1 \)

- \( P_1 \) interacts with \( P_2 \), \( P_1 \) is more important
  - \( P_1, P_2|P_1 \)
residue method

residue — syntactic fragment that anticipates impact, computed by partial subsumption

- Policy P₁
- Policy P₂

- Policy P₁ interacts with P₂, P₁ is more important

- P₁ is stronger?
- P₁ is stronger?
- P₁, P₂

- Residue is trivial (no resolution possible)
residue method

residue — syntactic fragment that anticipates impact, computed by partial subsumption

- Policy $P_1$ interacts with $P_2$, $P_1$ is more important
- “null” residue
- Residue is trivial (no resolution possible)
residue method

residue — syntactic fragment that anticipates impact, computed by partial subsumption

- Policy $P_1$
- Policy $P_2$

For non-trivial residue $r$, the residue method results in $P_1, P_2 \{r\}$.
moving forward

expressiveness of the IC representation
- facilitating template, translating tool

cyclic dependency in SDN
- break cycles

policies are private in interdomain
- obfuscate policies
disparate representations buried in the network

complementary policies in SDNs
- jointly satisfiable but not independent

conflicting policies in inter-domain routing
- overlooked conflicts within an AS

a unified knowledge representation
- policy as integrity constraints (ICs)

update orchestrator
- dependency analysis of policy ICs

policy negotiator
- deriving and merging impacts of policy ICs

recap
thank you

part of “Ravel database-defined networking” ravel-net.org
backup
residue computation by example

%%% shortest path
IC_{sp} : - \texttt{ro(x,y,z)}, \texttt{ri(x,y2,z2)}, \texttt{l(z)>l(z2)}.

%%% explicit path policy
IC_{ep}: \texttt{z=a :- ro(x,y,z), x=d}.

a non-trivial residue prescribes the impact of IC_{ep} — additional conditions that must be taken into account for IC_{sp}.

IC_{sp} affects IC_{ep}, as anticipated by the residue.