Probabilistic Fault Management in Networked Systems

REBECCA STEINERT

Doctoral Thesis
Stockholm, Sweden, 2014

TRITA-CSC-A 2014:06
ISSN-1653-5723
ISRN-KTH/CSC/A–14/06-SE
ISBN 978-91-7595-114-0

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i datalogi den 28:e maj klockan 14.00 i sal F3, KTH, Lindstedtsvägen 26, Stockholm.

Tryck: Universitetsservice US AB

SICS Dissertation Series 69
ISSN 1101-1335

Abstract

Technical advances in network communication systems (e.g. radio access networks) combined with evolving concepts based on virtualization (e.g. clouds), require new management algorithms in order to handle the increasing complexity in the network behavior and variability in the network environment. Current network management operations are primarily centralized and deterministic, and are carried out via automated scripts and manual interventions, which work for mid-sized and fairly static networks. The next generation of communication networks and systems will be of significantly larger size and complexity, and will require scalable and autonomous management algorithms in order to meet operational requirements on reliability, failure resilience, and resource-efficiency.

A promising approach to address these challenges includes the development of probabilistic management algorithms, following three main design goals. The first goal relates to all aspects of scalability, ranging from efficient usage of network resources to computational efficiency. The second goal relates to adaptability in maintaining the models up-to-date for the purpose of accurately reflecting the network state. The third goal relates to reliability in the algorithm performance in the sense of improved performance predictability and simplified algorithm control.

This thesis is about probabilistic approaches to fault management that follow the concepts of probabilistic network management (PNM). An overview of existing network management algorithms and methods in relation to PNM is provided. The concepts of PNM and the implications of employing PNM-algorithms are presented and discussed. Moreover, some of the practical differences of using a probabilistic fault detection algorithm compared to a deterministic method are investigated. Further, six probabilistic fault management algorithms that implement different aspects of PNM are presented.

The algorithms are highly decentralized, adaptive and autonomous, and cover several problem areas, such as probabilistic fault detection and controllable detection performance; distributed and decentralized change detection in modeled link metrics; root-cause analysis in virtual overlays; event-correlation and pattern mining in data logs; and, probabilistic failure diagnosis. The probabilistic models (for a large part based on Bayesian parameter estimation) are memory-efficient and can be used and re-used for multiple purposes, such as performance monitoring, detection, and self-adjustment of the algorithm behavior.

Keywords: probabilistic network management; probabilistic modeling; fault management; fault detection; event-correlation; change detection.

Sammanfattning

Utvecklingen inom molnbaserade nätverksteknologier, sensornät, heterogena radio-accessnät, samt den överlag ökande användningen av virtualiseringstekniker, sätter nya och höga krav på både hårdvaran och drifthanteringen i framtidens kommunikationsnätverk. De främsta utmaningarna gäller storskalig och flexibel nätverkskontroll under extrema nätverksförhållanden, vilket kräver effektiva och autonoma verktyg som kan hantera snabba tillståndsvariationer i nätet. Existerande metoder för övervakning och hantering av kommunikationsnätverk är otillräckliga för att möta dessa krav, eftersom styrning, kontroll och konfiguration av nätverksutrustning vanligtvis sker manuellt eller genom automatiserade skript. Ett lovande angreppssätt för att utveckla effektiva verktyg kapabla att möta kraven på tillförlitlig, skalbar och hanterbar nätverksdrift baseras på principerna för probabilistisk nätverkshantering, s.k. probabilistic network management (PNM).

En PNM-algoritm följer huvudsakligen tre principer, varav den första avser alla nivåer av skalbarhet, såsom decentralisering, effektiv användning av nätverksresurser, samt beräkningskomplexitet. Den andra principen avser förmågan att kontinuerligt reflektera tillståndet i nätverket korrekt genom att autonomt anpassa modellerna relativt förändringar i nätverksmiljön. Den tredje principen avser tillförlitlighet i modellerna för att minska graden av osäkerhet i de observationer som görs av nätverkstillståndet, och sker genom att en nätverksalgoritm anpassar beteende och operationer relativt prestandakrav specificerade av nätverksoperatören eller användaren.

Denna doktorsavhandling kretsar kring probabilistiska felhanteringsalgoritmer som följer ovanstående principer för probabilistisk nätverkshantering. Ett kapitel sammanfattar den senaste utvecklingen av nätverksteknologier och problemområden, följt av existerande metoder för probabilistisk nätverks- och felhantering. I avhandlingen redogörs PNM-konceptet samt praktiska tillämpningsaspekter som följer av att använda probabilistiska angreppssätt jämfört med deterministiska algoritmer. Det senare exemplifieras i en första studie av skillnaderna i prestanda mellan en probabilistisk feldetektionsalgoritm och traditionell feldetektion. Vidare presenteras sex probabilistiska felhanteringsalgoritmer, vilka implementerar olika aspekter av PNM.

Algoritmerna är till hög grad konstruerade för decentraliserad, adaptiv och autonom nätverksdrift och omfattar: probabilistisk feldetektion baserat på adaptiva mätningar relativt observerat nätverkstillstånd och specificerad detektionsprestanda; distribuerad och decentraliserad förändringsdetektion i modellerade mätningar i nätet; rotorsaksanalys och felisolering i multipla virtuella lager; korrelationsanalys av händelser i dataloggar; samt, probabilistisk diagnostik. Den övergripande ansatsen baseras till stor del på minneseffektiva Bayesianska modeller som under drift återanvänds för övervakning, detektion och adaptiva mekanismer.

Nyckelord: probabilistisk nätverkshantering; probabilistiska modeller; felhantering; feldetektion; korrelationsanalys; förändringsdetektion.

To Mother, Father, Maria, Albin and Hilda

Acknowledgements

This work has been supervised and contributed to by Dr. Daniel Gillblad, Swedish ICT SICS. Scientific advices during the period January 2010 - March 2012 were provided by Professor Stefan Arnborg, main supervisor at KTH in this period. For the development of this thesis, advices have been provided by Docent Anders Holst, academic supervisor at Swedish ICT SICS, and Professor Anders Lansner, main supervisor at KTH from March 2012.

I am especially grateful to Daniel Gillblad whom I have collaborated with throughout this work - it has been inspiring, fun and encouraging. Moreover, I would like to thank Björn Levin for giving me the possibility to work in interesting research projects at the Industrial Applications and Methods lab (now Decisions, Networks and Analytics). Anders Lansner was the one who first introduced me to scientific work at KTH and later to my current lab, and for this I am forever grateful. Thank you all for believing in me!

There are several colleagues and researchers who I specially want to thank for interesting discussions, inspiration, and contribution to this work: Sara Gestrelius, Anders Holst, Stefan Arnborg, Anders Gunnar, Per Kreuger, Björn Bjurling, Olof Görnerup, Catalin Meirosu, Diogo Ferreira, Avi Miron, and Alberto Gonzales Prieto. Additionally, I would like to thank Bengt Ahlgren, Rolf Stadler and Magnus Boman for kindly helping me to get in contact with the opponent and the members of the committee reviewing this thesis.

Finally, and most importantly, this work would not have existed without the support of my parents, my family, and my four-legged companions - Albin and Hilda. Your everyday encouragement and support have given me the willpower to follow this through and the ability to always find the spark in difficult moments.

Rebecca Steinert
Athens-Stockholm, March 2014
(at 30 000 feet)

List of Abbreviations

BGP border gateway protocol.

BIC Bayesian information criterion.

CDF cumulative density function.

CGF cumulative generating function.

EM expectation maximization.

GLR generalized likelihood ratio.

GMM Gaussian mixture model.

HetNet heterogeneous radio access network.

HMM hidden Markov model.

ICMP Internet control message protocol.

IoT Internet of Things.

IP Internet protocol.

KLD Kullback-Leibler divergence.

MaaS monitoring-as-a-service.

MCMC Markov chain Monte Carlo.

MEP maintenance endpoint.

MIB management information base.

MIP maintenance intermediate point.

MLE maximum likelihood estimate.

MoM method-of-moments.

MPE maximum probable explanation.

MPLS multiprotocol label switching.

NTP network time protocol.

OAM operations, administration and management.

OD origin-destination.

OMP orthogonal matching pursuit.

PCA principal components analysis.

PNM probabilistic network management.

QoS quality of service.

SDN software-defined networking.

SLA service level agreement.

SLO service level objective.

SNMP simple network management protocol.

VM virtual machine.

Contents

Chapter 1

Introduction

1.1 Motivation and Scope

The current development of different network technologies spans across several interesting areas, driven by the close collaboration between the network research community and telecom industry. *Cloud computing and networking*, for example, is the concept of seamless integration and management of compute, storage and network resources, needed for flexible handling of large-scale services and applications (Armbrust et al., 2010; Chowdhury and Boutaba, 2009; Dillon et al., 2010; Zhang et al., 2010). Managing complex cloud systems is technically challenging, and requires a wide range of advanced management algorithms and business models for dealing with e.g. inter-operability and security issues, service specifications, resource provisioning, service quality, and failure resilience.

Virtualization and software-defined networking are enabling and supporting technologies for flexible management of cloud systems and network infrastructures. Virtualization techniques allow for dynamic service deployment and implementation of virtual network functions on generic hardware. The increasing degree of virtualization has also lead to the development into new network controllers based on programmability under the paradigm of software-defined networking (SDN) (Lantz et al., 2010). Virtualization and SDN are complementary techniques that enable significantly greater flexibility in the control of traffic flows and management of more generic hardware and resources, compared to traditional network environments where the control is integrated with individual physical network devices designed for a certain purpose.

Internet of Things (IoT) is the evolving concept of seamless integration of communication entities or smart objects (such as mobile phones and other types of equipment carrying sensors, actuators or radio frequency identification tags) into information networks (Atzori et al., 2010). The increasing interest of the IoT-concept within different domains (i.e. industry, healthcare and social networking) drives the development of new applications, which requires network management

methods capable of handling the challenges related to inter-operability, adaptability, scalability and security (Atzori et al., 2010; Miorandi et al., 2012).

Mobile and radio access technologies are other areas under rapid development that require efficient means for managing heterogeneous network technologies and standards, in order to handle the increased user mobility and keep the number of disruptions in network access and connectivity to an absolute minimum (Andrews, 2013; Ghosh et al., 2012). In this context, one of the many interesting challenges is automated configuration of radio and cellular equipment for ensuring efficient resource usage and radio coverage (Hu et al., 2011).

These, and other technical advances in network communication, require revised network management methods. Currently, the general network management approach is primarily centralized and deterministic, meaning that management actions and decisions are executed from a central point and based on strict limits, thresholds and rules. In this setting, network devices and management algorithms are usually configured based on detailed knowledge about the network behavior combined with best practices. This approach has so far worked sufficiently well for medium-sized networks operating under fairly static network conditions. However, the development toward significantly more complex and large-scale networks imply technical management challenges related to resource-efficiency and service quality that cannot be sufficiently addressed by employing a centralized and deterministic management paradigm. One reason is that deterministic management is sensitive to variations and uncertainties in the network behavior and state - another reason is that a centralized management setup is susceptible to single-point failures, and does not scale well with increasing network sizes.

The next generation of communication networks will require efficient management tools for large-scale control and coordination of network equipment and resources, such that service guarantees to millions of users and clients can be fulfilled satisfyingly. In practice, it means that network management solutions need to be scalable and resource-efficient, and significantly more self-adaptive to variations in the network environment. A fundamental step in addressing these challenges, is the shift from a centralized and deterministic network management approach, toward more decentralized management operations based on adaptive and probabilistic models, that better can account for uncertainties and local network conditions. Enhanced efficiency of a decentralized probabilistic management framework may be achieved if the algorithms included also are designed to autonomously adapt in line with high-level objectives and performance guarantees.

On this background, the subject of this thesis is probabilistic fault management with focus on *scalability*, in terms of resource-efficient network management and operations; *adaptability*, in the sense of accurate reflection of the network state and behavior; and *reliability*, with respect to improved performance predictability and algorithm control.

1.2 Key Contributions

We define the probabilistic network management (PNM) concept and identify the necessary properties of probabilistic algorithms needed to meet the management challenges of future networks (Paper V). Essentially, a PNM-algorithm is based on a probabilistic model; operates in an adaptive manner; and is controlled by probabilistic guarantees or objectives. Practical differences and considerations between probabilistic and deterministic algorithms are illustrated in a first study evaluating different performance aspects of two types of probe-based monitoring algorithms (Paper VII). In addition, an overview of existing network management approaches based on probabilistic modeling is provided.

A major part of this thesis is focused on a set of six probabilistic fault management algorithms that implement various aspects of the following design goals:

- scalability, with respect to decentralized or distributed operation, and efficiency in network resource utilization and computations;

- adaptability, meaning the ability to update the probabilistic models to spatio-temporally varying network conditions (e.g. topological changes, or behavioral trends) in order to accurately reflect the network state and performance;

- reliability, in the sense that the algorithm behavior is autonomously adjusted to meet specified high-level objectives and performance guarantees, for the purposes of performance predictability and reduced configuration efforts.

The algorithms cover different functionality of the fault management cycle, including *data collecting and modeling* of data sources such as probe measurements, event logs and diagnostic input; *detection* of faults, changes and anomalies; *isolation* of faulty network states, including localization, root-cause analysis and diagnostics; and, *recovery*, here addressed in terms of reporting the analyzed network state. More specifically, the following problem areas are addressed: failure diagnostics (Paper I); event-correlation and pattern mining in data streams (Paper II); fault detection with controllable detection performance (Paper III); change detection in monitored link metrics (Paper IV); distributed root-cause analysis in virtual overlays (Paper VI); and, detection of performance degradations from end-to-end measurements (Paper VIII). Three of the six algorithms carry out probe measurements whereas the remaining algorithms operate on data from other sources.

