Managed dependability in interacting systems

Poul E Heegaard, Bjarne E Helvik, Gianfranco Nencioni, Jonas Wäfler

Abstract A digital ICT infrastructure must be considered as a system of systems in itself, but also in interaction with other critical infrastructures such as water distributions, transportation (e.g. Intelligent Transport Systems), and Smart Power Grid control. These systems are characterised by self-organisation, autonomous subsystems, continuous evolution, scalability and sustainability, providing both economic and social value. Services delivered involve a chain of stakeholders that share the responsibility, providing robust and secure services with stable and good performance.

One crucial challenge for the different operation/control centers of the stakeholders is to manage dependability during normal operation, which may be characterised by many failures of minor consequence. In seeking to optimise the utilisation of the available resources with respect to dependability, new functionality is added with the intension to help assist in obtaining situational awareness, and for some parts enable autonomous operation. This new functionality adds complexity, such that the complexity of the (sub)systems and their operation will increase. As a consequence of adding a complex system to handle complexity, the frequency and severity of the consequences of such events may increase. Furthermore, as a side-effect of this, the preparedness will be reduced for restoration of services after a major event (that might involves several stakeholders), such as common software breakdown, security attacks, or natural disaster.

This chapter addresses the dependability challenges related to the above mentioned system changes. It is important to understand how *adding complexity to handle complexity* will influence the risks, both with respect to the consequences and the probabilities. In order to increase insight, a dependability modelling approach is taken, where the goal is to combine and extend the existing modelling approaches in a novel way. The objective is to quantify different strategies for management of de-

Poul E Heegaard, Bjarne E. Helvik, Gianfranco Nencioni, Jonas Wäfler
Norwegian University of Science and Technology, Department of Telematics, Trondheim, Norway,
e-mail: {firstname.lastname}@item.ntnu.no

pendability in interacting systems. Two comprehensive system examples are used to illustrate the approach. A Software Defined Networking example addresses the effect of moving control functionality from being distributed and embedded with the primary function, to be separated and (virtually) centralised. To demonstrate and discuss the consequences of adding more functionality both in the distributed entities serving the primary function, and centralised in the control centre, a Smart Grid system example is studied.

1 Introduction

The private and public ICT service-provisioning infrastructure has developed over many years into a complex system and its interactions with other critical infrastructure systems such as water distributions, transportation (e.g. Intelligent Transport Systems), and Smart Power Grid control have created diverse digital ecosystems. Digital ecosystems are characterised by self-organisation, autonomous subsystems, continuous evolution, scalability, and sustainability, providing both economic and social value. Services delivered involve a chain of stakeholders that share the responsibility, providing robust and secure services with stable and good performance.

This evolution has been evident for some time. In spite of this, and the crucial role of such systems, not much research is directed toward ensuring the dependability of the services provided by such ecosystem of systems. The objective of this chapter is to address some of the issues that arise when we seek to mange the dependability of systems.

1.1 Challenges

One crucial challenge for the different operation and control centres of the different systems is to manage the dependability in normal operation with many failures of minor consequence. In seeking to optimise the utilisation of the available resources with respect to dependability [1], the complexity of the (sub)systems and their operation will increase due to increased interconnectedness and complexity.

Some issues to take into consideration include:

- The public ICT services are the result of the cooperation between a huge number of markets actors. The overall system providing these services are not engineered, and there is no aggregate insight into their design and operation.
- There is no coordinated management to deal with issues involving several autonomous systems, in spite of such issues being a likely cause of extensive problems and outages.
- It is necessary to prepare for restoration of service after a major event such as common software breakdown, security attacks, or natural disasters. This prepa-

ration must include technical and operational as well as organisational and societal aspects.

An additional challenge is the management of dependability over multiple network domains, with uncoordinated operations in each of the different domains. As a potential side-effect of this, the preparedness for restoration of services after a major event (that might involve several stakeholders) such as common software breakdown, security attacks, or a natural disaster will be reduced. In addition, the frequency and consequences of such events may increase. More focus on exercises and use of the improved situational awareness provided by the new operational functionality, will to some extent reduce the negative side effect.

Ensuring the dependability of services based on an interacting relationship between independent stakeholders in the provision is typically agreed upon through Service Level Agreements (SLAs), which give guarantees on the non-functional properties of the services, including dependability aspects such as interval availability. These are important means to ensure the dependability of the services, but are insufficient to prevent and handle dependability problems across providers, as outlined above.

New functionality is added to enhance and improve operation and management of complex digital ecosystems. This is done to rationalise the operation, save money, simplify resource management, and maximise utilisation. It also enables more timely and precise knowledge and information about system state, facilitating timely (proactive) maintenance, and reducing the frequency and consequences of failures. The operational cost is reduced by reduction in manual labour through better and quicker detection and diagnostic mechanisms, and more autonomous self-repair. The objective is to shorten the recovery time and to reduce the failure frequency through better proactive maintenance. It should be kept in mind that this functionality targets the frequent (everyday) failures which are anticipated in the system design and normally of low consequence. However, this increased maintainability is achieved by the introduction of new, and partly centralised functionality, that increases the total complexity and creates an interdependent system [8]. These systems not only have additional failures and failure modes [12, 22], but they may also manifest a more fragile behaviour in critical situations [2, 18].

Figure 1 illustrates a risk curve, where the events with high “probability” have low consequence and the events with low “probability” have high consequence. The introduction of ICT-based support system, to operate an ICT system, or a critical infrastructure such as Smart Grid, is expected to reduce the consequences and probability of daily events. Less human resources are needed for the daily operations. However, due to the introduction of another ICT-based system, the complexity and interdependency in a system will increase, with the potential consequence of increased probability of critical events with extensive and long lasting consequences. Such events affect large parts of the system and a take long time to recover from because of lack of understanding of the complexity (“we have not seen this failure before”), or the lack of maintenance support and coordination between the different subsystems and domains in the digital ecosystem (“who should do what?”). As indicated in the figure, it is not only necessary to increase the focus and manpower

on the events with larger consequences, but also increase the competence of the operation personnel.

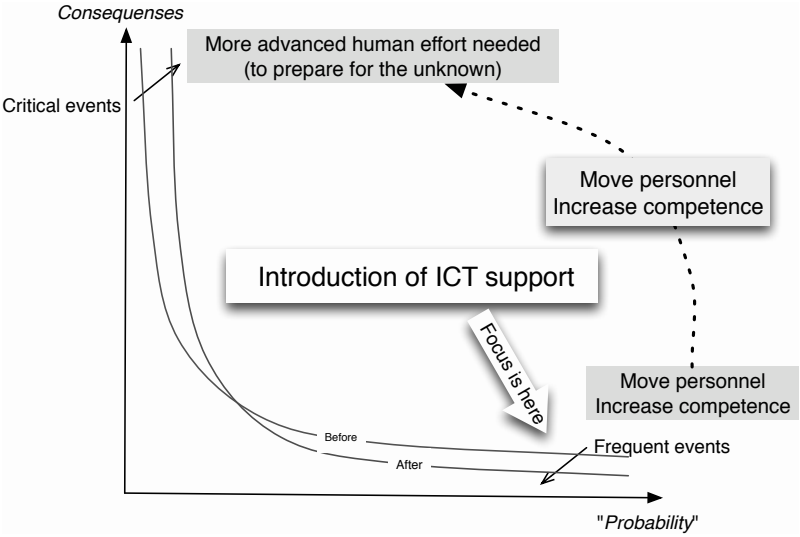


Fig. 1 Introducing ICT support to assist daily operations may increase the overall risk

There is a lack of theoretical foundation to control the societal and per service dependability of ICT infrastructure in the digital ecosystem. No foundation is established for optimisation, consolidated management, and provision of this infrastructure, neither from a public regulatory perspective, nor from the perspective of groups of autonomous (commercially) co-operating providers. A model of an ICT infrastructure must describe the structure and behaviour of the physical and logical information and network infrastructure, and include the services provided. Furthermore, through the modelling phases, it should be described how *resilience engineering* [9] can be applied to manage the robustness and survivability of the ICT infrastructure ecosystem.

1.2 Outline

This chapter describes the above mentioned challenges and outlines potential approaches to gain more insight into the risks. To increase the understanding and assess the risk (both consequences and probabilities), a holistic modelling approach is taken of service in systems of systems. The goal is to quantify different strategies for management of dependability in interacting systems. This should be addressed by different approaches:

- *System modelling*: Modelling of the functional interaction between embedded technical sub-systems in an ecosystem with multiple actors coordinated via business models only.
- *Management strategies*: Management and provisioning of (digital) ecosystems in a cost-efficient way, considering the trade-off between cost and quality.
- *Quantitative assessment*: Resource allocation optimisation (modelling, measurements, simulations) of robustness/dependability and performance in digital ecosystems.

