Strategies for Network Resilience: Capitalising on Policies

Paul Smith, Alberto Schaeffer-Filho,
Azman Ali, Andreas Mauthe and David Hutchison
Lancaster University

Marcus Schoeller
NEC Laboratories Europe, Heidelberg, Germany

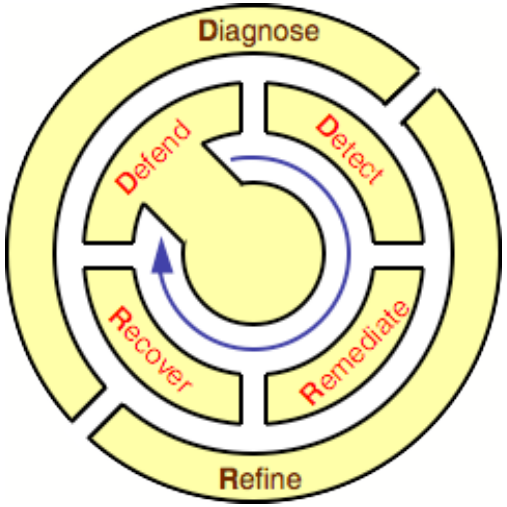
Nizar Kheir
France Telecom R&D Caen

Presented at the:
4th International Conference on Autonomous Infrastructure,
Management and Security (AIMS 2010)

Multi-Service Networks,
Cosener's House, July 8th 2010

Background

- To embed resilience into the future Internet
  - Conceptual framework
  - Mechanisms and algorithms
    - **Network resilience**
    - **Service resilience**
  - Experimentation in testbeds
- Network security and resilience framework: D^2R^2 + DR
  - Real-time control-loop (D^2R^2)
    - **Defend** against challenges to normal operation
    - **Detect** when adverse event occurs
    - **Remediated** the effects of adverse event
    - **Recover** to original normal operation
  - Offline control-loop (DR)
    - **Diagnose** what caused the challenge
    - **Refine** operation to prevent it from happening again



Motivation

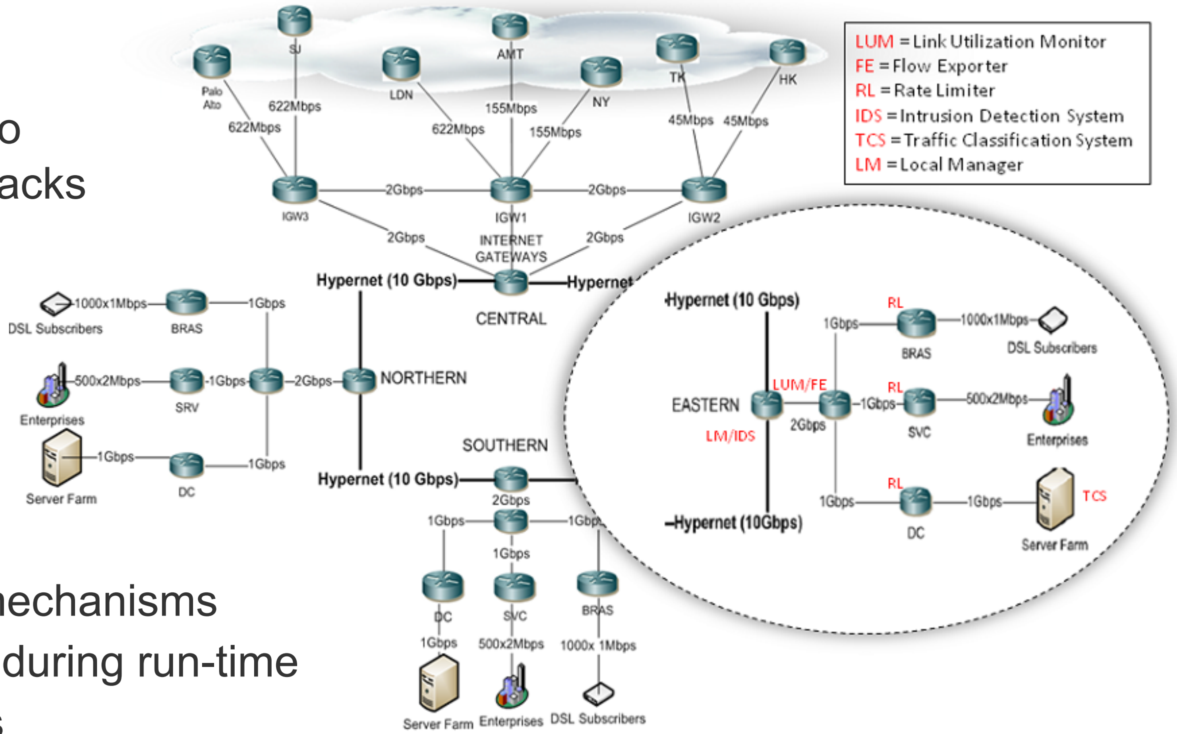
- Configuration criteria change over time
  - Requirements (e.g. SLAs)
  - Operation context (e.g. battery power, node churn)
  - Challenges (e.g. component faults, new types of attacks)
- Resilience strategy must be de-coupled from the mechanisms that implement it
- Difficulties in defining resilience configurations
  - Deriving configurations from high-level requirements
  - Identifying and resolving conflicting configurations
  - Learning resilience behaviour
- How policies can assist the specification of strategies for network resilience

