Causal Reasoning in SDNs (NetKAT)

Georgiana Caltais, University of Konstanz
Shonan Seminar -“Causal Reasoning in Systems”
24-27 June, 2019

Outline

1. NetKAT - the Language
2. Reasoning & Verification
3. Towards a Framework for Causality

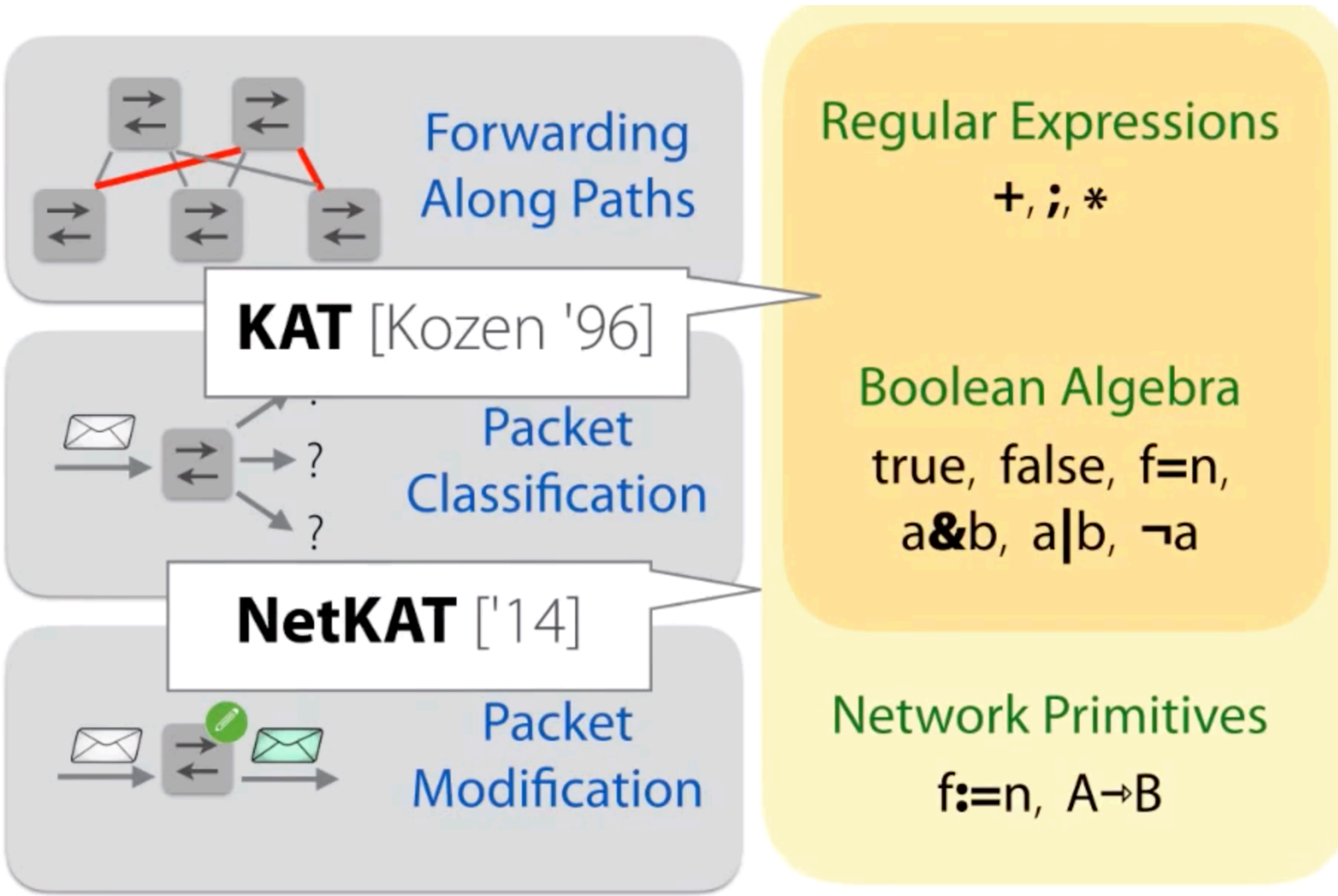
Sources:

“Programming, Modeling & Reasoning about Networks” (online tutorial by S.Smolka)

“NetKAT: Semantic Foundation for Networks” [C.J.Anderson et. al.], POPL’14

“A Fast Complier for NetKAT” [S.Smolka et. al.], ICFP’15

1. NetKAT - the Language



NetKAT Program - Example

$$\mathit{switch} = 6;\ \mathit{port} = 88;\ \mathit{dest} := 10.0.0.1;$$
$$(\mathit{port} := 50 + \mathrm{port} := 51)$$

“For all packets incoming on port 88 of switch 6, set the destination IP address to 10.0.0.1 and multicast the packet out of ports 50 and 51.”

NetKAT Syntax & Semantics

Syntax

Fields	$f ::=$	$f_1 \mid \cdots \mid f_k$	
Packets	$pk ::=$	$\{f_1 = v_1, \cdots, f_k = v_k\}$	
Histories	$h ::=$	$pk{::}\langle\rangle \mid pk{::}h$	
Predicates	$a, b ::=$	1	*Identity*
	$\mid$	0	*Drop*
	$\mid$	$f = n$	*Test*
	$\mid$	$a + b$	*Disjunction*
	$\mid$	$a \cdot b$	*Conjunction*
	$\mid$	$\neg a$	*Negation*
Policies	$p, q ::=$	a	*Filter*
	$\mid$	$f \leftarrow n$	*Modification*
	$\mid$	$p + q$	*Union*
	$\mid$	$p \cdot q$	*Sequential composition*
	$\mid$	p^*	*Kleene star*
	$\mid$	dup	*Duplication*

Semantics

$$[\![p]\!] \in \mathsf{H} \to \mathcal{P}(\mathsf{H})$$

$$[\![1]\!]\ h \triangleq \{h\}$$

$$[\![0]\!]\ h \triangleq \{\}$$

$$[\![f = n]\!]\ (pk{::}h) \triangleq \begin{cases} \{pk{::}h\} & \text{if } pk.f = n \\ \{\} & \text{otherwise} \end{cases}$$

$$[\![\neg a]\!]\ h \triangleq \{h\} \setminus ([\![a]\!]\ h)$$

$$[\![f \leftarrow n]\!]\ (pk{::}h) \triangleq \{pk[f := n]{::}h\}$$

$$[\![p + q]\!]\ h \triangleq [\![p]\!]\ h \cup [\![q]\!]\ h$$

$$[\![p \cdot q]\!]\ h \triangleq ([\![p]\!] \bullet [\![q]\!])\ h$$

$$[\![p^*]\!]\ h \triangleq \textstyle\bigcup_{i \in \mathbb{N}} F^i\ h$$

$$\text{where } F^0\ h \triangleq \{h\} \text{ and } F^{i+1}\ h \triangleq ([\![p]\!] \bullet F^i)\ h$$

$$[\![\mathsf{dup}]\!]\ (pk{::}h) \triangleq \{pk{::}(pk{::}h)\}$$

Encoding Switch Forwarding Tables

Pattern	Action
$\star$	$\mathsf{pt}{\leftarrow}2$

$pol_A \triangleq \mathsf{pt}{\leftarrow}2$

(a) An atomic modification

Pattern	Action
$\mathsf{dst}{=}\mathsf{A}$	*true*
$\star$	*false*

$pol_B \triangleq \mathsf{dst}{=}\mathsf{A}$

(b) An atomic predicate

Pattern	Action
$\mathsf{dst}{=}\mathsf{A}$	$\mathsf{pt}{\leftarrow}2$
$\star$	*false*