Figure 2 illustrates that to improve the operation and management (O&M) of complex systems (e.g. in the Smart Grids), new control logic and functionality must be added and in some cases also be centralised (e.g. in Software Defined Networking (SDN), and by the introduction of network function virtualisation NFV in next generation communication networks). This needs to be modelled, and the system models parametrised to quantify the effect on the dependability and to identify potential changes and improvements that can be made in O&M. The reason is that the new and/or moved functionality poses new risks and threats to the systems, and may have potential undesired side-effects that need to be qualitatively assessed to again identify potential changes and improvements that can be made during O&M, and to the O&M systems.

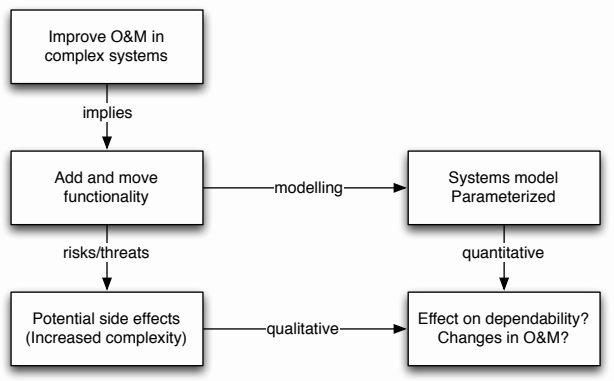


Fig. 2 Understanding the complexity

As a step towards gaining this understanding, Section 2 discusses how the complexity is changing by adding and moving control logic from being embedded and closely integrated with the functionality to be controlled to being separated and to some extents also centralised. Being able to deal with these issues, the ability to build representative, yet understandable and tractable dependability models are crucial. Seeking to build an entirely new theoretical approach does not seem feasible. Our approach is to extend and combine current approaches in novel manners to reach our objective. Hence, to illustrate this and to exemplify the effect of the changes in complexity, Section 2 includes two simple models with numerical exam-

ples. To demonstrate how the complexity might be modelled and assessed, Section 3 gives an example of modelling of the increase complexity in SDN, and Section 4 provides the same for a Smart Grid example. Finally, our concluding remarks are found in Section 5.

2 Complex digital ecosystems

As discussed in the previous section, digital ecosystems are complex systems, which are challenging to operate and control. This is due both to their tight integration with other technical systems and the necessity to perform management over multiple system domains where each domain has (partly) uncoordinated operations.

To enhance and improve the operation and maintainability of the complex digital ecosystems, new functionality is *added* and/or *moved and centralised*. As an example, in Software Defined Networking, the functionality of the control logic is separated from the forwarding functionality in the data plane and *moved* from the distributed control plane residing on the components to be controlled to a virtually centralised control plane. Another example is Smart Grid, where the ICT and power grids are tightly integrated and interdependent. New functionality is *added* both in a distributed manner to enable observability and controllability of the components in the power grid, and centralised in the control centres to implement the control and management.

Adding and moving functionality will contribute to changes in the complexity. The goal is to simplify, or assist handling of complexity. However, adding new hardware and software, or moving the existing, will change the interrelations between functional and logical "entities"/"components". This means that, even though the total complexity is the same or reduced, the system is less well understood and potentially contains new vulnerabilities and poses new management challenges.

Later in this chapter, two comprehensive system examples are introduced to demonstrate the modelling of this change in complexity. In Section 3, a model of Software Defined Networking is given and in Section 4 a Smart Grid example.

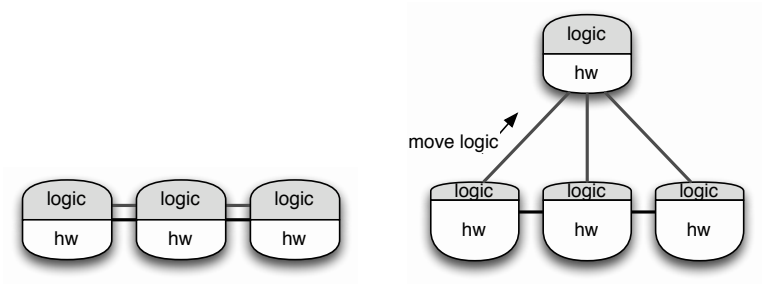
2.1 Centralising distributed functionality

IP networks are comprised of distributed, coordinated, but autonomous network nodes, where the control logic is embedded and closely integrated with the same forwarding functionality that is to be controlled, as illustrated in Figure 3(a).

In emerging networking technology, the trend is to separate the control and forwarding[1] and to move the control logic from the network nodes to a (virtually) centralised controller. The reduction in the distributed (control logic) functionality and

[1] This is similar to how it was done in telephony systems (PSTN) with separate data traffic and signalling traffic using Signalling System 7 (SS7) [10] and in B-ISDN [11]

a corresponding increase in the centralised functionality will potentially reduce the complexity in the (partly) autonomous network nodes and increase the complexity of the centralised systems, as illustrated in Figure 3(b).



(a) Distributed logic embedded on the forwarding engine of the network nodes

(b) Logically centralised control logic with simple distributed network nodes

Fig. 3 Moving control logic to enhance the resource utilisation and improve QoS

It is reasonable to assume that a simplification in the functionality will reduce the complexity of the network nodes. If the properties of the hardware platform is unchanged the network node will then be less error prone. However, if at the same time commodity hardware is used to reduce the node cost then there is a potential risk of decreasing the hardware availability. Then, it is not obvious whether the node availability will improve or not.

The centralisation of the complex functionality should increase the system availability, due to better global overview and coordination. The control logic has comparable (or the same) functionality to the functionality that is moved from the distributed nodes, but additional functionality is needed to coordinate and mitigate the central controllers. Furthermore, centralisation invites new more advanced functionality, for instance consult the motivation for SDN, [6, 20, 24]. It is therefore not known what effect the central controllers have on the system availability.

A separation of the forwarding and control functionality does not necessarily mean a separation of the hardware platform and its functionality. A common mistake is to forget that the underlying resources, such as the routing and switching hardware, are typically utilised not only by the primary information handled by the system, such as user packets, but also for the signalling of information exchange necessary to control and manage the very same resources. Such an interdependency has a negative effect on the overall system availability [4].

Whether the system availability is improved or not when centralising complex functionality depends on to what extent the reduced complexity of the functionality will have a positive effect and improve resource utilisation (due to the global system state being availability, which eases resource coordination) compared to the added complexity in the overhead associated with managing the centralised functionality.

Example 1: Availability requirement of the controller. To demonstrate the effect of moving the complexity on availability a very simple example can be considered. Assume that the conventional network in Figure 3(a) is modelled as a serial structure with three network nodes with availability A_{No}. The serial structure of the network nodes is assumed for simplicity and is not regarded as realistic. The new network is a serial structure consisting of the central controller with availability A_C and the three networks nodes with availability A_{Nn}. Since moving the complexity should improve the availability then $A_{No} < A_{Nn}$. The availability requirement of the controller is given by

$$A_C > (\frac{A_{No}}{A_{Nn}})^3 \tag{1}$$

If $A_{No} = 0.98$ and $A_{Nn} = 0.99$, then $A_C > 0.97$.

If we have some inherent redundancy in the distributed system the effect becomes radical. Assuming the elements in the network in Figure 3(a) operate in a ideal load-shared mode where on them can take the entire load. They will then constitute a parallel system and we get $A_C^* \cdot 1 - (1 - A_{Nn})^3 > 1 - (1 - A_{No})^3$, where $A_C^* > 0.999992$.

Later, in Section 3, a system model of Software Defined Networking is introduced to address in more detail the effect of moving control functionality from being distributed and embedded with the primary function to be separated and (virtually) centralised.

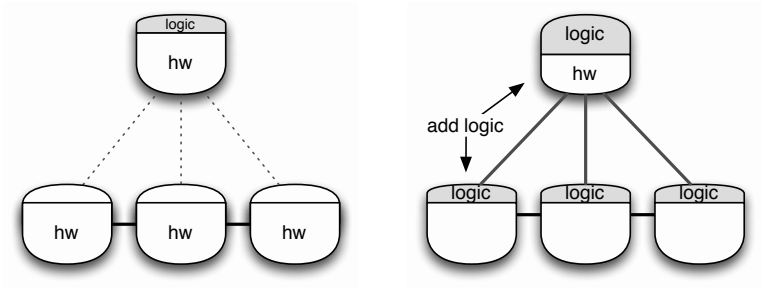
2.2 Add distributed and increase centralised functionality

The need for enhanced operation and control in the power grid is an excellent example where new ICT based control logic is added to the distributed power grid components. In power distribution grids, the grid components typically contain little or no automated control logic. This means that manual detection and recovery is required, which must be coordinated by the control centre, as illustrated in Figure 4(a).

