ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Network Verification and Monitoring

Khanh Nam Chu

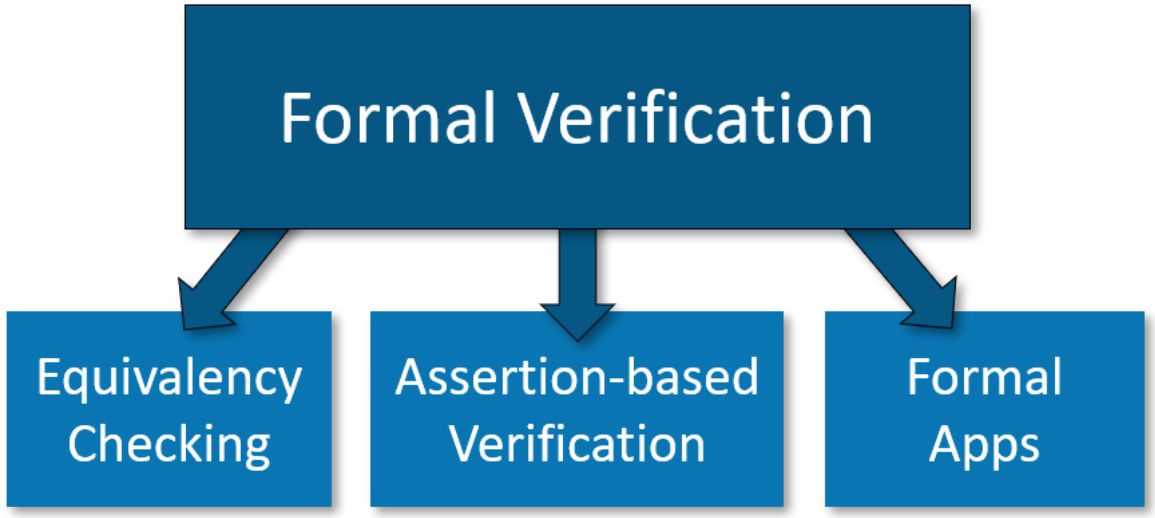
ONE LOVE. ONE FUTURE.

Contents

I. Network Verification

1. Formal Verification

Proving or disproving the correctness of a (software or hardware) system with respect to a certain formal specification or property using formal methods of mathematics.



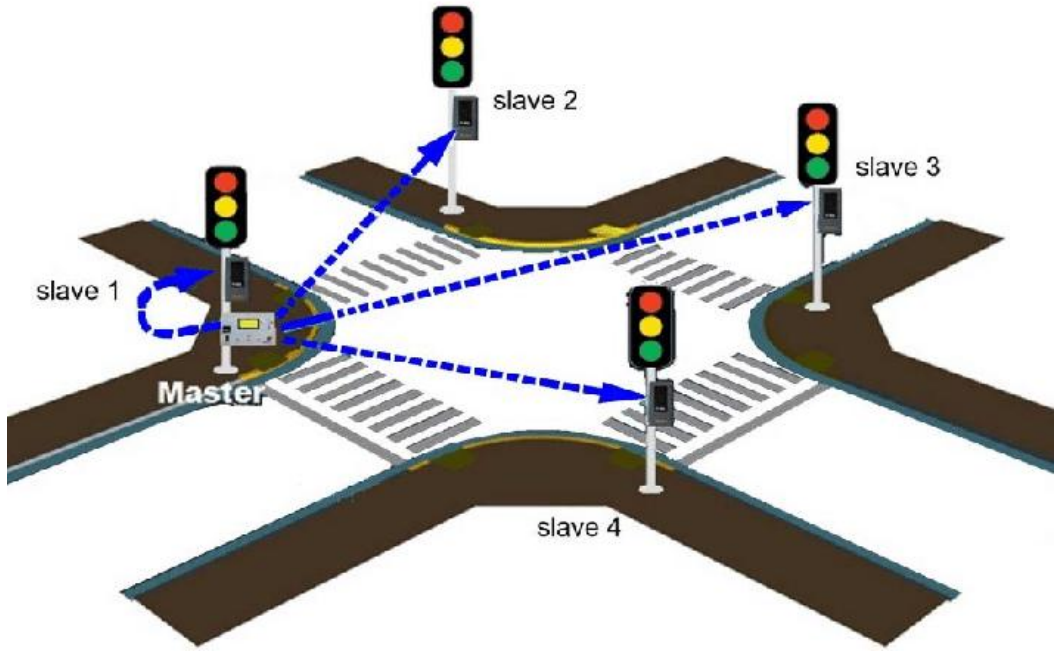
I. Network Verification

1. Formal Verification

Proving or disproving the correctness of a (software or hardware) system with respect to a certain formal specification or property using formal methods of mathematics.

For example, a traffic light controller **system** with **formal specification or property:** safety properties is nothing bad happens, or liveness property is something good happens.