Included algorithms implement memory-efficient parameter estimation methods based on the method-of-moments (MoM) and Bayesian techniques. The probabilistic models are used for multiple purposes, such as modeling of link metrics; self-adjusting algorithm behavior; and detection of faults and changes.

1.3 Thesis Outline

The remaining outline of the thesis is as follows: In Section 2, a literature survey is provided, covering the challenges of future networks, probabilistic management approaches, and existing fault management algorithms. Section 3 contains a presentation of the PNM-paradigm, and an overview of the fault management algorithms included in the thesis. Section 4 is a summary of included papers and contributions, followed by concluding remarks in Section 5.

Chapter 2

Current Development

This chapter provides a brief overview of the current development of network management methods in relation to the probabilistic management paradigm and the design goals described in Chapter 1. Current views on the challenges and requirements of future network management solutions relevant to probabilistic modeling are presented, followed by an overview of existing algorithms that to a high degree implement aspects of the probabilistic management paradigm. Finally, existing fault management approaches are briefly assessed with comments on how the algorithms relate to the PNM design goals.

2.1 Challenges and Requirements of Future Network Management

Within the area of cloud computing and networking, Zhang et al. (2010) highlight problem areas related to automated service provisioning, virtual machine (VM) migration and server consolidation, as well as traffic management and analysis. Automated service provisioning enables efficient resource allocation and simplified configuration, and is a necessary component to enforce specified service level objectives (SLOs). Parts of resource management, as well as dynamic service configuration and fulfilment, are based on virtual machine migration which enables e.g. load balancing across data centers. Moreover, the management of services requires further development of tools for managing and analyzing traffic because of the increasing scale and complexity of cloud infrastructures (which include computing, storage and networking). Similarly, Dillon et al. (2010) highlight some requirements relevant to cloud computing, such as on-demand self-service for instant service specification and deployment; broad network access for a variety of network equipment; resource pooling and rapid elasticity, where resources and services are continuously reconfigured across multiple providers and heterogeneous equipment; as well as service measurements for usage monitoring across services and individual users.

In other areas such as IoT and heterogeneous radio access networks (HetNets), similar challenges related to scalability, adaptability and reliability are also discussed. As described by Miorandi et al. (2012), the concept of IoT challenges traditional views on networking architectures, shifting the focus from point-to-point communication to data generation, information processing and exchange. One of the main challenges relates to different aspects of scalability (such as addressing and network communication, information handling, service provisioning and management), which can partially be addressed through the means of self-organizing management capabilities and autonomous mechanisms. The authors also mention that services are currently designed in detail for each application, which therefore requires further research on design patterns in order to achieve increased service flexibility and adaptability. Within the HetNet paradigm, Andrews (2013) lists several technical challenges and requirements, such as improved interference management through resource allocation based on traffic patterns instead of static frequency reuse; improved mobility robustness through handover optimization modeling instead of using deterministic thresholds; and, better methods to cell associations by the use of load and traffic models, instead of connecting to the base-station with the strongest signal.

Essentially, we see from these examples several problems regarding flexible network- and service-management, as well as demands for using more decentralized, self-managing algorithms that can adapt almost in real-time to changing network conditions. The following articles bring up requirements and possible approaches for solving many of these problems, closely related to a decentralized and probabilistic management paradigm. For example, Pras et al. (2007) provide an overview of the key research challenges and necessary properties of future network management solutions, such as new architectures capable of supporting autonomous, cooperative in-network management algorithms, while preventing loopy control behavior. The authors identify that future network management systems need to be capable of handling the scale and uncertainty in the networks, which suggests that an increased use of more distributed and probabilistic management algorithms is to be expected. Monitoring, for example, should according to the authors be performed in a distributed manner, at minimum cost and in relevant parts of the network while specified accuracy requirements and management objectives are met. Somewhat differently, Kind et al. (2008) discuss a holistic management approach to traffic analysis and monitoring based on adaptive and self-managing algorithms operating in large, complex network- and service-infrastructures. The authors list the key functions of an "awareness plane" necessary for providing accurate information about the network operation. The authors also list challenges related to monitoring, discussing the need for distributed processing algorithms that efficiently can process traffic measurements with low communication overhead, and that are adaptive in order to provide accurate and up-to-date estimates. Finally, Forell et al. (2011) discuss challenges and problems related more specifically to cloud management. The authors emphasize the need for a distributed architecture for flexible and scalable management methods that can handle the dynamic behavior in a cloud environ-

ment. Important components for closed loop management include, according to the authors, scalable and continuous online monitoring and analysis. The authors also mention the necessary development of autonomic solutions, based on statistical learning and control theory for the purpose of achieving self-adaptive system behavior relative to specified performance goals and policies.

The work in this thesis is in line with many of the aforementioned requirements - namely, that network management algorithms operating under highly dynamic network conditions need to be scalable, adaptive and reliable. Related work indicate that implementing network management functions following a decentralized and probabilistic paradigm is highly relevant and directly necessary, in order to successfully meet the aforementioned requirements and cope with the management challenges of current and future networked systems. In the next sections, examples of promising probabilistic algorithms are discussed with respect to the design goals under the probabilistic management paradigm.

2.2 Algorithms Implementing Probabilistic Management

In this section examples of management algorithms closely in line with the design goals (see Chapters 1 and 3) are discussed. The algorithms exemplify different important aspects of a probabilistic and autonomous management system, and are focused on resource management and network monitoring. In general, the algorithms either maintain a probabilistic model or operate on probabilistic input. Moreover, the algorithms are to a high degree adaptive to changes in the network environment, and may also self-adjust the algorithm behavior according to specified high-level objectives. In some cases, probabilistic performance guarantees can be derived from the model used, which generally provides means for improved predictability and controllability of the algorithm performance.

Resource Provisioning and Optimization

Gonçalves et al. (2012) propose a mathematical framework and an algorithm for probabilistic dynamic resource provisioning in cloud systems, controlled by high-level objectives and capable of adapting to observed load and user behavior. The method is based on inferring the probability of deviation from nominal values of workload time-series generated from Markovian processes satisfying the Large Deviation Principle (LDP). Within this setting, the purpose is to estimate the link capacity and the re-optimization frequency given probabilistic tolerances on link loss and workload overflows. Moreover, Elmroth et al. (2011) discuss management solutions for self-managing elastic cloud infrastructures, focused on predictive elasticity, admission control and placement of VMs. The admission control and VM-placement algorithm depend on estimated probability distributions of compute, storage, and aggregated service workloads, as well as probabilistic SLA constraints related to e.g. availability guarantees. Konstanteli et al. (2012) address the problem of allocating

resources for application workflows (consisting of real-time services requiring CPU, memory, network etc.) according to probabilistic guarantees on response-times and resource availability. The method exploits prior knowledge about the actual resource usage, which improves the overall resource efficiency as more processes can be scheduled on the same host.

The methods are designed to address dynamic resource provisioning given SLA requirements specified as probabilistic guarantees, such that service elasticity can be achieved. In the work by Gonçalves et al. (2012), the method used to estimate link capacity is fully probabilistic, whereas in the other two examples an optimization problem is solved given probabilistic input obtained from probabilistic performance models (Elmroth et al. 2011; Konstanteli et al. 2012). Scalability aspects are in this context generally addressed mainly in terms of reducing the overall computational complexity in dedicated computing equipment. Keeping the solutions up to date is achieved by using controllers or decision-making mechanisms that may trigger re-optimization upon changes in the network environment (Elmroth et al. 2011; Gonçalves et al. 2012; Konstanteli et al. 2012) (see also Murray et al. 2012). In these examples, the obtained solutions are also guaranteed to meet various service requirements and constraints (expressed as probabilities, fractions and deterministic limits). The quality of such solutions is often a trade-off between computational complexity and desired accuracy. With the increasing sizes of networked systems it is not unlikely that parts of future resource management solutions will include dynamic and decentralized resource provisioning in local parts of the infrastructure for improved resource-efficiency and service quality.

Distributed Estimation and Random Sampling

Prieto and Stadler (2006) present a distributed tree-based algorithm capable of estimating global metrics from in-network aggregates of local metrics across the network, controlled by probabilistic specifications on the tolerable estimation error. The aim is to optimize and compute local filters to minimize the communication overhead between child and parent nodes such that the estimation error of the aggregated metric at the root node does not exceed the specified limit. The local filters are updated periodically or when a change in the metric is detected. Further, Han et al. (2007) provide an adaptive algorithm for aggregation of min, max and mean values within a hierarchical sensor node model, capable of adjusting to probabilistic requirements on the query result. The authors address the problem of determining the smallest set of sensors that fulfills the specified level of accuracy in the monitoring process. The sampling rate, sample size and set of sensors used for aggregation in a network region are determined dynamically, as well as the set of regions used for aggregation, relative to specifications on the desired accuracy. Cohen and Landau (2009) propose a decentralized resource-efficient probabilistic approach to estimating the number of nodes affected by an event, by sampling a small amount of reports sent from several network nodes toward one management

node. The aim is to reduce the communication overhead and prevent message implosion at single points in the network. This is achieved by randomizing the time at which a report is sent from each node in combination with setting a limit on the number of reports to receive at the management node. For a varying number of affected nodes in the network, the algorithm produces predictable error estimates with respect to the number of reports received. Finally, Brunner et al. (2009) outline a management framework where functions are probabilistically activated and deactivated. Scalability is addressed in terms of distributed operation, low communication overhead in the network and reduced resource usage in managing nodes. Management functions controlled by important performance goals can obtain more running time by being activated more often with higher probability. The authors present a fault detection algorithm based on the above concept which illustrates the trade-off between the detection performance and activation probability.

These algorithms operate in a distributed and decentralized manner for improved resource-efficiency in the network, which is here achieved by reducing the number of message exchanges between the nodes (Cohen and Landau 2009; Han et al. 2007; Prieto and Stadler 2006), and by varying the time of operation for different functions in the managing network nodes (Brunner et al., 2009). In some cases the models maintained by the algorithms are autonomously adjusted over time - for example, the filter parameters used in Prieto and Stadler (2006) and the set of estimate components described in Han et al. (2007) are updated when there are changes in the monitored metric or in the network environment. Additionally, the behavior of the algorithms adapts according to specified accuracy requirements for increased reliability in algorithm performance and estimates (Brunner et al. 2009; Han et al. 2007; Prieto and Stadler 2006). The model proposed by Cohen and Landau (2009) allows for deriving lower and upper bounds on the estimation error relative to the number of reports received, demonstrating a high degree of performance predictability in the results under different conditions.

Self-Organization and Network Monitoring

Zhu and Ni (2008) introduce a distributed algorithm for balancing the resource consumption in sensor networks while meeting the requirements on detection probability and detection latency. The algorithm implements a cooperative behavior between the sensors for determining the probabilistic level of sensor activity needed for detecting an event under given detection quality specifications. Changes in the network topology trigger re-calculations of the activity probability in the affected sensors. Moreover, Rajagopal et al. (2008) suggest a distributed approach to fault detection in sensor networks and local correlation statistics between neighbors. Under the assumption that failed nodes exhibit lower correlation scores compared to its neighbors, faulty sensor behavior can be detected. The energy consumption and node density adapt in accordance with specified performance guarantees on the detection delay and false alarm probability. For cloud-specific applications, Meng and

Liu (2012) introduce the concept and key functional requirements of monitoring-as-a-service (MaaS) in a cloud system, addressing problems related to adaptive monitoring. The authors propose a resource aware topology planning method for increased scalability in the sense of reduced communication overhead among the monitoring nodes. Moreover, the authors address the trade-off between scalability and uncertainty in the network by offering an algorithm for flexible monitoring, capable of adjusting the measurement intensity in line with the specified performance requirements on detecting threshold violations. A distributed algorithm for reporting violations of local thresholds is introduced in the same paper, where the authors employ a window-based filtering mechanism in order to reduce the number of false alarms on local and global levels. An EM-algorithm is suggested for automatically setting the window size and the thresholds, given a cost model and the expected communication cost by running the algorithm. Further, Badonnel et al. (2006) present a distributed self-organization method for identification of high-connectivity node clusters and autonomous selection of managing nodes (or cluster heads) in ad-hoc networks. A cluster component is formed by the set of neighboring nodes that are connected for some percentage of time, such that spatio-temporally connected cluster components contain nodes with high adjacency probability. A K-means classification method is applied to the connectivity measurements limited to the spatio-temporally connected nodes within a cluster for the selection of a managing node. The approach allows for deriving probabilistic guarantees on the proportion of managed nodes. Additionally, a mechanism for detecting abnormal mobility behavior in a cluster is developed based on comparing the spatio-temporal node connectivity between sliding time-windows.

The algorithms are designed to address different aspects of scalability and resource-efficiency by distributed operation in the network, such that the energy-consumption and the amount of management traffic are reduced (Badonnel et al., 2006; Meng and Liu, 2012; Rajagopal et al., 2008; Zhu and Ni, 2008). Adaptation and parameter updates are in these examples performed continuously in cycles (Meng and Liu, 2012), or when topological changes (Zhu and Ni, 2008) and abnormal behavior in the nodes (Badonnel et al., 2006; Rajagopal et al., 2008) have been detected. The reliability aspect is in general addressed by adjusting different parameter settings (e.g. sampling rate and sensor activities) to high-level requirements on the detection quality such as detection probability or accuracy (Meng and Liu, 2012; Rajagopal et al., 2008), or tolerable level of false alarms (Zhu and Ni, 2008). In the case of Badonnel et al. (2006), the algorithm performance (i.e. the proportion of managed nodes in the network) can be predicted at known confidence levels observed through simulations.