Figure 4(b) shows that new functionality must be added to the centralised controller to be able to utilise the new distributed functionality (remote control logic). Centralising functionality to achieve better decisions will provide a single point of failure, performance bottleneck, and expose targeted attacks.

The ICT based control functionality is not only supporting the operations, but needs to be operated in addition to the primary functionality. The technology and functionality will in many cases be new to the organisation and might change the workflows and result in a need for enhanced knowledge and competence in operation.

From a dependability perspective, adding ICT based control seems to be a bad idea since all the negative side-effects pointed out in the previous subsection apply, with functionality added both in the distributed nodes and in the centralised controllers. This produces less positive effects compared to moving and centralising



(a) Centralised logic and loosely coupled distributed network nodes

(b) Add logic to both the centralised controller and the distributed network nodes

Fig. 4 Adding control logic to enhance the maintainability and improve service reliability

functionality. However, the new ICT based control functionality will increase the maintainability through more timely and precise knowledge and information about system state, so timely (proactive) maintenance can be carried out, and hence, the frequency and consequences of the most frequent faults (failures) are reduced. The operational cost is reduced by reduction in manual labour through better and quicker detection mechanisms and more autonomous (self-)repair. The results are reduced recovery times and better proactive maintenance.

It is not guaranteed that the system availability will increase from added (ICT-based) functionality or not. Even though the maintainability is significantly improved, which makes both proactive and reactive maintenance more effective, it is an uncertainty in that the control functionality itself adds complexity that might affect the system availability.

Example 2: Mean component down time. Adding more logic to the components is assumed to reduce the components recovery time, but at the same time increase the component failure intensity. The hardware failure intensity is assumed unchanged, but the added logic might also fail.

To compare the two systems we should consider the requirements of mean down time (MDT), mean time to failures (MTTF), and availability. In this example, we say that the new system should have the same availability requirement and will then determine the maximum MDT requirement of the component for a given set of failure intensities for the hardware, λ_H, and software, λ_S.

The availability of the original system is:

$$A_{No} = A_{So} \cdot A_H^3 = \frac{\mu_S}{\lambda_S + \mu_S} \cdot \left(\frac{\mu_H}{\lambda_H + \mu_H} \right)^3 \tag{2}$$

while for the modified system with added functionality it is:

$$A_{No} = A_{Sn} \cdot (A_{HS} \cdot A_H)^3 = \frac{\mu_S}{\mu_S + \lambda_{SS}} \cdot (\frac{\mu_S \mu_{SS}}{(\lambda_S + \mu_S)(\lambda_H + \mu_{SH})})^3 \quad (3)$$

To retain the same availability level in the new system, the maximum mean down time $MDT = 1/\mu_{HS}$ is determined by $A_{No} < A_{Nn}$. Let the software failure intensity [in minutes^{-1}] for the centralised control logic be $\lambda_{SS} = 0.5\lambda_S$, and $\lambda_H = 1/24$, $\mu_S = 60$, $\lambda_H = 1/168$, $\mu_H = 1$ then $\mu_{HS} > 1.18529$, which means that MDT < 50.6 minutes.

In Section 4, a Smart Grid example is introduced to demonstrate and discuss the consequences of adding more functionality, both in the distributed entities serving the primary function and centralised in the control centre.

3 Example: Availability in Software Defined Networking

The purpose of this section is to present a case study that highlights how the complexity changes by moving the control logic of a system from distributed to centralised. To illustrate this, we extend and combine current approaches in order to model and assess the availability of a new network paradigm. The results show how the management of complex systems is critical from a dependability perspective. In the following, we introduce some details about Software Defined Networking (SDN) and describe the problem addressed, then we present a two-level hierarchical model for to evaluate the availability of SDN. Finally, we perform a simple sensitivity analysis on a selected set of parameters that will potentially affect the dependability of SDN.

3.1 Software Defined Networking

During the recent years, the SDN has emerged as a new network paradigm, which mainly consists of a programmable network approach where the forwarding plane is decoupled from the control plane [6, 14]. Despite programmable networks having been studied for decades, SDN is experiencing a growing success because it is expected that the ease of changing protocols and provide support for adding new services and applications will foster future network innovation, which is limited and expensive in todays legacy systems.

A simplified sketch of the SDN architecture from IRFT RFC 7426 [6] without the management plane is depicted in Figure 5. The control plane and data plane are separated. Here the control plane is logically centralised in a software-based controller (“network brain”), while the data plane is composed of the network devices (“network arms”) that conduct the packet forwarding.

The control plane has a northbound and a southbound interface. The northbound interface provides an network abstraction to the network applications (e.g. routing protocol, firewall, load balancer, anomaly detection, etc...), while the southbound

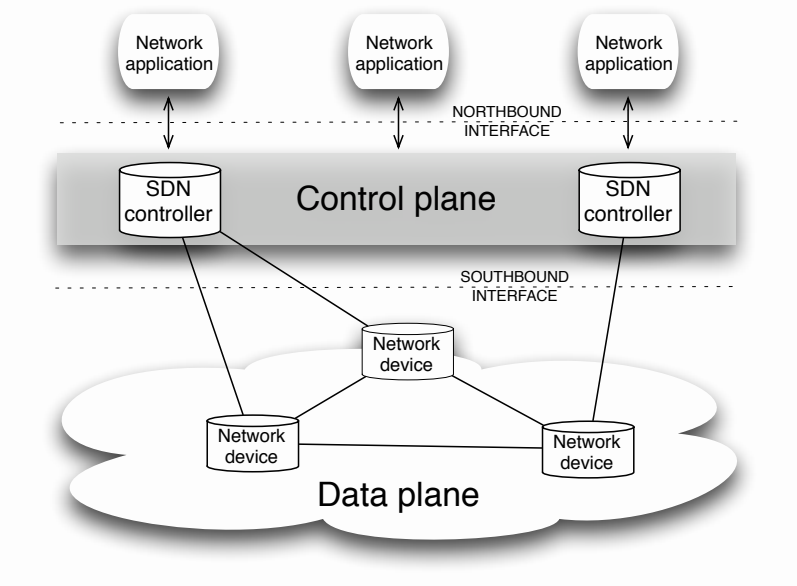


Fig. 5 SDN architecture (exclusive the management plane)

interface (e.g. OpenFlow) standardises the information exchange between control and data planes.

In [20], the following set of potential advantages of SDN were pointed out:

- centralised control;
- simplified algorithms;
- commoditising network hardware;
- eliminating middle-boxes;
- enabling the design and deployment of third-party applications.

However, from a dependability perspective, the SDN poses a set of new vulnerabilities and challenges compared with traditional networking, as discussed in [7]:

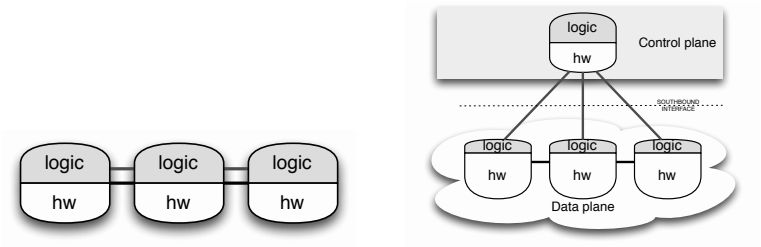
- consistency of network information (user plane state information) and controller decisions;
- consistency between the distributed SDN controllers in the control plane;
- increased failure intensities of (commodity) network elements;
- compatibility and interoperability between general purpose, non-standard network elements
- interdependency between path setup in network elements and monitoring of the data plane in the control plane;
- load sharing (to avoid performance bottleneck) and fault tolerance in the control plane have conflicting requirements;

3.2 Problem description

Traditional IP networks consist of a set of interconnected nodes that include both the data and control planes. Each network node is a complex device that has the functionality of both data forwarding and networking control. To increase the availability and performance of such devices, manufacturers have focused on specialised hardware and software over the past few decades.

As discussed in Section 2, SDN has the potential to change the principles of networking and to enhance network flexibility. This implies moving the control logic from the network nodes to a (virtual) centralised controller, and to open up the controllers to a third party via an API (northbound interface), as illustrated in Figure 6. The transition from a *distributed* network with a focus on establishing and maintaining the connectivity between peering points, to a *centralised* network with a focus on QoS and resource utilisation, will potentially lead to much simpler network nodes with less control logic. The centralised control logic, such as the routing decisions, might be simpler and can even be made more advanced, without making it more complex compared to the distributed solution. The controller has the potential to set up data flows based on a richer set of QoS attributes than in traditional IP networks. However, the coordination and handling of the consistency between the SND controllers, will require new, and complicated logic that will be a critical element to also make SDN a good solution from a dependability perspective.

In the example in this section, we study how the SDN paradigm modifies the overall availability of the network relative to the traditional distributed IP network and analyse which factors dominate in this new scenario.