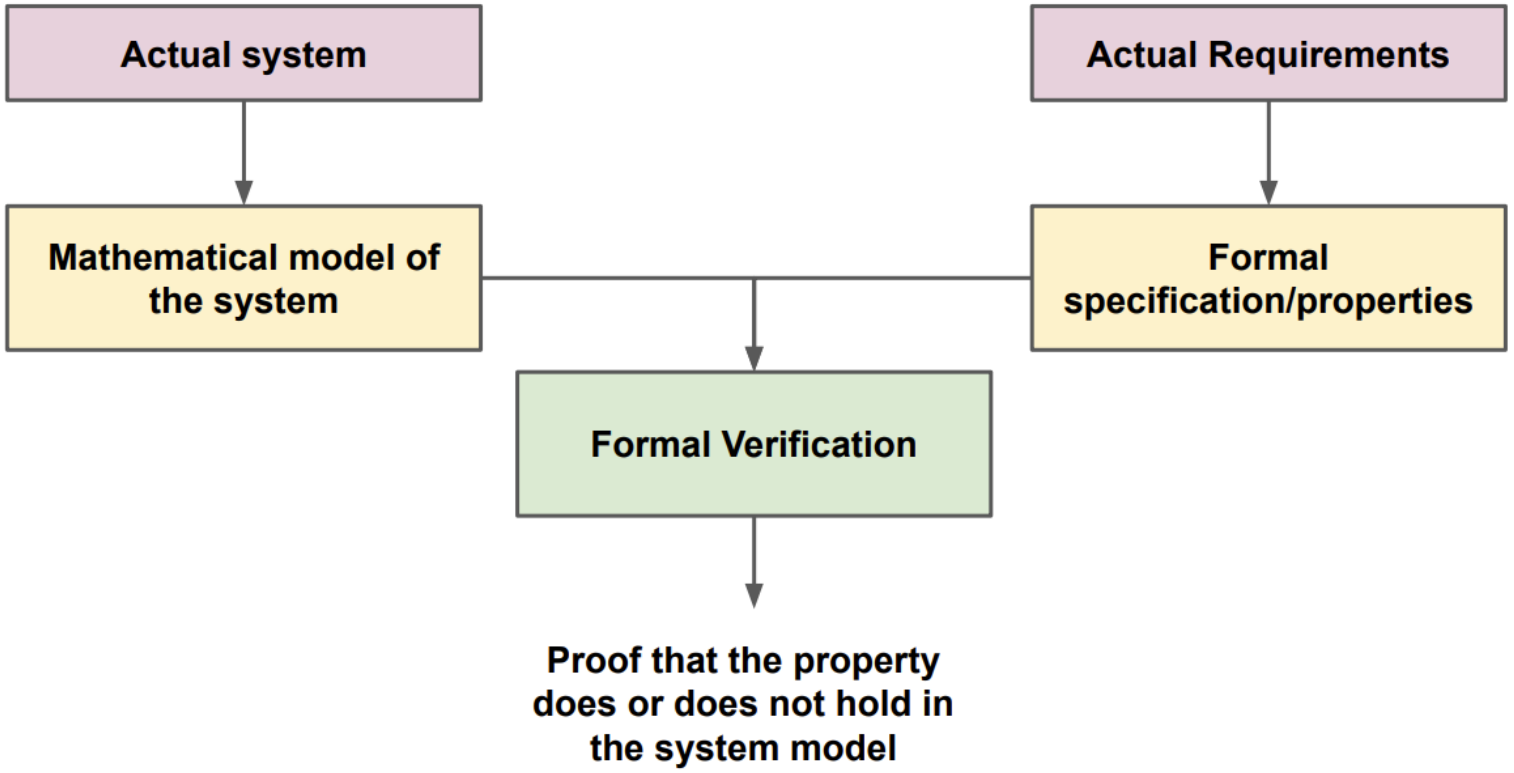
→ **Applicable for Network Verification.**



I. Network Verification

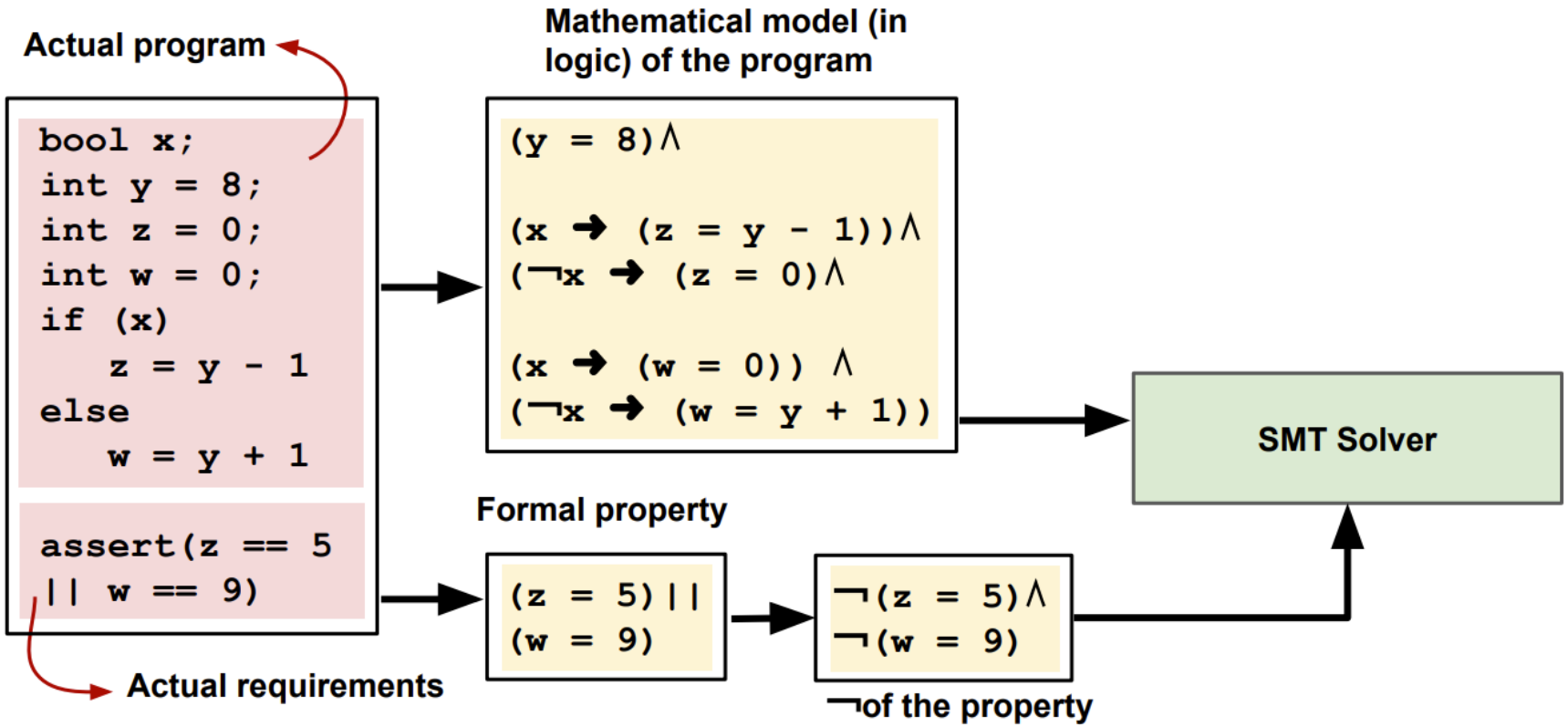
1. Formal Verification

How can we verify a system?



