Complexity Framework Discussion

Michael Behringer (ed)

29 July 2013

draft-irtf-ncrg-complexity-framework-00

Summary

- Each network has N design criteria / metrics

  Explicit: Cost, bandwidth, delay, ...

  Implicit: Extensibility, de-bug-ability, ...

- We see those criteria are axes in an N-dimensional graph

  Each network can be mapped into this space

- There are tradeoffs: You can't optimise all axes

Tradeoffs and Complexity

cost

Adding gratuitous complexity

Solution

optimization

Curve of optimal complexity

scale
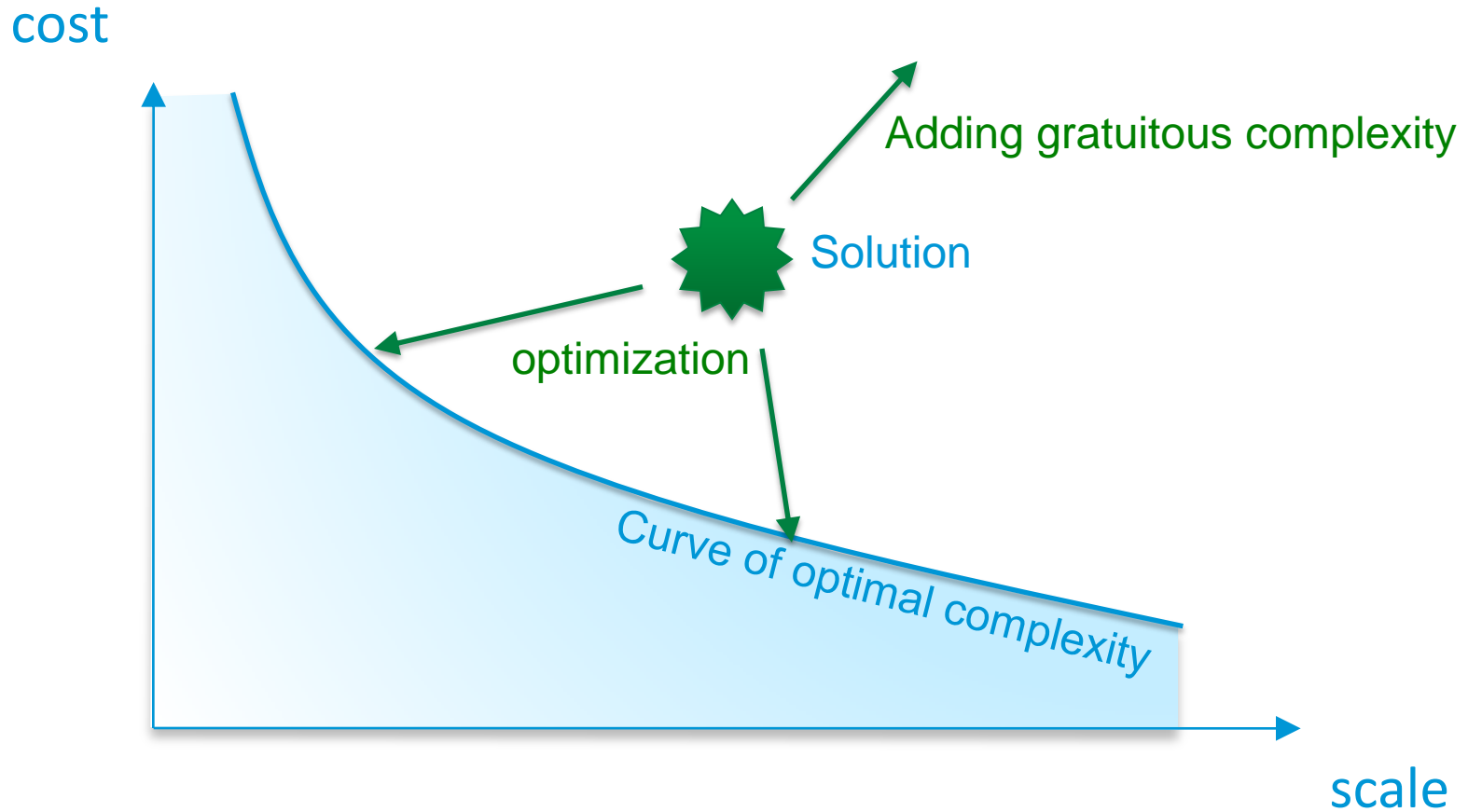
Source: John Doyle

Obvious Metrics

- Cost

  How much does the network cost to build (capex) and run (opex)

- Bandwidth / delay / jitter

  Traffic characteristics between two points (average, max, ...)

