Dynamic Gossip in NetKAT with Simulated Switch States

Malvin Gattinger (University of Groningen)
Jana Wagemaker (CWI Amsterdam & University College London)

RADICAL workshop, 26 August 2019, CWI Amsterdam

Dynamic Gossip

NetKAT

NetKAT with Simulated Switch States

Dynamic Gossip in NetKAT

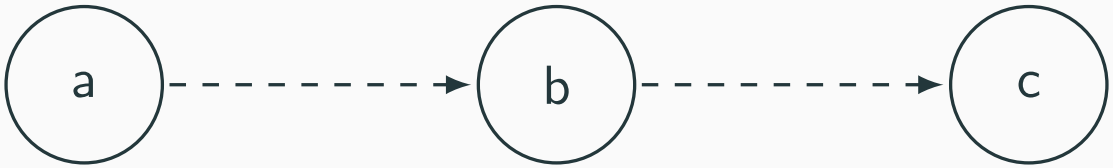
Dynamic Gossip

Dynamic Gossip

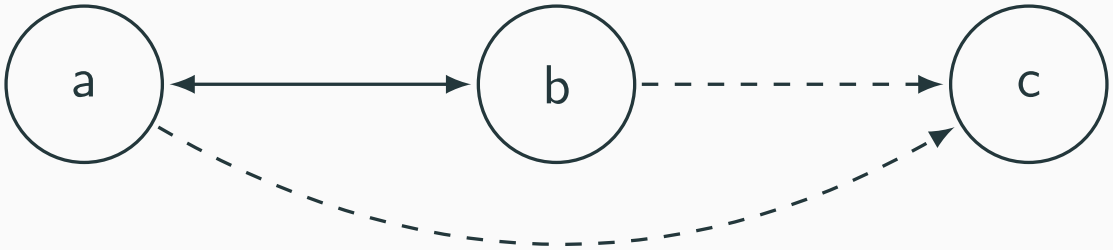
- The *telephone problem*: Given n agents who each have a secret, how many phone calls are needed until everyone knows everything?
- *Dynamic* gossip: also exchange phone numbers ([vDvEP+18])
- Gossip protocol: who is allowed to call whom?
- Goals:
  - avoid redundant calls
  - ensure termination

Dynamic Gossip: Example

Suppose we have this



. . . and now *a* calls *b*. Then we get:



where *a* also learned the number of *c*.

Dynamic Gossip: Research Questions

- For which gossip graphs is a given protocol successful? [vDvEP$^+$18]
- What is the expected execution length of a protocol? [vDKS17]
- How can protocols be improved? [vDGKP19]

Why analyse (Dynamic) Gossip in NetKAT?

- There was no sound and complete logic that captured dynamic gossip in a satisfactory way. [vDGH$^+$16, vDGKP19].

- NetKAT (Anderson et al. 2014 [AFG$^+$14] and Foster et al. 2015 [FKM$^+$15]) has a sound and complete axiomatization to describe packet-processing behaviour of networks.

NetKAT

NetKAT

Definition (Syntax)

NetKAT uses *predicates* a and *policies* p:

$$\begin{aligned} a &::= 1 \mid 0 \mid f = n \mid a + b \mid a \cdot b \mid \neg a \\ p &::= a \mid f \leftarrow n \mid p + q \mid p \cdot q \mid p^* \end{aligned}$$

where f ranges over a finite set of fields including a switch field sw and a port field pt and n is a value from a finite domain.

NetKAT

Definition (Syntax)

NetKAT uses *predicates* a and *policies* p:

$$\begin{aligned} a &::= 1 \mid 0 \mid f = n \mid a + b \mid a \cdot b \mid \neg a \\ p &::= a \mid f \leftarrow n \mid p + q \mid p \cdot q \mid p^* \end{aligned}$$

where f ranges over a finite set of fields including a switch field sw and a port field pt and n is a value from a finite domain.

Example

The NetKAT expression

$$\mathrm{sw} = A \cdot \mathrm{pt} = 4 \cdot \mathrm{dst} \leftarrow H \cdot \mathrm{pt} \leftarrow 7$$

can be read as "if the packet is located at port 4 of switch A, then set its destination to H and move the packet to port 7".

Semantics of NetKAT

A *packet* is a tuple, e.g. $\mathsf{pk} = (f_1 = 23, f_2 = 42)$.

The interpretation of an expression maps packets to sets of packets:

$$
\begin{aligned}
[\![1]\!](\mathsf{pk}) &:= \{\mathsf{pk}\} \\
[\![0]\!](\mathsf{pk}) &:= \varnothing \\
[\![\neg a]\!](\mathsf{pk}) &:= \{\mathsf{pk}\} \setminus [\![a]\!](\mathsf{pk}) \\
[\![p + q]\!](\mathsf{pk}) &:= [\![p]\!]\mathsf{pk} \cup [\![q]\!]\mathsf{pk} \\
[\![p \cdot q]\!](\mathsf{pk}) &:= \bigcup \{ [\![q]\!](\mathsf{pk}') \mid \mathsf{pk}' \in [\![p]\!](\mathsf{pk}) \} \\
[\![p^*]\!](\mathsf{pk}) &:= \textstyle\bigcup_{i \in \mathbb{N}} [\![p]\!]^i(\mathsf{pk}) \\
[\![f = n]\!](\mathsf{pk}) &:= \begin{cases} \{\mathsf{pk}\} & \text{if } \mathsf{pk}.f = n \\ \varnothing & \text{otherwise} \end{cases} \\
[\![f \leftarrow n]\!](\mathsf{pk}) &:= \{\mathsf{pk}[f := n]\}
\end{aligned}
$$

Example

Example

Consider policy

$$(\mathsf{sw} \leftarrow B \ \cdot \ f \leftarrow 23) + (\mathsf{sw} \leftarrow C \ \cdot \ f \leftarrow 42)$$

Applied to $\mathsf{pk} = (sw = A, f = 17)$, this policy outputs

$$\{ \ (\mathsf{sw} = B, f = 23), \ (\mathsf{sw} = C, f = 42) \ \}.$$

NetKAT with Simulated Switch States

Can we encode Dynamic Gossip in NetKAT?