Policy-Based Resilience Strategy

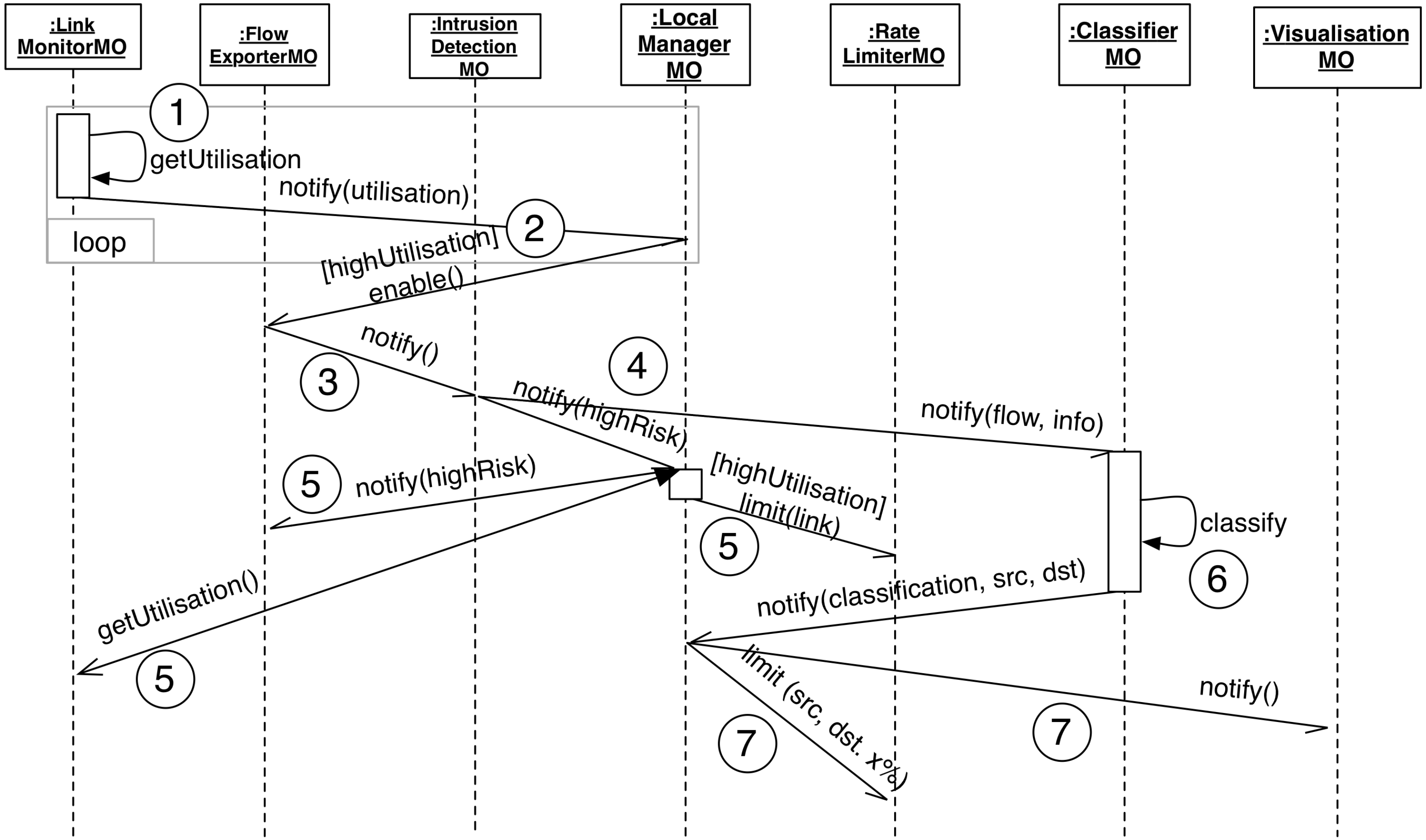
- Case study on “network high-traffic volume”
  - Mechanisms must co-operatively enforce the resilience of the network
  - Includes: flow exporter, rate limiter, anomaly classifier