(a) Current IP networks: Distributed logic embedded on the forwarding engine of the network nodes

(b) SDN: Logically centralised control logic combined with simplified network elements

Fig. 6 Software Defined Networking is an example where the control logic is moved from distributed to virtually centralised (see Fig. 3)

Although dependability must be regarded as an important issue to make SDN a success, to the best of our knowledge, very limited work on modelling the dependability in SDN availability has been performed. In [17], a model of SDN controllers is developed, while [7] discusses potential dependability challenges with SDN, which is partially illustrated by a small case study with a structural analysis

of SDN enabled network. In this section, we study a comprehensive system model of SDN with respect to dependability.

3.3 Modelling

A two-level hierarchical model is introduced to evaluate the dependability of SDN in a global network. In this example, the dependability is measured in terms of steady state availability, in the following referred to as availability. The two-level hierarchical modelling approach consists of

- *upper level:* a structural model of the topology of network elements and controllers
- *lower level:* dynamic models (some) of network elements

The approach seeks to avoid the potential uncontrolled growth in model size, by compromising the need for modelling details and at the same time modelling a (very) large scale network. The detailed modelling is necessary to capture the dependencies that exists between network elements and to described multiple failure modes that might be found in some of the network elements and in the controllers. The structural model disregards this and assumes independence between the components considered, where a component can be either a single network elements with one failure mode or a set of elements that are interdependent and/or experience several failure modes and an advanced recovery strategy. For the former we need to use dynamic models such as a Markov model or Stochastic Petrinet (e.g., Stochastic Reward Network [3]), and for the latter structural models such as reliability block diagram, fault trees, or structure functions based on minimal cut or path sets.

In the following section, we will demonstrate the use of this approach.

3.3.1 Model case

In this example, we analyse the availability of a nation-wide backbone network that consists of 10 nodes across 4 cities, and two dual-homed SDN controllers. See Figure 7 for an illustration of the topology. The nodes are located in the four major cities in Norway, Bergen (BRG), Trondheim (TRD), Stavanger (STV), and Oslo (OSL). Each town has duplicated nodes, except Oslo which has four nodes (OSL1 and OSL2). The duplicated nodes are labelled, X_1 and X_2, where X=OSL1, OSL2, BRG, STV, and TRD. In addition to the forwarding nodes, there are two dual-homed SDN controllers (SC_1 and SC_2), which are connected to TRD and OSL1.

The objective of the study is to compare the availability of SDN with a traditional IP network with the same topology of network elements (SDN forwarding switched and IP routers). We assume that nodes, links, and controllers in the system may fail. The peering traffic in a city is routed through an access and metro network with a connection to both (all four) nodes in the city. The system is working (up), when

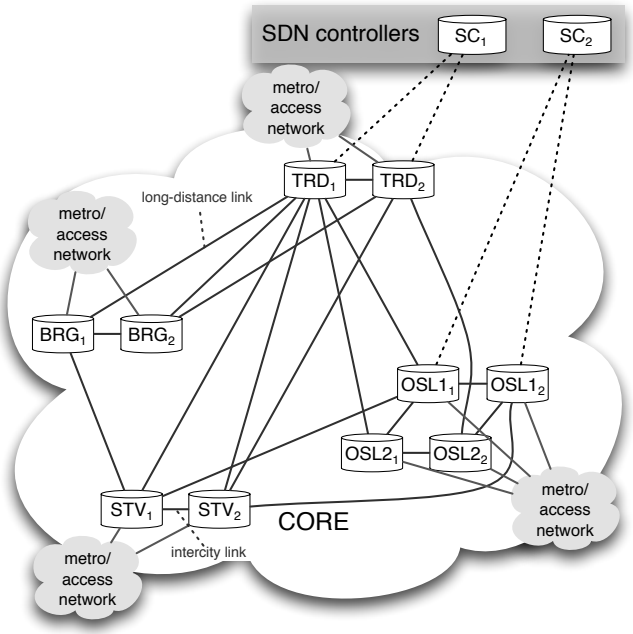


Fig. 7 Case study: nation-wide backbone network

all the access and metro networks are connected. Note that for SDN, at least one controller must be reachable from all nodes along a working path.

3.3.2 Structural analysis

The critical parts of the connection between the traffic origins and destinations can be determined using structural analysis based on either *minimal cut sets*, S, or *minimal path sets*. The sets are defined as follows.

Definition 1. *Minimal cut set:* The system is failed if and only if all the subsystems in a minimal cut set are failed, given that all the other subsystems that are not in the set are working.

Definition 2. *Minimal path set:* The system is working if and only if all the subsystems in a minimal path set are working, and given that all the subsystems that are not in the set are failed.

We use the *minimum cut sets*, S, to form the basis for a *structure function*, Φ (minimum path sets can also be applied).

Definition 3. *Structure function:* Each max-term of the structure function expressed in a minimal product-of-sums form corresponds to a minimal cut set.

The following connections in SDN must be considered:

- *flow triggering*: a path for the trigger message that should be sent from the source node (at least one node of each city) to at least one SDN controller on arrival of a new flow;
- *network state update and route directives*: a path from the SDN controller to each node;
- *forwarding*: forwarding path from/to each city (6 combinations).

The structural analysis for all the possible connections in the SDN example, shows that the cardinality of the set of minimal cut set S is $\|S\| = 2916$. The cardinality $c_j = \|s_j\|$ of each of the minimal cut sets, $j = 1, \cdots, 2916$ is given in Table 1. Each column contains the number of sets that is $C_k = \|\{s_j \in S | c_j = k\}\|$, $k = 1, \cdots, 13$. The table compares the minimal cut sets of SDN with a conventional IP network where the control plane is embedded in the nodes, and hence, no controllers are needed.

Table 1 Distribution of cardinality of the minimum cut sets for the IP network and SDN

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	sum
IP network	0	3	8	91	304	360	356	189	70	13				1394
SDN	0	4	15	107	340	520	780	584	302	170	59	31	4	2916

The number of minimal cut sets with cardinality one is equal to zero because traffic sources are at least dual-homed and there are two dual-homed control sites.

The number of minimal cut sets C_2 increases from 3 to 4 due to the control nodes. Note also that the number of minimal cut sets C_3 almost doubles. This indicates that in this example, a significant increase in vulnerability is observed for the SDN case that is not explained solely by the introduction of a control node, but the fact that a controller must be reachable from every node across the backbone in order for the network to work.

3.3.3 Markov model of networks elements

In order to evaluate the availability of each network element, we develop Markov models of each of the links, traditional routers/switches, SDN routers/switches, and the SDN controllers.

Links

The network model of a link is assumed to be dominated by hardware failures. Therefore, a simple two-state Markov model is used. The links are either up or down due to hardware failure. We use the same model for both traditional networks and

SDN. Given failure rate λ_L and repair rate μ_L, the availability of a link is $A_L = \frac{\mu_L}{\lambda_L+\mu_L}$. This model is assumed for each of the components in the structural model.

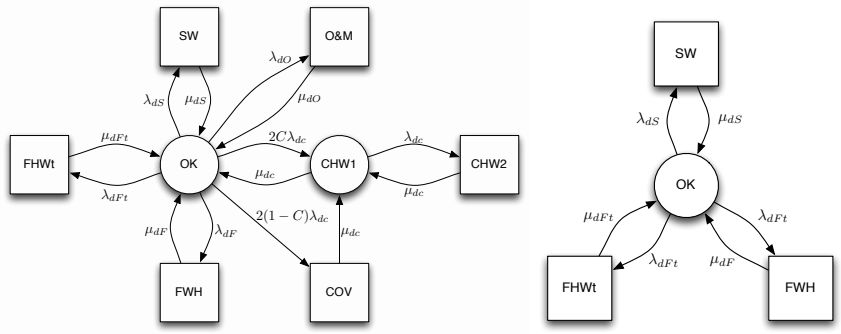
Routers

The model of a traditional router/switch is depicted in Figure 8(a), where the states are defined in Table 2.

Table 2 State variables for traditional IP router

state	up/down	description
OK	up	System is fault free
OM	down	Operation and Maintenance state
CHW1	up	Hardware failure of one controller
CHW2	down	Hardware failure both controllers
COV	down	Coverage state, unsuccessful activation of the stand-by hardware after a failure; manual recovery
FHW	down	Permanent hardware failure in forwarding plane
FHWt	down	Transient hardware failure in forwarding plane
SW	down	Software failure

Multiple failures are not included in the model since they are rare and will have an impact significantly smaller than the expected accuracy of the approach.



(a) Traditional network (b) SDN

Fig. 8 Markov model of a router/switch

SDN forwarding nodes

Figure 8(b) shows the model of the forwarding node, i.e., router or switch in an SDN, which corresponds to the traditional IP router. It is significantly simpler. The states related to the control hardware and O&M failures are not contained in this model, since all the control logic is located in the controller. The software is still present but its failure rate will be very low since the functionality is much simpler.

