Purpose of the draft

• “Intent-Defined Networking” is the latest buzzword
  • Basic idea: Define what you want, not how to get it
  • This sounds good, but is this idea really new? (rhetorical question)

• What is intent, and how does it differ from what came before?
  • Policy-based management: Define high-level policies, leave it to policy renderers to do the rest
  • Service models and service provisioning: Define services, mapping to low-level configurations and objects left to a system

• Is Intent a reincarnation of policy? Are they synonymous? Do the mean different things? Why all those terms and how do they relate?

• Compare confusion with “data models” and “information models”

This Draft
Intent vs Policy vs Service Models

• Policy:
  • “Policies are rules governing the choices in behavior of a system” (Morris Sloman 1994)
  • “Policy is a set of rules that are used to manage and control the changing and/or maintaining of the state of one or more managed objects” (John Strassner 2003)
  • “A definite goal, course or method of action to guide and determine present and future decisions”. “A set of rules to administer, manage, and control access to network resources”. (RFC 3198)

• Intent: RFC 7575: “An abstract, high-level policy used to operate the network”

• Service Models: A model that represents a service that is provided by a network to a user. (And: which service to provide to a user is a form of intent)

• From these definitions, “intent” and “policy” are practically synonymous (and a “service” may just be another type of intent, and hence policy)
Key concepts in modeling that may be useful to explain this

**Abstraction**
- Hide irrelevant details
- Flesh out “concept”
- Data normalization

A concept
Implementation independent
Omits irrelevant details

**Information hierarchies**
- Hierarchy of higher-level model abstractions
- Build on top of and combine + aggregate lower-level model abstractions into new concepts
- Typically, from fine grained to coarser grained

End-to-end service
Svc Order params
(from OSS, outside network)

Network
Allocate

Device/Network Element
Compose + aggregate

Policy and intent are here – “renderers”

Service models are here – “controllers”

Compare TMN

A set of low-level details
e.g. a specific data model, CLI, etc

rendering
Other aspects worth mentioning

• Declarative vs Procedural
  • Declarative: Desired outcome, goals, post-conditions
  • Procedural: (programmed) workflow
  • Some middle ground (e.g. rule-based)

• Rendering:
  • Predetermined: well-defined rendering rules, deterministic, “programmed”
  • Dynamic: obtained e.g. via control loops, negotiation and planning, trial-and-error, ...
  • Centralized vs Distributed (control hierarchy) vs. Decentralized (e.g. peer-to-peer, autonomic)

• What about intent-based management and policy-based management and the resulting frameworks
  • Mouli + Kaarthik draft alludes to this
  • Intent = SDN Controller API? Policy = Device Automation? Service model = non-SDN services?
  • Combine this into one big “treatise” on the subject, or have narrower drafts?
A second attempt at terminology

• Why not try to distinguish related but different concepts when we have different terms? Define things a little more narrowly for a clearer distinction:

• **Intent**: A high-level operational goal for a network or a service, whose precise mapping into lower-level parameters may be non-deterministic and unknown
  *(Abstraction of operational goals for network operators, focusing on network layer concepts)*
  Note: this makes IDS a candidate for cognitive management and distributed algorithms

• **Policy**: An abstracted rule of what to do (obligation policy) or what to permit (permission policy), given a set of well-defined events, conditions, and actions
  *(Focus on common abstractions and declarative rules, can apply at each layer of hierarchy)*
  Note: this makes PBM essentially rule-based systems

• **Service model**: A higher-level model abstraction, representing services provided to an end user, with a set of component resources and well-defined dependencies on network and device model abstractions.
  (and **Network model**: A model abstraction, representing a network as a whole and holistic cross-device concepts (such as paths or connections), with well-defined dependencies on device model abstractions)
  Note: This includes service function chains. It is what many controllers fundamentally provide
Structure of the draft

• Introduction

• Explanations of concepts on their own (building on existing definitions)

• Distinction between them (with examples) with refinements of terms as needed for an IRTF/NMRG definition of terms

• Is this useful – to what extent do those distinctions matter?

• If so, how do we gain broad consensus?
Backup
Examples

• Intent:
  • “Optimize my network for energy efficiency”
  • “Prioritize higher levels of service over resource consumption”
  • “Ensure that gold level users receive the highest quality-of-experience of all users”

Q: Does intent involve quantities? If there are quantities, where do they come from?

• Policy:
  • “When resource utilization falls below 20%, migrate workload and shut down server”

• Service Model:
  • “John gets residential internet access with 100 Mbps down 10 Mbps up”