2.3 Fault Management and Monitoring

This section provides an overview of general fault management and network monitoring algorithms available in the existing literature. The related work include

different methods for detection and isolation of faults and anomalies given network measurements and statistics as input. The selected algorithms are centered around probabilistic modeling and machine learning approaches. As the focus in this section is mainly to provide an overview of existing fault management methods, the degree to which these algorithms match the design goals (see Paper V or Chapter 3) under the probabilistic management paradigm varies[1].

Network Tomography

Network tomography includes a set of algorithms used for inferring link metrics (such as link delays and loss) and network topologies from end-to-end probe measurements. Derived link metrics can then be used for further analysis, such as anomaly detection. For example, Duffield and Lo Presti (2000) provide a non-parametric estimator for deriving the delay variance of individual links in multicast trees, obtained by modeling and comparing end-to-end delays over common links in a multicast measurement session. In a different approach by Duffield et al. (2001), the authors outline a method for deriving link delays in a tree-topology by discretized non-parametric modeling of unicast back-to-back measurements. In their model, variable bin sizes are used at several levels to capture the link delay distribution, from which the parameters of individual links can be derived via expectation maximization (EM). Similarly, Tsang et al. (2003) employ a non-parametric model to derive link delays in tree-topologies via unicast back-to-back packet pair measurements. Here, a fixed number of bins is used and set equal to the number of measurements performed in order to model delay distributions. Individual link delays are estimated using a modified EM-algorithm based on fast Fourier-transforms. Lin et al. (2010) propose another method for deriving link delays in multicast trees. In their method the parameters of exponentially distributed link delays are estimated from the observed end-to-end delays between the root node and the leaf nodes via the MoM. Similarly, Shih and Hero (2001) have developed a bias-corrected MoM-estimator for the cumulative generating functions (CGFs) of intermediate link delays. The authors employ a least-squares method for modeling unicast end-to-end probe measurements in the given network topology. Final estimates are obtained from the empirical averages of MoM-estimates within a sliding time-window. The estimated CGFs can then be used for analysis of individual link behavior and detection of bottlenecks by the use of probabilistic thresholds.

Zhang et al. (2005) present a two-step method for dynamic network anomography for the purpose of identifying which origin-destination (OD)-flows contribute to anomalies in observed link loads. The mathematical framework provided by

[1] Topics on more complex fault-handling or recovery actions (in addition to simple reporting and triggering of alarms) are not covered in the following section, as such methods generally include e.g. programmability and virtualization techniques which here are out of scope (see Chapter 3).

the authors encompasses linear matrix transformations and several signal processing techniques (e.g. time-frequency analysis and principal components analysis (PCA)) and greedy algorithms (e.g. orthogonal matching pursuit) for filtering out deviating link loads. The deviating link loads are used for identifying which OD-flows should be selected for further diagnosis. Firooz and Roy (2010) employ a compressed sensing method for estimation of link delays under the assumption that the distribution of significant delays in the network is sparse. The compressed sensing framework allows for designing a measurement matrix (i.e. the routes of the probes) that fulfills the specified conditions on tolerable estimation error and measurement costs. Moreover, Burch and Chase (2005) propose a method for estimating propagation and queueing delays per link by solving a linear system of tunnel measurements obtained via one probing host. The estimates are obtained by finding optimized solutions (via linear programming and least-squares methods) under certain assumptions on propagation and queueing delays, and on the degree of link delay asymmetry.

Cunha et al. (2009) extend a binary tomography method to blackhole identification by adding mechanisms for failure confirmation and aggregation of observed loss measurements, for the purpose of creating a reachability matrix after each probing cycle. The failure confirmation step is based on solving an optimization problem which gives the number of probes and intertransmission times needed to identify a fault relative to a threshold on the tolerable detection error. The authors derive probabilistic expressions and bounds for obtaining a consistent reachability matrix, corresponding to different conditions under which failures are observed.

Finally, as an alternative to network tomography via active probe measurements, Mao et al. (2005) present a method for passive inference of link loss in sensor networks. Assuming that individual link loss follows a Bernoulli distribution, the authors estimate the loss probability using a sum-product message passing algorithm in a factor graph representing observed binary link states within a certain time frame. Additional network tomography algorithms can be found in the surveys by Coates et al. (2002) and Lawrence et al. (2006).

These methods often rely on a certain topology (Duffield et al., 2001; Tsang et al., 2003) and on specific protocols (such as multicast) for performing measurements (Duffield and Lo Presti, 2000; Lin et al., 2010). In practice, these requirements may be limiting with respect to the characteristics and measurement capabilities of the networked system in which the algorithms can be deployed. However, scalability issues and modeling are (for the most part) the main problems addressed and relate here both to the probing approach (for reducing the communication overhead; Cunha et al. 2009; Firooz and Roy 2010) as well as the computational complexity in the analysis of obtained measurements (Lin et al., 2010; Mao et al., 2005). Adaptability is to some degree addressed by the use of sliding windows (e.g. Shih and Hero 2001). In some of the approaches, the reliability aspect is addressed in terms of probabilistic detection thresholds (Shih and Hero, 2001) and probe planning, in order to minimize probing costs (Firooz and Roy, 2010) or to meet specifications on the detection error (Cunha et al., 2009).

Traffic Characterization and Anomaly Detection

An important part in network fault management is the processing and analysis of data streams, which enables early detection of network performance degradations, anomalies, and faults. This section contains several examples of common data processing methods used for characterization of the network behavior and detection of changes and anomalies.

Barford et al. (2002) apply wavelet analysis on Internet protocol (IP)-flows and simple network management protocol (SNMP) measurements for the purpose of characterizing network anomaly classes related to outages, flash crowds, attacks and measurement failures. Anomalies can be detected by analyzing different levels of time-frequency representations of the monitored signal relative to a deviation score, computed from the combined variability of the high- and mid-frequency parts.

Thottan and Ji (2003) introduce a different signal processing method for detecting changes in SNMP-management information base (MIB) variables via autoregressive modeling of time-series. Changes from the normally observed behavior are detected by comparing adjacent time-windows via hypothesis testing by the use of the generalized likelihood ratio (GLR). Somewhat differently, Lv et al. (2007) use a combination of GLR and wavelet transforms applied to network traffic in order to detect anomalies and failures. Under the assumption that segments of network traffic within a time-window can be modeled as autoregressive processes, the failure point can be detected and isolated based on hypothesis testing against a GLR-threshold, followed by wavelet analysis applied to the deviating segment.

For multivariate input, a common approach to detection of deviating traffic behavior is the subspace method involving e.g. PCA (Haykin, 1999), for the purpose of separating normal and anomalous traffic patterns. Lakhina et al. (2004a) employ the subspace method to detect and characterize anomalies in multivariate time-series of OD-flow traffic in terms of byte counts, packet counts and IP-flow counts. In a similar fashion, the authors Lakhina et al. (2004b) propose a method for detecting volume anomalies in network-wide multivariate OD-flows, followed by identification of the flow causing the anomaly. In both methods, the idea is to decompose the flows and use the top principal components for reconstruction, followed by measuring the residual between the original and reconstructed signals. The detection threshold is determined by computing a test statistic together with a specified percentile. Huang et al. (2007) outline a method for detection and identification of network disruptions in which network-wide correlations in routing updates are modeled. The authors use the PCA-subspace method for analyzing time-series data from border gateway protocol (BGP) updates, by expressing the original data as a linear sum of normal and anomalous terms, reconstructed from different sets of the principal components. For detection purposes, a Q-statistic (Jackson and Mudholkar, 1979) is computed and used to test whether the norm of the anomalous term exceeds a certain threshold.

Another approach includes modeling of network traffic flows in terms of sketch data structures, where a hash function maps each flow to a certain sketch. For

example, Li et al. (2006) suggest to model aggregates of IP-flows using sketches for anomaly detection. For each network router, local histogram-sketches are constructed in order to model flow features, which in turn are used for the construction of global sketches representing the empirical distributions of each flow feature across the network. Anomalies are detected via the PCA subspace-method using the computed entropies of the empirical distributions as input. For increased detection confidence, the outcome of the subspace analysis is decided via voting. Further, Schweller et al. (2004) make use of reversible hashing algorithms for the purpose of isolating the set of keys mapped to deviating signals (e.g. IP-flows) of the observed data stream. As the exact keys are not preserved when mapped into sketches, the authors here try to infer the set of suspect keys by analyzing the heavy bucket intersections in each hash table. Similarly, Krishnamurthy et al. (2003) use a sketch-based model used for change detection. Sketches representing associated network flows are constructed from a data stream. The current sketch is compared to a forecast sketch computed from historical sketches. A change is detected if the residual sketch crosses an internally computed threshold.

Other approaches to entropy-based anomaly detection has also been proposed - for example, Wang et al. (2010a) employ a hierarchical model used for entropy-time series analysis of monitoring data. Monitoring metrics are binned at each component in the hierarchy and represented as a monitoring sample at each time step. From the monitoring samples, global and local entropy-time series are created and aggregated for further analysis and anomaly detection by the use of the subspace method. Moreover, Wang et al. (2011) use a similar non-parametric statistical model but use hypothesis testing between distributions of discretized data observed in time-windows in order to detect changes. Essentially, the similarity between the distributions is computed in terms of the relative entropy. As the number of samples grows, the similarity measurements converge toward a chi-squared distribution. A fraction of the cumulative density function (CDF) of the chi-squared distribution is used as a threshold for detecting deviations from the normal data, and reflects the upper bound on the acceptable probability of false negatives.

Finally, some anomaly detection algorithms are developed under the assumption that various aspects of multivariate network traffic can be characterized in terms of mixture models following a certain probability distribution. An example is the work by Guo et al. (2006) who apply a Gaussian mixture model (GMM) for characterizing probabilistic correlations between network flow-intensities. For the purpose of estimating the parameters of each model (or cluster), the authors employ an online recursive variant of the EM. Once the model parameters are estimated, a boundary in the Mahalanobis distance following a chi-squared distribution is computed based on a coverage parameter, specified as the fraction of data points enclosed in the cluster, and used for anomaly detection. The coverage parameter basically specifies the acceptable probability for misclassification of a data point. A data point is classified as anomalous whenever its probability density is lower than the corresponding density of the boundary points. Hajji (2005) also present a finite GMM for detecting anomalies in network traffic. Each variable of the observed net-

work traffic is assumed to be generated from a weighted mixture of known regimes related to the time of day. Similarly to Guo et al. (2006), model parameters are estimated via an online-variant of the EM-algorithm. A baseline residual variable, following a normal distribution with zero mean, is used to model the difference between the estimated parameters of a component from one time-point to the next. Changes are detected by observing the GLR of the baseline residuals (representing normal conditions) given current and previous parameter estimates. Additionally, an expression is derived for how the detection threshold can be automatically set given an upper limit on the acceptable false alarm rate.

Addressing the scalability of data processing algorithms is essential for efficient network management. The diversity and increasing size of the data produced in the network from measurements and statistics, require computationally simple and adaptive algorithms that can be easily configured without detailed knowledge of the data stream. Many of the aforementioned methods are designed to reduce the computational complexity of data stream analysis (Krishnamurthy et al., 2003; Lv et al., 2007; Schweller et al., 2004; Wang et al., 2011), and are based on centralized, or possibly decentralized, processing. However, as the data volumes rapidly increase, online distributed data processing will be fundamental in future management systems to produce close to real-time results in a resource-efficient manner. Hierarchical models and aggregation as described in Wang et al. (2010a) is one such example of distributed analysis performed in the network with low communication overhead. Adaptation of the model estimates produced by the processing algorithms is also addressed in many of the papers, in terms of e.g. window-based techniques and decay functions (Hajji, 2005; Thottan and Ji, 2003; Wang et al., 2011). Online processing of data (as in Guo et al. 2006; Hajji 2005; Krishnamurthy et al. 2003; Wang et al. 2010a) can further contribute to real-time adaptability, instead of fixed interval processing of large data sets. Moreover, many of the papers include automatic definition of thresholds and boundaries, specified as fractions or probabilities, for simplified configuration and increased robustness in performance (Guo et al., 2006; Huang et al., 2007; Lakhina et al., 2004b; Lv et al., 2007; Wang et al., 2011).

Event Correlation and Pattern Mining

Temporal and spatial event correlation and pattern mining are useful tools for log-analysis and alarm-filtering purposes, and can also be used for detecting faulty network states from baseline models as well. For example, Julisch (2001) applies a heuristic clustering algorithm for alarm filtering and root-cause analysis. The aim is to cluster similar alarms into general alarms. The problem of finding the clusters that minimize the total average distance between similar alarms within a cluster is NP-complete. For this reason, the authors apply attribute-oriented induction (Heinonen and Mannila, 1996) in order to obtain an approximate solution.

Fu and Xu (2007) propose an algorithm for clustering failure events with respect to temporal and spatial correlation, that can be used for failure prediction. Temporal correlation between failure events at different time-scales is computed using

a spherical covariance model. A probabilistic model is employed for clustering spatially correlated failure events, and for analyzing the dependency between observed types of failures and certain network nodes. Failure statistics for nodes within and across clusters are aggregated in a hierarchical manner and used for failure prediction on different levels (system-wide, cluster-level, and node-level). Moreover, Jiang and Cybenko (2004) investigate several approaches to modeling spatio-temporally correlated observations of distributed events mapped to various attack scenarios. Identification of temporal correlations is formulated as a target tracking problem and modeled using Kalman-filters or hidden Markov models (HMMs) for continuous and discrete cases, whereas spatial correlations are modeled as Bayesian networks. The authors suggest a joint probabilistic correlation framework for modeling both temporally and spatially correlated events from multiple observation spaces, for the purpose of detecting the most likely attack scenario given the observations.

