Network verification and synthesis

CSE 599N1

Sep 25, 2019

Who are we?

Ratul Mahajan

- UW → MSR → Intentionet → UW
- One of the first paper was “Understanding BGP misconfiguration” (2002)

Ryan Beckett

- Princeton → MSR
- Recently finished thesis: Network Control Plane Synthesis and Verification
  - Won the ACM SIGCOMM dissertation award and ACM Honorable Mention

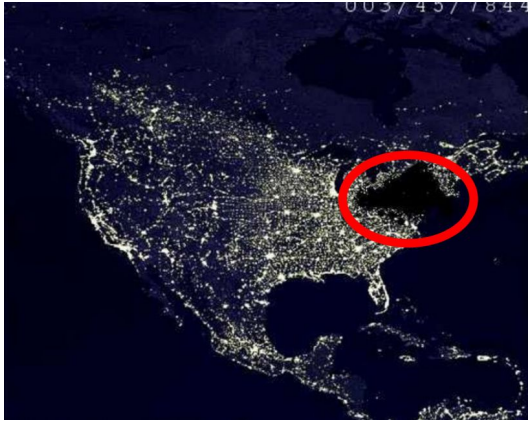
What is verification?

“Mathematical analysis of a system to determine rigorously if it meets some end-to-end goal”

Why bother with verification?

Mission critical systems

Ariane-5 self-destruction
software interface issue



Northeast Blackout
power control software

Boeing 737 Max crash
control software bug

Why bother with network verification?

Another massive outage takes down many of the internet's biggest

Jacob Siegal @JacobSiegal

TIME WARNER CABLE INTERNET OUTAGE AFFECTS MILLIONS

NEWS

United Airlines flights resume after 'network connectivity'

'It's been chaos': Sacramento airport network outage resolved after frustrating morning

Target stores suffer network outage; stores back online

Nationwide internet outage affects CenturyLink customers

Sutter Health's computer network outage forces surgery cancellations

Julie Spitzer - Wednesday, May 16th, 2018 Print | Email

Why bother with network verification?

[A]n **unplanned data center outage costs companies more than $7,900 per minute**, and the cost continues to rise. The cost of downtime per minute has risen an incredible 41% since 2010…**the average cost per incident is now at a staggering $690,200**.

https://www.linktek.com/cost-of-network-downtime/

"Networks have become **the infrastructure for the infrastructure**… the cloud is holding up the computation that supports the planet so **it is mission critical and can not have glitches**." -- Albert Greenberg (head of Azure)

https://www.youtube.com/watch?v=b94Iv-oN91s

Course Logistics

Why this course?

Lots of research activity in the past few years

- Has opened a new sub-field of networking
- Hard to make sense of it all

Important (and fun!) topic

- Combines networking with formal methods and programming languages

Course goals

We will collectively

- Synthesize work in this area
- Identify open research problems and promising new directions

Stretch goal: Write a survey paper with our experience and findings

Course organization

Primarily paper reading and projects

- One main paper per lecture
  - Each student will lead the discussion of at least one paper
    - Sign up now! Via Canvas → Collaborations → Paper signup
- Highly encourage you to read additional material

- Projects in groups of 2-3
  - Follow recommended plan (next slide)
  - Or, you may pick your own -- come talk to us first

Recommended project plan

Based on a small language for data plane and control plane (later today)

1. Dataplane verification [2 weeks]
2. Incremental **or** scalable dataplane verification [2 weeks]
3. Control plane simulation [2 week]
4. Control plane verification [2 weeks]
5. Control plane synthesis [2 weeks]

(Deadlines will go on the Web page soon)

Turning in projects

Code (pointer) and a short report

- Ideal: host on GitHub with a README.md
  - We should be able to clone and run (easily)
- Report should be no more than 2 pages
  - Detail the approach you took
  - Benchmark performance as a function of network size

