Automatic Inference of High-Level Network Intents by Mining Forwarding Patterns

Ali Kheradmand

University of Illinois at Urbana-Champaign
Typical network configuration procedure

- Mostly manual in today’s networks
- Alas, the semantic gap!
Microsoft: Misconfigured Network Device Caused Azure Outage

Google Cloud Went Down Because It Was Misconfigured

Wired
How a Tiny Error Shut Off the Internet for Parts of the US

A simple misconfiguration spiraled into outages for internet service providers and large internet platforms around the US.
How to bridge the gap?

High-level Intents (informal) → Operator → Configurations → Network
Current approaches

**Verification**
- Configured network
- High-level intents
- Formal specification
- Model of system S
- Property \( \Psi \)
- Verifier
- Does S satisfy \( \Psi \)?

**Synthesis/Compilation**
- High-level intents
- Formal specification
- Compiler
- Configurations
- Network
Reality

• Formal specifications not existing / maybe not even known
  • Operator is given an informal description of networking objectives
    • Intents are implicit

• Operators inherit legacy networks
  • Asked to maintain it
  • 82% concerned that changes would cause problems with existing functionality [Kim et al, NSDI’15]
A new approach

- Idea: automatically infer high-level intents by looking at the low-level network forwarding behavior

How about going this way?
Automatic Network Intent Miner (Anime)

Observed behavior (low-level) → Anime → Inferred intents (high-level)

Data plane/control plane/configuration analysis
Live traffic monitoring...

Possible
Inferred
Observed
Applications

• Streamline “Intent Based Networking”
  • Verification/Synthesis
  • Automatic migration from legacy networks to cloud, SDN, ...
  • Transparent optimizations, automatic repair, etc.

• Network behavior summarization
  • Debugging and management

• Anomaly analysis
  • Misconfiguration detection
Example

Observed low-level behavior

dstIP: 10.0.1.2, start: U1, waypoint: F1, end: S1

dstIP: 10.0.1.2, start: U2, waypoint: F1, end: S1

dstIP: 10.0.1.2, start: U3, waypoint: F2, end: S1

dstIP: 10.0.1.3, start: U1, waypoint: F1, end: S2

dstIP: 10.0.1.3, start: U2, waypoint: F2, end: S2

dstIP: 10.0.1.3, start: U3, waypoint: F2, end: S2

dstIP: 10.0.1.2/31, start: User, waypoint: Firewall, end: Server

Inferred high-level intent

Inferred higher-level intent
Expressing behavior and intents

• Using **features**
  • Each corresponding to one aspect of an observed behavior
    • Devices: e.g. start, waypoint, end, entire forwarding path
    • Header information: e.g. source/destination address or port
    • Conditions/state, e.g. temporal (snapshot timestamp), topological (link failures). device (connection state)
  • Each has a set of **labels** associated with it:
    • Device: U1, U2, U3, S1, S2, FW1, FW2, User, Firewall, Server, Any
    • IP: 10.0.1.2, 10.0.1.2/31, 10.0.0.0/8, ...

\[
\text{dstIP: 10.0.1.2, start: U1, waypoint: F1, end: S1}
\]
Insight

- Networks are hierarchical
  - E.g. IP hierarchy (CIDR), device role hierarchy
  - Idea: use hierarchical labels

![Diagram showing IP and device hierarchy]

- IP:
  - 0.0.0.0/0 : $2^{32}$
  - 10.0.1.2/31 : 2
    - 10.0.1.2/32 : 1
    - 10.0.1.3/32 : 1

- Device:
  - Any: 7
    - User: 3
      - $U_1:1$  $U_2:1$  $U_3:1$
    - Firewall: 2
      - $F_1:1$  $F_2:1$
    - Server: 2
      - $S_1:1$  $S_2:1$
Library of feature templates

DAG $<V,E>$

Flat $\{L_1,L_2,\ldots,L_n\}$

Range

IP Prefix

Set $\{L_1,L_2,\ldots,L_n\}, b$

TBV $<n>$

Hierarchical Reduced Regex (HRE)
Expressing behavior and intents

• Combine multiple features (Tuple<$F_1, \ldots, F_n$>) to express behavior and intents
  • E.g. Tuple<dstIP, start, waypoint, end>