Dynamic Gossip is

- non-local: a call ab can be followed by cd.
- stateful: a call ab influences a much later call ac.
- dynamic: we change who can call whom at runtime.

All three seem impossible in NetKAT at first sight.

Can we encode Dynamic Gossip in NetKAT?

Dynamic Gossip is

- non-local: a call *ab* can be followed by *cd*.
- stateful: a call *ab* influences a much later call *ac*.
- dynamic: we change who can call whom at runtime.

All three seem impossible in NetKAT at first sight.

Solution:

- simulate switch states in packets
- use a *total* NetKAT network

$\Rightarrow$ A packet may only "interact with itself" when passing by the same switch again, but not with other packets!

This is sufficient for dynamic gossip, which is still *sequential*.
No actual concurrency is needed.

NetKAT with Simulated Switch States

Idea: for each agent/switch i, add a field state(i) to all packets.

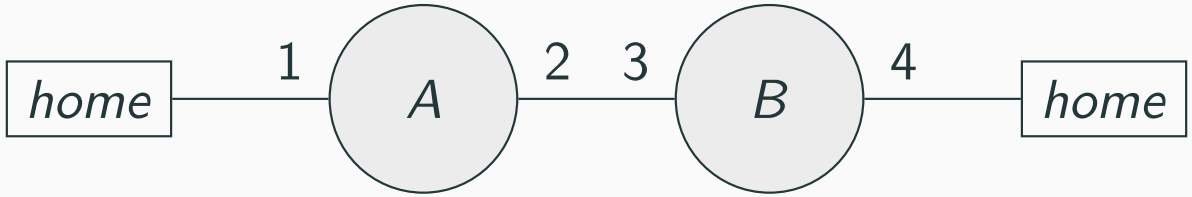
The same axiomatization from [AFG$^+$14] is still sound and complete for NetKAT with such simulated states.

A policy is *topology respecting* iff it only modifies the state of the switch where it currently is. Formally: any state(i) $\leftarrow \ldots$ is prefixed by sw $= i$.

Dynamic Gossip in NetKAT

A NetKAT model for gossip

For two gossiping agents we use the following NetKAT model:



The initial gossip graph is translated to NetKAT via the initial packet.

Instead of a single state(i) we use multiple fields: Sij, Nij.

We can also add fields to keep track of the call history.

Dynamic Gossip in NetKAT

Definition

LNS: agent a may call b iff a does not know b's secret.

Theorem

The LNS protocol is weakly successful on gossip graph G if and only if the following NetKAT equivalence holds:

$$pol_G \cdot pol_{\mathrm{LNS}} \cdot pol_{\mathrm{success}} \not\equiv 0$$

We can also encode *strong success* and the following theorem.

Theorem 20 in [vDvEP+18]

The LNS protocol is strongly successful on an initial gossip graph G if and only if G is a sun graph.

Summary

- NetKAT can simulate switch states "for free".
- Dynamic Gossip decision problems reduce to NetKAT equivalences.
- Implementation available! — Help?
  `https://github.com/janawagemaker/GossipKATS`
- Parallel Gossip?

Full paper: [GW18] available at `https://malv.in`

References

C. J. Anderson, Nate Foster, Arjun Guha, Jean-Baptiste Jeannin, Dexter Kozen, Cole Schlesinger, and David Walker.
Netkat: Semantic foundations for networks.
SIGPLAN Not., 49(1):113–126, January 2014.

N. Foster, Dexter Kozen, Matthew Milano, Alexandra Silva, and Laure Thompson.
A coalgebraic decision procedure for netkat.
SIGPLAN Not., 50(1):343–355, January 2015.

Malvin Gattinger and Jana Wagemaker.
Towards an analysis of dynamic gossip in netkat.
In Stef Joosten Jules Desharnais, Walter Guttmann, editor, *RAMiCS 2018*, 2018.

Hans van Ditmarsch, Davide Grossi, Andreas Herzig, Wiebe van der Hoek, and Louwe B. Kuijer.
Parameters for epistemic gossip problems.
In *Proceedings of the Twelfth Conference on Logic and the Foundations of Game and Decision Theory*, 2016.

Hans van Ditmarsch, Malvin Gattinger, Louwe B. Kuijer, and Pere Pardo.
Strengthening gossip protocols using protocol-dependent knowledge.
Journal of Applied Logics - IfCoLog Journal of Logics and their Applications, 6(1), 2019.

Hans van Ditmarsch, Ioannis Kokkinis, and Anders Stockmarr.
Reachability and expectation in gossiping.
In *Proceedings of PRIMA 2017*, 2017.

Hans van Ditmarsch, Jan van Eijck, Pere Pardo, Rahim Ramezanian, and François Schwarzentruber.
Dynamic gossip.
Bulletin of the Iranian Mathematical Society, 9 2018.