Merlin: Programming the Big Switch

Shrutarshi Basu[1], Robert Soulé[2], Robert Kleinberg[1], Emin Gün Sirer[1], Nate Foster[1]

[1]Cornell University   [2]University of Lugano

Network administration is complicated and error-prone.

- Code in many languages runs on diverse hardware.
- Policy enforcement must scale over large networks.
- Tenants have diverse needs and usage patterns.

Networks lack unified abstractions.

- Target a single type of device or platform.
- Assume a single point of control.
- Disconnect between intended policies and actual implementation mechanisms. For example:

  "Ensure that all web traffic traverses a firewall" vs.
  "Match HTTP traffic & forward out switch port 4"

Merlin: Specify policies in a high-level language that is compiled to low-level code for network devices.

Policies may be:
- **Partitioned** into components for network devices,
- **Distributed** for finer grained enforcement,
- **Safely delegated** to tenants for network federation.

Examples

Bandwidth Guarantees

```
( ipSrc = 10.1.1.1 and
  ipDst = 10.1.1.2 and
  ipProto = 0x06 and
  tcpDst = 50060 )
  -> .* at min(100Mb/s)
```

Flexible Defense-in-Depth

```
forall
true
-> (.* fire1 .* fire2 .*) |
   (.* fire2 .* fire1 .*)
```
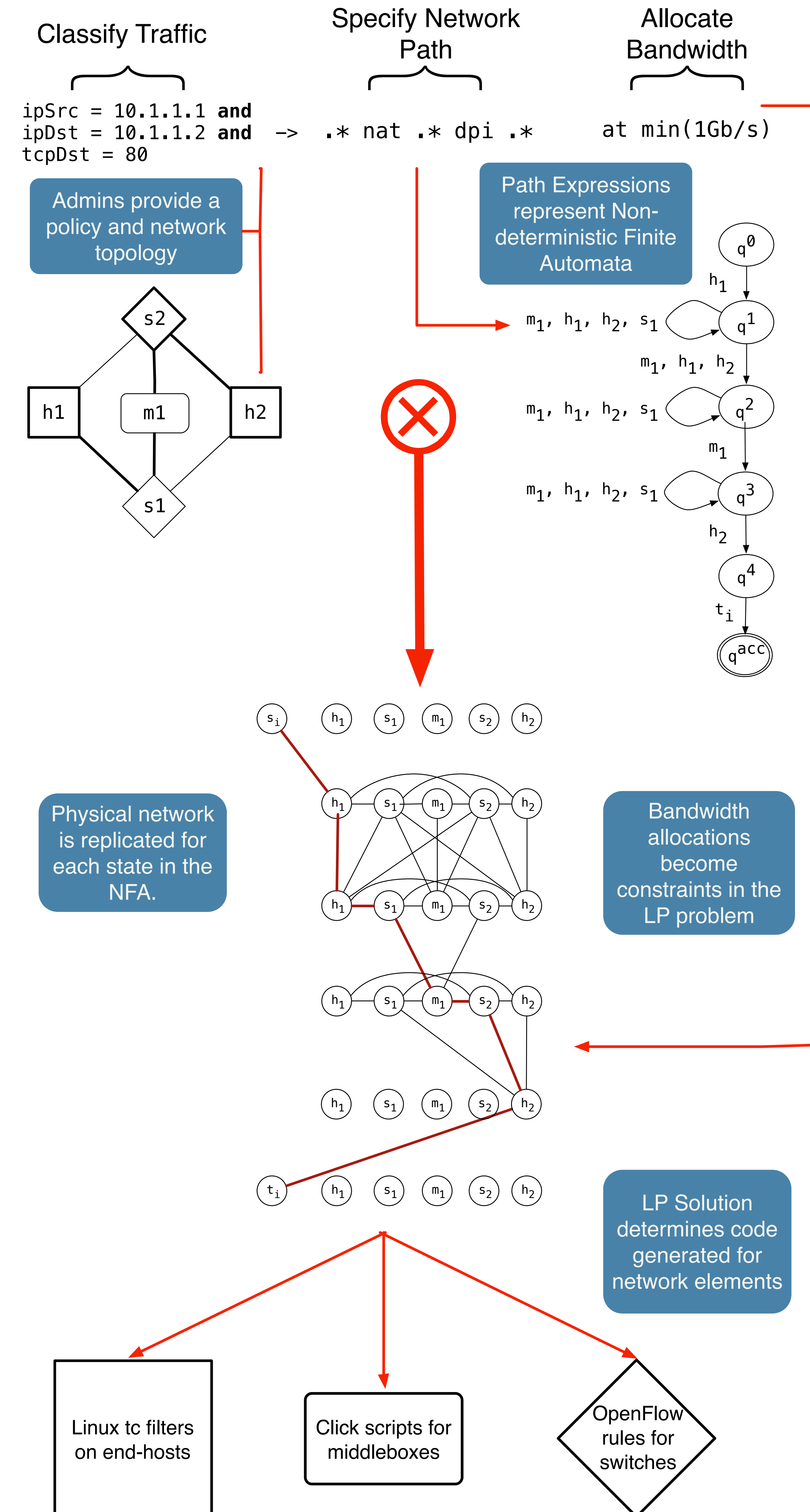
IP Multicast Control

```
forall
( ipDst = 224.0.0.1 or
  ipDst = 224.0.0.2 or
  ipDst = 224.0.0.3 )
  -> compress .*
     at max(10Gb/s)
```