$pol_B \cdot pol_A$

(c) Forwarding to a single host

Pattern	Action
$\mathsf{dst}{=}\mathsf{A}$	$\mathsf{pt}{\leftarrow}1$
$\mathsf{dst}{=}\mathsf{B}$	$\mathsf{pt}{\leftarrow}2$
$\star$	*false*

$$pol_D \triangleq \begin{array}{l} \mathsf{dst}{=}\mathsf{A} \cdot \mathsf{pt}{\leftarrow}1 + \\ \mathsf{dst}{=}\mathsf{B} \cdot \mathsf{pt}{\leftarrow}2 \end{array}$$

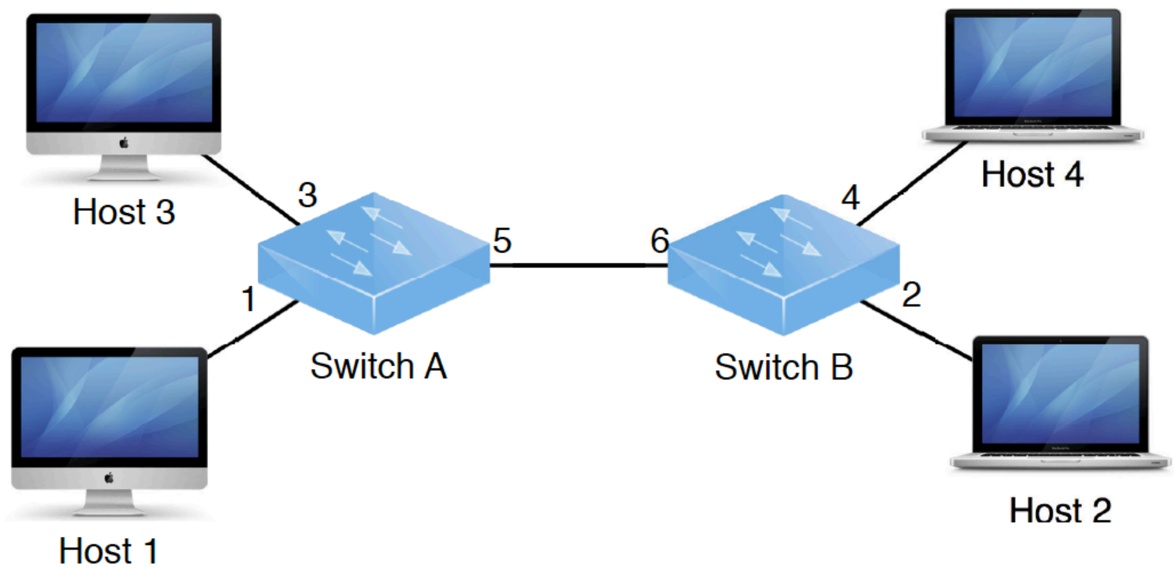
(d) Forwarding traffic to two hosts

Pattern	Action
$\mathsf{dst}{=}\mathsf{A}$	$\mathsf{pt}{\leftarrow}3$
$\mathsf{proto}{=}\mathsf{ssh}$	$\mathsf{pt}{\leftarrow}3$
$\star$	*false*

$$pol_E \triangleq \left(\begin{array}{c} \mathsf{proto}{=}\mathsf{ssh} + \\ \mathsf{dst}{=}\mathsf{A} \end{array}\right) \cdot \mathsf{pt}{\leftarrow}3$$

(e) Monitoring SSH traffic and traffic to host A

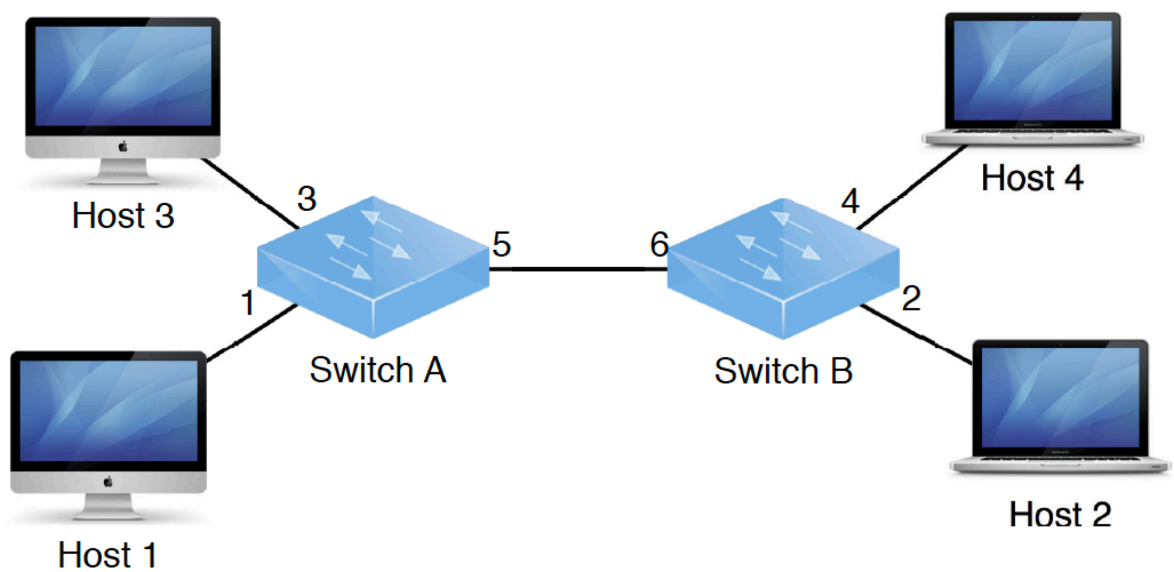
Encoding Network Topologies (I)



topology, e.g.,

$$
\begin{aligned}
t \equiv\ & sw = A \cdot pt = 5 \cdot sw \leftarrow B \cdot pt \leftarrow 6\ + \\
& sw = B \cdot pt = 6 \cdot sw \leftarrow A \cdot pt \leftarrow 5\ + \\
& sw = A \cdot (pt = 1 + pt = 3)\ + \\
& sw = b \cdot (pt = 2 + pt = 4)
\end{aligned}
$$