SDN controller

The model of the SDN controller is composed of two sets of states. One set captures the software and hardware failures. The second set captures the O&M failures in combination with the hardware states of the system. We have assumed that the SDN controller is a cluster of M processors and the system is working, i.e., posesses sufficient capacity if K out of the M processors are active, which means that both software and hardware are working. To represent this scenario, each state is labelled by four numbers $\{n,i,j,k\}$, where n is the number of active processors, i the number of processors down due to hardware failures, j the number of processors down due to software failures, and k the state of the O&M functionality ($k=1$ if O&M mistake, $k=0$, if not). Figure 9 shows the *outgoing* transitions from a generic state $\{n,i,j,k\}$. The main characteristics of the model are:

- single repairman for a hardware failure;
- load dependency of software failure when the system is working, $\lambda_S(n)=\lambda_S/n$, where the meaning of λ_S is explained in more detail in Section 3.4;
- load independence of software failure when the system has failed, $\lambda_S(n)=\lambda_S$;
- when the entire system fails, only processors failed due to hardware failures will will be down until the system recovers.

3.3.4 Using inclusion-exclusion principle to evaluate the system availability

The inclusion-exclusion principle is a technique to obtain the elements in the union of finite sets. Using the inclusion-exclusion principle on the structure function, we can write the system availability as the probability of the union of all minimal paths:

$$A_S = P\left(\bigcup_{i=1}^{n} Q_i\right) = \sum_{k=1}^{n} (-1)^{k-1} \sum_{\substack{\emptyset \neq I \subseteq [n] \\ |I|=k}} P\left(\bigcap_{i\in I} Q_i\right), \tag{4}$$

where $\{Q_1, Q_2, \ldots, Q_n\}$ is the set of all minimal paths, and $P(Q_i)$ is the probability of set Q_i.

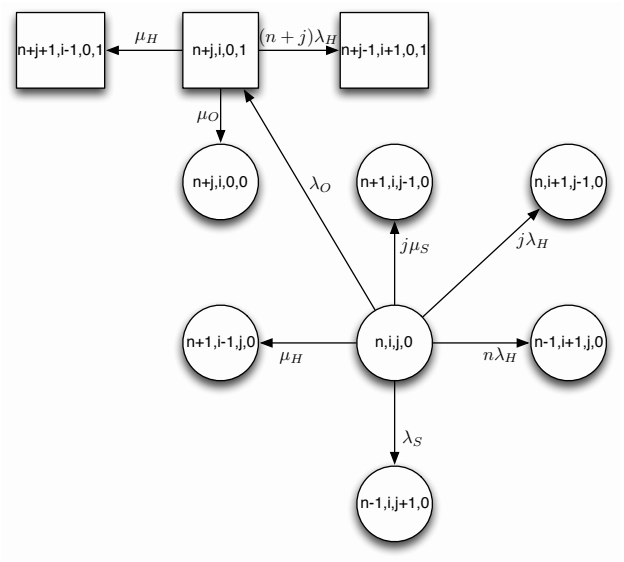


Fig. 9 Generic states of the model of SDN controller

To compute the probability of the intersection of minimal paths, we need to know the availability of each network element. To this emd, we can calculate the element availability by using the proposed Markov models.

3.4 Numerical evaluation

To evaluate the availability of traditional networks, we consider the typical parameters in Table 3, which are inspired by and taken from several studies [5, 15, 23].

All SDN parameters are expressed relative to the parameters for the traditional network (Table 3). The parameters for the SDN switch you find in Table 4 and for the SDN controller in Table 5. The parameters α_H, α_S, and α_O are proportionality factors that are studied in this example.

Using these parameters in the models described in this section, we can compare the (un)availability of traditional IP and SDN networks. Failures with the same cause, have the same intensities in both models. However, we assume that the software on an SDN switch/router will be much less complicated than on a traditional IP router, and we have set the failure rate to zero, for the sake of simplicity. In an SDN controller, all failure rates are N-times larger than in the traditional network, where N is the number of network nodes. This is because we assume that the centralised system needs roughly the same processing capacity and amount of hardware. Therefore, the failure intensity is assumed to be proportional to N, and of the

Table 3 Model parameters for the IP network

intensity	[time]	description
$1/\lambda_L = 4$	[months]	expected time to next link failure
$1/\mu_L = 15$	[minutes]	expected time to link repair
$1/\lambda_{dF} = 6$	[months]	expected time to next permanent forwarding hardware failure
$1/\mu_{dF} = 12$	[hours]	expected time to repair permanent forwarding hardware
$1/\lambda_{dFt} = 1$	[week]	expected time to next transient forwarding hardware failure
$1/\mu_{dFt} = 3$	[minutes]	expected time to repair transient forwarding hardware
$1/\lambda_{dC} = 6$	[months]	expected time to next control hardware failure
$1/\mu_{dC} = 12$	[hours]	expected time to repair control hardware
$1/\lambda_{dS} = 1$	[week]	expected time to next software failure
$1/\mu_{dS} = 3$	[minutes]	expected time to software repair
$1/\lambda_{dO} = 1$	[month]	expected time to next O&M failure
$1/\mu_{dO} = 3$	[hours]	expected time to O&M repair
$C = 0.97$		coverage factor

Table 4 Model parameters for SDN switch/router

intensity	description
$\lambda_F = \lambda_{dF}$	intensity of permanent hardware failures
$\mu_F = \mu_{dF}$	repair intensity of permanent hardware failures
$\lambda_{Ft} = \lambda_{dFt}$	intensity of transient hardware failures
$\mu_{Ft} = \mu_{dFt}$	restoration intensity after transient hardware failures
$\lambda_{sS} = 0$	intensity of software failure

Table 5 Model parameters for SDN controller

intensity	description
$\lambda_H = \alpha_H \, \lambda_{dC} \, N/K$	intensity of hardware failures
$\mu_H = \mu_{dC}$	hardware repair intensity
$\lambda_S = \alpha_S \, \lambda_{dS} \, N$	intensity of software failures
$\mu_S = \mu_{dS}$	restoration intensity after software failure
$\lambda_O = \alpha_O \, \lambda_{dO} \, N$	intensity of O&M failures
$\mu_O = \mu_O$	rectification intensity after O&M failures

same order of magnitude as the total failure intensity of the traditional distributed IP router system.

The results of a numerical example are given in the plot in Figure 10. The overall unavailability, i.e., the probability that not all cities in Section 3.2 are connected (for SDN this requires also a connection to a controller) is given for different values of α_O. The figure shows that the unavailability increases with about one order of magnitude when α_O changes in the range from 0.1 to 1. The sensitivity of α_H and α_S are far less significant. This indicates that O&M failures are dominant and most critical to the dependability of SDN.

As a preliminary conclusion from this study, it seems as the use of commodity hardware and centralised control has a moderate effect on the availability of the overall network. However, the O&M failures and software/logical failures that

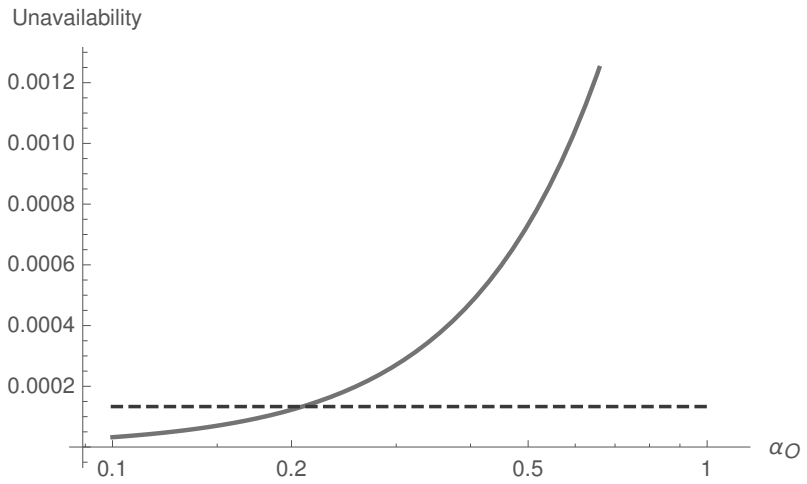


Fig. 10 Unavailability of SDN (solid line) and of traditional network (dashed line) by varying α_O ($\alpha_H = 1$ $\alpha_S = 1$)

causes a control cluster to fail are very important in order to improve the dependability when changing from the traditional distributed IP network to SDN.

4 Example: Restoration in Smart Grid

The purpose of this example is to show how the automation of process steps changes the dependability of a system. The system under consideration is a power grid and we focus on the restoration process after a physical failure.

A power grid is a critical infrastructure and its reliability is critical to the smooth operation of a resilient society. Power grids are due to undergo modernisation in the coming years. This next generation power grid is commonly called the smart grid. One of the biggest differences compared to the current grid is additional monitoring information about the current state of the grid and new control abilities throughout the grid. These improvements allow the introduction of more automated processes with the goal of increasing the overall dependability of the system.

