Automation in the Days of the Software Network

Diego R. Lopez
Telefonica I+D

IEEE ETR-RT. Masala, August 2019
The Automation Target

- The main raison d'être for all efforts around Software Networks
  - Elasticity
  - Homogeneity
  - Programmability
  - Abstraction
- In a changing network landscape
  - Pervasive encryption
  - Internet stack evolution
  - And the advent of 5G
The Complexity Challenge

- Networks becoming increasingly complex
  - 5G foresees a x10 densification of sites compared to 4G
  - Best user experience demands heterogeneity in access technologies
  - Multiple coexisting RATs (2G/3G/4G and 5G), creating interoperability challenges
- And not suitable to be managed using traditional operation
  - Virtualization
  - Efficient orchestration
  - Towards zero-touch service management
  - Adapting results from the IT experience
  - And exploring new paths
The Ossified Network

- Many complex functions
  - Baked into the infrastructure
- Pervasive standardization
  - Much beyond a lingo for interoperability
  - Limit differentiation
- An industry with a “mainframe-mentality”
  - Reluctant to change
  - As a whole
  - A path of lowest resistance for all actors
The Momentum of Software Principles

- Software Networks allow for applying IT and software engineering principles to networks
  - Compositional mechanisms
  - Model-driven development and management
  - DevOps and continuous integration
  - Integrated environments
  - Cognitive methods
  - . . .
The Role of Virtualization

- **NFV**: Separate functionality from capacity
  - Increase network elasticity
  - Address heterogeneity

- **SDN**: Decouple the control and forwarding functions
  - Gain programmability
  - Abstract infrastructure
Design Principles for an Evolutionary Approach

- **Elasticity**
  - Automated deployment and management
  - Evolve towards Zero Touch and E2E network parametrization

- **Common shared infrastructure**
  - Telco grade
  - Agile and open to innovation

- **Cloud nativeness**
  - Lightweight and highly distributed
  - As-a-service
    - Infrastructure
    - Transport
    - Platform
Model-Based Network Orchestration and Management

LOCAL DEVELOPMENT & TESTING
- Open development environment
- Functional tests
- Low cost
- Integration from the beginning

TEST POOL FOR DEVELOPERS
- Real servers and switches
- Performance tests (EPA can be enforced)
- Cost-effective shared infrastructure
- Move the value to VNF services

SERVICE PROVIDER
- Production/pre-production environment
- Real network scenarios
- Final service configuration
- Fast deployment
- Low final integration cost

Applying network-aware CI/CD principles
- Development and testing
- Deployment description and sharing
The Integration Goal

- Tenant controllers define the forwarding rules for VNFs
  - Service chaining
  - Identity-based services
  - On-demand provisioning
  - OSS integration
  - BSS becomes a SDN application
- Several tenant controllers may interact with the same infrastructure controller
  - Conflict resolution has to be addressed
  - Tenant identity must be exchanged to apply policies
- Support for network slicing
  - Relaying on SDN composition
Micro-Serviced NFV

- Oblivious components
  - Supported by connecting tiers of nodes
  - And platforms
- Requiring
  - Node discovery
  - Load-balancing
  - Retry on failure
  - Overload detection
  - Security
- Ongoing work
  - Service meshes: Istio, NSM…
  - More radical approaches based on RINA

- A few challenges
- Topology awareness
  - Within the service
  - At the attachment points
- The conservation principle
  - Data plane performance
- Openness
  - Functional and operational
- Integrity and auditability
  - Trusted paths
- Isolation
  - Beware the noisy neighbors
The Promise of Programmable Planes

• A generalized approach to virtualization
  o Homogeneous (not unique) infrastructure
  o Full programmability
  o Infrastructure offloading

• Addressing the performance issue
  o Supporting integral orchestration
  o Incorporate whitebox approaches

• A convergence path
  o Manage software images
  o Connect with intent approaches
The Essential Closed Loop

- Not a radical change
  - AI as a tool to improve policy enforcement
  - Apply extended capabilities, but do not expect Skynet

- The key issues are not in the engine(s)
  - But in the data and action flows
  - Including distribution and placement of the engine(s)
At Any Layer and Segment
The Data Stream

- No matter how intelligent: Crap in means crap out
  - Usable: Adaptation (formats, scales…)
  - Sufficient: Topology (sources, aggregators…)
  - Safe: Provenance (origin, timestamps…)
  - Steady: Continuity (pace, availability…)
- Not just data
  - Metadata becomes essential, including semantic mappings
  - What seems to claim for a data stream ontology
  - Not that far away: data modeling is a first step
- An enhanced data fabric seems the logical approach
  - Supporting resource, orchestration and function sources
  - Combining current network monitoring tools and recent telemetry developments
The Action Stream

- OAM actions at a wide variety of different domains
  - Challenging, given the current state-of-the-art
- Initial strategies
  - Domain specific
  - Recommendation systems
  - Autonomic protocols
- Capability models
  - Reusable functionality description
  - Abstractions of network element functionalities usable as building blocks
  - Combined to provide more powerful features
  - Registration mechanisms to support CI/CD
  - Inter-domain collaboration for E2E management
The Human in the Loop

• The dialectic way
  o Thesis: Translate intent into action
    ▪ Understanding intent statements
    ▪ Mapping onto technologies
  o Antithesis: Support environment constraints
    ▪ Policies provided by network management
    ▪ The archetypal SLA enforcement
  o Synthesis: Conflict resolution
    ▪ Among action requests
    ▪ And with management constraints

• Audit track and intelligibility
  o The who, the what, the when
  o And the why
  o Not only as a scapegoat

• And security
  o Deal with adversarial AIs
  o And consider circuit breakers
Trustworthy Datasets

• Lack of usable datasets
  o For training or validation
  o Data as an asset
  o Privacy concerns
  o None or limited tagging

• Generation of synthetic datasets
  o Traffic samples generated in a controlled way
  o Configurable mixes of synthetic and real traffic

• And metadata management
  o Different scenarios, from high loads to security threats
  o Training and validation loops

• Relying on Software Network principles
  o Repeatability and reproducibility
  o Controlled variations
Making Serious Science (and Engineering)

- Independent verification and *reproducibility* are essential to the scientific method
  - “Non-reproducible single occurrences are of no significance to science” (K. Popper)
- Complicated in many cases because different reasons
  - Ethical
  - Nature of the research field
- Recent computing and network results
  - Complexity
  - Disparate conditions
- Corroboration
  - Avoid (un)intentional idiosyncratic results
  - Repeatable results
- Transparency
  - Avoid (un)intentional biases in environment and measurements
  - Repeatable methods
- Robustness
  - Avoid (un)intentional best-of-breed results
  - Repeatable causes
Fitting Pegs and Holes

- Automation as the gist of Software Network technologies
  - Additional degrees of freedom
  - Smoother migration paths
- Elements required for application and interoperability
  - Framework components
  - Stream protocols, APIs and models
  - Topology and knowledge sharing
  - Multi-language, multi-solution, multi-thread...
  - The human in the loop
- We are talking standardization, one way or another
  - Specifications in documents
  - Reference implementations
  - Slowness is in consensus formation
- Avoid re-ossification
  - Keep it ductile/moldable/plastic/elastic/…
HAVE YOU TRIED NOT BEING BORING?

GOOD IDEA. I'LL MAKE FIFTY SLIDES OF PURE EXCITEMENT.