Ye (2000) uses a discrete-time first order Markov chain model for discovering the temporal profile of normal behavior observed from audit data logs. The observed short-term behavior of the system is analyzed and tested probabilistically against the learned long-term normal profile in order to detect anomalous behavior and intrusion attempts. A probabilistic expression is derived for testing how well the observed sequence matches the learned Markov model - if the support is below a certain probabilistic threshold an intrusion has been detected. Similarly, Yamanishi and Maruyama (2005) apply an on-line algorithm for detecting symptoms of failures and for discovering sequential alarm patterns in syslogs. The behavior of the syslog time-series data is modeled as a mixture of HMMs and the parameters are estimated via EM. The number of mixture components and the estimated parameters are dynamically adjusted over time, which keeps the model up-to-date under varying conditions. Anomalies are detected by the use of a universal test statistic based on Shannon information - the detection threshold is optimized relative to the histogram of observed scores. Khreich et al. (2009) evaluate a model of multiple HMMs for detecting anomalies in system calls to the operating system kernel. The authors address the problem of improving the detection performance when estimating the parameters for multiple HMMs in the case when the exact number of hidden states is unknown. The idea is to learn several HMMs (via EM) with different number of hidden states, and identify the models that best discriminate between anomalous and normal states. The model selection is performed by fusing the HMM classification responses in the receiver operating characteristics (ROC) space, followed by selecting the HMMs that maximize the area under the ROC curve (AUC).

The use of event correlation and pattern mining methods is mainly to identify and model the relation between cause and effect (i.e. network states and failures) (Julisch, 2001), and to recover underlying source models that can explain a certain system behavior (Ye, 2000). However, the computational complexity increases with the data size and the number of model parameters, which makes many existing approaches practically limited for large-scale analysis. Scalability issues can be addressed by the use of more distributed methods - for example, Tasoulis

and Vrahatis (2004) and Bhaduri and Srivastava (2009) suggest methods for distributed clustering and a local distributed EM-algorithm for data mining purposes in peer-to-peer networks, respectively. From the adaptability perspective, the system behavior and failure patterns can often be assumed to be more or less static, but in the cases where there is a certain degree of drift, on-line learning and adaptive mechanisms can be applied, as described for example in Yamanishi and Maruyama (2005) and Khreich et al. (2009). From a reliability perspective, the algorithms generally require relatively detailed knowledge about the domain and understanding of the relation between a certain configuration and the result (e.g. Julisch 2001; Yamanishi and Maruyama 2005). New infrastructures, protocols and applications will lead to increasingly complex models of the system behavior, and therefore the development toward more or less zero-configuration methods will be of increasing importance in order to reduce configuration efforts and management costs.

Fault Localization and Diagnosis

A common method to fault isolation is hypothesis updating, performed via e.g. incremental selection of the most informative measurements given a set of symptoms, that effectively can be used to verify or identify a certain fault or network state with low communication overhead. For example, Rish et al. (2005) provide a model for real-time diagnosis where the most informative measurements (or probes) are incrementally selected under the current failure hypothesis, such that the fault mapped to the observed network state can be identified with as few probes as possible. The next probe in the test sequence is selected based on an information-theoretic framework given the outcome of previous probes. The most likely diagnosis is found through probabilistic inference in a Bayesian network in terms of the maximum probable explanation (MPE), and is obtained via local approximation for reducing the computational complexity. Similarly, Steinder and Sethi (2004) perform incremental hypothesis updating using a probabilistic symptom-fault map expressing the causal relation between a fault and observable symptoms. Differently from the aforementioned model, the most probable set of faults are incrementally isolated to a set of hypotheses (ranked by a goodness of fit function) as symptoms arrive, instead of actively sending probes. In an earlier work by Steinder and Sethi (2002), identification of the most likely cause of a detected network state is performed by the use of a layered belief network model, encoding the mapping between possible failures and services or network functions. The authors investigate several approximation approaches for finding the MPE, such as bucket elimination and Bayesian inference techniques.

Tang et al. (2005) propose a combination of passive and active methods for isolating faults via probing, involving heuristic fault-reasoning and fidelity measurements for the purpose of decision making and evaluation of which faults contribute the most to observed symptoms. The localization process is terminated when a satisfactory hypothesis has been found according to a probabilistic decision threshold. The hypothesis is confirmed by performing a minimum-cost set of actions that

verify the existence of the symptoms. A somewhat different approach to anomaly detection and localization is suggested by Barford et al. (2009). Anomalies are here breaches against service level agreements (SLAs) on link performance (loss, jitter, delay). Probes are actively injected on selected paths based on certain counters in order to reduce the link load caused by probing. Localization of a fault is performed by narrowing down the set of suspected paths, selected for additional probing in order to isolate the origin of the anomaly. Determination of which paths to probe is formulated as the minimum set cover problem, and is solved using heuristics.

For localization of faults, Kandula et al. (2005) provide a method for isolating a set of failed links to a specific group of logical entities at the IP-layer. Failure probabilities, as well as the correlation between failed links and the logical groups, are modeled using a Bayesian network. The authors use a fast approximation algorithm for inferring a diagnosis in polynomial time, under the assumption that an event is a subset of the most probable failures, thereby reducing the problem space. Finally, Kompella et al. (2007) outline a method for detection and localization of silent faults, appearing from the interaction between the multiprotocol label switching (MPLS) and IP layers. Failure detection is performed by observing the losses of injected probes between OD-pairs within a specified time-window. Fault localization is based on the spatial correlation between OD-pairs sharing common link segments. The aim is to identify multiple failures and their causes from the observations, by iterative partitioning of the network links associated with the largest set of observations.

As observed from these examples, resource-efficiency is mainly addressed by reducing the number of probes and steps needed to reach a diagnosis (e.g Barford et al. 2009; Rish et al. 2005; Steinder and Sethi 2004; Tang et al. 2005). Additionally, different approximation approaches for inferring diagnoses from the observed symptoms are employed to reduce the computational complexity (Kandula et al., 2005; Rish et al., 2005; Steinder and Sethi, 2002). From an adaptability perspective, continuous updating of the models due to structural changes over time are generally not addressed as the set of faults, as well as the dependency relations between observations and failures, are assumed to be fixed. Initiating diagnostic tools currently requires significant configuration efforts based on expert knowledge of the domain, but would be simplified by adaptively learn the structure from data over time in a varying system. Moreover, noise and inconsistencies in the observations are often taken into account in these models, for the purpose of increasing the robustness in obtained hypotheses (Kandula et al., 2005; Kompella et al., 2007; Rish et al., 2005; Steinder and Sethi, 2004). Inconsistency-handling in combination with incremental hypothesis updating (as in Rish et al. 2005; Steinder and Sethi 2004) that terminates in line with a probabilistic confidence threshold (Tang et al., 2005) can further increase the reliability in obtained diagnoses.

Chapter 3

General Approach and Problem Areas Addressed

This chapter describes probabilistic fault management, the general principles of PNM, the problem areas addressed, and modeling aspects of the algorithms included in this thesis. The purpose is to set context and provide necessary preliminaries and motivations behind the algorithm designs, in addition to included papers. A summary of the included papers is provided in Chapter 4, and further details of each algorithm are provided in the papers appended to this thesis.

3.1 Scope and Design Principles of Probabilistic Fault Management

The main functions of network fault management are to provide resilience to network failures, to enable and support proactivity against performance degradations, and to provide accurate information about the network state. The term *fault management* here refers to the cycle of:

- *data collecting and modeling*, given input from various network monitoring tools and logs;
- *detection* of network faults, changes, and anomalies;
- *isolation*, meaning fault localization, diagnosis, and root-cause analysis;
- *recovery*, performed manually or through automatic recovery actions triggered by alarms or other types of signals.

Data collecting and modeling encompasses processing and probabilistic modeling of measurements, log data, and other types of input from manual or automatic readings of the network state. *Detection* is the process of finding faults, changes, or anomalies in monitored equipment, based on observing the network behavior in relation to baseline models representing the current regime or normal conditions.

Here, a *fault* is an unplanned event or state that appears in the network as an effect of physically or logically malfunctioning network entities; a *change* is an abrupt or gradual transition from an observed behavioral trend to another (caused by e.g. a performance degradation); an *anomaly* is a deviation from a baseline model representing normal system behavior. *Isolation* refers to the processes of finding and explaining the origin of the detected fault, change or anomaly. More specifically, *localization* is the process of pin-pointing the fault-location to any hardware or software in the network (such as a link or node on a monitored path). A localized fault or a faulty network state may be classified through *diagnosis*, or may be further analyzed with *root-cause analysis* explaining the origin of the faulty condition, e.g. through event correlation and identification of common alarm patterns. *Recovery* actions include reporting about the network state as well as signaling to other parts of the network for the purpose of triggering relevant management actions, such as reconfiguration or optimization of available resources.

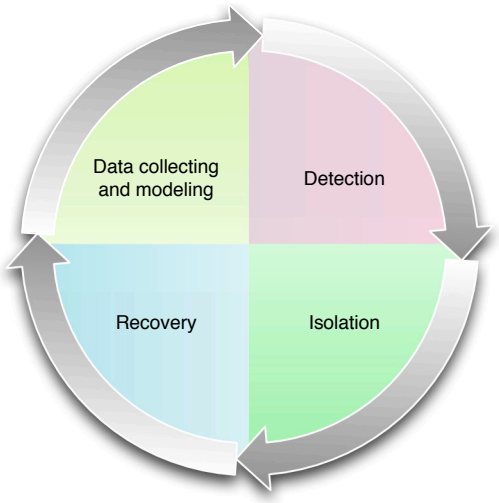

Figure 3.1: Summary of the fault management cycle.

The largest contribution of this thesis consists of a set of fault management algorithms addressing decentralized (or distributed) detection of network connectivity faults (Paper III and VII), performance degradations and changes (Paper IV and VIII), as well as incremental diagnosis (Paper I) and event correlation (Paper II and VI). Each algorithm covers different parts of the fault management cycle, with emphasis on data collecting and modeling, detection and isolation (Figure 3.1). Recovery actions are within the scope of external signaling for automated or manual correction of the problem.

Principles of Probabilistic Network Management

The algorithms are designed in line with the principles of the PNM-paradigm, targeting *scalability*, *adaptability* and *reliability*. More specifically, we define in this thesis the concept of the PNM-paradigm (Paper V), where such algorithms

- are based on probabilistic modeling of the network states, in terms of parametric and non-parametric models of varying complexity;
- allow for probabilistic input and output, being probabilities or parameters of a probability distribution;
- can autonomously adapt the algorithm behavior and adjust models relative to observed variations in the network environment;
- can be configured by high-level performance specifications, expressed as deterministic or probabilistic performance objectives, limits or guarantees.
- may operate in a distributed or decentralized manner.

The first property is inherent to all PNM-algorithms, whereas the remaining properties are optional depending on the application. Probabilistic modeling of network states includes, for example, the parameter estimation of a simple distribution or more complex models such as mixtures (Bishop, 2007).

A probabilistic input or output here refers to, for example, a conditional probability of observing a probe response after a series of lost probes; a fraction of a CDF; or, simply the parameters of a probability distribution, such as mean and variance. Whereas deterministic limits are strict (and related directly to e.g. the number of lost probe responses, bandwidth, timeouts, etc.), the idea of using probabilistic input or output is the possibility to express and reflect uncertainty about the network state. Uncertainty can be expressed through probabilistic thresholds, limits, or guarantees, or by specifying a prior distribution over the model parameters to be estimated. Partial or prior knowledge about the system can thus be accounted for in the algorithm behavior or model, allowing for more accurate or reasonable results when the network observations are few or insufficient.

A PNM-algorithm is typically designed to autonomously adapt in line with high-level goals, specified by both deterministic and probabilistic input. The algorithm can for example vary the number of measurements performed in order to achieve tolerable levels of uncertainty or estimation error. Essentially, planned network performance or management objectives can be better reflected by deterministic and probabilistic high-level goals, compared to micro-managing the low-level parameters of the algorithm. In addition to simplified algorithm configuration, such high-level objectives contribute to improved performance reliability and predictability, as well as to increased robustness to noise.

Finally, a PNM-algorithm operates mainly in a decentralized or distributed manner for the purposes of scalability, robustness, resilience and accuracy, achieved

by modeling the locally observed network behavior. The scope of the decentralized operation depends on how well the modeling or analysis can be divided between network nodes, and the computational capacity of the nodes in the network. In addition, the degree of assumed or known statistical independence between observed network states, events, measurements or equipment also contributes to the level of decentralization possible to achieve.

Compared to current, highly centralized and deterministic network management approaches, the decentralized PNM-paradigm offers autonomous, resource-efficient and robust management methods that are configurable in terms of high-level performance objectives, which altogether contributes to reduced management costs. The drawbacks are in the scope of less strict control over managed devices and network equipment; no hard guarantees on the accuracy of measurements and observations; and occasionally, suboptimal solutions to management problems. These side-effects follow from the introduction of uncertainty and decentralized operation. Nevertheless - the next generation of networked systems requires management solutions that support flexible operation of network infrastructures and services. To ensure service quality, network management actions need to be dynamically executed within extremely short time-scales (i.e. milliseconds and seconds). Employment of the deterministic management paradigm will not, under such conditions, provide the operator with self-managing and automated processes at the level needed to ensure service quality at low management costs. For these reasons, it is rather likely that we will observe a paradigm shift toward a less exact but, in the overall perspective, more cost-effective and robust approach to network management in line with the PNM-concepts.