Encoding Network Topologies (II)

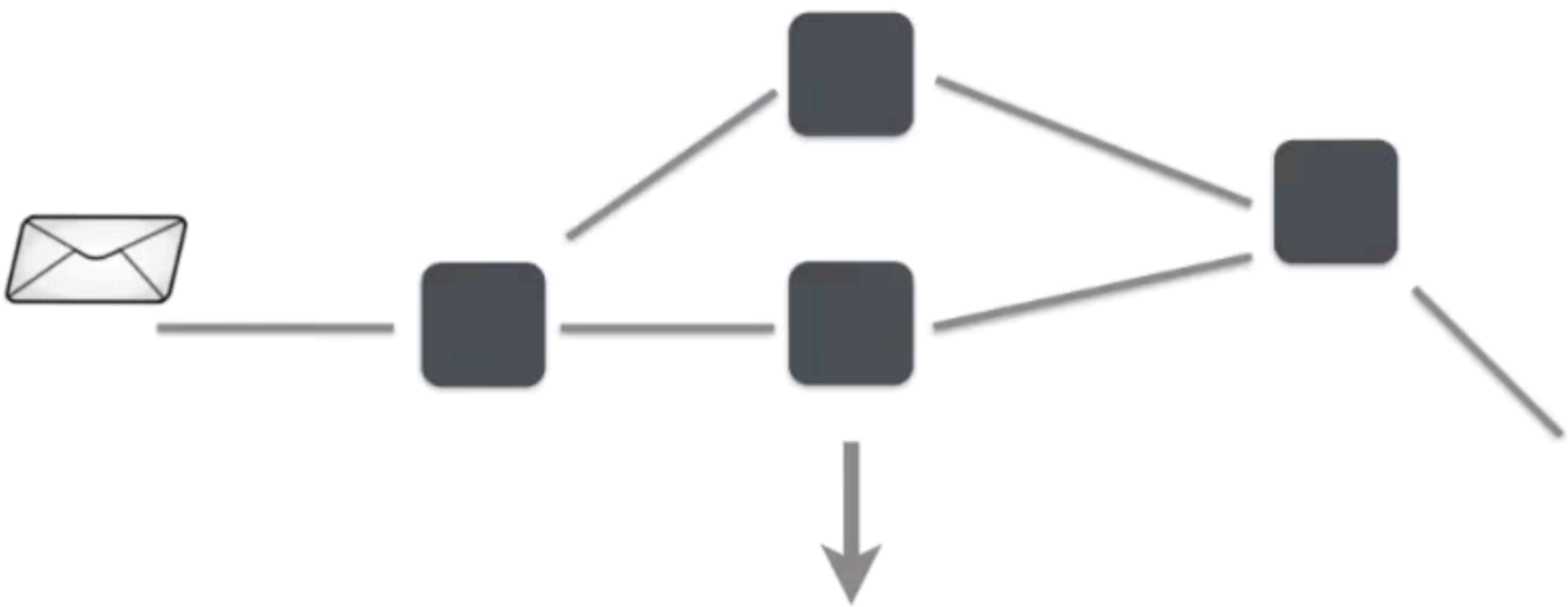


topology (assuming all ports different):

$$t = pt=5 \cdot pt \leftarrow 6 + pt=6 \cdot pt \leftarrow 5 + pt=1 + pt=2 + pt=3 + pt=4$$

Encoding Networks

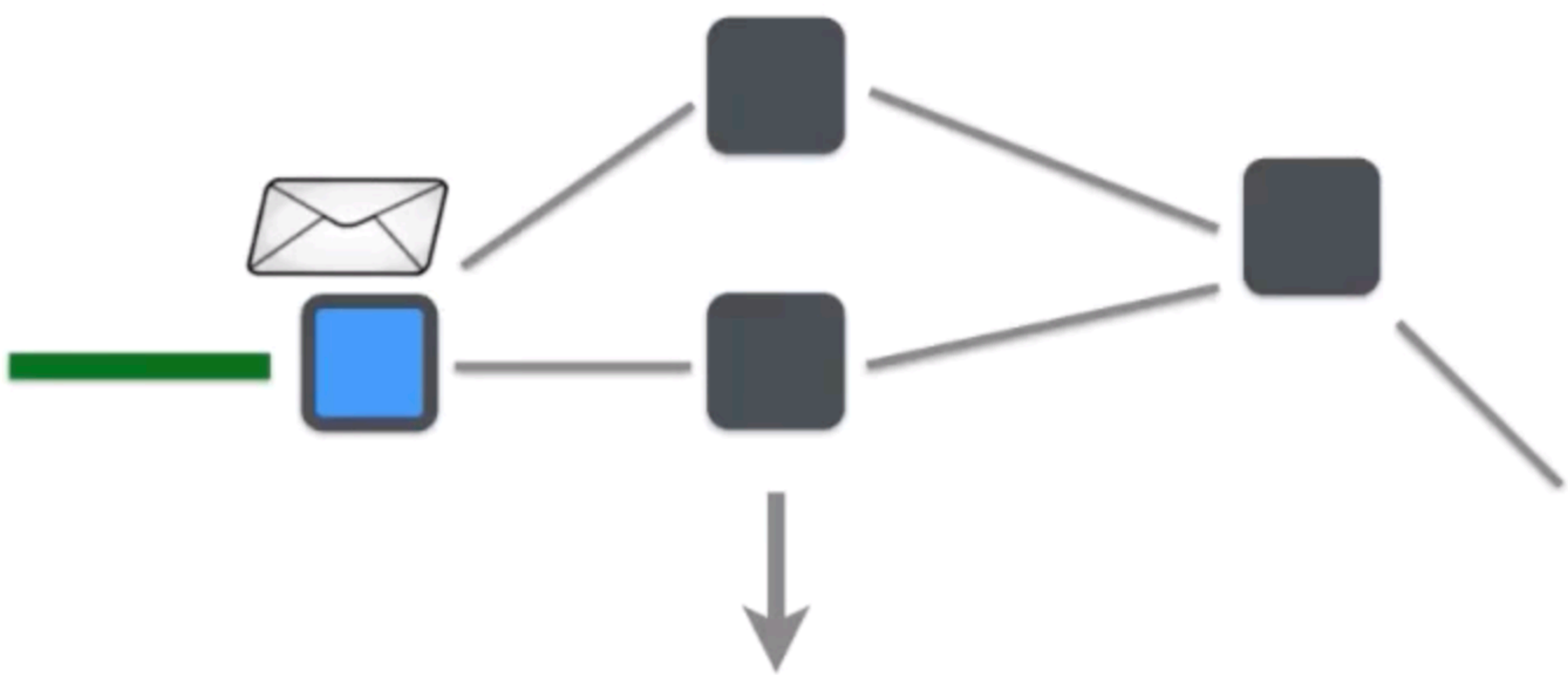
A network can be encoded in NetKAT by interleaving steps of processing by switches and topology