Resource Isolation

```
forall
( ipSrc = 10.1.1.1/8 )
  -> .* m1 .*
forall
( !ipSrc = 10.1.1.1/8 )
  -> !(.* m1 .*)
```

Specify, Partition & Generate

Classify Traffic

```
ipSrc = 10.1.1.1 and
ipDst = 10.1.1.2 and  ->  .* nat .* dpi .*     at min(1Gb/s)
tcpDst = 80
```

Specify Network Path

Allocate Bandwidth

Admins provide a policy and network topology

Path Expressions represent Non-deterministic Finite Automata

Physical network is replicated for each state in the NFA.

Bandwidth allocations become constraints in the LP problem

LP Solution determines code generated for network elements

Linux tc filters on end-hosts

Click scripts for middleboxes

OpenFlow rules for switches

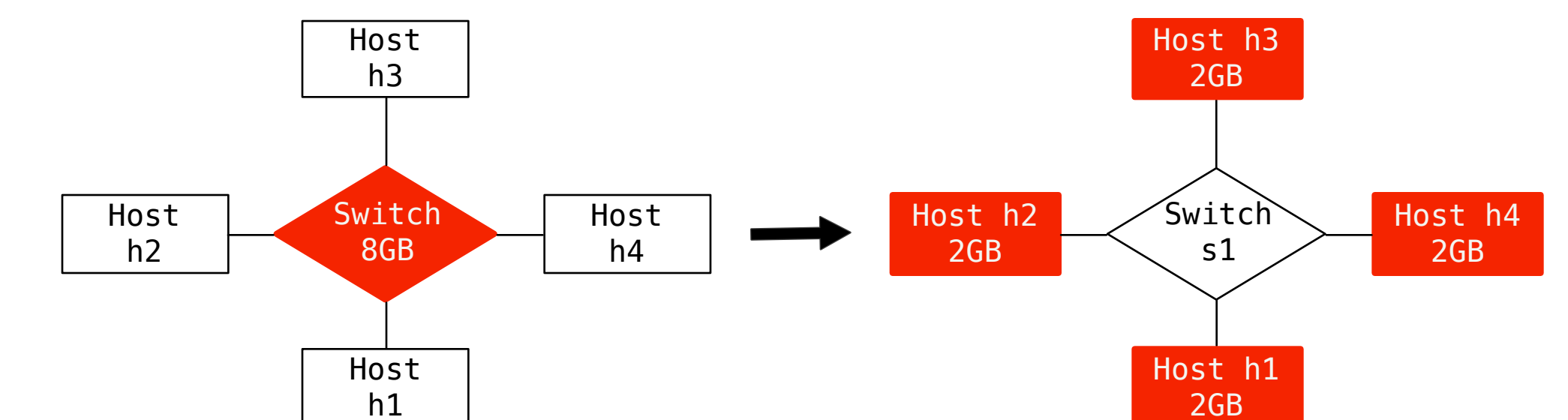## Delegate, Distribute & Verify

Delegate

```
ipSrc = 10.1.1.0
-> .* nat .*
   at min(1Gb/s)
```

```
ipSrc = 10.1.1.0 and tcpDst = 80
-> .* nat .*
   at min(700Mb/s)
```

```
ipSrc = 10.1.1.0 and !(tcpDst = 80)
-> .* dpi .* nat .*
   at min(300Mb/s)
```