Further discussions about these aspects and the PNM-paradigm can be found in Paper V. An investigation about some of the practical differences between using a probabilistic fault detection algorithm compared to deterministic monitoring is presented in Paper VII.

Addressing Scalability, Adaptability, and Reliability

The design goals of scalability, adaptability and reliability cover many of the problems common to different challenges of network management (see Chapters 1 and 2). The algorithms included in this thesis map to different aspects of these goals - in general, they are designed for distributed or decentralized operation and are self-adjusting in order to produce reliable results under changing network conditions.

More specifically, scalability is here addressed mainly in terms of reducing the link load as well as the computational demands. The algorithms are intended to run either in a distributed (Paper III, IV, VI and VII) or decentralized manner (Paper I, II and VIII), which contributes to efficient usage of resources in the network, deployment flexibility, and accuracy in the produced output from the models with respect to local network conditions and temporal changes. The use of self-adjusting probing intervals in Paper III, IV and VII, reduces the overall probing traffic in the network, as the probing intervals adapt to local link conditions. Additionally, the

probabilistic models used for network monitoring (Paper III, IV, VII, and VIII) are here based on a limited set of simple counters, which allows for obtaining parameter estimates at a very low computational cost.

Many of the algorithms include a mechanism for gradual adaptation to new data, which is used to address the temporal aspect of adaptability (see Section 1.2). In Paper IV, VII and VIII, the parameter estimates are gradually adjusted such that recent probe measurements have stronger influence on the estimates than older measurements (see Section 3.2). Similarly, the parameters of the data model used for incremental diagnosis (Paper I) are gradually updated as new symptoms or states are observed.

The reliability goal is addressed by designing the algorithms such that high-level objectives are used to express the desired performance of the algorithm, instead of directly controlling its low-level parameters. In the case of incremental diagnosis (Paper I), the length of each query-session adapts to the input and the desired confidence in the obtained hypothesis. In Paper III, the fault detection algorithm continues to probe a suspected node until a probabilistic level of the acceptable false alarm rate has been reached. Moreover, the probing algorithms in Paper III, IV and VII, autonomously adjust the probing intervals relative to locally observed link delays and a probabilistic threshold. This simplifies configuration as the value of the probabilistic parameter can be applied to the entire network while local network conditions are taken into account.

3.2 Data Collecting and Modeling

From a modeling perspective the fault management algorithms in this thesis can be divided into two categories. In the first category of algorithms, probe measurements are used as input to the probabilistic models. A *probe* is a test packet (such as the Internet control message protocol (ICMP); Postel 1981) sent between nodes in the network for the purpose of testing the availability of a certain network equipment or measuring the quality of the connection[1]. In the second category of algorithms, measurements, logs, observed symptoms or states, are modeled and obtained by request either manually or automatically, in a passive or active manner. For example, data may be obtained from active measurements (Rish et al., 2005) or manual input (as in Paper I) and used for diagnosis, or may be retrieved from logs for event correlation (such as in Paper II and VI).

The remainder of this section covers the general approach for modeling probe measurements used in Paper III, IV, V, VII and VIII. Modeling aspects related to Paper I, II and VI are covered in Sections 3.6-3.8.

[1] In Paper III and VII the term heartbeat probing or monitoring refers to frequent signaling as in sending e.g. probe packets, but should not be confused with the "heartbeat monitoring"-method, in which a central node collects frequent signals from network devices (Sterritt, 2003).

Modeling Link Metrics From Probe Measurements

For the purpose of modeling link metrics such as delay and loss from probe measurements, we here estimate the parameters of Gamma and Bernoulli distributions, respectively. Estimating the parameters of a certain probability distribution is in many cases a memory-efficient way to model observed stochastic behaviors, but requires analysis and prior knowledge of the data in order to select a matching probability density function. A non-parametric model may allow for relaxed assumptions about the underlying distribution (and may therefore be an attractive modeling alternative in some applications), but instead the number of free parameters needed to accurately model the observed behavior needs to be decided (e.g. Haykin 1999). From this perspective, parametric models can in some cases be estimated with less memory and computational capacity in comparison with non-parametric modeling, which is advantageous especially when resource-efficiency is of high importance in various types of network equipment.

It is assumed that the link delay modeled here includes propagation and queuing delays, as well as other processing times. Previous studies (e.g. Choi and Limb 1999; Hernandez and Magafia 2007; Kalman and Girod 2004; Mukherjee 1992) indicate that the Gamma distribution (often used for modeling queuing delays) can be used for modeling different types of delay at different network levels. For modeling of link delays, the Gamma parameters, scale α and shape β, are estimated via MoM (Bishop, 2007; Gut, 2009; Jaynes, 2003), based on the first $s_1 = E(X)$ and second $s_2 = E(X^2)$ moments of measured delay $x_j \in X$ (Kumar, 2006):

$$\alpha = \frac{s_2 - s_1^2}{s_1}, \quad \beta = \frac{s_1^2}{s_2 - s_1^2} \tag{3.1}$$

There are a number of alternative ways to estimate the parameters, involving e.g. numerical integration (Damsleth, 1975; Miller, 1980) or Markov chain Monte Carlo (MCMC) sampling (Pradhan and Kundu, 2011). These methods generally provide more accurate estimates (compared to MoM that for example is sensitive to outliers), but are computationally more demanding. The MoM-approach is here used as a trade-off between accuracy and computational requirements, which allows for algorithm deployment on a wider range of network equipment with varying computational limitations (such as sensors).

We define link loss as the fraction of missed probe responses relative to the amount of sent probes. Lost probe responses usually appear from transmission disturbances (causing damaged packets) and traffic congestion (such that packets are dropped due to buffer overflows). A basic and common approach to modeling link loss is based on estimating the success rate parameter of the Bernoulli distribution (e.g. Fragouli and Markopoulou 2005; Gaeta et al. 2003; Yajnik et al. 1999; Yang et al. 2001) from observed probe responses. Bernoulli-distributed link loss can be assumed under the condition that the probes are not temporally correlated and do not have significant impact on the link load (Bolot, 1993; Yajnik et al.,

1999). In practice, these requirements are fulfilled if the delay between the probes is sufficiently long, which is here assumed.

The link loss P_D is here modeled in terms of estimating the λ-parameter of the Bernoulli distribution, such that $P_D = 1 - \lambda$. The maximum likelihood estimate (MLE) of λ corresponds to the average number of successes of n observed probe responses $x_j \in \{0, 1\}$ (Bishop, 2007):

$$\lambda = \frac{1}{n} \sum_{j=1}^{n} x_j \tag{3.2}$$

Further details regarding how the delay and loss models have been applied can be found in Paper III, IV, VII, and VIII. In the next section, modified estimators used for adaptive learning are presented.

Adaptation to New Data

In practice, the network behavior tends to drift over time due to various changes in the network environment, such as shifting traffic patterns, maintenance, varying user behavior, new applications and software, etc. For the purpose of maintaining accurate models, additional mechanisms are needed that can put more trust to recent measurements while older data points are gradually forgotten. Common approaches for adapting estimates to new observations include the application of decay functions on a set of observations or measurements in a sample window (Cohen and Strauss, 2006; Hajji, 2005), aggregation within fixed (Datar et al., 2002) and variable window sizes (Bifet and Gavaldà, 2007), exclusion (Gençata and Mukherjee, 2003) or replacement of expiring data points (Dasu et al., 2009).

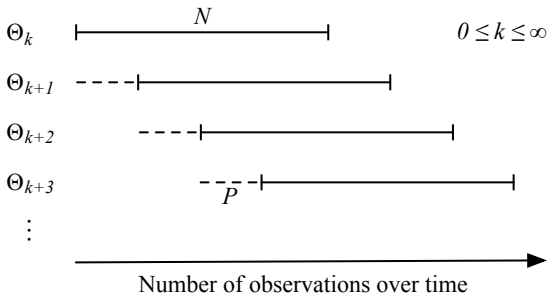

Figure 3.2: Overlapping estimates.

In the probing algorithms described in Paper IV, VII and VIII, the parameter estimates are adjusted via *overlapping estimators*. The approach is inspired from Bayesian estimation techniques and is based on using information from previous estimates as prior input to each new overlapping estimator, Θ_k. In general, the overlapping estimators are configured by two parameters, related to the number of

samples N in each estimator, and the number of samples P included in the prior input from one estimator to the next (Figure 3.2).

Adaptation of the Gamma parameters scale α and shape β is performed by updating the moments s_1 and s_2 in Equation 3.1 similar to an *m-estimate* (Mitchell, 1997), where the prior estimate is weighted in terms of w samples (usually set $w = 1$) relative to the current number n of observed delay measurements x_j:

$$s_1^{(i)} = \frac{\sum_j^n x_j + w s_1^{(i-1)}}{n+w}, \quad s_2^{(i)} = \frac{\sum_j^n (x_j)^2 + w s_2^{(i-1)}}{n+w}, \tag{3.3}$$

The impact of older observations is thus reduced exponentially, which is similar to controlling the importance of each observation in accordance with some decay function (Cohen and Strauss, 2006). In comparison with explicitly using a decay function, overlapping estimators are computationally simpler to use and do not require storage of samples - instead, the memory utilization scales with the number of counters needed for computing the parameter estimates.

In the case of link loss estimation, the success rate parameter λ of the Bernoulli distribution is updated via a Bayesian estimator, obtained by using the Beta distribution with parameters (a^*, b^*) as a conjugate prior (Bishop, 2007) distribution over the $\lambda^{(i)}$. Given that $E(X) = \frac{1}{n}\sum_{j=1}^n x_j$, and $a = a^* + nE(X)$, $b = b^* + n - nE(X)$ (e.g. Lee 2004) and that $E(\lambda^{(i)}) = \frac{a}{a+b}$ and $w = \frac{b^*}{(1-\lambda^{(i-1)})}$, the success rate can be updated as follows:

$$E(\lambda^{(i)}) \quad = \quad \frac{nE(X) + w\lambda^{(i-1)}}{n+w}, \tag{3.4}$$

The Bayesian model for updating the link loss estimators was used in Paper VII and VIII. A similar approach of using conjugate priors for adjusting the models to new data has also been used for the purpose of incremental diagnosis, described further in Section 3.6 and Paper I.

Autonomous Adjustment of Probing Intervals

The algorithms described in Paper III and IV are based on adaptive probing intervals, meaning that the delay between each probe adjusts to the measured link delays. The probing interval is controlled by a fraction τ of the CDF computed from the estimated Gamma distribution $P(x; \alpha, \beta)$:

$$F(\Delta x) = \int_0^{\Delta x} P(x)dx = \frac{1}{\Gamma(\beta)}\gamma(\beta, \frac{\Delta x}{\alpha}), \tag{3.5}$$

where γ is the lower incomplete Gamma function (Bishop, 2007). The CDF here indicates at which delay Δx the fraction τ of all probe responses has been observed (Figure 3.3). The probing interval is set given a fraction of the inverted CDF

$$c_\tau F_{cdf}^{-1}(\tau) \tag{3.6}$$

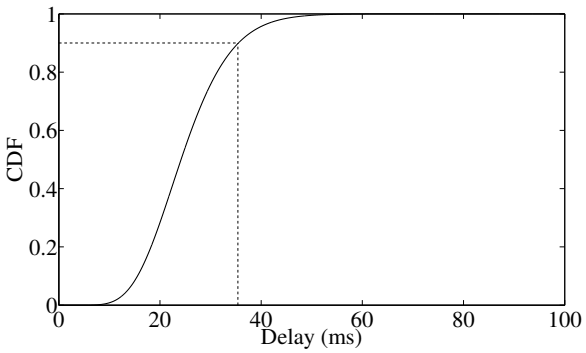

Figure 3.3: An example of the selection of adaptive probing intervals based on the fraction τ (dotted line) of the Gamma CDF of observed link delay.

and a multiplier c_τ, which can be used to further control the time-scale of the interval. This parameter can also be regarded as a "cost" of sending a probe. However, the meaning of c_τ as being a cost parameter is a matter of interpretation, and does not directly reflect an expense that in this sense should account for the observed resource availability and usage in the network.

The advantage of autonomous and adaptive probing intervals was demonstrated in Paper VII - the study show that the resource usage can be significantly reduced, compared to when ordinary monitoring methods with fixed probing intervals are used (e.g. Andersen et al. 2001; Postel 1981).

3.3 Change Detection Using Overlapping Estimators

A change is a deviation from the currently observed behavior or trend, and can be positive (e.g. reduced link load) or negative (e.g. increased link loss). Changes may appear as direct effects of local failures, or may be symptoms of faults in remote parts of the network. Apart from being an important tool to proactively manage faults and performance degradations, change detection is also relevant for resource management in e.g. cloud infrastructures, as it can be used for initiating re-allocation of compute, storage, and network resources (Murray et al., 2012).