- Key problem
  - Discriminate overload due to legitimate requests from attacks (e.g. flash crowd vs. DDoS)
  - Apply countermeasures

- Policy-based resilience strategy
  - Configure and coordinate mechanisms
  - Modification of the strategy during run-time
  - Adding or removing policies



Policy-Based Resilience Strategy



Complexities in Defining Configurations

- Policy frameworks can assist in defining resilience strategies for multi-service networks

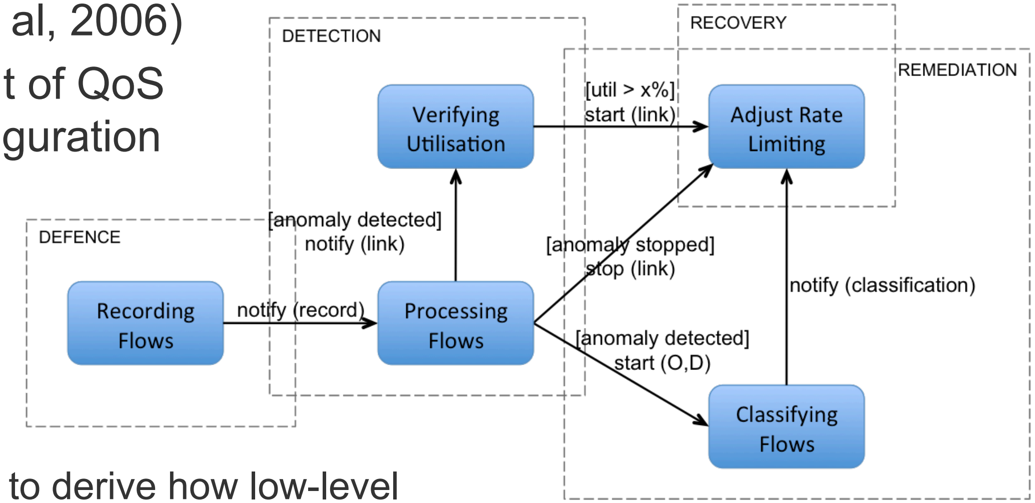
  ① **Deriving configurations from high-level requirements**
  ② **Identifying and resolving conflicting configurations**
  ③ **Learning resilience behaviour**

Complexities in Defining Configurations (1st)

Deriving configurations from high-level requirements

- Policies realise a high-level requirement to ensure resilience
  - E.g. in terms of the availability of a server farm and the services it provides
  - Complex scenarios would make deriving concrete policies by hand intractable
  - Derive implementable policy configurations from high-level specifications

- Policy refinement (Bandara et al, 2006)
  - Goal elaboration & refinement of QoS requirements into policy configuration



a) Transform high-level goals into more concrete ones, until they can be expressed as implementable operations

b) Use logical reasoning and abduction to derive how low-level operations need to be executed sequentially or in parallel