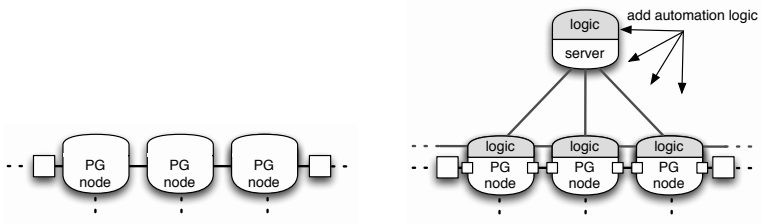
This is the starting point of our example. We model the restoration process with and without automation and conduct a dependability analysis. Our results show that the introduction of automation yields benefits like a reduction of down time, but it also extends the system into a compound and more complex system. This system has new failure modes as the automation may malfunction and thus, without taking the appropriate measures, may partially negate benefits.

4.1 Problem Description

The power grid (PG) has traditionally contained only a few monitoring and controlling devices distributed throughout the grid. Mostly they are deployed in the higher voltage levels. In the lower voltage levels monitoring and controlling devices are, depending on the country, virtually absent. In case of a failure a distributed and autonomously working protection system automatically disconnects a whole protection zone by opening a circuit breaker, causing a power outage to all customers inside this protection zone.

The future power grid, the so called smart grid, will possess monitoring and control systems widely deployed throughout the power grid. These devices detect failures automatically and send failure diagnostics to a central control, operation, and management system. The central system then attempts to isolate the failure by opening other circuit breakers closer to the failure and connecting the rest of the protection zone again to the grid. It is assumed that the power grid at this voltage level has an open ring topology that allows reconnection to the non-isolated parts after a single failure. Figure 11 shows a protection zone in the current PG and in the smart grid, consisting of three PG nodes and two protection devices represented by large squares. The small squares represent new circuit breakers controlled by the centralised control system.

In the following, we study how the introduction of detection and isolation automation changes the characteristics of the restoration process. More precisely, we study the downtime and the energy not supplied (*ENS*), which is the accumulated energy that could not be delivered due to outages, i.e., down time weighted with the load during the outages. Both the lines and the PG nodes can fail, but only larger outages that require a repair crew to go on cite are considered.



(a) Current power grid: no automated detection and controlling.

(b) Smart grid: added logic for automated detection and controlling (distributed and centralised).

Fig. 11 Schematic view of a protection zone in the current power grid and smart grid.

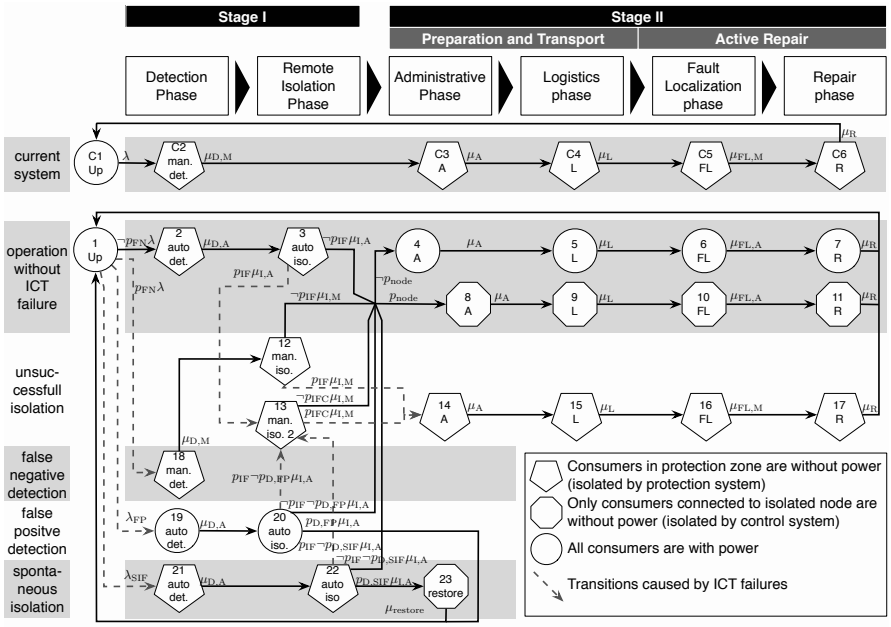


Fig. 12 Phases during the restoration process. For readability reasons, the transitions into state 4 and 8 are displayed in a compact form, it is read as follows; States 3, 12, 13, 20 and 22 each have a transition to 4 and 8, The first is multiplied with $\neg p_{\text{node}}$, i.e. $(1 - p_{\text{node}})$, the latter with p_{node}

4.2 Modelling

The restoration process of a power grid failure consists of two stages containing a total of six phases, as shown in Figure 12. The phases are:

Detection Time period between a failure and its detection in the monitoring system. It is assumed that the protection system disconnects the protection zone containing the failure immediately after the incident. In reality, there is a short delay of several milliseconds. The disconnection leads to a black out in the whole protection zone.

Remote Isolation The failed element is isolated more precisely, either automatically by the central system or manually by a controller at the control centre. The rest of the protection zone is powered up again.

Administrative Failure diagnostics from the monitoring devices are evaluated, the recovery is planned, and a repair crew is assigned.

Logistic Repair crew is equipped with the necessary material and moves to the incident location.

Fault Localization Precise localisation of the failure, both geographically and in the system.

Repair Actual repair, all isolated network elements are restored to normal operation.

The difference between the current power grid and the smart grid lies mainly in *Stage I*. In the current power grid, detection occurs manually, i.e., the failure is detected by a controller or through a call by a consumer. There are no remote isolation capabilities, so this phase is skipped. Throughout the entire restoration phase, the whole protection zone is without power in the model in Figure 12. This is denoted by pentagonal states.

In the smart grid, the distributed devices detect the failure automatically and send an alarm together with fault diagnostics to the central system. Now, the failure is isolated automatically and remotely from the central system and *Stage II* begins. If a PG node is affected by the failure, and now isolated, then the system proceeds to state 8. If only a line is isolated then it proceeds to state 4. In the first case, there are still consumers without power. In the latter case, the power supply has been reinstated to all consumers. This difference is indicated in the model by the different shapes of the states. In both cases, the number of consumers affected is smaller than in the current system. An additional difference is the sojourn time of the fault localisation phase. It is shorter for the smart grid, as the detection devices provide fault diagnostics that accelerate this phase.

So far, we have described the process during operation without any failures in the new system. In the following, we consider failures in the information and communication technology (ICT) subsystem used for the automation. It is assumed that all the other systems, e.g., the protection system, work perfectly. The following failures in the detection system are considered:

- **false positive detection failure:** there is no failure, but the detection system reports one.
- **false negative detection failure:** there is a failure but the detection system does not notice it.

A *false positive detection failure* is modelled with a new transition out of state 1 with an additional failure intensity leading to state 19. The failure is detected by the system as before. If the system discovers the false positive failure the restoration process is interrupted and the system goes back to state 1, otherwise it continues.

A *false negative detection failure* is modelled by splitting the transition from state 1 to 2 into two, pointing one to state 18 and weighting the rate by the false negative probability p_{FN}. The new state 18 indicates a manual detection because of the non-detection in the system. The isolation is then done manually by an operator. If the isolation is successful it proceeds as before either in state 4 or 8 depending on whether a line or a node is affected. If the isolation is not successful the entire protection zone remains without power for *Stage II* of the restoration process.

In the isolation system, the following failures are considered:

- **isolation failure:** there is a failure, but isolation is unsuccessful because of problems with communication or systems. The whole protection zone remains unpowered.

- **spontaneous isolation failure:** there is no failure, but a network element is falsely isolated by the system.

An *isolation failure* is modelled in the system by splitting the transitions from the isolation states 3, 12, 13, 20, and 22 into two, and weighting the rate by the probability of an isolation failure p_{IF}, except for the transitions from 13, which uses a higher probability p_{IFC}, because the system already suffered one ICT failure and is in a critical state.

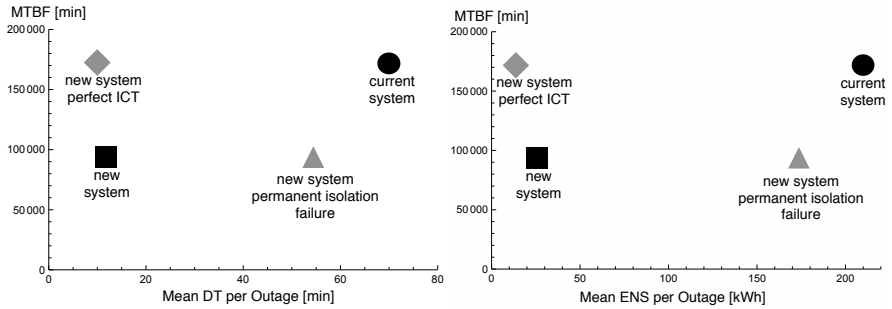
A *spontaneous isolation failure* is modelled with a new transition out of state 1 with an additional failure intensity leading to state 21. The failure is detected by the system as before. If the system discovers that the failure originates from the isolation system and not the power grid it restores the system (state 23) and goes back to the up state, otherwise it continues.