Tenants may refine the traffic predicates for policies and the allowed paths and resource allocations
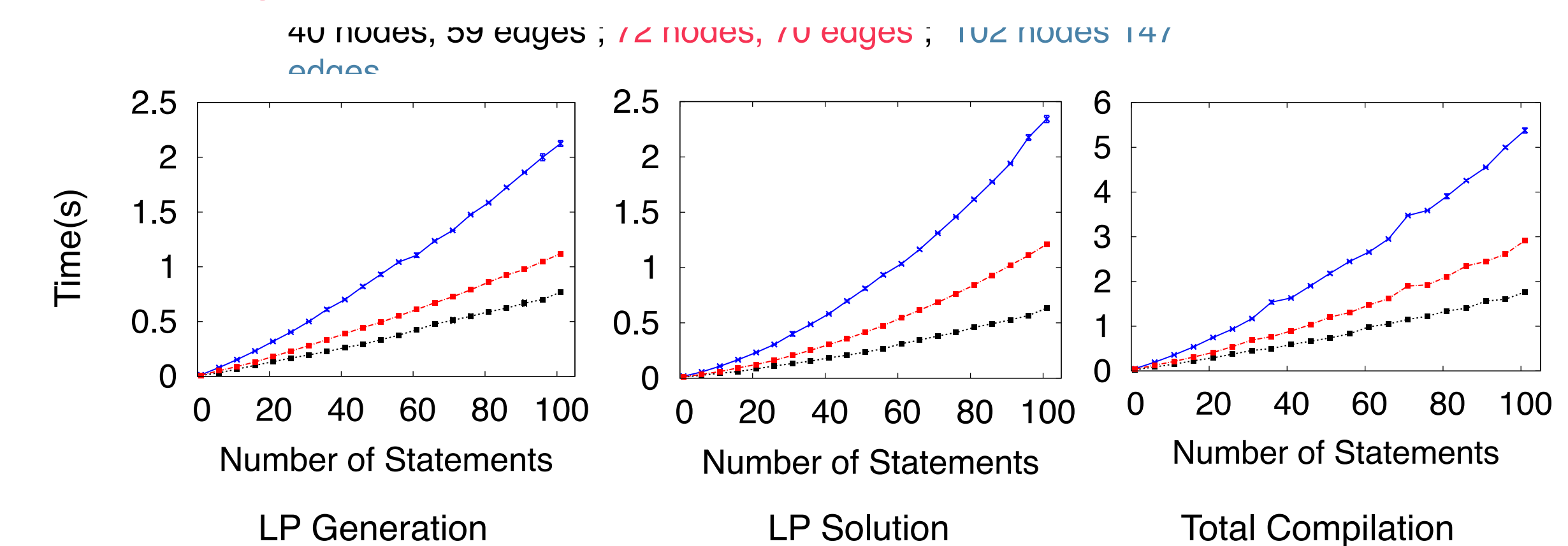
Distribute

Tenants may distribute allocations among themselves, and negotiate allocations with other tenants
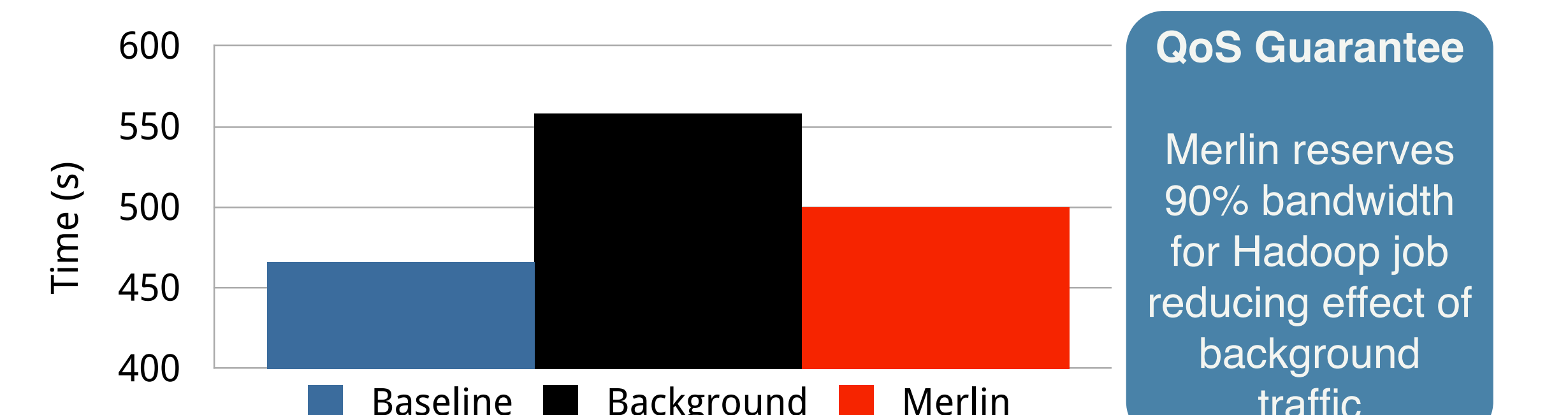
Verify

- Check for language inclusion between the regular expressions in statements with overlapping predicates.
- Check that the sum of the bandwidth constraints in all overlapping predicates implies the original constraint.

Deployment & Results

40 nodes, 59 edges ; 72 nodes, 70 edges ; 102 nodes 147 edges

LP Generation   LP Solution   Total Compilation

Scalability: Compiler completes in <6s for 100 node,147-edge topology & 100 statement policy

Baseline   Background   Merlin

QoS Guarantee

Merlin reserves 90% bandwidth for Hadoop job reducing effect of background traffic