(topology; switch)*

Encoding Networks

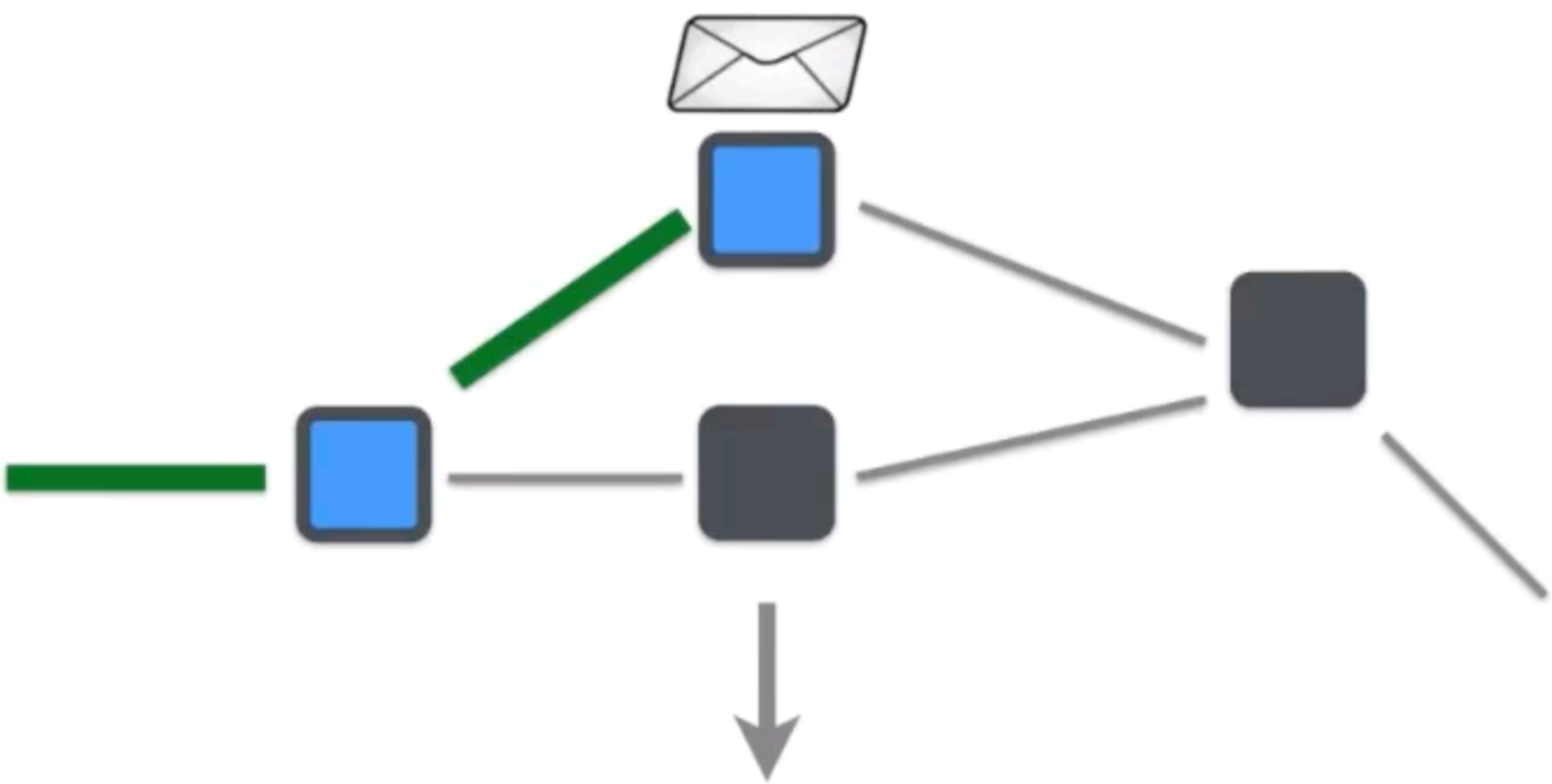
A network can be encoded in NetKAT by interleaving steps of processing by switches and topology



(topology; switch)*

Encoding Networks

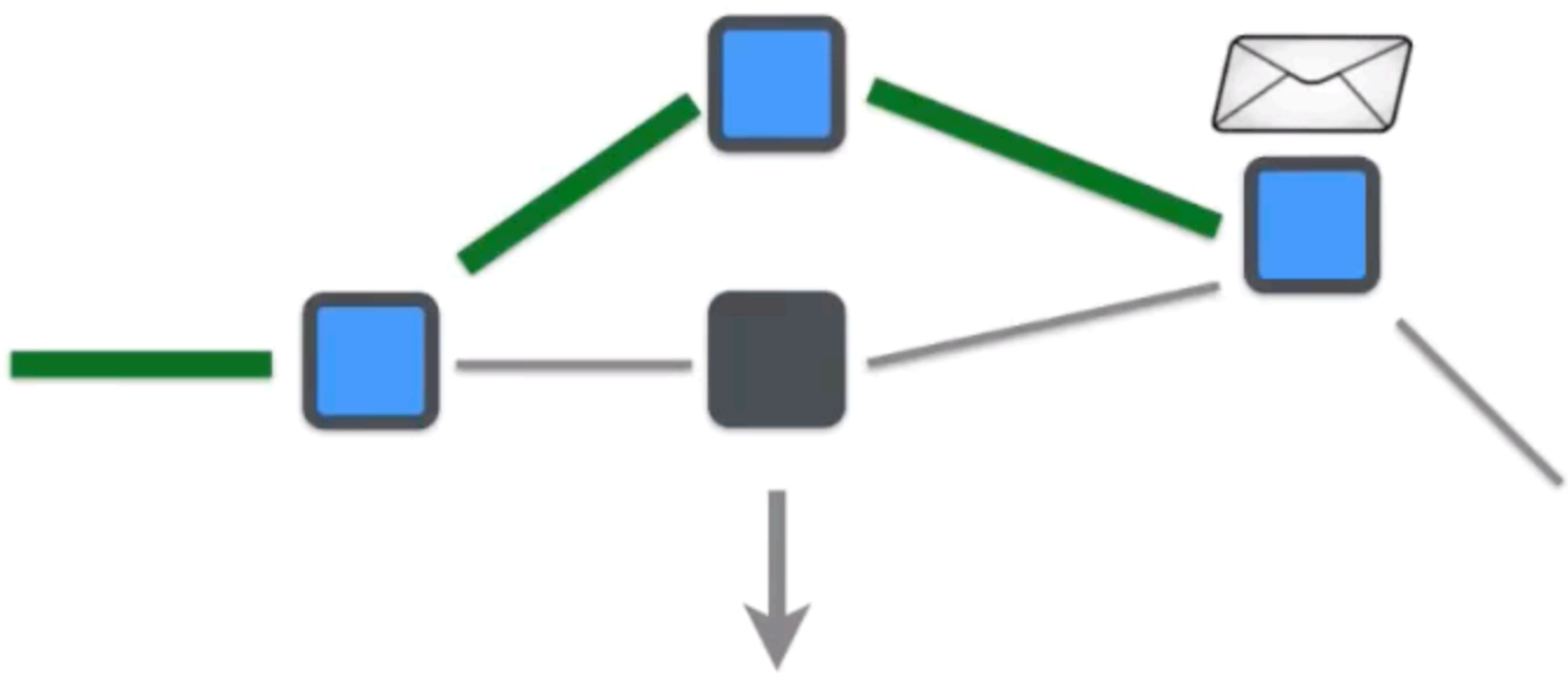
A network can be encoded in NetKAT by interleaving steps of processing by switches and topology