1. Formal Verification

An example of code verification



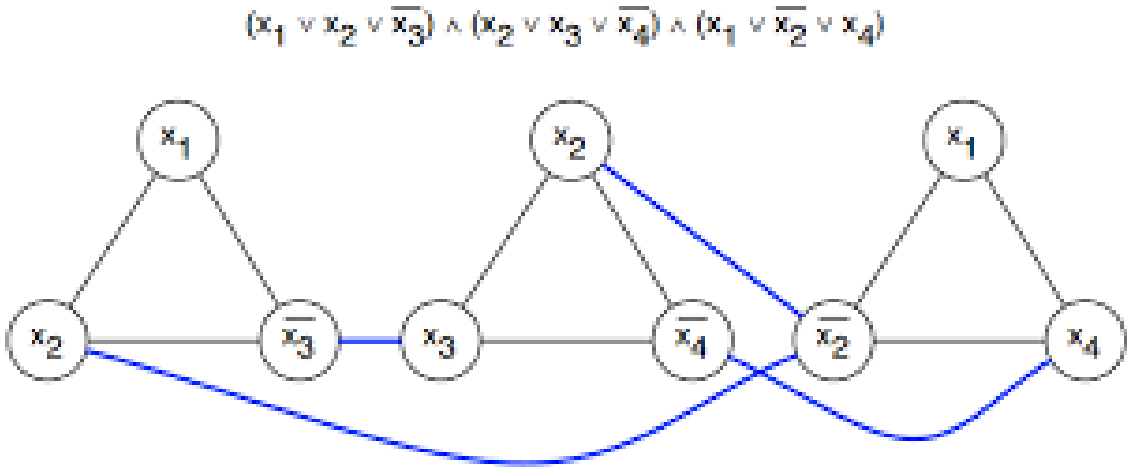
I. Network Verification

2. SMT Solver (Satisfiability Modulo Theories)

The Boolean Satisfiability Problem (SAT)

- Suppose you have a Boolean formula, e.g., (A and B) or (A and C).
- You can assign true or false to each variable.

→ **SAT problem**: Is there an assignment that will make the entire formula evaluate to true?

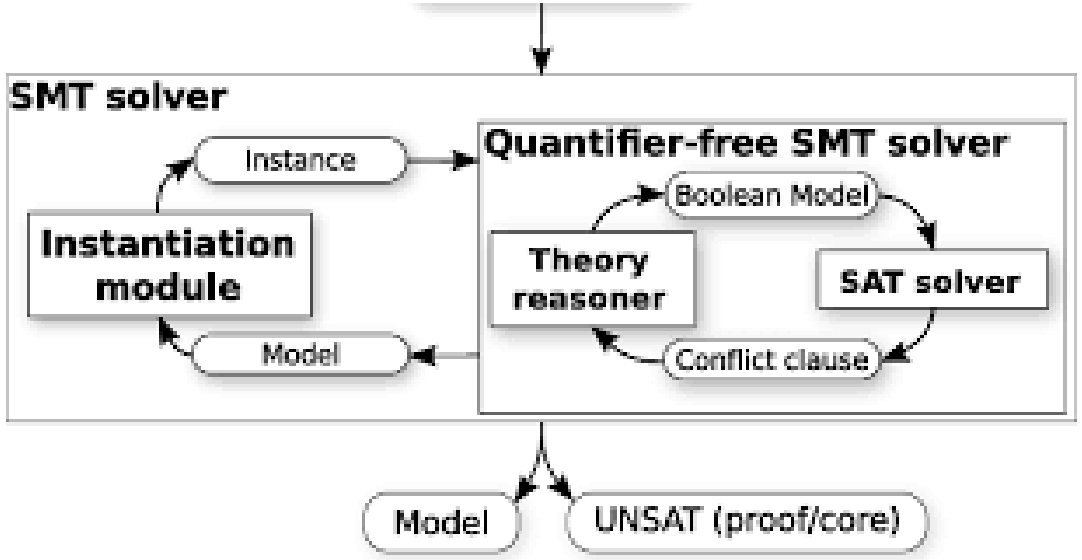


2. SMT Solver (Satisfiability Modulo Theories)

The SMT Solver

Same as SAT problem, but for more complex (first-order-logic) formulas:

- integer variables, real variables,…
- arrays, bit vectors, lists, strings,…
- function such as equality, addition, subtraction,…

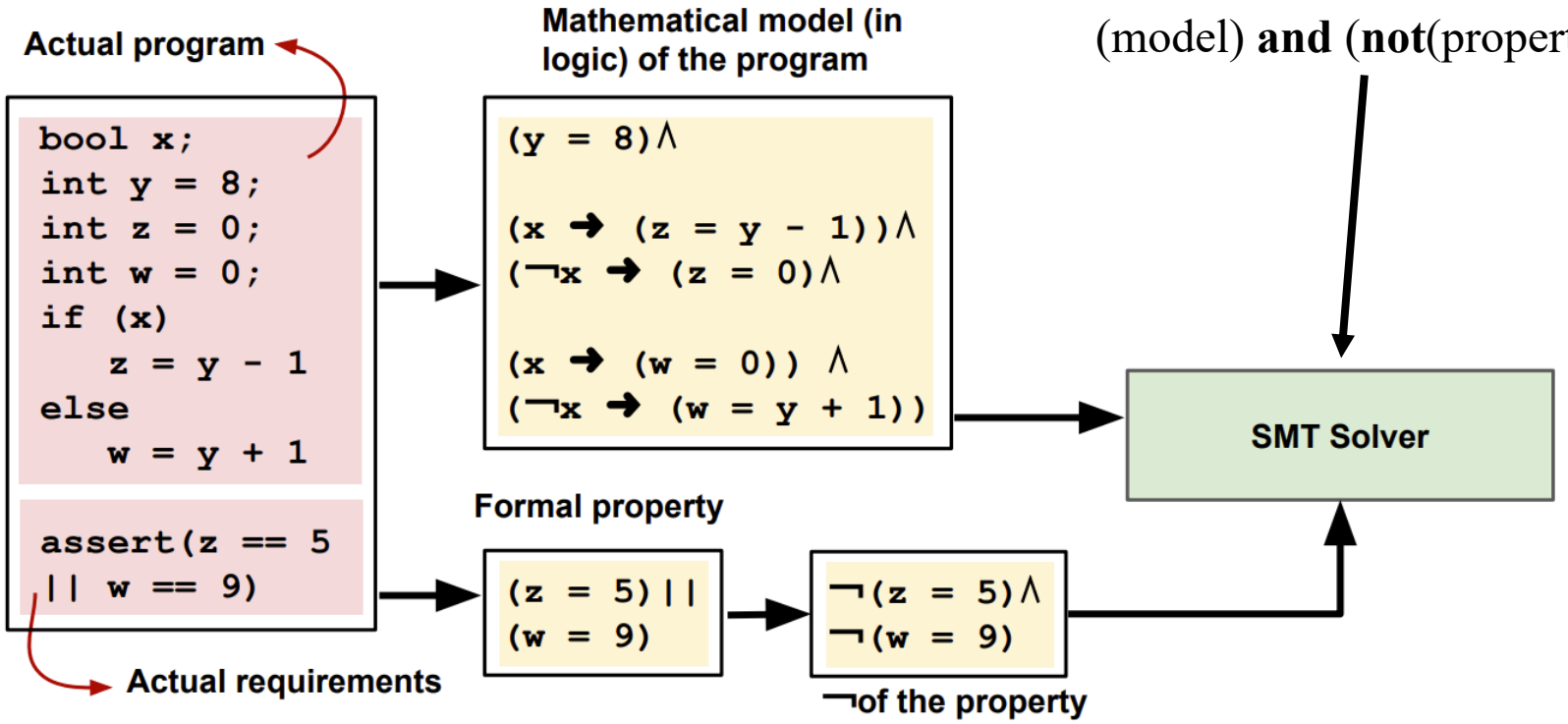


2. SMT Solver (Satisfiability Modulo Theories)

The SMT Solver

An example describing SMT Solver

Can we find any assignment to the variables x, y, z, and w to make the following formula evaluate to **TRUE**?