4.3 Numerical Example

All event times in the system are assumed to be exponentially distributed with the following expected values. The event times are based on data for longer outages from the Norwegian regulator [21].

Table 6 Model parameters for the IP network

intensity	[time]	description
$1/\lambda = 4$	[months]	expected time to next PG failure inside this protection zone
$1/\lambda_{\mathrm{FP}} = 6$	[months]	expected time to next false positive detection failure
$1/\lambda_{\mathrm{SIF}} = 12$	[months]	expected time to next spontaneous isolation failure
$1/\mu_{\mathrm{D,M}} = 20$	[minutes]	expected manual detection time
$1/\mu_{\mathrm{D,A}} = 1$	[minutes]	expected automatic detection time
$1/\mu_{\mathrm{I,M}} = 5$	[minutes]	expected manual isolation time
$1/\mu_{\mathrm{I,A}} = 1$	[minutes]	expected automatic isolation time
$1/\mu_{\mathrm{A}} = 5$	[minutes]	expected time in administrative state
$1/\mu_{\mathrm{L}} = 15$	[minutes]	expected time in logistics state
$1/\mu_{\mathrm{FL,M}} = 20$	[minutes]	expected manual fault localisation time, i.e. without fault diagnostics from the detection devices.
$1/\mu_{\mathrm{FL,A}} = 10$	[minutes]	expected automatic fault localisation time
$1/\mu_{\mathrm{R}} = 10$	[minutes]	expected repair time
$1/\mu_{\mathrm{restore}} = 10$	[minutes]	expected restoration time for discovered spontaneous isolation failure
$p_{\mathrm{node}} = 0.1$		probability of failure affecting a node
$p_{\mathrm{FN}} = 0.01$		probability of false negative detection failure
$p_{\mathrm{D,FP}} = 0.2$		probability of discovering a false positive in isolation phase
$p_{\mathrm{D,SIF}} = 0.2$		probability of discovering a spontaneous isolation failure in isolation phase
$p_{\mathrm{IF}} = 0.1$		probability of unsuccessful isolation
$p_{\mathrm{IFC}} = 0.5$		probability of unsuccessful isolation (ICT failure)



(a) *MTBF* against *MDT* per outage (b) *MTBF* against *Mean ENS per outage*

Fig. 13 Mean values per outage

First, we compute *MDT* and the mean time between failure (*MTBF*) for the model in Figure 12. All states in which there is a power outage are considered as down states, i.e. all states but the round states. *MTBF* is computed with

$$\text{MTBF} = 1/(\sum_{i \in \Omega_{\text{Up}}} \sum_{j \in \Omega_{\text{Down}}} \lambda_{ij} p_i)$$

where p_i is the steady state probability of being in state i, λ_{ij} is the transition rate from state i to j, and Ω_{Up} and Ω_{Down} are the sets of up and down states respectively. *MDT* is computed by $\text{MDT} = U \cdot \text{MTBF}$, where the unavailability U is computed with $U = \sum_{i \in \Omega_{\text{Down}}} p_i$.

The results are presented in Figure 13(a). Four scenarios are computed:

1. current system, which is today's power grid system
2. new system,
3. new system with perfect ICT, i.e. $p_{\text{FN}} = 0$, $p_{\text{IF}} = 0$, $\lambda_{\text{FP}} = 0$, $\lambda_{\text{SIF}} = 0$, and
4. new system with a permanent isolation failure, i.e. $p_{\text{IF}} = 1$.

The *MDT* of the new system is smaller than the current system, due to the reduced event times. However, when considering the new system with imperfect ICT, the *MTBF* is reduced as well. Hence, the reduction in *MDT* comes at the expence of more frequent failures. In case of a permanent isolation failure, the *MDT* increases significantly but is still shorter than the current system, as the time in the detection phase is reduced.

MDT gives a one-sided picture of the situation, as the down states have different consequences for the system. The consequences are marked in the model with three different shapes. To incorporate this information, we use the concept of Energy Not Supplied (*ENS*). *ENS* is used in outage reports in power engineering and plays a central role in the Norwegian regulation framework [13]. As the name suggests, it indicates the amount of energy that could not be supplied due to an outage. For our example, we assume that each PG node has a constant energy consumption of 1 kWh per minute. In the pentagonal states, three nodes are down. Therefore, the

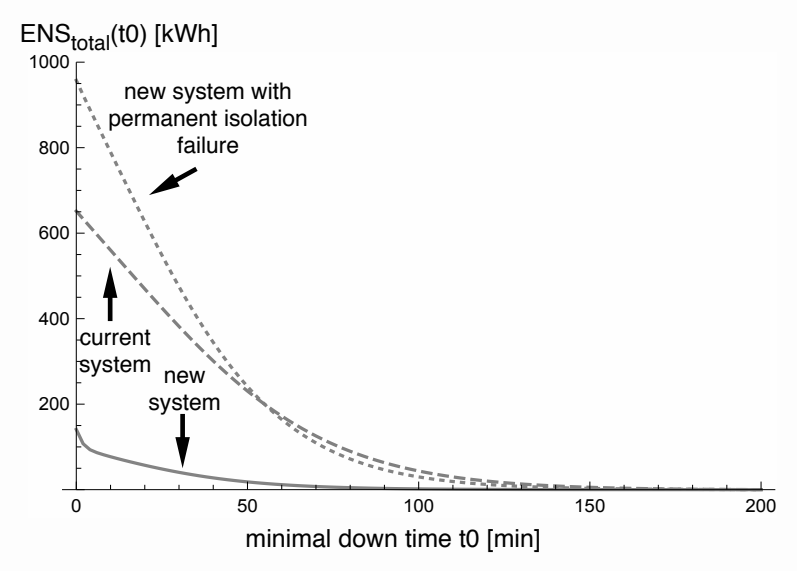


Fig. 14 $\mathrm{ENS}_{\mathrm{total}}(t_0)$ of failures with downtimes $> t_0$

ENS is 3 kWh per minute. The octagonal states have an *ENS* of 1 kWh per minute and the round states 0 kWh per minute.

We use a Markov reward model to obtain the instantaneous *ENS* $e(t)$, i.e., the energy that cannot be delivered at time instance t. First, state 1 is defined to be absorbing. When the system is in steady state, a down period starts in state $j \in \Omega_{\mathrm{Down}}$ with probability $p_j(0) = \mathrm{MTBF} \cdot \sum_{i \in \Omega_{\mathrm{Up}}, j \in \Omega_{\mathrm{Down}}} \lambda_{ij} p_i$. Now the instantaneous *ENS* is computed with $e(t) = \sum_{i \in \Omega_{\mathrm{Down}}} p_i(t) \cdot e_i$, where $p_i(t)$ and e_i are the instantaneous state probability and the energy consumption per minute of state i respectively.

Integrating $e(t)$ over time yields *Mean ENS per outage* $= \int_0^\infty e(t)$, which is plotted in Figure 13(b). The *MTBF* is the same as in Figure 13(a). Compared to MDT, the improvement achieved by automation is even larger in this metric because *ENS* weighs the downtime according to the consequences. However, this is not true for the case with a permanent isolation failure because the down states are all pentagonal like in the current system.

Finally, we extend *downtime-frequency curves* [19] to characterise how the total *ENS* per year of all failures in this protection zone depends on the down time. Let us denote the total *ENS* per year with $\mathrm{ENS}_{\mathrm{total}}$. Counting only the *ENS* of those outages that are longer than time t_0, it becomes time dependent and is computed by:

$$\mathrm{ENS}_{\mathrm{total}}(t_0) = \frac{d(t_0)}{\mathrm{MTBF}} \left(\int_{t_0}^{\infty} \frac{e(t)}{d(t_0)} dt + e^*(t_0) \right)$$

where $(\mathrm{MTBF})^{-1}$ is the number of failures per year, $d(t)$ the probability that the system is down at time t, computed by $d(t) = \sum_{j=\Omega_{\mathrm{Down}}} p_j(t)$, and $e^*(t_0)$ is the

energy not supplied up to time t_0 given that the system has not yet been restored. In order to compute $e^*(t_0)$, the Markov model is modified so there is no transition out of the subspace formed by Ω_{Down} because no complete restoration takes place before t_0 by definition. The transition rates are defined as

$$\lambda_{ij}^* = \begin{cases} \lambda_{ij} & \text{if } i,j \in \Omega_{\text{Down}} \\ 0 & \text{otherwise} \end{cases}$$

The initial state vector of the system is $\mathbf{p}^*(0) = \mathbf{p}(0)$, as before. Thus, $\mathbf{p}^*(t)$ and $e^*(t)$ are computed in the same way as explained above.