- Configuration complexity

  How hard to configure and maintain the configuration

- Susceptibility to Denial-of-Service

  How easy is it to attack the service

- Security (confidentiality / integrity)

  How easy is it to sniff / modify / insert the data flow

- Scalability

  To what size can I grow the network / service

Other Metrics

- Extensibility

  Can I use the network for other services in the future?

  Positive example: IP

  Negative example: Traditional telephony

- Ease of troubleshooting

  How hard is it to find and correct problems?

  Negative example: Manually configured IPsec overlay networks

  Positive example: Dynamic IPsec overlay networks

- Predictability

  If I change a parameter, what will happen?

  Negative example: Configuration

- Clean failure

  When a problem arises, does the root cause lead to deterministic failure

  Negative example: Coax Ethernet; browser certificate problems

  Positive example:

Metrics from draft-retana-network-complexity-framework-00

- Control Plane State verses Optimal Forwarding Paths (or it's opposite measure, stretch)

- Configuration State verses Failure Domain Separation

- Policy Centralization verses Optimal Policy Application

- Configuration State verses Per Hop Forwarding Optimization

- Reactivity verses Stability

Ideas / Discussions

- Include text to describe the various forms (?) of complexity

  Operational complexity

  deployment complexity; configuration complexity, trouble shooting

  Network system complexity

  s/w complexity; h/w complexity; Protocol / algortihm complexity

  Management system complexity

  OSS systems, etc.

- Intrinsic complexity of the system vs complexity from the user base?

- Include the various scales of metrics

  Illustrate trade-offs between various scales

New Outline – Proposal

- What constitutes complexity?

- Where is it?

- What does it depend on?

- How is it managed?

- How do users interact?

New Outline – Proposal

- What constitutes complexity?

    State

    config / protocol state / OS state

    Churn

    Algorithms

    packet exchanges / state machine complexity / …

- Where is it?

- What does it depend on?

- How is it managed?

- How do users interact?

New Outline – Proposal

- What constitutes complexity?

- Where is it?

  Topology

  various network elements / NMS system / AAA server / …

  Layer

  Layer 1, 2, 3, 4, app layer, …

- What does it depend on?

- How is it managed?

- How do users interact?

New Outline – Proposal

- What constitutes complexity?

- Where is it?

- What does it depend on?

>>> local depencencies

>>> network wide

>>> network external

- How is it managed?

- How do users interact?

New Outline – Proposal

- What constitutes complexity?

- Where is it?

- What does it depend on?

- How is it managed?

  Configuration

  Troubleshooting

  Monitoring

  System Integration

- How do users interact?

New Outline – Proposal

- What constitutes complexity?

- Where is it?

- What does it depend on?

- How is it managed?

- How do users interact?

  Open loop control

  Local configuration (PC)

Mapping Old Draft → New Draft

1. Introduction . . . .
2. Current Understanding of Network Complexity
  2.1. The Behavior of a Complex Network
  2.2. Robust Yet Fragile . . . .
  2.3. The Complexity Cube . . . . .
3. Towards Defining Network Complexity .
  3.1. General Observations . . . . .
  3.2. The Problem Space . .
  3.3. Technical Debt . .
  3.4. Layering considerations .
4. Possible Directions of Research
  4.1. Definitions and Metrics
  4.2. Comparative Analysis
  4.3. Containment, Control or Reduction of Complexity
  4.4. Use Cases

new

1. Introduction . . . .
2. General Considerations.
3. Structural complexity
4. Components of complexity
  state (config/protocol/OS)
  churn
  algorithms
5. Location
  topology
  layer
6. Dependencies
7. Management interaction
  configuration
  troubleshooting
  monitoring
  system integration
8. User interaction
9. Examples / how to use this doc.