Changes are detected by measuring the symmetric Kullback-Leibler divergence (KLD) (Kullback and Leibler, 1951) between the current and previous overlapping estimates Θ_i, Θ_j (see Section 3.2), followed by comparing the divergence against the threshold η:

$$D(\Theta_i||\Theta_j) = D_{KL}(\Theta_i||\Theta_j) + D_{KL}(\Theta_j||\Theta_i) > \eta \tag{3.7}$$

The expressions for the asymmetric KLD of the estimated Gamma and Bernoulli parameters $\Theta = (\alpha, \beta)$ and λ, respectively, are shown in Equation 3.8 (e.g. Kwitt

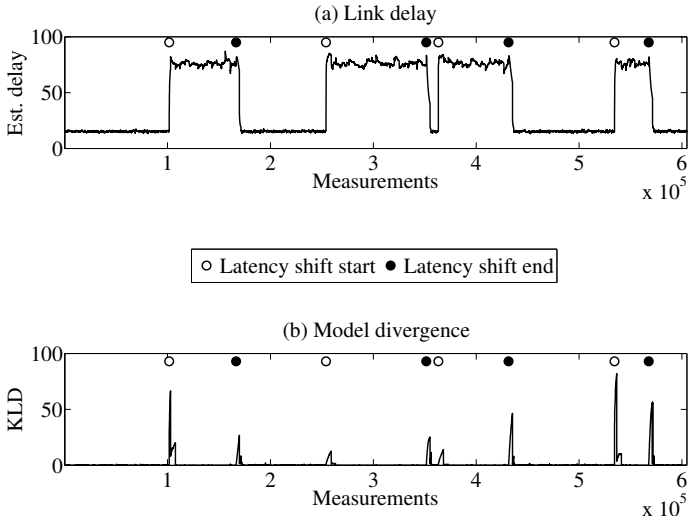

Figure 3.4: Example of change detection in the estimated link delay and the observed KLD when using overlapping estimators.

and Uhl 2008) and Equation 3.9 (e.g. Roychowdhury and Khurshid 2011). In this case, it is not necessary to differentiate the direction of changes between the overlapping estimates, and therefore the symmetric KLD is used for change detection.

$$\begin{aligned} D_{KL}(\Theta_i||\Theta_j) &= \psi(\beta_i)(\beta_i - \beta_j) - \beta_i + \\ &\quad \log\frac{\Gamma(\beta_j)}{\Gamma(\beta_i)} + \beta_j \log\frac{\alpha_j}{\alpha_i} + \frac{\alpha_i\beta_i}{\alpha_j}. \qquad (3.8) \\ D_{KL}(\lambda_i||\lambda_j) &= \lambda_i \log\frac{\lambda_i}{\lambda_j} + (1-\lambda_i)\log\frac{1-\lambda_i}{1-\lambda_j} \qquad (3.9) \end{aligned}$$

Alternative methods for detecting changes in the parameter estimates include e.g. likelihood ratio tests (Hajji, 2005; Neyman and Pearson, 1933; Thottan and Ji, 2003), Bayesian information criterion (BIC) (Chen and Gopalakrishnan, 1998; Liu and Yang, 2011; Schwarz, 1978), or moving average methods (Basseville and Nikiforov, 1993). The KLD is practically useful for measuring the degree of dissimilarity between estimated parameters, since closed form expressions exist for many continuous and discrete distributions. Moreover, it provides a robust estimate of the relative information loss between two distributions, based on the *expected* log-likelihood ratio (Eguchi and Copas, 2006).

Further details about the change detection algorithm and how it has been used can be found in Paper IV and VIII. An example of how changes in the estimated link delay are detected is shown in Figure 3.4.

3.4 Distributed Fault Detection with Controllable Rate of False Alarms

In the context of detecting connectivity problems between nodes, simple probing algorithms are mainly configured in terms of a probing interval and a detection threshold related to the number of consecutively lost probe responses (Hofstede et al., 2011; ITU-T Rec. Y.1731, 2008; Postel, 1981). Deterministic monitoring algorithms operating in line with strict limits or thresholds exhibit several drawbacks, such as unpredictable detection performance, false alarms, resource inefficiency (e.g. caused by frequent probing), and sensitivity to varying network conditions. In Paper III and VII, a probabilistic and self-adjusting algorithm for distributed fault detection and localization is presented.

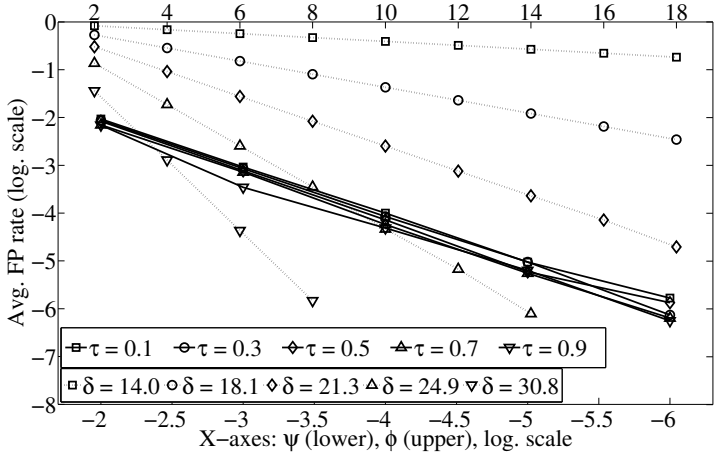

Figure 3.5: Comparison of performance predictability between a probabilistic (solid) and deterministic (dotted) monitoring algorithm for varying parameters ψ, τ of the probabilistic algorithm, and probing interval δ (time units) and detection threshold ϕ (i.e. number of lost probes) of the deterministic algorithm. The example shows that the probabilistic algorithm performs with higher predictability in the false positives rate than the deterministic algorithm under varying conditions (see Paper VII and V).

The approach is based on modeling the link delay $P(x)$ and loss P_D as Gamma and Bernoulli distributions, respectively, as described in Section 3.2. The measurements are performed in a distributed manner via probing between the nodes in the network. The algorithm takes as input the parameters of the probing interval and the detection threshold, and operates under two types of probing intervals controlled by two parameter sets, τ, c_τ and θ, c_θ (see Section 3.2). The first set of parameters corresponds to longer probing intervals used during normal operation when no symptoms of faults have appeared, whereas the second set is mapped to significantly shorter intervals in order to verify a suspected fault (i.e. a connectivity

problem). The use of two probing schemes can further reduce the link load induced by probing traffic, in addition to the autonomous adjustment of the interval - a similar approach was used by Andersen et al. (2001), but with fixed probing intervals. Moreover, the detection threshold is indirectly controlled by ψ, reflecting *the acceptable rate of false alarms* observed on a link.

For the purpose of detecting a fault, it is assumed that the probability of observing a probe response $R_{\Delta x}$ within delay Δx can be computed from the joint probability distributions of observed link delay and loss (Equation 3.10). Assuming that probe responses are statistically independent, the detection mechanism is based on the probability of not observing a probe response given a set of already sent probes $\boldsymbol{\Delta x} = \{\Delta x^{(1)}, \Delta x^{(2)}, \ldots, \Delta x^{(n)}\}$ (Equation 3.11).

$$P(R_{\Delta x}) = (1 - P_D) \int_0^{\Delta x} P(x; \alpha^*, \beta^*) dx \tag{3.10}$$

$$P(\neg R | \boldsymbol{\Delta x}) = \prod_i^n \left(1 - P(R_{\Delta x}^{(i)})\right) < \psi \tag{3.11}$$

A fault is detected when the probability in Equation 3.11 has reached below ψ. When a fault has been detected toward a node, the problem is isolated to a link or node failure, based on collaborative fault localization between the neighbors of the suspected node.

The actual number of lost probe responses needed for detection adjusts relative to ψ and the observed link behavior. An example of the performance predictability is shown in Figure 3.5, where the observed amount of false alarms on one link well matches the acceptable false alarm rate ψ. In practice this means, for example, that a high degree of link loss or a small value on ψ will require more probes before a fault with certainty can be concluded. Self-adjusting probing behavior (in terms of probing intervals and detection) offers several advantages over strict parameter settings, such as overall reduction in bandwidth usage, increased robustness to false alarms, and improved performance controllability.

Details of the distributed fault detection and localization method can be found in Paper III. An investigation of the practical differences between this fault detection algorithm and a deterministic probing algorithm was presented in Paper VII.

3.5 Localizing Performance Degradations from End-to-End Measurements

Communication networks contain passive, traffic-forwarding devices (such as network switches) and significantly fewer measurement endpoints capable of performing various management actions. The level to which monitoring can be performed in passive parts of the network, depends on the availability and topological placement of dedicated and expensive probing equipment. The general strategy for increasing the observability in the network is to derive individual link metrics based on

comparisons of probe measurements traversing common paths, and is referred to as network tomography (Coates et al., 2002).

In many of the existing network tomography methods (see also Section 2.3), multicast tree measurements are used for modeling of link delay and loss (Cáceres et al., 1999; Coates et al., 2002; Duffield and Lo Presti, 2000; Lin et al., 2010). As multicast protocols are rarely used in practice, a wide range of unicast-based probing approaches to modeling of link metrics exists as well (Burch and Chase, 2005; Coates and Nowak, 2000; Duffield et al., 2001; Tsang et al., 2001). The decentralized fault localization algorithm in Paper VIII, transmits simple unicast end-to-end loopback messages for probabilistic modeling of link delay and loss. Compared to many of the aforementioned approaches, the method is simple, adaptive, and memory-efficient, as only counters for obtaining MoM-estimates are needed. From the end-to-end measurements, changes in delay and loss can be directly localized to an individual link on the monitored path, based on the change detection approach outlined in Sections 3.2 and 3.3.

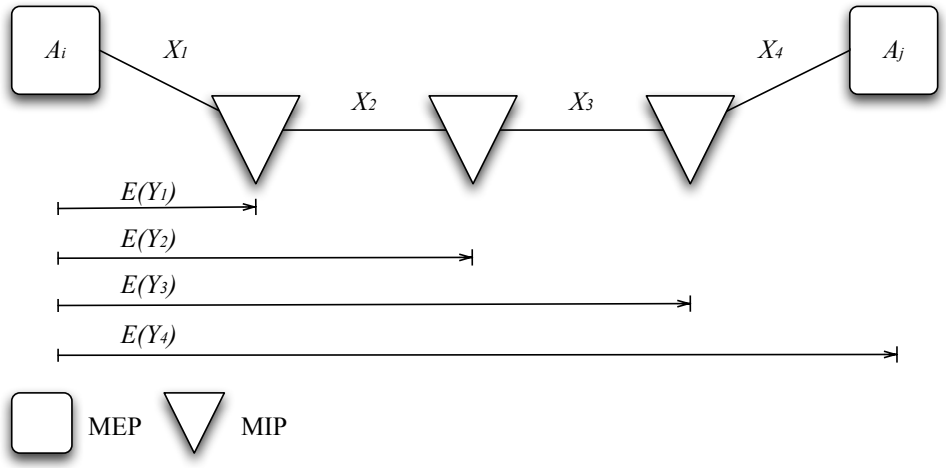

Figure 3.6: A probe test performed between active MEPs, consisting of successively incremented probes (one way or round-trip) over passive MIPs.

The link delay and loss models in Paper VIII are based on input from incremental unicast probes on a monitored path. Segments $Y_i = \sum_{j=1}^{i} X_j$ on the monitored path consisting of X_i $(i = 1 \dots n)$ links are successively measured, either via a *hop-by-hop probes* or a *probe burst* (Figure 3.6). The probing strategy partially depends on the available OAM tools offered by the network protocol. The degree to which the measurements are consistent affects the quality of the estimates, and depends on the probing strategy and the measurement conditions. A forwarded hop-by-hop probe naturally keeps the successive measurements consistent toward the endpoint

of the monitored path, whereas the measurement consistency in a probe burst depends on the level of stationarity during the measurement session. The probing strategy and the measurement conditions, together with the statistical modeling assumptions related to a certain link metric, have influence on how the parameters for individual link segments are estimated. For these reasons, two models in the case of link delay addressing stationary and non-stationary measurement conditions are proposed:

$$E(X_i) = E(Y_i) - E(Y_{i-1}) \tag{3.12}$$

$$Var(X_i) = Var(Y_i) + Var(Y_{i-1}) - 2Cov(Y_i, Y_{i-1}) \tag{3.13}$$

$$Var(X_i) = Var(Y_i) - Var(Y_{i-1}) \tag{3.14}$$

The first link delay model (Equation 3.13) accounts for covariance, as it is assumed that the successive measurements are statistically dependent and consistent, which implies stationary measurement conditions if probe bursts are used. In the second link delay model (Equation 3.14), it is assumed that the measurements are statistically independent. This model is used when the successive link delay measurements are obtained via probe bursts under non-stationary measurement conditions. For both models, the mean is derived as in Equation 3.12.

Under the condition that link loss follows a Bernoulli distribution (see Section 3.2), it is assumed that the measurements are statistically independent and that a probe response from one hop to the other is conditionally independent from the previous successful outcomes. Provided that $Y_i^D = \prod_{j=1}^{i} X_j^D$, the loss for one individual link X_i^D can thus be estimated directly from:

$$E(X_i^D) = \frac{E(Y_i^D)}{E(Y_{i-1}^D)} \tag{3.15}$$

From the estimated mean and variance, the moments for each individual link are derived and plugged into Equation 3.3 and 3.4 in Section 3.2. The final estimates of link delay and loss are then used for detection and localization of performance degradations as described in Section 3.3.

The algorithm is a cost-efficient alternative to monitoring of passive network equipment, as it is flexible to topology changes and can be combined with existing operations, administration and management (OAM) tools. Further details about the method can be found in Paper VIII, where different aspects of the measurement conditions are also discussed.

3.6 Incremental Diagnosis of Network States

Efficient diagnosis of network states can be done via active probing, which means that classification of a network state is performed by incrementally sending probes that are the most informative given previous probe outcomes. For example, Rish et al. encode the relation between network states and probe types in a Bayesian

network, and propose a method for selecting the next probe given an information theoretic metric that represents the highest information gain. In this respect, the overall approach to incremental diagnosis in Paper I is somewhat similar to the work by Rish et al. (2005), but incorporates mechanisms for autonomously updating the model parameters as real cases are observed.