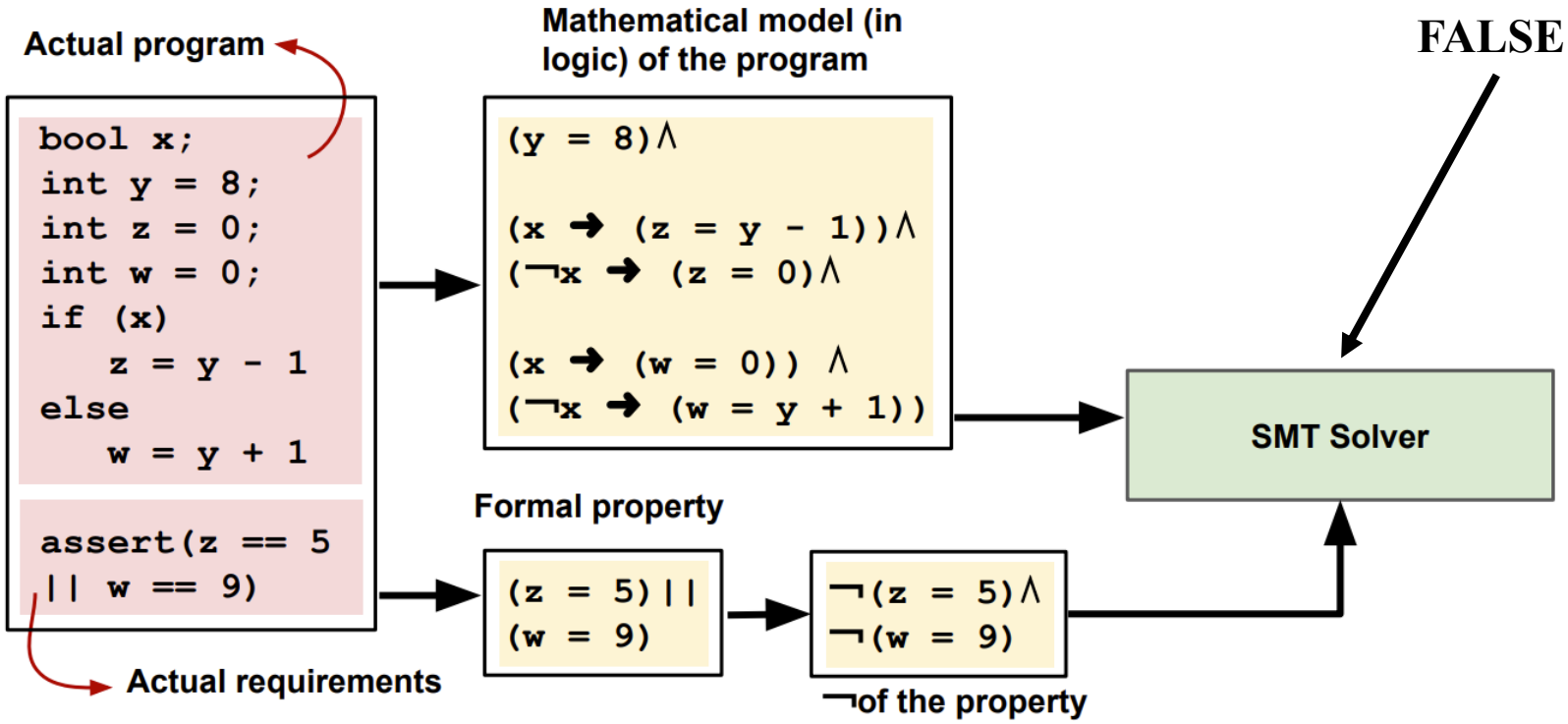


2. SMT Solver (Satisfiability Modulo Theories)

The SMT Solver

An example describing SMT Solver

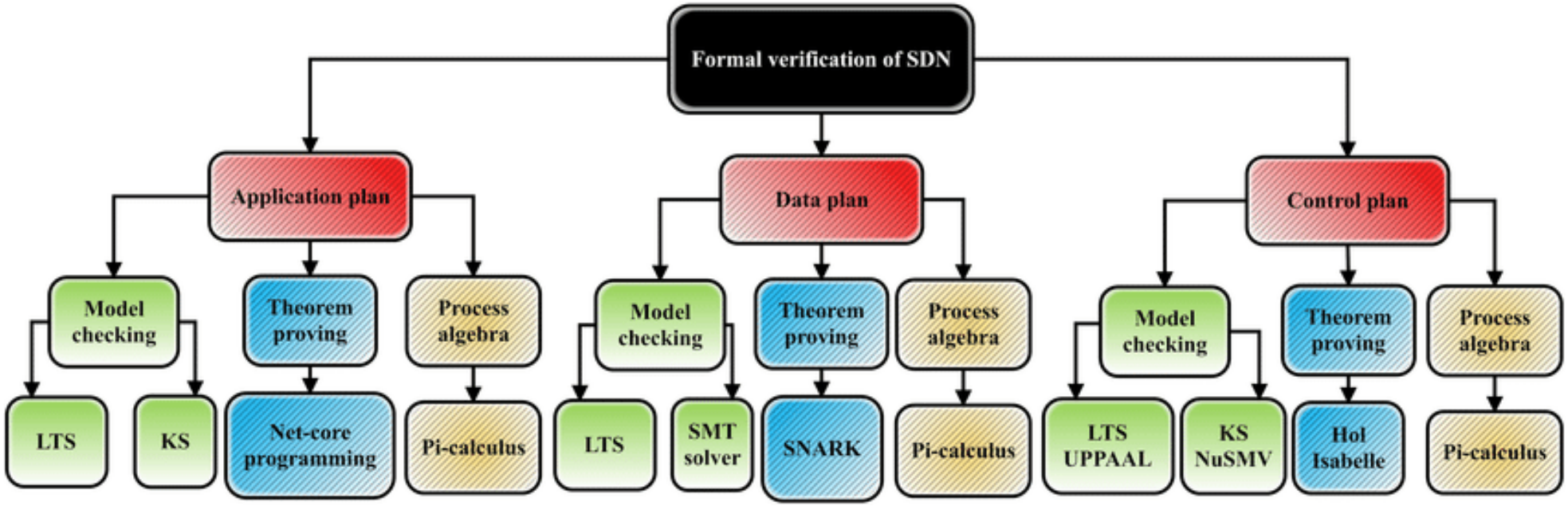
(model) **and** (**not**(property))
The above formula equal to **TRUE** when x = true, y = 8, z = 7, w = 0.
This is when the model = **TRUE** and the property = **FALSE**.