The results for $\text{ENS}_{\text{total}}(t_0)$ are shown in Figure 14. In the current system, the relation between downtimes and $\text{ENS}_{\text{total}}$ is approximately linear during the first 50 minutes. In the new system, however, there is a drop in the beginning, indicating that short down times contribute disproportionately to $\text{ENS}_{\text{total}}$. The drop corresponds to *Stage I* of the model. After that, there are either no consumers without power or the system is in the restoration process with the octagonal or pentagonal states and behaves similarly to the current system but at a reduced level. In the case of a permanent isolation failure, $\text{ENS}_{\text{total}}(t_0)$ is larger than in the current system for $t_0 <$ 55 minutes, mainly because of the shorter *MTBF*. For larger t_0, this is compensated for by the effect of shorter *MDT* due to automatic detection. The results show that the automation possesses significant potential to reduce $\text{ENS}_{\text{total}}$. However, in case of longer failures, this may become a disadvantage.

4.4 Observations from the example

The automation of the detection and isolation phase is introduced with the goal of reducing *MDT* and *mean ENS per failure*. However, as the new supporting ICT systems may fail as well, the failure characteristics of the system are changed. First, the *MTBF* decreases significantly, i.e. the number of failures per year increases. Second, outages are on average shorter, and short outages become an important factor when the total *ENS* per year is considered. Third, in case of a longer permanent failure in the ICT system, the consequences increase temporarily and, thereby, adversely affect of the benefit of automation.

The introduction of automation should, therefore, be accompanied by two crucial steps. First, additional training is necessary for the staff covering the new failure characteristics and failures, including the scenario of a malfunctioning ICT system [16]. Second, it is necessary to acquire the skills to maintain and quickly restore the new ICT system to assure a high dependability and thus achieve the positive effects for which the automation was originally introduced.

5 Concluding remarks

The focus of this chapter has been the increasing complexity in digital ecosystems, which are system-of-systems of ICT infrastructures or interact with other critical infrastructures such as water distribution, transportation (e.g. Intelligent Transport Systems), and Smart Power Grid control. There is a lack of theoretical foundation to control the societal and per service dependability of ICT infrastructure in the digital ecosystem. No foundation has been established for optimisation, consolidated management, and provision of this infrastructure, neither from a public regulatory perspective, nor from the perspective of groups of autonomous (commercially) cooperating providers.

More ICT based operation support and control functions are included to manage digital ecosystems, with the objective to reduce the frequency and consequences of daily events. However, it is important to be aware of the potential side-effects that might increase the frequency and consequences of critical and catastrophic failure events. The reason is that the added support enables interaction and integration of even more complex and heterogeneous systems, changes workflows in organisations, and ICT based support systems may fail.

To enhance and improve the operation and maintainability of complex digital ecosystems, new functionality is *added* and/or *moved and centralised*. Two examples are considered in this chapter: (i) Software Defined Networking, which separates the control logic from the forwarding functionality and *moves* the logic from the distributed network elements to a virtual centralised controller, (ii) Smart Grid integrates ICT and power grids which make them more interdependent. Here, new functionality is *added* both in a distributed manner to enable observability and controllability of the components in the power grid and centralised in the control centres to implement the control.

How the changes in complexity affect the overall system dependability is less understood, contains potential vulnerabilities, and poses new managements challenges. This chapter emphazises the importance of being able to model ICT infrastructures. A model must describe both the structure and behaviour of the physical and logical information and network infrastructure, including the services provided. Furthermore, through the modelling phases, it should be explained how *resilience engineering* can be applied to manage the robustness and survivability of the ICT infrastructure. This is the research focus of the research lab on *Quantitative modelling of dependability and performance*, NTNU QUAM Lab (`www.item.ntnu.no/research/quam/start`).

Acknowledgements This work is partly funded by Telenor-NTNU collaboration project *Quality of Experience and Robustness in Telecommunications Networks*, NTNU project *The next generation control centres for Smart Grids* (`https://www.ntnu.edu/ime/smartgrids`), COST Action ACROSS (IC1304), and the research lab on *Quantitative modelling of dependability and performance*, NTNU QUAM Lab (`www.item.ntnu.no/research/quam/start`).

References

1. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing **1**, 11–33 (2004)
2. Buldyrev, S.V., Parshani, R., Paul, G., Stanley, H.E., Havlin, S.: Catastrophic cascade of failures in interdependent networks. Nature **464**(7291), 1025–1028 (2010)
3. Ciardo, G., Trivedi, K.S.: A decomposition approach for stochastic reward net models. Perf. Eval **18**, 37–59 (1993)
4. Cristian, F., Dancey, B., Dehn, J.: Fault-tolerance in the advanced automation system. In: Fault-Tolerant Computing, 1990. FTCS-20. Digest of Papers., 20th International Symposium, pp. 6–17 (1990)
5. Gonzalez, A.J., Helvik, B.E.: Characterization of router and link failure processes in UNINETT's IP backbone network. International Journal of Space-Based and Situated Computing (2012)
6. Haleplidis, E., Pentikousis, K., Denazis, S., Salim, J.H., Meyer, D., Koufopavlou, O.: Software-defined networking (SDN): Layers and architecture terminology. Request for Comments RFC 7426, Internet Research Task Force (IRTF) (2015)
7. Heegaard, P.E., Mendiratta, V.B., Helvik., B.E.: Achieving dependability in software-defined networking - a perspective. In: 7th International Workshop on Reliable Networks Design and Modeling (RNDM). Munich, Germany (2015)
8. Heller, M.: Interdependencies in civil infrastructure systems. The Bridge **31**(4) (2001)
9. Hollnagel, E., Woods, D.D., Leveson, N.: Resilience engineering: Concepts and precepts. Ashgate (2006)
10. ITU-T: Recommendation Q.700: Introduction to Signaling System No. 7 (1994)
11. ITU-T: Recommendation I.371: Traffic control and congestion control in B-ISDN (1996)
12. Kirschen, D., Bouffard, F.: Keeping the lights on and the information flowing. IEEE Power and Energy Magazine **7**(1), 50–60 (2009). DOI 10.1109/MPE.2008.930656
13. Kjølle, G., Samdal, K., Brekke, K.: Incorporating short interruptions and time dependency of interruption costs in continuity of supply regulation. In: CIRED, Prague, Czech Republic, pp. 1–4 (2009)
14. Kreutz, D., Ramos, F.M.V., Veríssimo, P.J.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: A comprehensive survey. Proceedings of the IEEE **103**(1), 14–76 (2015)
15. Kuusela, P., Norros, I.: On/off process modeling of ip network failures. In: Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on, pp. 585–594 (2010). DOI 10.1109/DSN.2010.5544427
16. Line, M.B.: Understanding information security incident management practices: A case study in the electric power industry. Ph.D. thesis, Norwegian University of Science and Technology (NTNU) (2015)
17. Longo, F., Distefano, S., Bruneo, D., Scarpa, M.: Dependability modeling of software defined networking. Computer Networks **83**, 280 – 296 (2015)
18. Morris, R.G., Barthelemy, M.: Interdependent networks: the fragility of control. Scientific Reports **3** (2013). DOI 10.1038/srep02764
19. Norros, I., Pulkkinen, U., Kilpi, J.: Downtime-frequency curves for availability characterization. In: IEEE/IFIP Dependable Systems and Networks (DSN), pp. 398 – 399 (2007)
20. Nunes, B., Mendonca, M., Nguyen, X.N., Obraczka, K., Turletti, T.: A survey of software-defined networking: Past, present, and future of programmable networks. Communications Surveys Tutorials, IEEE **16**(3), 1617–1634 (2014). DOI 10.1109/SURV.2014.012214.00180
21. NVE, Norwegian Water Resources and Energy Directorate: Avbrotsstatistikk 2013. [Outage statistics 2013] (2014)
22. Rinaldi, S., Peerenboom, J., Kelly, T.: Identifying, understanding, and analyzing critical infrastructure interdependencies. IEEE Control Systems **21**(6), 11–25 (2001). DOI 10.1109/37.969131

23. Verbrugge, S., Colle, D., Demeester, P., Huelsermann, R., Jaeger, M.: General availability model for multilayer transport networks. In: Proceedings.5th International Workshop on Design of Reliable Communication Networks, 2005. (DRCN 2005), pp. 85 – 92. IEEE (2005)
24. Xia, W., Wen, Y., Foh, C.H., Niyato, D., Xie, H.: A survey on software-defined networking. Communications Surveys Tutorials, IEEE **17**(1), 27–51 (2015). DOI 10.1109/COMST.2014.2330903