The data consist of vectors of attribute values representing a network state or observed symptoms, and are divided into *prototypes* and *cases*. A prototype vector is an archetypical representation of a network state consisting of attribute values known a priori when the system is put into operation, whereas a case vector consists of real observations of the attribute values. An attribute represents the state of a network aspect or entity, such as connectivity or measured link metrics. The attribute values are either discrete or continuous. The attribute vectors encode combinations of measurements and observations and the relation between them, similar to states reached within tree-based troubleshooting schemes. However, in the incremental diagnosis system the relation between different combinations of attribute values is modeled by the use of a hierarchical probabilistic model. Over time, the diagnostic system will autonomously adjust the model parameters in order to put more trust into observed cases compared to prototypical data, which is a significantly more cost-efficient alternative to altering tree-based or rule-based troubleshooting schemes.

Probability distributions are represented as hierarchical graph mixtures (Gillblad, 2008), meaning that the final diagnosis is a weighted combination of naïve Bayes classifiers for each prototype under the assumption that all attributes are statistically independent. The aim is to incrementally find the diagnosis with maximum probability given a set of attribute values x_i and the distribution of classes Z. The probability of observing a certain class given a set of attributes is here formulated as:

$$p(Z|\boldsymbol{X}) \propto \sum_{z \in Z} p(Z = z) \sum_{k \in P_z} \left(\pi_{z,k} \prod_{i=1}^{n} p_{z,k}(x_i|Z = z) \right) \tag{3.16}$$

where P_z is the set of prototypes that are labelled with class z, and $\pi_{z,k}$ denotes the mixing proportion corresponding to prototype k for class z. Throughout the diagnostic session, the next attribute value is incrementally selected with respect to the already known attribute values and the highest information gain, expressed as the highest conditional mutual information between the class distribution and the remaining unknown attribute values.

Each data class is updated as new cases arrive, in proportion to how likely it is that the case $\boldsymbol{x}$ was generated from the parameters Θ of a specific class:

$$p(\Theta|\boldsymbol{x}) = \frac{\pi_\Theta \prod_{i=1}^{n} p_{z,\Theta}(x_i|Z = z)}{\sum_j \pi_j \prod_{i=1}^{n} p_{z,\Theta}(x_j|Z = z)} \tag{3.17}$$

The trust to prototypes relative to the observation of new cases is weighted through a parameter ζ similar to the w parameter in the overlapping estimates described in

Section 3.2. Differently to the w parameter, ζ is here usually set larger than one. The parameters of the data classes are updated via Bayesian estimators obtained by the use of conjugate priors. In the discrete and continuous cases, the multi-Beta and Wishart-Gaussian distributions are used, respectively (Gillblad, 2008).

The method has been successfully evaluated on real-world data for troubleshooting of vehicles (Paper I), but can just as well be used for diagnosing the state in networks and network devices. The algorithm efficiently produces a diagnosis within very few troubleshooting steps, and does not require explicit updates (as in rule-based systems) in the case of drifting representation of symptoms or states over time (compared to e.g. rule-based diagnostic systems). Additionally, the method is robust to noisy or inconsistent input. Further details and preliminaries of the incremental diagnosis model can be found in the work by Gillblad (2008) and Holst (1997), in addition to Paper I.

3.7 Sequential Pattern Mining in Event Logs

Automatic discovery of patterns in data logs is an important part in fault management, and can be used for e.g. detection of anomalous behavior in the network, or for explaining the order of events which leads to a certain state in the network or network device. Common approaches to data log analysis and pattern mining in data streams include different variants of Markov modeling, such as simple Markov chains (Jha et al., 2001; Ye, 2000) and HMMs (Hoang and Hu, 2004; Khreich et al., 2009). Alternatively, data can be modeled as *mixtures of Markov models*, where the data stream is assumed to be an interleaved sequence of symbols generated from several hidden source models. Markov mixture models have been employed for web-related applications, such as anomaly detection in web traffic (Song et al., 2009), and clustering of web-user behavior (Liu et al., 2004). In general, EM-algorithms are used for estimating the model parameters of the underlying sources that are usually assumed to be non-identical (Figure 3.7).

An alternative approach to parameter estimation under the assumption of *identical* sources is presented in Paper II. The aim is to estimate the selection and transition probabilities from the interleaved sequence of symbols (representing events, network states, etc.) such that the hidden source models can be recovered. In the general case, given K sources and their transition matrix $A_k(x_{n-1}, x_n)$ describing the transition from a state x_{n-1} to x_n, the probability of observing a state can be formulated as in Equation 3.18, where π represents the source selection probability and $e(x_n)$ is the steady-state emission probability of x_n. Assuming identical sources and uniform source selection probability $\pi_k = \pi = 1/K$ (meaning that the next symbol will be drawn from any of the sources with equal probability at step n), Equation 3.18 can be further reduced into Equation 3.19.

$$P(x_n|x_{n-1}) = \sum_k \pi_k^2 A(x_{n-1}, x_n) + \sum_k \sum_{l \neq k} \pi_k \pi_l e(x_n) \tag{3.18}$$

$$A^+(x_{n-1}, x_n) = \pi A(x_{n-1}, x_n) + (1 - \pi) e(x_n) \tag{3.19}$$

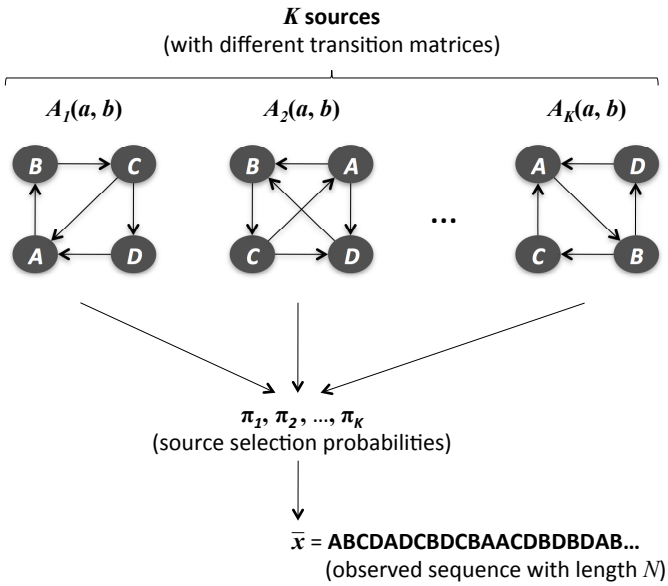

Figure 3.7: A model consisting of K Markov sources and a source selection mechanism. The sources that produced each symbol in an observed sequence are unknown and are to be recovered.

The recovery of the underlying transition matrix $A^+(x_i, x_j)$ can in the case of identical sources be obtained through maximum-likelihood estimates of p_{ij} (i.e. transition probabilities). In practice, this allows for counting the observed state transitions $i \rightarrow j$ directly from the data sequence. For a more robust estimate of p_{ij}, the parameters are computed as the posterior expectation of P_{ij} based on a Bayesian estimator, where the Hyper-Dirichlet distribution is assumed as a conjugate prior distribution for the parameters. The most likely number of identical sources generating a sequence can be predicted by finding the K that maximizes the expression in Equation 3.20.

$$\arg\max_K = P(K)P(\boldsymbol{x}|K). \tag{3.20}$$

Provided that the sources can be assumed identical, the parameter estimation is fast as only simple counter updates are needed. The method could therefore be favorably applied to real-time processing of data streams, compared to more computationally complex methods based on for example the EM-algorithm (Liu et al., 2004; Song et al., 2009). Further details and preliminaries about the approach are presented in Paper II.

3.8 Root-Cause Analysis in Virtual Overlays

Faults detected within a virtual overlay are not necessarily isolated to the layer itself, but may be symptoms of a problem in some other level of the overlay stack. Fault management tools applied to virtual network overlays therefore need to be capable of analyzing the complex relations between faults, anomalies and other events that occur in the stack of virtual overlays. Different approaches addressing event-correlation across overlays already exist, but these are often centralized (Appleby et al., 2001; Wang et al., 2010b) or require expensive dedicated aggregation equipment (Tang et al., 2007).

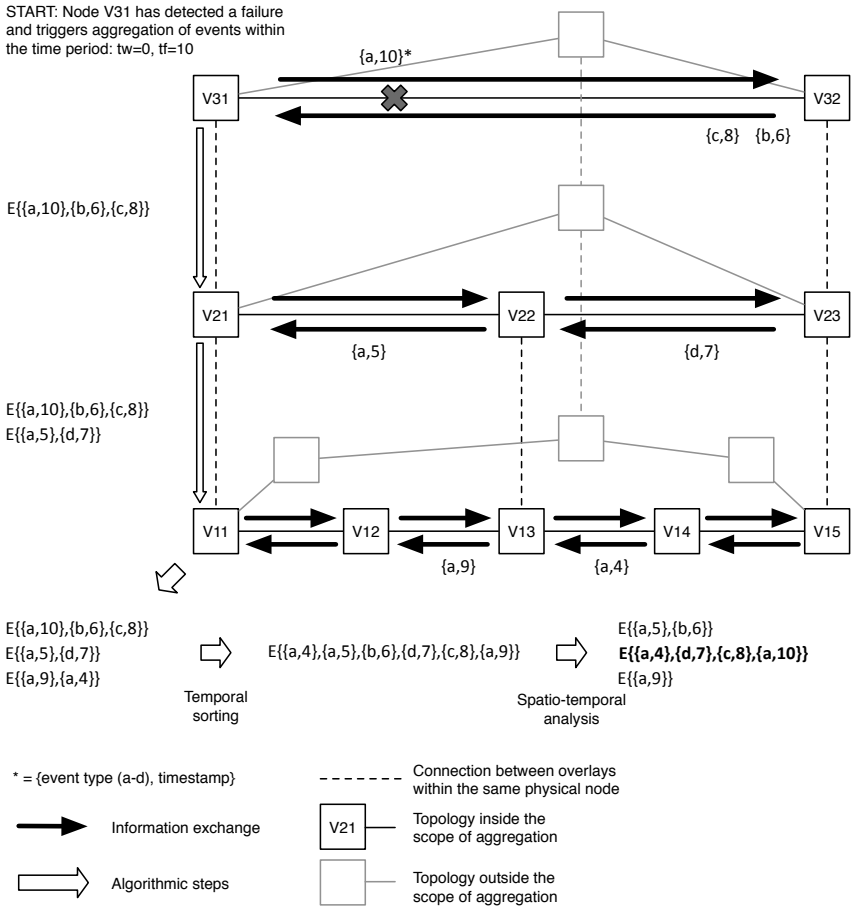

Figure 3.8: An overview of the algorithm for spatio-temporal correlation within and across overlays as described in Paper VI.

The algorithm for correlating spatio-temporally relevant events in Paper VI operates in a distributed manner in stacked virtual overlays, where each overlay relies on and is configured based on the network capabilities of the underlay[2]. It is assumed that all nodes store historical events within a specified time-window. An *event* is a reported alarm triggered by e.g. fault and change detection mechanisms as described in Paper III and IV. An overview of the algorithm is shown in Figure 3.8. When an event has occurred in a node, the protocol collects all events during the time period measured from when the network was working (from the detecting node perspective) until the current event. The protocol operates top down, and collects all temporally relevant events in the current layer, and transfers the set of events to the underlay, together with the time-window parameters and the spatially relevant endpoint nodes of the overlay. The underlay is aware of the topological mapping toward the overlay, and maps the overlay to the nodes in its own topology. The process continues either until no additional events are found within the time-window or until the physical layer has been reached. When the aggregation process has ended, further analysis is performed where the events are spatio-temporally sorted with respect to the assumption that problems in the lower layer cause a symptom in the upper layer[3]. Through this analysis, the chain of events relevant to the triggering event can be isolated and used in probabilistic modeling for the purpose of classification or diagnosis of network state patterns (see e.g. Paper II and Section 3.6). Additionally, the protocol accounts for relative time differences that may appear in networks without a central solution for time synchronization between the nodes, such as network time protocol (NTP) (Mills, 1991). An example is ad-hoc networks (Munaretto et al., 2004), including virtual layers on top of a wireless or cellular infrastructure (Hoebeke et al., 2006). Relative time differences can be estimated by using the methods in the work by e.g. Römer (2001) and Solis et al. (2007).

The distributed operation makes the algorithm flexible to changes in the topology and cost-efficient, as no dedicated aggregating network equipment is required. The lack of a probabilistic model currently makes the event-correlation analysis sensitive to the parameter settings of the algorithm. However, this could be compensated for by modeling observed event-patterns over time. Further details and evaluation of the event-correlation algorithm can be found in Paper VI.

[2]This algorithm deviates from the definition of PNM-algorithms, as it does not maintain a probabilistic model. The current method is a first step toward an extended approach where the aggregated data may be used as input to a probabilistic model.

[3]This assumption may be somewhat simplifying in practice but is reasonable as higher layers usually are affected by events in the underlying layers (see also e.g. Steinder and Sethi 2002). Higher-order correlations and other more complex relations can be modeled via e.g. pseudo-causal graphs (Mahimkar et al., 2009) or clusters (Wang et al., 2010b).

Chapter 4

Summary of Included Papers

4.1 Included Papers

Paper I: Fault-Tolerant Incremental Diagnosis with Limited Historical Data

D. Gillblad, R. Steinert and A. Holst, "Fault-Tolerant Incremental Diagnosis with Limited Historical Data". In *Proceedings of the International Conference on Prognostics and Health Management (PHM)*, Denver, CO, USA, 6-9 October, 2008.