Potential high-level intents

Expressible low-level behavior
Problem definition

• Given
  • A set of observed behavior \( P = \{p_1, \ldots, p_n\} \)
  • Limit \( k \) on the number of inferred intents

• Find
  • Intents \( I = \{i_1, \ldots, i_{k'}\} \) \( (k' \leq k) \)

• Such that
  • Each behavior in \( P \) is represented by at least one intent from \( I \)
  • Minimizes \( \sum_{i \in I} cost(i) \)
Heuristic solution

\( \text{join}(g) \): **Single** best intent representing **all** behavior in \( g \)

\[ \{p_1, ..., p_n\} \rightarrow \text{cost}(\text{join}(p_1,p_2)) \rightarrow \text{join}(g_1), \text{join}(g_2), \text{join}(g_3), ... \]

Insight: cost of \( \text{join}(g) \) inversely related to similarity of behavior in \( g \)

Efficient

\( \text{Least cost common ancestor} \)

Clustering methods

\[ \text{Any}:7 \]

User:3

\[ U_1:1 \quad U_2:1 \quad U_3:3 \]

Firewall:2

\[ F_1:1 \quad F_2:1 \]

Server:2

\[ S_1:1 \quad S_2:1 \]

\( \text{join}(U1,U2) = \text{User} \text{ (cost: 3)} \)

\( \text{join}(U1,FW1) = \text{Any} \text{ (cost: 7)} \)
Evaluation

Quality of inferred intents  Performance

Refer to the paper
Evaluation (objective quality metrics)

\[
\text{Precision: } \frac{TP}{TP+FP} \\
\text{Recall: } \frac{TP}{TP+FN}
\]

Specificity (exclusion of impossible behavior)

Coverage (inclusion of possible behavior)

Whitelist assumption: any behavior not explicitly allowed by any intent in a set of intents is disallowed by that set.
Evaluation (comparison with Net2Text)

• Re-implemented Compass algorithm from Net2Text [Birkner et al, NSDI18]
  • Summarize forwarding traffic
    • “as much as possible”
  • No use of hierarchy

• Net2Text dataset
  • Simplified ISP, Real-world topologies, IPv4 RIB, and AS-to-organization information
  • No hierarchy (to be fair)
  • AT&T topology, 25 nodes, 5 egresses, 100 prefixes, 2500 paths
  • Perfect observation (possible = observed)
    • Goal: summarization
Evaluation (effect of hierarchical values)

- Multiple groups of servers
- Synthetic access control policies
  - “group/server A can communicate with group/server B”
- 100 nodes, 5 groups of size bw 5-30, 10 intents, 435 paths
- Perfect observation (possible = observed)
  - Goal: summarization
Evaluation (partial observation)

“train” on observed (60% of possible), evaluate on possible

Near perfect F-score (1 FN, 0 FP)
9/10 actual intents correctly inferred
Concluding remarks

• A new approach towards bridging the semantic gap
• Anime, a framework to express network behavior and infer intents
  • Fits the hierarchical nature of networks
  • Enables application of ML-toolbox to network intent inference
• Prototype produces (objectively) high-quality results
  • Acceptable performance

• Future
  • Incorporating user feedback
  • Automatic anomaly detection
  • User study with real-world network operators

• Interested?
  • Let’s get in touch kheradm2@illinois.edu
Backup slides
Related work

• Network behavior summarization
  • Net2Text [Birkner et al, NSDI18]

• Network invariant inference
  • Network analysis [US patent 15/860,558]
  • Config2Spec [Birkner et al, NSDI20]

### Anime:

snapshot: 1, path: \(X.A.Y\)
snapshot: 2, path: \(X.B.Y\)

snapshot: Any, path: \(X\{A,B\}.Y\)
Example 2


**time: Any**, failed links: [0-1], dstIP: 128.174.0.0/16, path: AS1.R1.**Internal**+.R5.AS2