(topology; switch)*

Encoding Networks

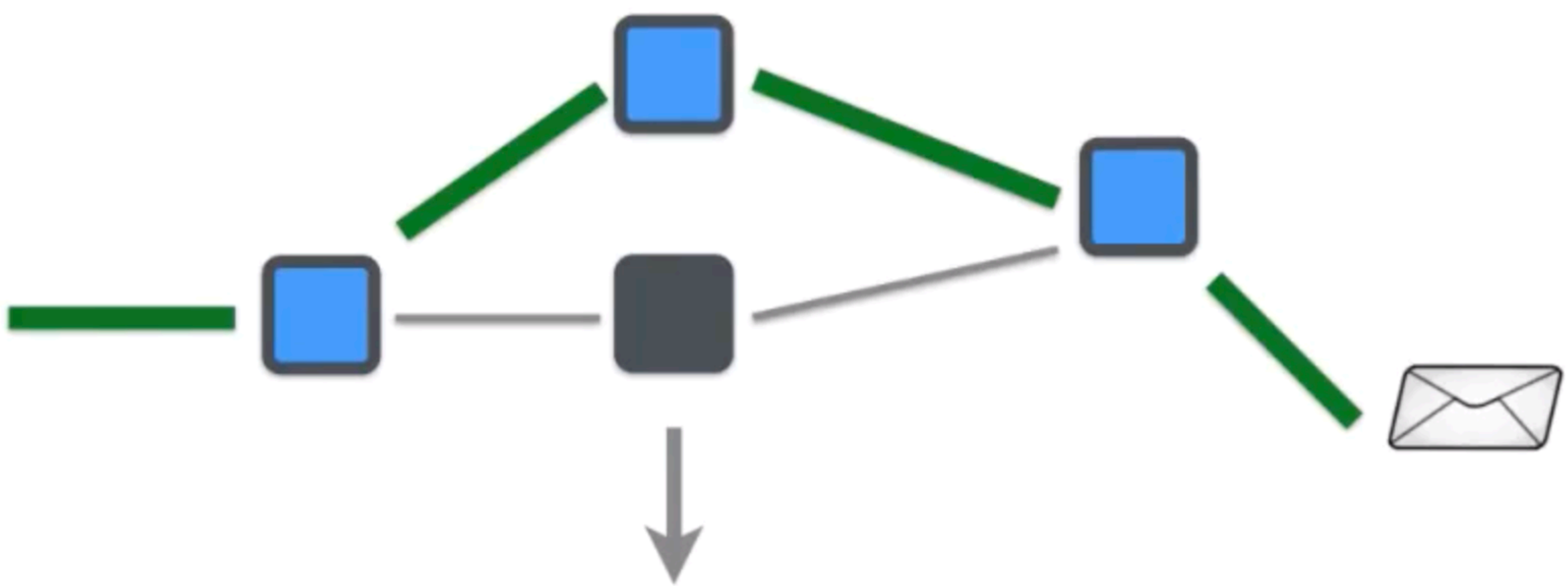
A network can be encoded in NetKAT by interleaving steps of processing by switches and topology



(topology; switch)*

Encoding Networks

A network can be encoded in NetKAT by interleaving steps of processing by switches and topology



(topology; switch)*

2. Reasoning & Verification

Network Verification

- Sound & Complete Axiomatisation [C.J.Anderson et. al.]

Kleene Algebra Axioms

$p + (q + r) \equiv (p + q) + r$	KA-PLUS-ASSOC
$p + q \equiv q + p$	KA-PLUS-COMM
$p + 0 \equiv p$	KA-PLUS-ZERO
$p + p \equiv p$	KA-PLUS-IDEM
$p \cdot (q \cdot r) \equiv (p \cdot q) \cdot r$	KA-SEQ-ASSOC
$1 \cdot p \equiv p$	KA-ONE-SEQ
$p \cdot 1 \equiv p$	KA-SEQ-ONE
$p \cdot (q + r) \equiv p \cdot q + p \cdot r$	KA-SEQ-DIST-L
$(p + q) \cdot r \equiv p \cdot r + q \cdot r$	KA-SEQ-DIST-R
$0 \cdot p \equiv 0$	KA-ZERO-SEQ
$p \cdot 0 \equiv 0$	KA-SEQ-ZERO
$1 + p \cdot p^* \equiv p^*$	KA-UNROLL-L
$q + p \cdot r \leq r \Rightarrow p^* \cdot q \leq r$	KA-LFP-L
$1 + p^* \cdot p \equiv p^*$	KA-UNROLL-R
$p + q \cdot r \leq q \Rightarrow p \cdot r^* \leq q$	KA-LFP-R

Network Verification

- Sound & Complete Axiomatisation [C.J.Anderson et. al.]

Additional Boolean Algebra Axioms

$a + (b \cdot c) \equiv (a + b) \cdot (a + c)$	BA-PLUS-DIST
$a + 1 \equiv 1$	BA-PLUS-ONE
$a + \neg a \equiv 1$	BA-EXCL-MID
$a \cdot b \equiv b \cdot a$	BA-SEQ-COMM
$a \cdot \neg a \equiv 0$	BA-CONTRA
$a \cdot a \equiv a$	BA-SEQ-IDEM

Network Verification

- Sound & Complete Axiomatisation [C.J.Anderson et. al.]

