

Management Breakout

High Order Bits

- Need a federated approach
 - I.e., consider both local and global tasks
- Want to automate as much as possible
 - Ideally, everything after setting initial policy
- Need a measurement component
- Need to include management considerations across all of our new technologies

High Order Bits (cont.)

- Need some linkage between apps and the underlying delivery mechanisms
- Evolvability is important

What is Management?

- FCAPS
 - Fault detection
 - Configuration
 - Accounting
 - (Accountability)
 - Performance
 - Security

Fault Management

- Need to first figure out when something is wrong
- Also involves
 - performance management
 - configuration management
- Critical piece:
 - some way to indicate a desire to get data and have that data flow to me
 - e.g., ~pub/sub

F (cont.)

- Control / signaling plane
- Information recorded / used needs to be tailored to different situations
- Need to take organizational boundaries into account
 - perhaps a hierarchy for information flow

F (cont.)

- Control plane vs. data plane
 - perhaps a detail; just want information
 - perhaps out-of-band is key
 - could use virtualization to have multiple paths for data and control

F (cont.)

- Lying
 - reputation system?
 - need to get caught
 - detection (with some probability) is key
 - do we need to design for corroboration of information?
 - has to do with what the threat model is

F (cont.)

- Where are the boundaries?
 - one boundary?
 - vs. bunches

F (cont.)

- Granularity
 - who is asking?
 - different granularity depending on the context

F (cont.)

- If there is a fault we should converge to some sort of stability
- Want to understand where the system can recover itself and where it can not
 - Where are the limits
 - Multiple parts of the system can react to the same problem; need communication

F (cont.)

- Is it management's function to fix all the ills caused by the rest of the parts of the system?
- What is the timescale of management?
 - minutes to tens of minutes?
- “Self management” is a useful touchstone
- Timescale + information needed + actions required

Configuration

- We need some notion about dealing with configuration errors
 - each component might be working, but end-to-end services might not be working
 - diagnosis -> some fix
- No manual device-level configuration
 - need humans to articulate policy
 - need to design protocols that are amenable to high level policy descriptions

C (cont.)

- How do we change networks without disrupting operations?
- Configurations should be useful to debugging
 - The network itself should not be the configuration database

Accountability

- Need to have some sort of notion of identity to understand who is responsible
 - sometimes it is easy; e.g., person at the other end of the wire
 - often more difficult
 - tough when some actor wants to be anonymous
- What?
- The business model plays a part

A (cont.)

- Information gathering for accounting is also useful in other management contexts
- Perhaps we don't care as much about who has data, but rather how it is handled and used
- Trust plays a part
 - ways to corroborate useful
 - e.g., that I am being billed correctly for my usage

A (cont.)

- Management system needs to be lightweight
- Cant be more costly to run than the network itself

Performance

- Not too different from fault management
 - instead of black or white it is grey
- Need to make sure the information being collected is appropriate for performance monitoring
- volume of data could be an issue
 - stresses scalability

Security

- Two problems:
 - securing the overall system
 - securing the management system

S (cont.)

- Need to make sure there is reliability and availability
- Need to make sure the infrastructure is protected
 - How do we protect the control traffic? E.g., ensure only authorized folks can configure.
- Need security policy for each service you run
 - Consistency across policies

S (cont.)

- Second alternative
- Aspects of security
 - Authentication
 - Integrity
 - Privacy
- Consider implications on each aspect of management
 - E.g., may need privacy in some cases and not in others

S (cont.)

- What are the threats?
- How much do I care about the threats?
- How open is the system?

- Again, what about liars?
- What can we say about how things work in the face of a very aggressive adversary?

S (cont.)

- Challenge: need to consider federations and cross-organizational policies
- Policies can conflict
 - System might not work in face of the conflict
 - Need negotiation
 - (Not specific to security, but crosses into other areas of management)

S (cont.)

- What are the right aggregations of data to share across organizations?
- understanding where the limits of sharing and aggregation are
- Different security requirements for different applications

S (cont.)

- How much security will laws allow to be shared and how?
- What is the cost of the security mechanisms?
How does that impact usability?

Wondering

- How can we eliminate redundancy?
- Need a framework that allows one to compose tools
- Perhaps, a programming language for combining these things

Dear Ellen

- How management schemes scales is key
- Policy is important and is federated
 - GENI should have some notion of being federated
- Every device in GENI should have something next to it to provide computation and storage
 - Allows for experimentation
- Need enterprise networks

Dear Ellen (cont.)

- Allow us to try out automatic configuration
 - And, let us play with that
- GENI should be able to be easily rolled back to previous state
- GENI itself be well managed and extensible
 - Might be hard so let people play

Dear Ellen (cont.)

- More people should be able to watch how actual management is done
 - Expose students to the real problems
- Should expose students to reality
 - “Why all we taught you is useless”

Non-Technical Issues

- What are the political requirements on management?
- Much cross-organizational cooperation required
- What needs to be collected to enforce accountability
- How do we allow the system to accommodate social networks that may form?

Related Areas

- Game theory and cryptography
 - multiple administrative entities cooperating to diagnose a problem without divulging sensitive information
- Control theory
 - collecting information from many places, with time delays, and tuning the system in a way that remains stable

Related Areas (cont.)

- Streaming algorithms
 - data-reduction techniques (e.g., sampling, hashing, sketches, etc.) for collecting measurement data at high speed
- Optimization theory
 - setting the tunable parameters in a protocol/system to meet a target objective; deriving new protocols from scratch that achieve precisely stated objectives (and proving that such protocols are stable, optimal, etc.)

Related Areas (cont.)

- Statistics, inference
 - tomography problems to infer network properties (e.g., link loss rates, or traffic matrix) from incomplete or sampled measurement data
- Machine learning
 - identifying previously unknown relationships in measurement data
- Programming languages
 - specifying policies

Related Areas (cont.)

- Logic
 - reasoning about behavior across layers/ protocols/systems, specifying protocols themselves
- Databases
 - for storing, slicing-and-dicing, and managing measurement data, configuration data, etc.
- Lots of venues focused on network management

Going Forward

- Seems like an awful lot of requirements
- Can we design something to accomodate them all?
- Tradeoff between what we might be able to do vs. the cost
- Goes to what we consider management
- Want to get hooks into the other components of the systems
 - What to do with the hooks in a coherent fashion?

Next Steps

- One approach: wait to see what architectures are developed
- Another meeting with more structure?
 - People tasked with particular deliverables
- Perhaps a strawman on what is necessary for network management