I. Network Verification

3. Why use formal verification in Networking?

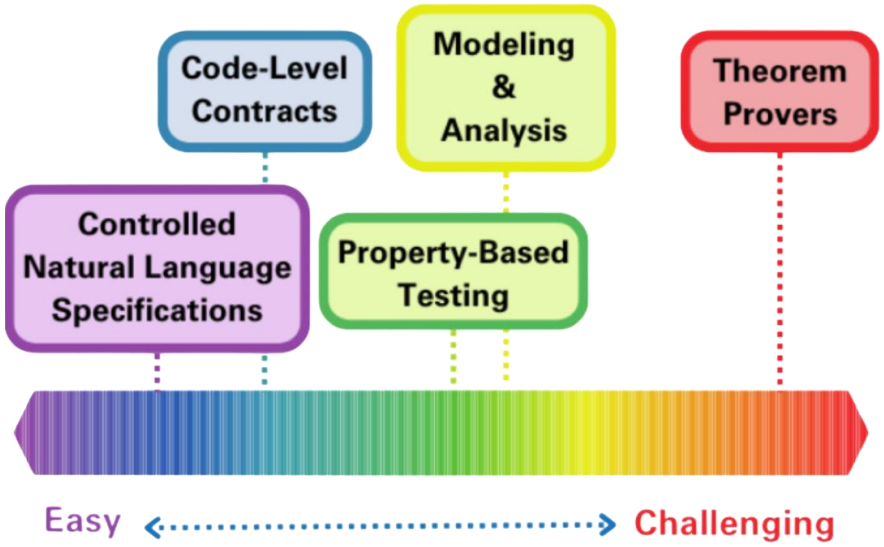
- Networks are growing increasingly complex.
- Networks are becoming a critical infrastructure.
- We need to catch bugs proactively before going into production.



I. Network Verification

4. Formal Methods in Networking

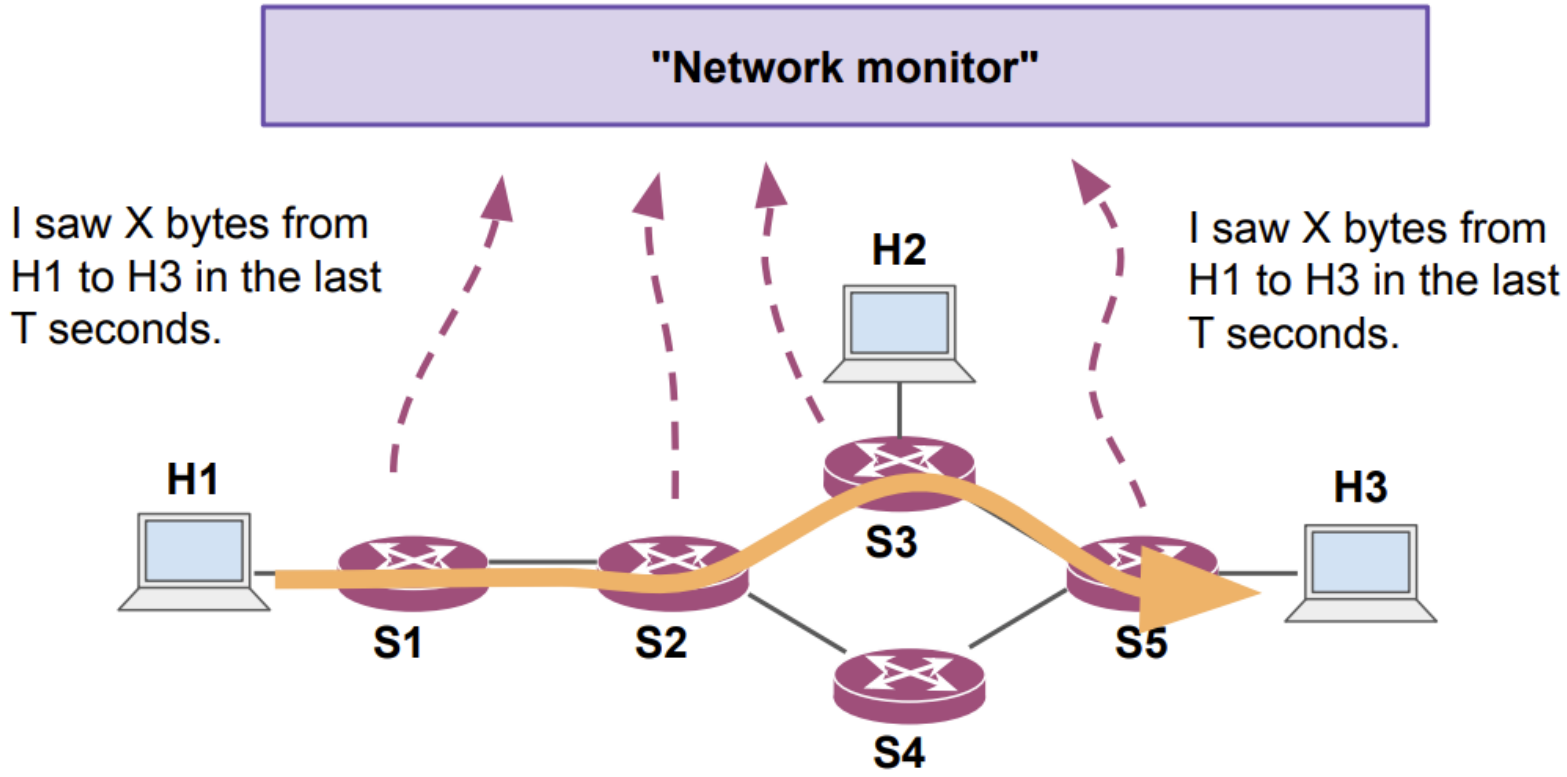
- Data plane Verification: model and analyze the forwarding rules on the data plane.
- Control plane Verification: model and analyze the control plane protocols that configure the data plane.
- Stateful and programmable data planes.
- Analyzing DNS: Is there a query under our domain that is sent for resolution to a name server, not under our domain?
- Analyzing Performance: Is there an input traffic pattern under which the network provides high latency?