Grades

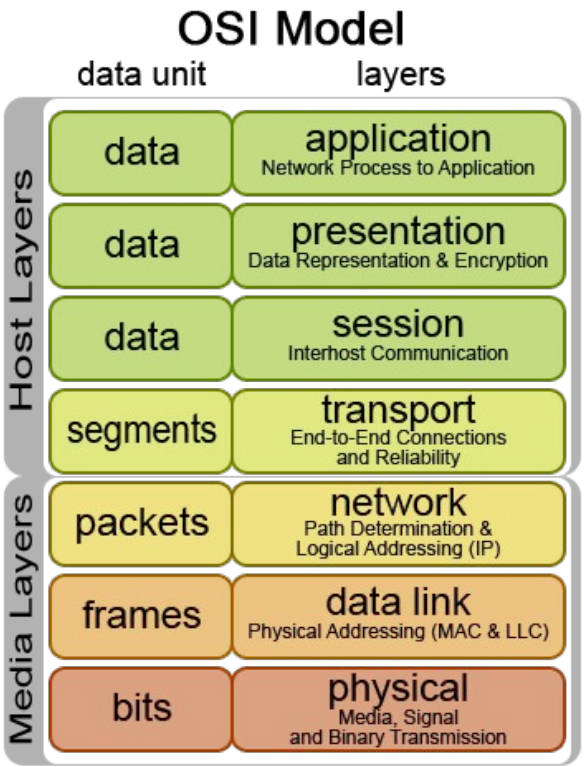
Class participation: 40%

- Offline and in-class discussion
  - Additional reading is excellent fodder for offline discussion
- Paper presentation

Projects: 60%

Networking Background

Networking primer



How to enable human interaction with applications

How to translate data for the application

How to set up and terminate application conversations

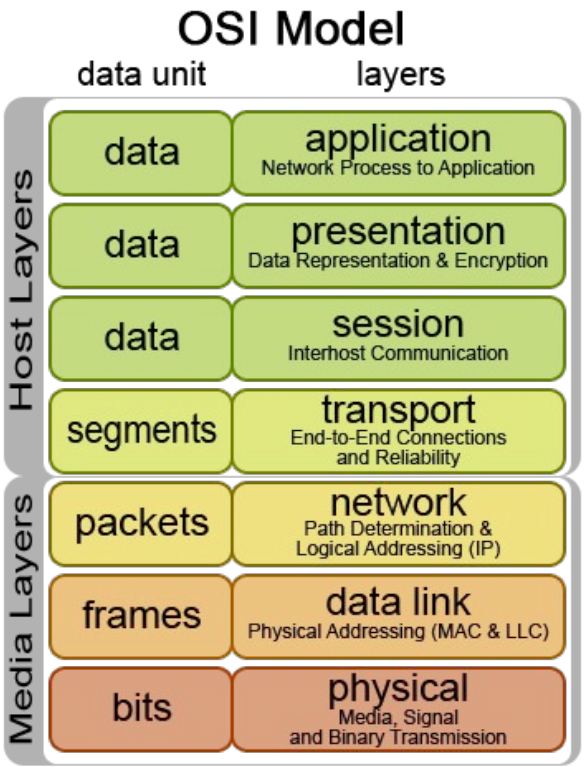
How to get data across the network reliably

How to get data packets across the network

How to get bits across a wire reliably

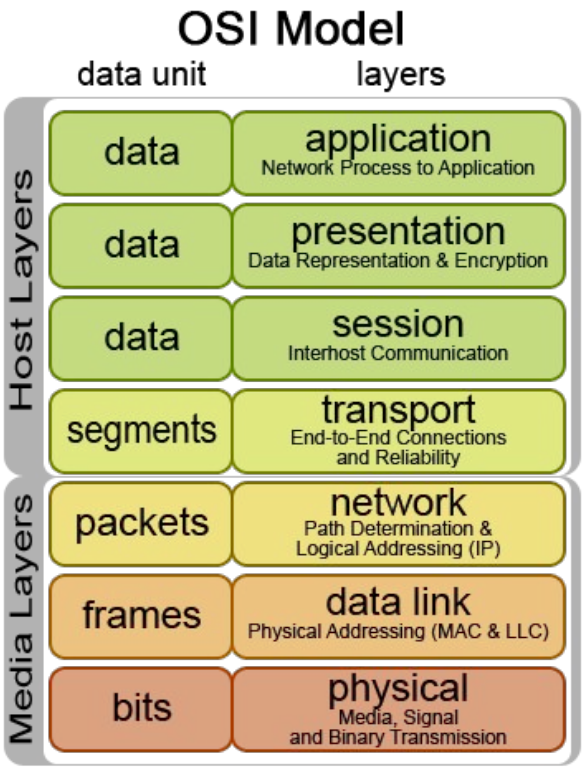
How to transport across electrical or mechanical interface