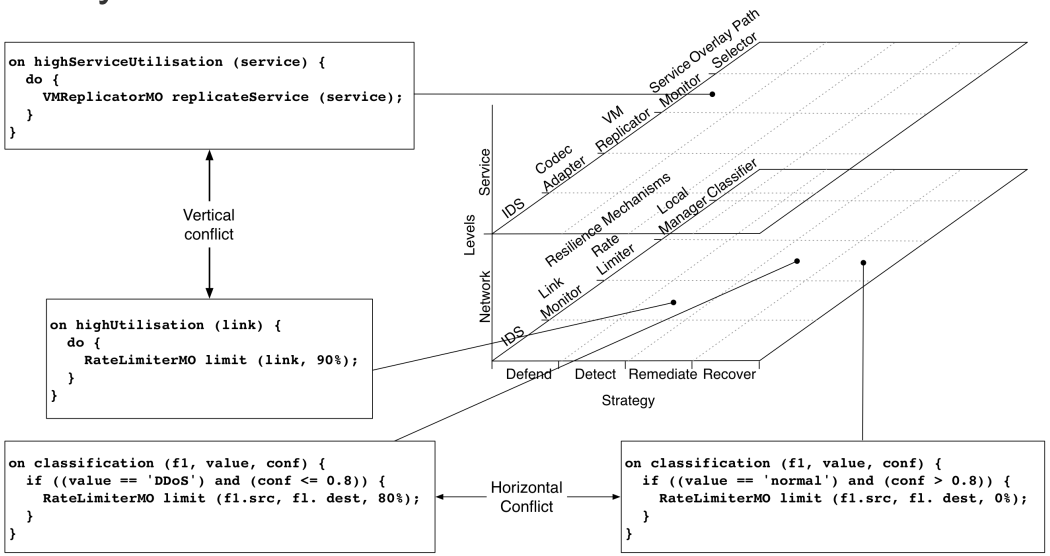
Complexities in Defining Configurations (2nd)

Identifying and resolving conflicting configurations

- Complex multi-service networks where conflicts can occur
  - Requirements of a set of services being met at the expense of another set
  - No requirements being met for any service

- Conflicting configurations

a) **Vertically, *across levels*:** in concurrent challenges - e.g. flash crowd and DDoS. Rate limiting started on routers (network) as well as replicating service during flash crowd (service)



```
on highServiceUtilisation (service) {
  do {
    VMReplicatorMO replicateService (service);
  }
}
```

```
on highUtilisation (link) {
  do {
    RateLimiterMO limit (link, 90%);
  }
}
```

```
on classification (f1, value, conf) {
  if ((value == 'DDoS') and (conf <= 0.8)) {
    RateLimiterMO limit (f1.src, f1. dest, 80%);
  }
}
```

```
on classification (f1, value, conf) {
  if ((value == 'normal') and (conf > 0.8)) {
    RateLimiterMO limit (f1.src, f1. dest, 0%);
  }
}
```

b) **Horizontally, *along the D^2R^2 strategy*:** detection mechanisms at the server farm may (wrongly) determine that node has ceased to behave maliciously, and initiate a recovery configuration

Complexities in Defining Configurations (3rd)

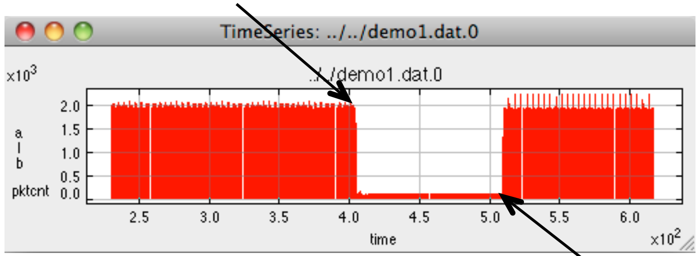
Learning resilience behaviour

- Resilience configurations will need to evolve over time
  - Attacks may change and new agreements may cause high-level priorities to shift
  - Strategy may prove to be sub-optimal or incorrect
- Background loop in the D^2R^2 + DR strategy: Diagnose and Refine
- Policy-based learning (Corapi et al, 2008)
  - Logical rules for knowledge representation and reasoning
  - Policies can be easily translated into a logical program
  - Allow user to understand (and correct) what has been learned
- Rules can be iteratively amended to represent better resilience practices based on how successful previous attempts were
  - E.g. during football final, high link utilisation is better remediated by replication of the server streaming the live match, rather than rate limiting link capacity

