

# Slicing & IOT

Jari Arkko  
Ericsson Research  
IETF 99, July 2017

# Use Cases



Mass deployment of sensors



Critical services



Control of remote equipment



Factories




Networked traffic systems

# Useful Access Network Properties from an Internet of Things Perspective

- Separation
- Additional control
- Scaling up or down
- Flexibility

# Useful Access Network Properties from an Internet of Things Perspective

- Separation
- Additional control
- Scaling up or down
- Flexibility

- 
- Traffic separation
  - Resource guarantees

# Useful Access Network Properties from an Internet of Things Perspective

- Separation

- Traffic separation
- Resource guarantees

- Additional control

- E.g., for low-latency cases
- Topology control, e.g., mobility endpoint
- Detnet-style reservations

- Scaling up or down

- Flexibility

# Useful Access Network Properties from an Internet of Things Perspective

- Separation

- Traffic separation
- Resource guarantees

- Additional control

- E.g., for low-latency cases
- Topology control, e.g., mobility endpoint
- Detnet-style reservations

- Scaling up or down

- Flexibility

- E.g., run your own 5G network inside a factory

# Useful Access Network Properties from an Internet of Things Perspective

- Separation

- Traffic separation
- Resource guarantees

- Additional control

- E.g., for low-latency cases
- Topology control, e.g., mobility endpoint
- Detnet-style reservations

- Scaling up or down

- Flexibility

- E.g., run your own 5G network inside a factory

- Tailor services to the situation
- Allow evolution (features, sw)

# What is Network Slicing?

- One possible minimal definition:
- “Network slicing is network management pertaining to treating different traffic categories in separate virtual networks, with independent resource, technology, and topology choices”



# Observations

- What's needed? Much is doable with existing tools: network virtualisation, management tools, SDN, service chaining, ...
- Are there missing ones? Unclear... can always do more
- Learn to walk before running (short timelines, incremental process?)
- Be careful with assumptions of silos... difficulties of deploying QoS... and keep a system perspective (see draft-arkko-arch-low-latency)