II. Network Monitoring

1. What is network monitoring?

Understanding what is happening in the network at run-time and in real-time.

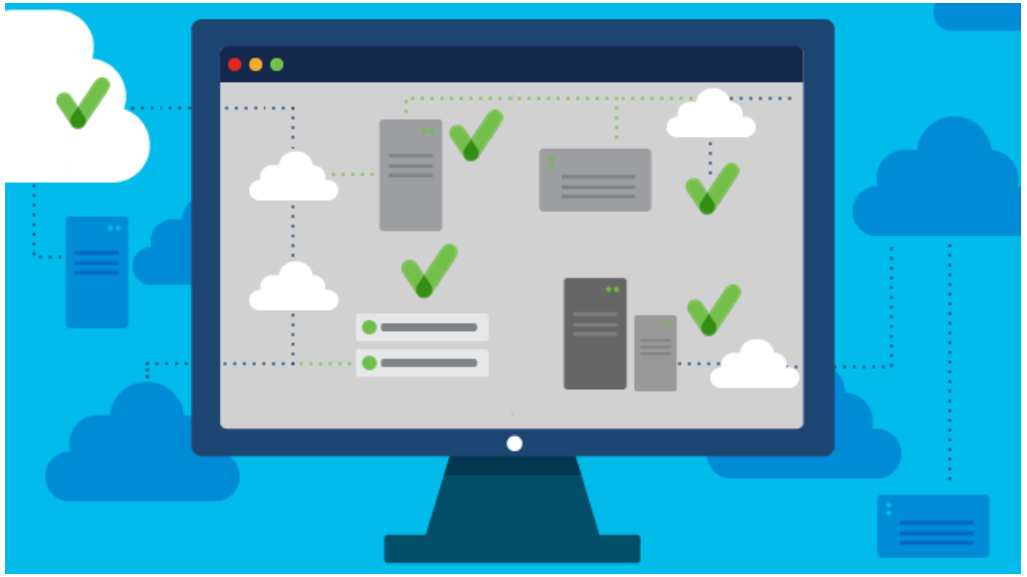


II. Network Monitoring

1. What is network monitoring?

It is important to monitor how network behaves.

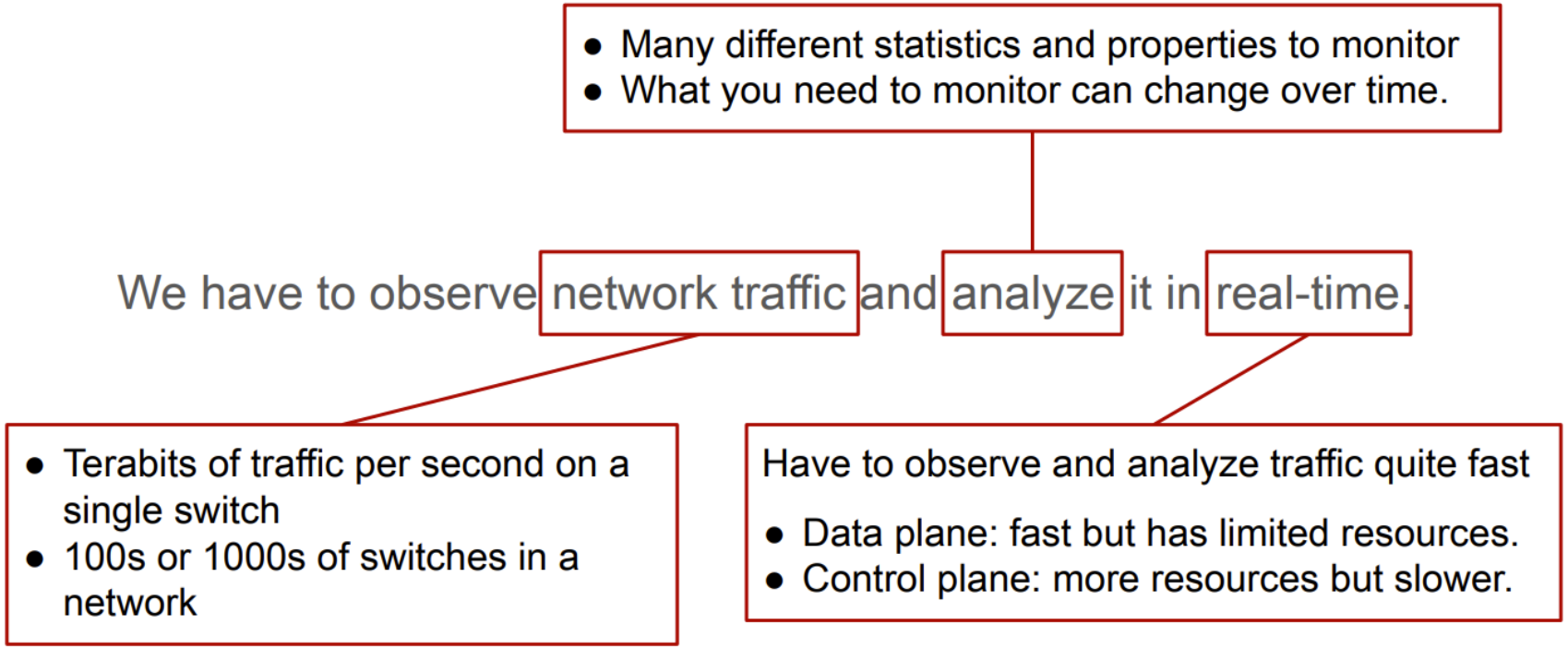
- Networks are quite dynamic: traffic patterns change, links and devices fail,…
- More often than not, we need to dynamically re-consider how to network should process traffic in response to changes at run-time.
- An accurate understanding of the state of the network is crucial for making good decisions.