Networking primer



How to get data packets across the network

Networking primer

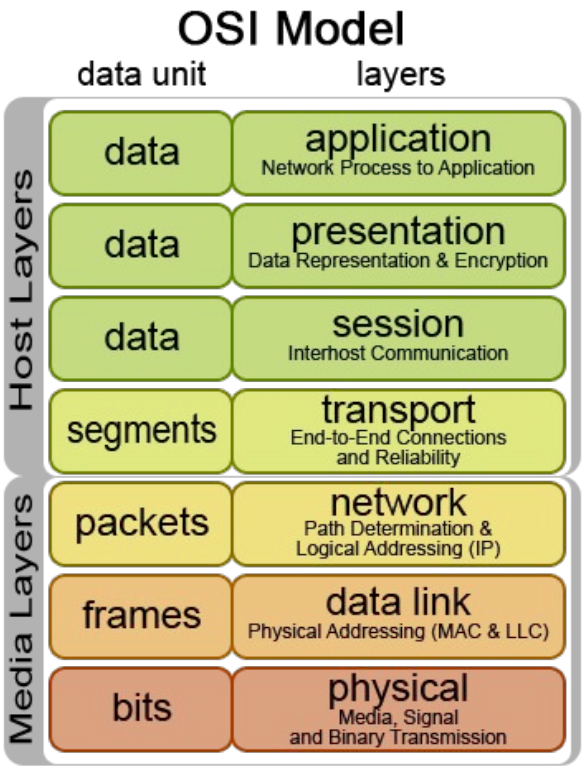


Complications
- Distributed protocols
- Complex interactions
- Vendor languages
- Middleboxes

Rich policies
- Business preferences
- Security
- Traffic engineering
- Fault tolerance

How to get data packets across the network

Networking primer



Formal Methods Toolbox

Model Checking BDDs

SAT Ternary symbolic execution

Abstract Interpretation SMT

Symmetry reduction Bisimulation

Assignment 1

Our dataplane language

Dataplane:

- Abstracts away many details
- Topology, forwarding tables, ACLs
- YAML based format

Specification:

- Collection of reachability statements
- Specifies packet headers, ingress + egress locations
- YAML based format

Fields are always fully specified for simplicity

Dataplane format

Device interfaces

Forwarding table rules as an ordered list

Access control lists


```yaml
Devices:
  - Name: r1
    Interfaces:
      - Name: r1@Eth0
        Neighbor: null
        InAcl: null
        OutAcl: null
      - Name: r1@Eth1
        Neighbor: r2@Eth1
        InAcl: null
        OutAcl: r2_outbound_host_permit
      - Name: r1@Eth2
        Neighbor: r3@Eth1
        InAcl: null
        OutAcl: r2_outbound_host_permit
    ForwardingTable:
      - Prefix: 70.4.193.0/24
        Interface: r1@Eth1
      - Prefix: 70.4.193.0/24
        Interface: r1@Eth2
      - Prefix: 10.0.0.1/32
        Interface: r1@Eth1
      - Prefix: 10.0.0.1/32
        Interface: r1@Eth2
      - Prefix: 70.4.194.0/24
        Interface: r1@Eth0
    Acls:
      - Name: r2_outbound_host_permit
        DefaultAction: Deny
        Rules:
          - Description: "allow srcip for host"
            DstIp: 0.0.0.0/0
            SrcIp: 70.4.194.0/24
            Protocol: 0-255
            DstPort: 0-65535
            SrcPort: 0-65535
            Action: Allow
```

Query format

If a packet enters one of these ingress interfaces

Then the packet must exit one of these egress interfaces

So long as the packet has one of these headers

```
- Ingress: [r1@Eth0]
  Egress: [r4@Loopback0]
  DstIp: [10.0.0.1/32]
  SrcIp: [0.0.0.0/0]
  Protocol: [0-255]
  DstPort: [0-65535]
  SrcPort: [0-65535]
```

Questions?