Packet Algebra Axioms

$f \leftarrow n \cdot f' \leftarrow n' \equiv f' \leftarrow n' \cdot f \leftarrow n, \text{ if } f \neq f'$	PA-MOD-MOD-COMM
$f \leftarrow n \cdot f' = n' \equiv f' = n' \cdot f \leftarrow n, \text{ if } f \neq f'$	PA-MOD-FILTER-COMM
$\mathsf{dup} \cdot f = n \equiv f = n \cdot \mathsf{dup}$	PA-DUP-FILTER-COMM
$f \leftarrow n \cdot f = n \equiv f \leftarrow n$	PA-MOD-FILTER
$f = n \cdot f \leftarrow n \equiv f = n$	PA-FILTER-MOD
$f \leftarrow n \cdot f \leftarrow n' \equiv f \leftarrow n'$	PA-MOD-MOD
$f = n \cdot f = n' \equiv 0, \text{ if } n \neq n'$	PA-CONTRA
$\sum_i f = i \equiv 1$	PA-MATCH-ALL

Network Verification

- Sound & Complete Axiomatisation [C.J.Anderson et. al.]

$$[[p]] = [[q]] \text{ iff } \vdash p = q$$

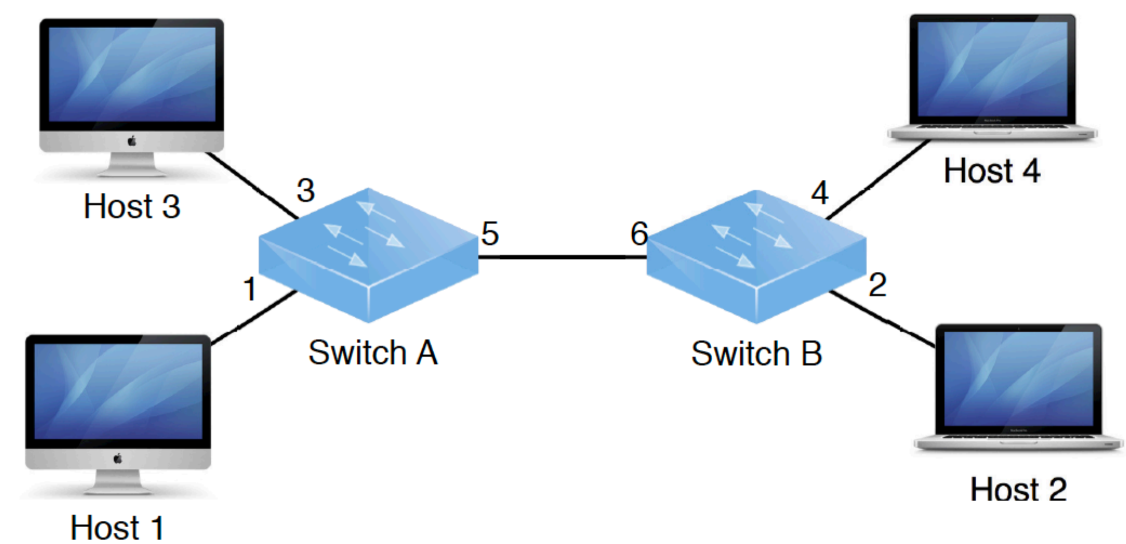
- E.g., Reachability:

"Does the network forward from ingress (in) to egress (out)"?

NO iff $\vdash$ in . (switch.topology)* . out = 0

YES iff $\vdash$ in . (switch.topology)* . out =/= 0

Reasoning About Correctness of NetKAT Programs

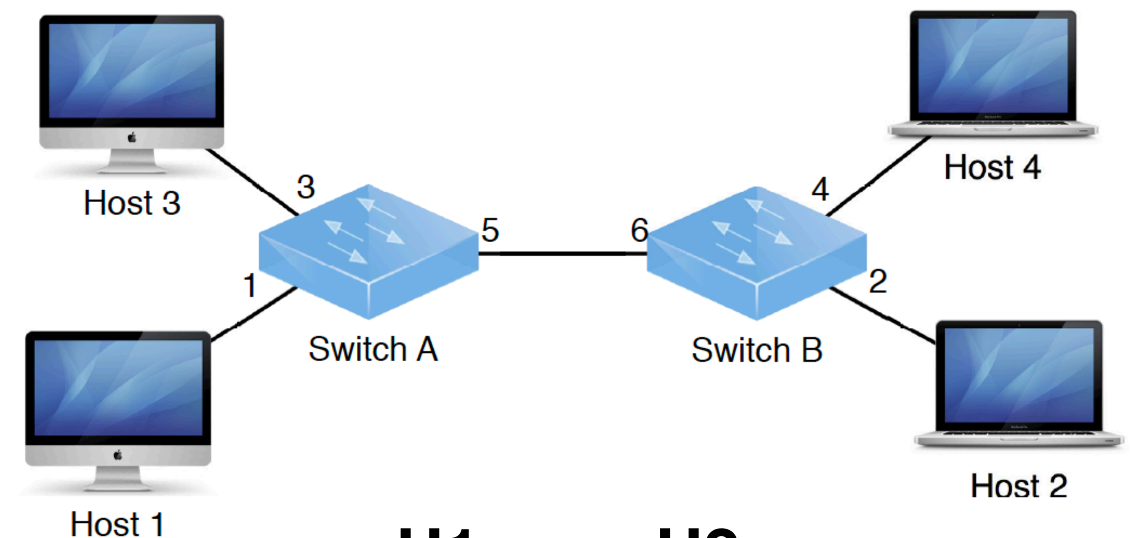


- Programmer 1 has to implement a switch policy s.t.:

"H1 can only forward to H2"

- Correctness:
  - H1 can forward to H2 (H1 —>> H2)
  - H1 cannot forward to H3 or H4 (H1 -/->> H3,4)

Reasoning About Correctness of NetKAT Programs



H1 —>> H2

H1 -/->> H3,4

Proven correct based on the axioms!