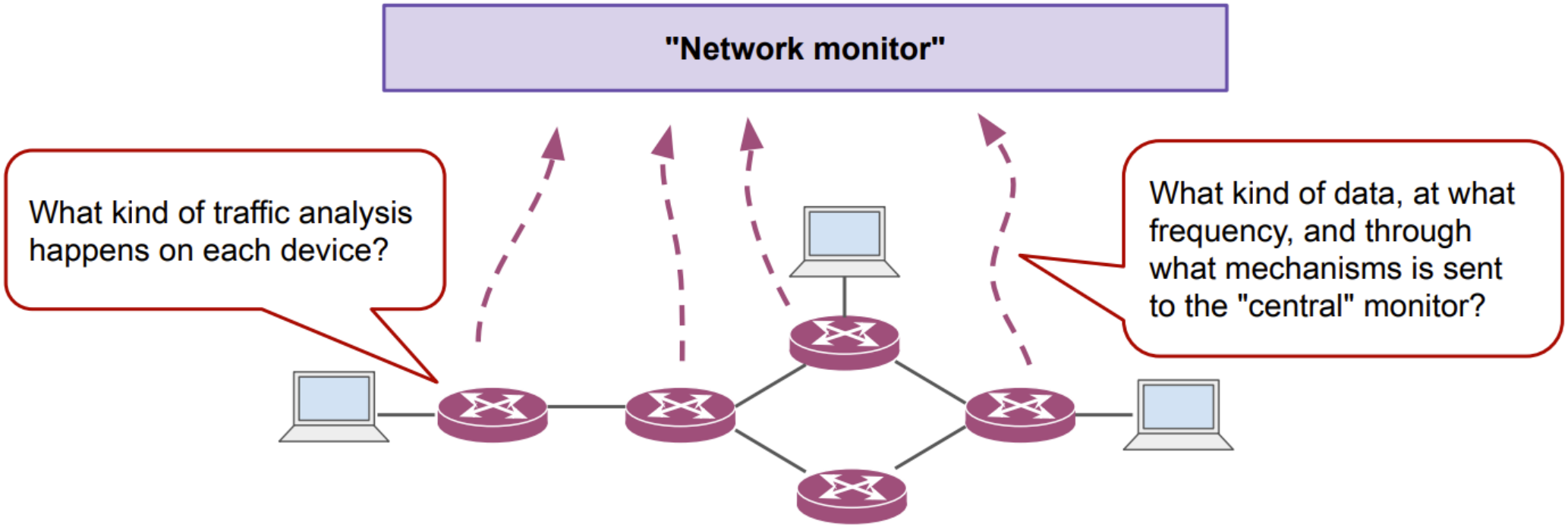
II. Network Monitoring

1. What is network monitoring?

It is important to monitor how network behaves.



2. The design space



II. Network Monitoring

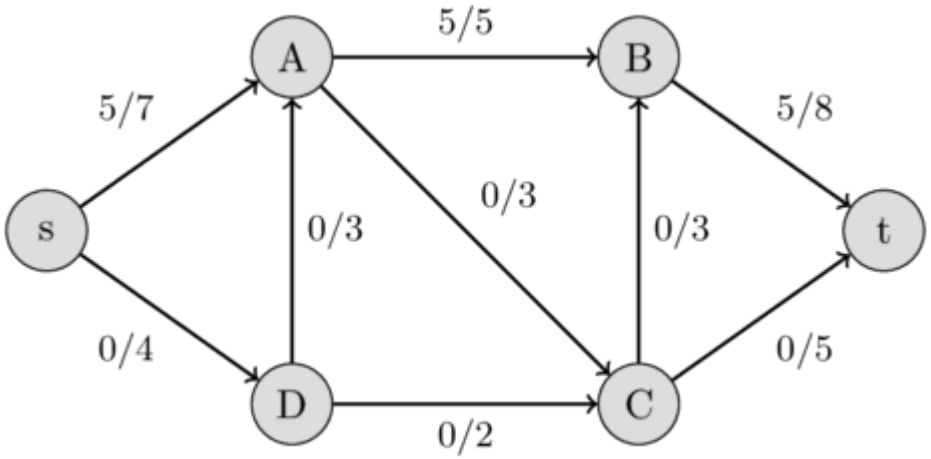
3. Single query: K largest flows

Suppose you want to know which K flows are the largest (has sent the largest amount of traffic) in your network.

You can send the flow identifier and size of each packet to the central network monitor.

You can count how many bytes of each flow you see in the switch and send a report of that to the network monitor every X seconds.

You can keep track of only the heavy hitters (as opposed to every flow) on each device and have the network monitor pull the info when needed.



II. Network Monitoring

4. Flexible and fine-grained monitoring in network programmability

- Program the data plane to gather the data that you want
- Program the data plane (and the run-time) to have the data pushed to/pulled form a central monitor when you want.
- Create top-down programmable monitoring frameworks:

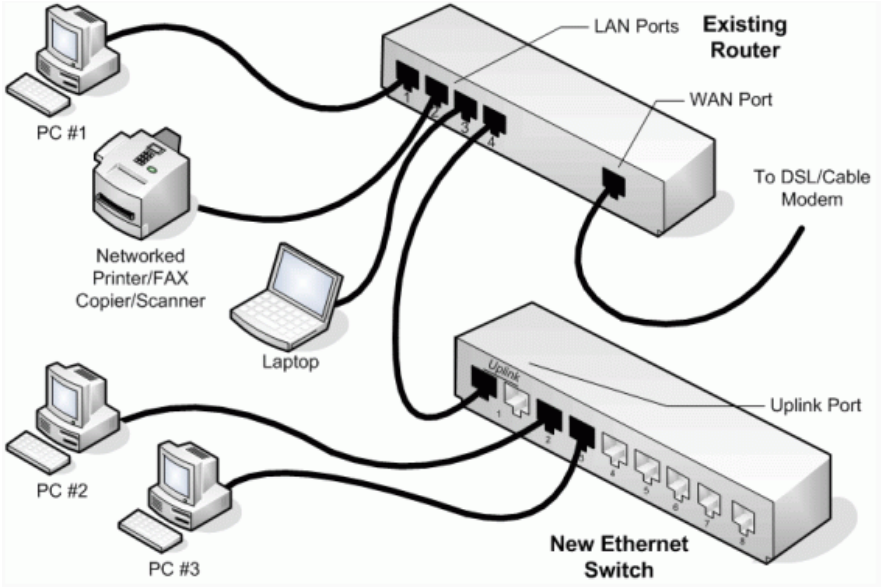
    - Users specify the information they are interested as queries.

    - The compiler and runtime figure out how to configure each device to collect and report information according to the query.

II. Network Monitoring

5. Sketches

- Modern high-speed switches can observe terabits of traffic every second.
- Keep information about a large amount of data in a substantially smaller amount of space.
- Can answer certain queries about it in an approximate way.
- They typically provide a trade-off between resource usage and accuracy.
- If you give them more space, they'll provide a more accurate result.