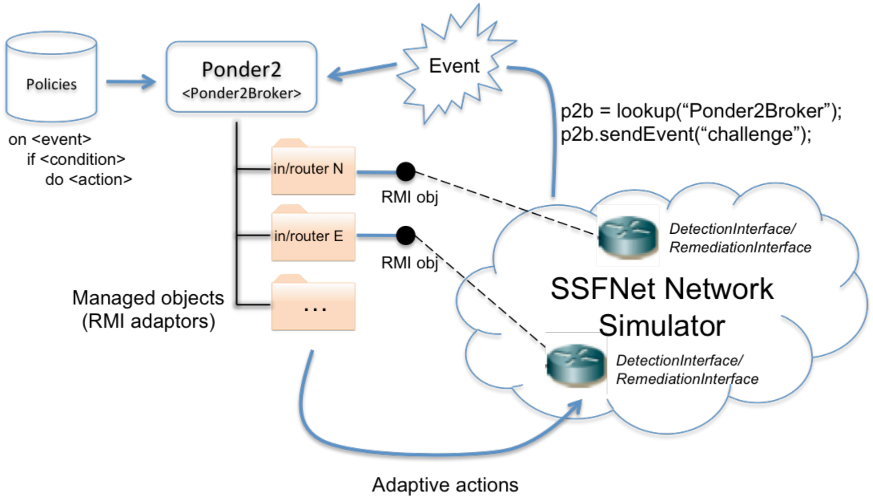
Implementation: Policy-based Network Simulator

- Basic idea
  - Combine network simulator and policy framework, and then use policies to adapt the behaviour of the simulation during run-time
    - **Implement different network topologies**
    - **Analyse different threat and anomaly scenarios**
    - **Implement different detection and remediation strategies**
- Current status
  - Evaluation of different toolsets: OMENet++, SSFNet, NS-3
  - Architectural Work
  - Preliminary testbed based on
    - **SSFNet and Ponder2**

start rate limiting



stop rate limiting



```
p2b = lookup("Ponder2Broker");
p2b.sendEvent("challenge");
```

Conclusion

- Network resilience is difficult to ensure
  - Configuration of systems is complex
  - Spans across several levels
  - Subject to a wide range of challenges
- D^2R^2 + DR strategy
  - Conceptual framework
  - Network- and service-level mechanisms
- Policies-based resilience provide flexibility in configuring components that implement this strategy
  - Changes in application requirements
  - Context changes
  - New types of challenge manifestation
- Policy-based approaches to make the problem more tractable

Strategies for Network Resilience: Capitalising on Policies

Alberto Schaeffer-Filho
asf@comp.lancs.ac.uk

Thank you

**Multi-Service Networks,
Cosener's House, July 8th 2010**

Policy-Based Resilience Strategy

```
on classification(fl,value,conf)
    if ((value == “DDoS”) and (conf < 0.4))
        do
        {
            VisualisationMO notify(alert(high));
            RateLimiterMO limit(fl.src,fl.dest,x%);
        }
    if ((value == “DDoS”) and (conf >= 0.4))
        do
        {
            VisualisationMO notify(alert(high));
            FirewallMO block(fl.src,fl.dest);
        }
```

ManagerMO policy, configure remediation based on root cause

Policies written in terms of the interface of managed objects

```
on lowRisk(link,src,dst)
    if ((list del(link,src,dst)) isEmpty(link))
        do
        {
            FlowExporterMO notify(lowRisk(link));
            RateLimiterMO limit(link, 100%);
        }
```

ManagerMO policy, configure recovery