"H1 can only forward to H2"

- Policy p1 : (pt = 1 . pt <— 5) + (pt = 6 . pt <— 2)

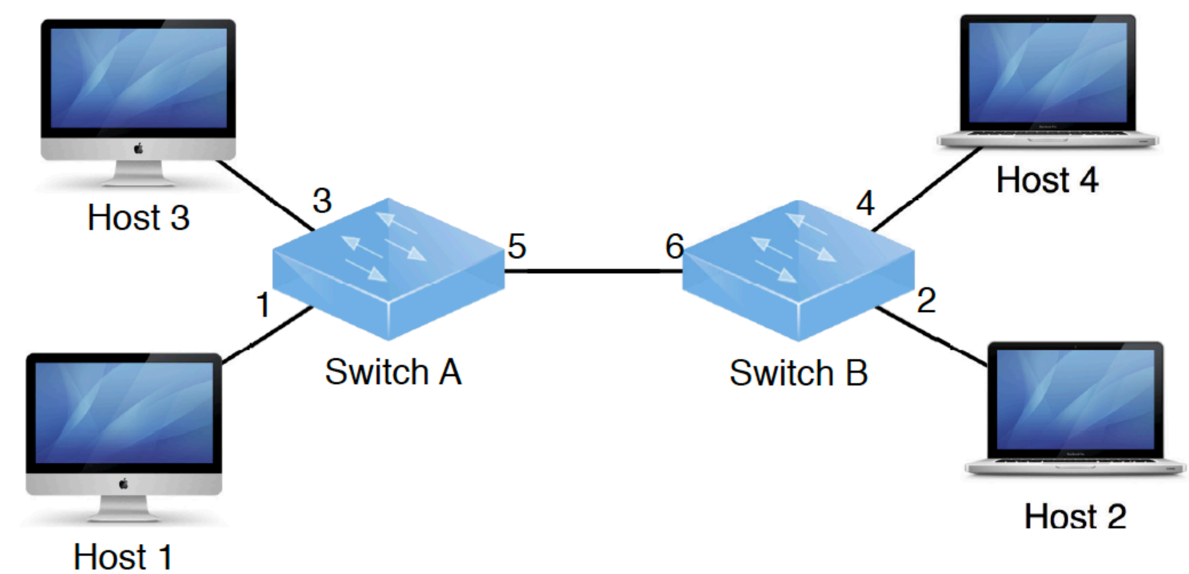
H1 can forward to H2 (H1 —>> H2)

- |— (pt = 1) . (p1 . t)* . (pt = 2) =/= 0

H1 cannot forward to H3 or H4 (H1 -/->> H3,4)

- |— (pt = 1) . (p1 . t)* . (pt = 3 + pt = 4) = 0

Reasoning About Correctness of NetKAT Programs

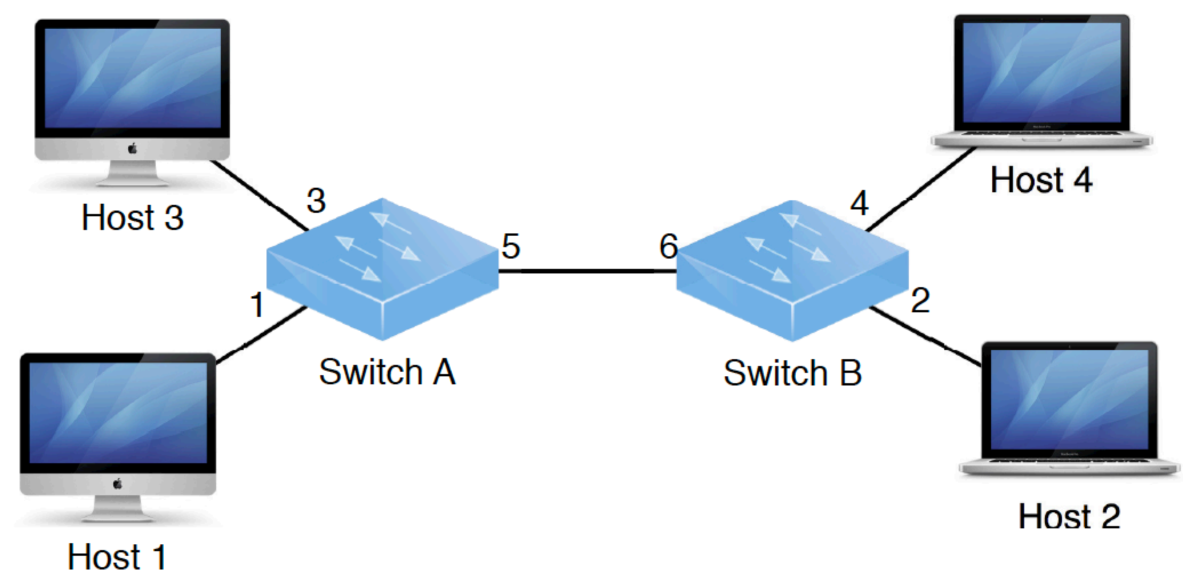


- Programmer 2 has to implement a switch policy s.t.:

"H3 can only forward to H4"

- Correctness: … shown in a similar fashion…
    - H3 can forward to H4 (H3 —>> H4)
    - H3 cannot forward to H1 or H2 (H3 -/->> H1,2)

Reasoning About Correctness of NetKAT Programs

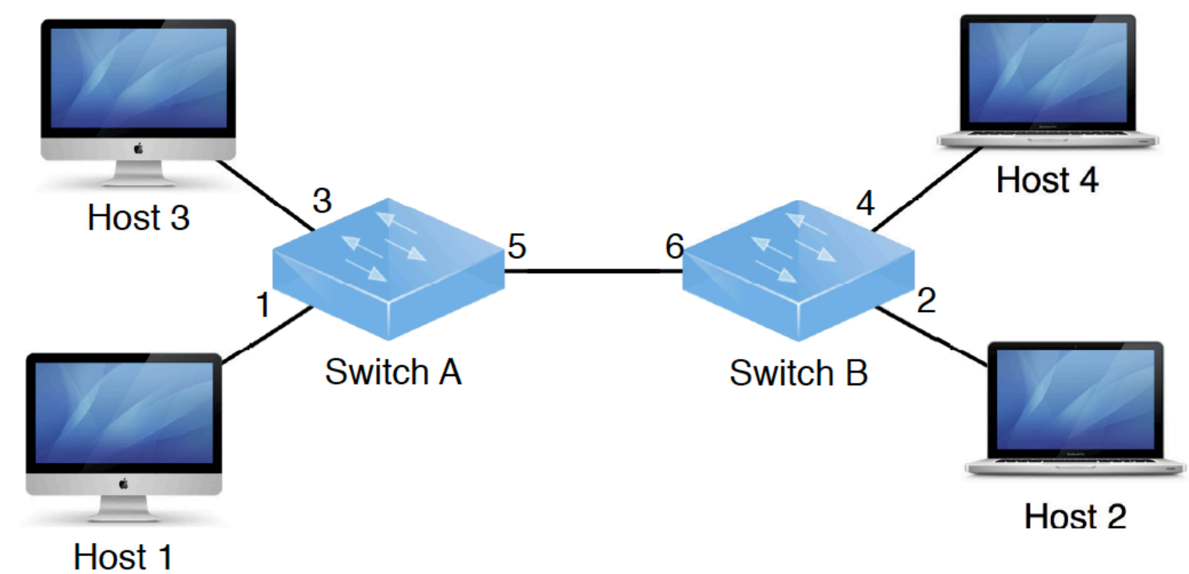


- Programmer 1: “H1 can only forward to H2” / switch policy p1
- Programmer 2: “H3 can only forward to H4” / switch policy p2
- Assume Programmer 3 implements p as the union of the two correct policies p1 and p2

p = p1 + p2

- Network becomes (p . t)* = ((p1 + p2) . t)*
- Does H1 -/->> H3,4 still hold?