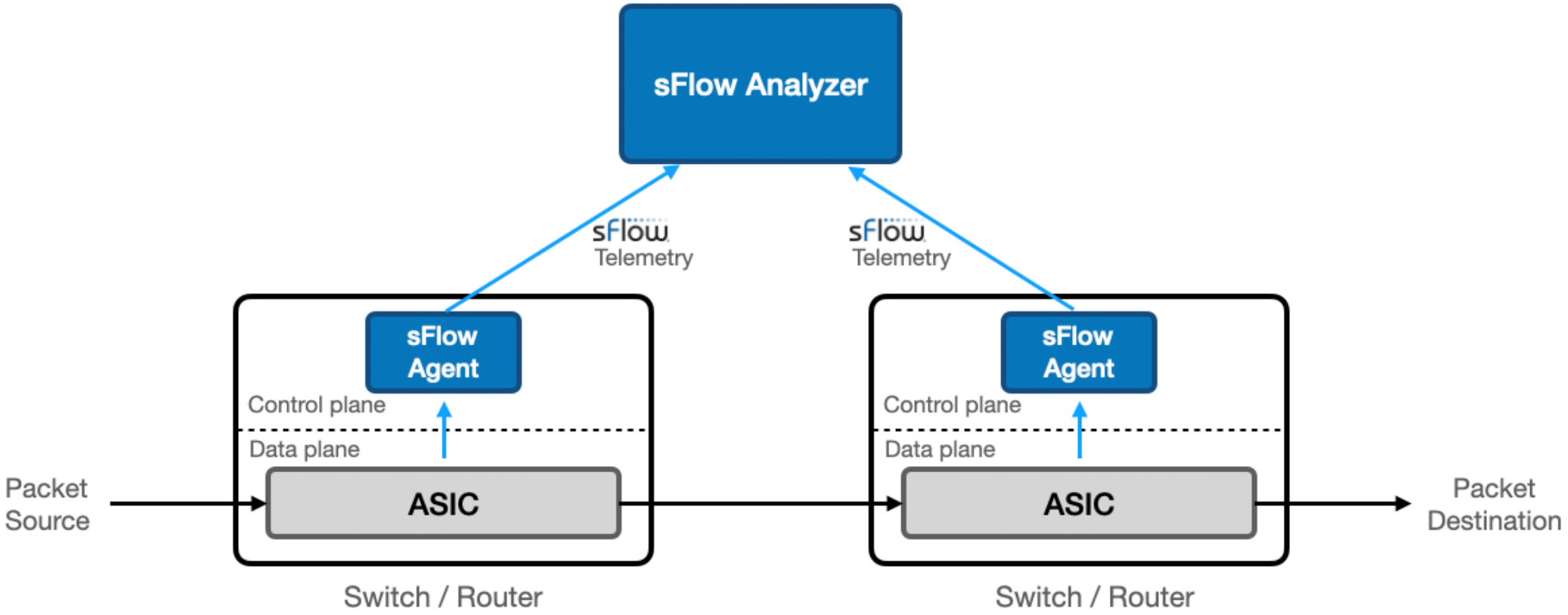


II. Network Monitoring

6. In-Band Network Telemetry (INT)

- In programmable data planes, you can define custom headers and process them however you want.
- INT proposes to add a "telemetry" header and have switches populate it with information that will help network monitoring .
- Once the packet gets to its destination, the information in the INT header can be analyzed there and/or sent to a central monitor.
- Having fine-grained information about what happened to the packet as it traverses a network is extremely useful.
- Specially for detecting and debugging transient and subtle problems.

6. In-Band Network Telemetry (INT)

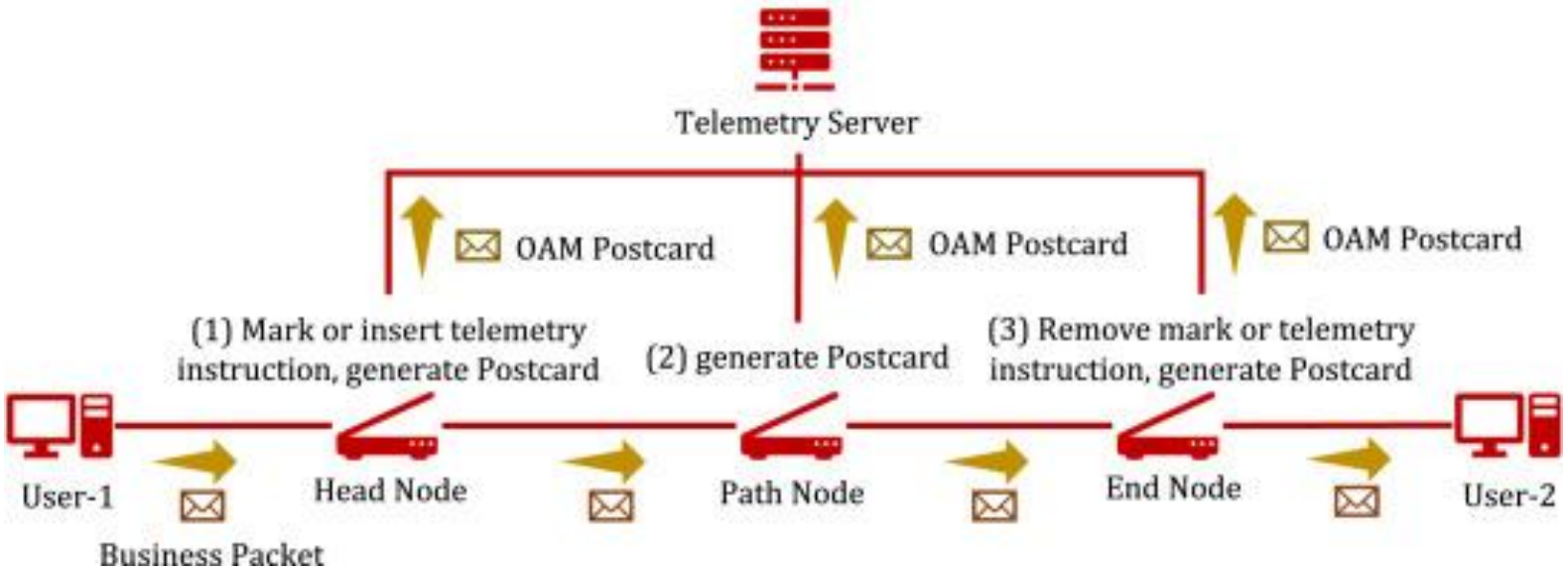


II. Network Monitoring

7. Connections to network verification

To ensure our networks satisfy our desired properties, we need

- scalable proactive analysis to catch as many violating scenarios as possible before production.
- flexible and fine-grained run-time monitoring to continuously watch for property violations at runtime.



HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !