TITLE
SDN and The Engineering Team of the Future *(also known as Disruptive Technologies and the new Breed of Nerd)*

PRESENTER
Richard Nelson (University of Waikato) and TBC (REANNZ)

AFFILIATION
University of Waikato, REANNZ (Research and Education Advanced Network of NZ)

FIVE KEYWORDS
Faucet, Engineering, Software, Systems, Disruption SDN

AUTHORS NAMES (if different from the presenter)
Richard Nelson (University of Waikato) and Angela Nash (REANNZ)

AUTHORS AFFILIATIONS
University of Waikato, REANNZ (Research and Education Advanced Network of NZ)

PRESENTERS BIOGRAPHY

REANNZ TBA*

PRESENTATION DESCRIPTION
It is not news that Software Defined Networking, along with other disruptive technologies is rapidly changing how we build and manage our networks. What is starting to become apparent is how this is affecting what you need your technical teams to do and what they need to know. This talk will explore how the world of SDN is rapidly evolving to create tangible, real-world applications, such as Faucet, that are increasingly more usable and easy to implement.

Faucet is an OpenFlow controller for multi-table OpenFlow 1.3 switches, that implements layer 2 switching, VLANs, ACLs, and layer 3 IPv4 and IPv6 routing, static and via BGP. The Openflow switch is deployed as a drop-in replacement for a L2/L3 switch in the network to enable extra SDN based functionality.

Faucet is a project from REANNZ, the University of Waikato and Google to build an enterprise style SDN controller. It aims to be as simple as possible where simple means easy to deploy, operate and support as well as working on the code base. Nevertheless it aims to be fully featured.

As part of it's simplicity, Faucet includes no driver code, it uses vanilla multi-table OF 1.3. To ensure this it has a comprehensive integration test suite. OpenVSwitch is used as a reference implementation and hardware switches will work with Faucet if they also pass the test suite. This means that buying compatible hardware requires no reading of specs or features but simple deployment of an automated process. To date four vendors have supplied hardware to pass the tests.

Faucet uses a simple YAML configuration file and is instrumented for monitoring through Prometheus and Grafana. It can easily be automated with common orchestration packages. Future versions will incorporate Google’s gNMI and gNOI for platform configuration and certificate management. This will enable true Zero-Touch network operation.

The development of Faucet follows Dev-Ops principles. Releases are regular - normally every Monday. Faucet upgrades without any pause in network function.

So what does an automated SDN network look like?

- It can be multi-vendor. Our network has equipment from Allied Telesis, HP Enterprise (Aruba) and Cisco as well as using OVS. It interoperates with the standard networks that use IP and Ethernet VLANs.
- It is completely managed using automated tools rather than SSH. We use Ansible for every network element. Updates are Git commits. Review and rollback are standard git features. Different ansible inventories separate staging from production so we can test features on a staging network, then validate configuration on a canary network then Push-on-green to deploy to the production network.
- It supports very fine grained security. ACLs can be deployed against ports or run intra-VLAN as well as between them or on routed interfaces. New ACLs can be deployed at any time very simply. Various security projects will automate this process in response to alerts. We can also filter and mirror any traffic and direct it for inspection in hardware so we don't need to inspect entire links.
- It is well monitored using Gauge - Faucet's companion monitoring application or Prometheus to grab instrumented variables direct from Faucet.

This has been achieved with a small network for a research group and a network administrator who happens to be a Faucet developer. So the next question is how to scale this to a team and an enterprise and eventually a national backbone?

The on-flow effect of these new technologies is that simply being able to understand infrastructure, networks, software, systems and operations as a speciality is no longer enough.

There is a revolution on the horizon, there is a new breed of nerd needed….. one that knows a how to code *AND* build a network. The business needs are evolving, the way we deliver to them are evolving and your people need to evolve to meet these new challenges and skillsets.

How do you build a team for this? How do you ensure that you have sufficient depth in your engineers and software developers? Traditional training and team structures tend to separate these activities and manage them within different business areas thereby creating very real walls between the teams you need to collaborate with. How can you begin to manage this transition, set your teams up for success and create value both for them and the business?

REANNZ not only utilises Faucet internally but they also have a very unusual engineering team structure that is successfully straddling the current needs of the business while setting themselves up to create and embed the engineer of the future. There are a few key things they are looking at to help them create a world-leading team and approach:

● People	(What’s in it for them)
● Business	(What’s in it for us)
● Dynamic	Resourcing (how to resource with a small team to deliver broad services)
● Training	(How to best use your experts to support each other)
● Community	(Change is coming, how can you help support it)