Reasoning About Correctness of NetKAT Programs



H1 -/->> H3,4 holds iff

$\vdash pt = 1 . ((p1 + p2) . t)^* . (pt = 3 + pt = 4) = 0$ iff

(acc. to NetKAT axioms)

$\vdash pt = 1 . pt \leftarrow 4 + P = 0$

What is the cause?

3. Towards a Framework for Causality

What Is the Cause?
- Obvious Challenges -

H1 -/->> H3,4 holds iff

$\vdash$ pt = 1 . ((p1 + p2) . t)* . (pt = 3 + pt = 4) = 0 iff

(acc. to NetKAT axioms)

$\vdash$ **pt = 1 . pt <— 4 + P = 0**

provides too little information

contains *

What Is the Cause?
- Obvious Challenges -

H1 -/->> H3,4 holds iff

|— pt = 1 . ((p1 + p2) . t)* . (pt = 3 + pt = 4) = 0 iff

(acc. to NetKAT axioms)

|— **pt = 1 . pt <— 4 + P = 0**

provides too little information

“Star Elimination”
in [C.J.Anderson et. al]
assumption: no dup, no sw <—
uses all axioms to build the Normal Form of P, NF (P)
|— P ~ NF(P)
… provides too little information as well…

What Is the Cause?
- Possible Solution -

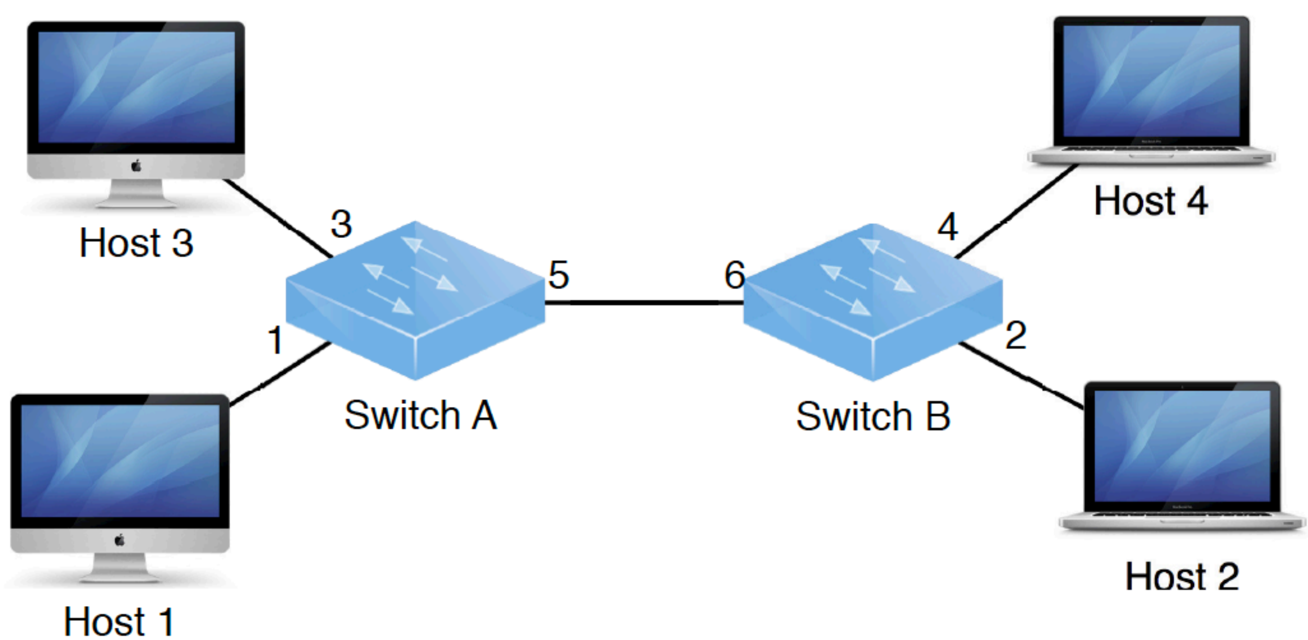
|— pt = 1 . ((p1 + p2) . t)* . (pt = 3 + pt = 4) = 0 iff (… axioms)

|— **pt = 1 . pt <— 1 . pt <— 5 . pt <— 6 . pt <— 4 + P_{sf} = 0**

Inhibit some of the axioms, e.g.:
f <— n . f <— n' = f <— n' [PA-MOD-MOD]

"Approximate" *
(p.t)* = (1 + p.t)^n
for some n…

and remove *-unfolding axioms



* "Approximation"

A - the set of all tests $f = n$

Π - the set of all assignments $f \leftarrow n$

Assume $p = \sum_{i=1}^{n} \alpha_i \cdot \pi_i$, with $\alpha_i \in A^*$, $\pi_i \in \Pi^*$

t - a topology

Observation: p entails a loop-free path from in to out crossing at most n switches

$\Downarrow$ new axiom

$$(p \cdot t)^* \equiv (1 + p \cdot t)^n$$

Some Terminology…

Let $\vdash_{*}^{=}$ be the entailment relation over the new axiomatization

p in Tree-Form iff $p = \sum_{i=1}^{n} p_i \,;\, p_i \in (A^* . \pi^*)^*$

notation: $p_i \in p$, $TF(p)$

support $(q) = \{ \bar{q} \in q \mid \cancel{\exists}\, \tilde{q} \in q .\ \tilde{q} \subset \bar{q} \}$, q in Tree Form

Consider the SAFETY property:

$$\boxed{\vdash \mathrm{in}.(p.t)^*.\mathrm{out} \equiv 0} \quad (*)$$

The CAUSE w.r.t. the violation of $(*)$ is

$$c = \sum_{\bar{q} \in \mathrm{support}(q')} \bar{q} \quad \text{where}$$

$$\vdash_* \mathrm{in}.(p.t)^*.\mathrm{out} \equiv q, \quad q \not\equiv 0$$

$$q' = TF(q)$$

Conjecture: $\vdash c \equiv NF\left(\mathrm{in}.(p.t)^*.\mathrm{out}\right)$

Questions?

- Current & Future Work:
  - Trace back the cause into the original code
  - How does the counterfactual look like?
  - Handling other interesting network properties
    - E.g., waypointing…
  - Responsibility, blame