A Bayesian approach to incremental diagnosis is presented. The diagnostic system is trained from prototypical and case-based vectors, which represent sets of continuous and discrete attributes for each diagnose based on prior knowledge and real observations. Compared to rule-based methods, where structures of troubleshooting knowledge are built up and restructured for new cases (at additional management costs), the proposed system is capable of automatically adjusting to new cases as they arrive. This means that the diagnostic system initially puts more trust into the prototypes (as there usually are very few or no case vectors) when finding a diagnosis. Over time, the parameters of the statistical model are gradually updated to reflect new observed real cases. During a diagnostic session, the system incrementally guides the user to provide the value of the most informative attribute (measuring the information gain after each step), which reduces the number of troubleshooting steps compared to providing input to all attribute values. The system can compensate for noisy or inconsistent attribute input, by computing the likelihood of observing a certain attribute value in combination with other inputs obtained during the diagnostic session.

The method has been evaluated on both continuous and discrete valued attributes, by the use of one synthetically generated dataset and three real datasets. The synthetic dataset was used for classifying different types of animals. The two first real datasets were used for diagnosing mechanical faults in military vehicles, whereas the third real dataset was extracted from the evaporation stage of a paper mill. The results indicate that between 80% and 99% of the diagnoses are correctly

classified at a noise level of 20%, and that the use of inconsistency-checks in general can improve the classification rate.

Overall contribution: The diagnostic system is flexible and cost-efficient - the model is self-adjusting to new cases and can provide accurate diagnoses in very few troubleshooting steps. Moreover, the method includes mechanisms for improved robustness against noise.

My contribution: I edited the paper in collaboration with Daniel Gillblad and Anders Holst, and performed the experiments and analysis for the evaluation of the method under the supervision of Daniel Gillblad.

Paper II: Estimating the Parameters of Randomly Interleaved Markov Models

D. Gillblad, R. Steinert and D. Ferreira, "Estimating the Parameters of Randomly Interleaved Markov Models". In *IEEE International Conference on Data Mining Workshops (ICDMW)*, Miami, FL, USA, 6 December, 2009.

In this paper, the problem of estimating the transition probabilities of hidden underlying sources from a sequence of observed symbols is addressed. It is assumed that a sequence of symbols is generated as a mixture of K sources of underlying Markov models. The symbols of the sequence are unlabelled, meaning that the source from which a particular symbol was drawn is unknown. In the context of network management, each symbol may here be a reported event in an alarm log. In the special case of identical sources studied in the paper, the method proposed is computationally simple in the sense that the hidden sources can be estimated from counting the appearance of each symbol in the sequence. In addition, the method includes means for predicting the number of sources needed for maximal estimation accuracy. The results produced from synthetically generated data indicate that the transition matrix can be successfully reproduced, and that the accuracy can be increased when observing longer symbol sequences. The algorithm was also evaluated on a real dataset with the aim of discovering the behavior of a technical support process. The dataset consists of seven different symbols that represent the states within a support issue. In the experiment, three sources were used for successful recovery of the underlying transition matrix, given a support issue from a sequence of 3127 symbols representing a total of 1211 issues.

Overall contribution: Under the assumption of identical sources, the method can be used to recover the underlying sources based on Bayesian parameter estimation. The model has been evaluated using data from a different domain, but can be applied to network data streams in a similar fashion for the purpose of discovering

processes and event patterns. The method is fast (scaling linearly with the sequence length), which makes it applicable for analytic network management tasks where large amounts of data need to be processed fast and efficiently.

My contribution: In collaboration with Daniel Gillblad and Diogo Ferreira, I contributed in the editing efforts of the paper. Moreover, I performed the experiments and analysis in Section IV.A of the paper under the supervision of Daniel Gillblad.

Paper III: Towards Distributed and Adaptive Detection and Localisation of Network Faults

R. Steinert and D. Gillblad, "Towards Distributed and Adaptive Detection and Localisation of Network Faults". In *Proceedings of the Sixth Advanced International Conference Telecommunications (AICT)*, IARIA, Barcelona, Spain, 9-15 May, 2010.

We present a probabilistic probing algorithm for distributed detection and localization of network faults. The algorithm is designed to be configured by probabilistic parameters related to the probing delay and detection accuracy. The algorithm autonomously adapts in line with the probabilistic parameters and the observed link behavior. Link delay and loss are modeled via parameter estimation from observed measurements. The statistical models are used for controlling the local probing interval and the detection threshold. A detected fault is isolated to a node or link by collaborative fault localization. The algorithm performance was evaluated in synthetic and real network topologies under varying settings of the algorithm parameters and different failure conditions. The simulation results indicate that more than 95% of the generated faults can be detected with only very few false alarms. The localization rate was at least 80% for the link faults and 65% for the node faults. The synthetic and real-world topologies that were used in the experiments consisted of a small set of single connections, which influenced negatively on the localization rate for node faults. With the possibility of sending algorithm control messages over alternative paths, the localization rate of node faults can be improved.

Overall contribution: The fault detection approach is fully distributed and based on probabilistic modeling of local network measurements. The method offers increased performance controllability, specified in terms of the acceptable rate of false alarms for one link. It is a clear example of a distributed, resource-efficient, and autonomous PNM-algorithm, capable of adjusting its probing behavior to local network conditions and specified performance goals.

My contribution: I wrote the paper under the supervision of Daniel Gillblad. I implemented the algorithm, and performed the evaluation experiments and analysis

of the method. The practical details of the distributed localization scheme was refined by me. The use of moments for estimating the parameters was a joint decision based on studies of related work.

Paper IV: Long-Term Adaptation and Distributed Detection of Local Network Changes

R. Steinert and D. Gillblad, "Long-Term Adaptation and Distributed Detection of Local Network Changes". In *IEEE Global Telecommunications Conference (GLOBECOM)*, Miami, FL, USA, 6-10 December, 2010.

A statistical approach to distributed detection of local latency shifts in networked systems is presented in this paper. Similar to Paper III, link delay measurements are performed between neighboring nodes via probing, and modeled by estimating the parameters of a Gamma distribution via MoM. Adaptation to drifting delays is accounted for by the use of overlapping estimators, such that the previous estimates are partially used as input to the next model. Shifts in the link delay can be detected by comparing the estimated parameters of the current and previous models by the use of the symmetric KLD. The detection is bounded by an upper and lower divergence threshold for the purpose of reducing the number of alarms that may appear during the development of a change. The method is memory-efficient as only a small set of counters are used for estimating the parameters of the probability distribution without the need of storing samples. In addition, the method is flexible in the sense that it can be applied to other suitable statistical distributions if necessary (see also Paper VIII). The algorithm was evaluated in the OMNeT++ network simulator using synthetically generated link delays and under varying change detection conditions. In this setting, over 90% of the shifts were detected. A change may sometimes go undetected as a result of overlapping shifts. To which degree changes developing under different time-scales can be detected is controlled by the algorithm parameters.

Overall contribution: Based on overlapping estimators, the distributed monitoring algorithm is capable of self-adjusting to local link variations, and can report gradual or abrupt changes relative to the current regime. This work is one step further toward a more adaptive monitoring algorithm than the one described in Paper III.

My contribution: I wrote the paper under the supervision of Daniel Gillblad. I implemented the algorithm, and performed the evaluation experiments and analysis of the method. Further, the expression for updating the overlapping moment estimators for the Gamma parameters was formulated by me.

Paper V: Toward Decentralized Probabilistic Management

A. G. Prieto, D. Gillblad, R. Steinert and A. Miron, "Toward Decentralized Probabilistic Management". In *IEEE Communications Magazine*, vol. 49, no. 7, pages 80-96, July, 2011.

In this paper we argue that the adoption of a decentralized and probabilistic paradigm is crucial for meeting the management challenges of future networks, such as scalability, robustness and adaptability. We define the concept of probabilistic management and discuss the potential advantages and disadvantages of employing probabilistic approaches from a management perspective. As examples of the probabilistic management paradigm, we present three algorithms used for real-time monitoring and anomaly detection. The advantages of a decentralized probabilistic paradigm can be summarized in terms of increased resource-efficiency, configurability, and performance predictability. As effects of the introduction of uncertainty, the potential drawbacks include less exact control of the network devices and over the behavior of the management algorithms, and sometimes, suboptimal solutions to management problems.

Overall contribution: A first definition of the PNM-paradigm with focus on in-network management, as well as an introduction to the concept of designing algorithms that adapt the behavior relative to high-level objectives and observed measurements.

My contribution: I participated in close collaboration with Daniel Gillblad and the other co-authors in defining the concepts of PNM and in identifying the implications of employing PNM-algorithms. I contributed to the editing efforts of the paper, and wrote the Section "A Probabilistic Anomaly Detection Algorithm" (based on Paper III and IV).

Paper VI: A Distributed Spatio-Temporal Event Correlation Protocol for Multi-Layer Virtual Networks

R. Steinert, S. Gestrelius and D. Gillblad, "A Distributed Spatio-Temporal Event Correlation Protocol for Multi-Layer Virtual Networks". In *IEEE Global Telecommunications Conference (GLOBECOM)*, Houston, TX, USA, 5-9 December, 2011.

From the Paper III and IV, we take a step toward detection, localization and diagnosis in virtual overlays, by designing a distributed spatio-temporal event correlation protocol for multi-layer networks. In Paper VI, we present a cross-layer protocol designed for distributed operation capable of compensating for asynchronous timestamps. It is assumed that events in one layer may arise from a series of events in lower layers. The detecting virtual node triggers aggregation of all spatio-temporally relevant events within a time-window, specified by the time of normal

operation and the detected event. Spatially related events in one layer are aggregated using a gossip-like protocol. The set of aggregated events in one layer is disseminated to lower layers and used for temporal correlation in order to find a root cause. We have tested the scalability and the performance of the distributed event protocol, using both synthetically generated and real-world topologies. The performance of the algorithm, in terms of the average match between synthetically generated and spatio-temporally correlated events, is sensitive to the rate of generated events and amount of historic data available. Moreover, the results indicate that the average overhead produced for collecting events within and across the overlays, scales in line with the graph-theoretic properties of the generated network and the number of layers. The algorithm and the associated protocol are intended to be part of a larger system in which previous fault and change detection algorithms provide basic events. Currently, the algorithm operates on events generated from other probabilistic algorithms - part of future work is to extend the method to include a probabilistic model of obtained root-causes or event patterns for improved robustness.

Overall contribution: The cross-layer protocol is fully distributed and does not rely on dedicated monitoring stations or centralized collection points. It is designed to compensate directly for asynchronous timestamps upon aggregation and dissemination of event information, which reduces the need for centralized time synchronization in different network environments.

My contribution: I wrote the paper under the supervision of Daniel Gillblad. I implemented and refined the details of the algorithm and added the compensation mechanisms for asynchronous timestamps. The experiments and analysis were performed by Sara Gestrelius under my supervision.

Paper VII: Performance Evaluation of a Distributed and Probabilistic Network Monitoring Approach

R. Steinert and D. Gillblad, "Performance Evaluation of a Distributed and Probabilistic Network Monitoring Approach". In *Proceedings of the 8th International Conference on Network and Service Management (CNSM)*, Las Vegas, NV, USA, 22-26 October, 2012.

Probabilistic and adaptive monitoring algorithms generally offer better performance controllability, adaptability and reliability, compared to deterministic monitoring methods. However, the differences between the two opposite approaches with respect to configuration efforts, algorithm performance under varying conditions, and resource-efficiency, seem not to have been studied. An investigation of these aspects is needed to gain further understanding of the implications of moving from deterministic management approaches to a probabilistic management paradigm.

As a first step, we perform a non-exhaustive investigation of the practical aspects of employing a probe-based PNM-algorithm for fault detection (based on Paper III and IV) in comparison with a deterministic network monitoring algorithm.

Experimental results based on synthetically generated and real link measurements, indicate that the PNM-algorithm is more resource-efficient with respect to the average amount of probing traffic generated in the network. We also see that the probabilistic algorithm is much more predictive in the performance than the deterministic algorithm, in the sense that the observed false positive rate closely follows the specified acceptable rate of false positives used to configure the probabilistic algorithm. The deterministic algorithm may produce fewer false positives than the probabilistic algorithm given that the configuration exactly matches the conditions under which it is applied. For a given parameter configuration, the probabilistic approach is in general more robust to variations in the observed measurements, which gives a better detection performance compared to when the deterministic algorithm is used.

Detailed knowledge about the measurement conditions is required in order to configure the deterministic algorithm for optimal performance. The results indicate that the potential gains in using the probabilistic algorithm are in the scope of improved robustness, performance prediction, and resource-efficiency, with less need for detailed knowledge about the measurement conditions. In a practical setting these factors may altogether lead to reduced management costs. However, this is a conjecture based on the study of two specific algorithms - for more conclusive results, additional comparative studies of other probabilistic algorithms are needed.

Overall contribution: An investigation in some of the practical effects in employing a PNM-algorithm compared to deterministic probing, with respect to adaptability, performance controllability, resource utilization, and configuration efficiency.

My contribution: I wrote the paper under the supervision of Daniel Gillblad. I implemented the experimentation framework within which the algorithms were evaluated. I planned and performed the experiments and analysis of the results. I extended the algorithm from Paper III with Bayesian overlapping estimators for loss following a Bernoulli distribution.

Paper VIII: Direct In-Network Localization of Performance Degradations

R. Steinert and D. Gillblad, "Direct In-Network Localization of Performance Degradations". *Submitted to IEEE Transactions on Network and Systems Management (TNSM)*, 2013.

We propose a unicast approach to statistical modeling of link delay and loss on network paths consisting of active measurement endpoints and passive intermediate

nodes, for the purpose of direct localization of performance degradations. Compared to other network tomography approaches, the proposed method is designed without the specific requirements on tree topologies and multicast measurements. Moreover, many existing algorithms operate under the assumption that all measurements are consistent, but in practice the degree of measurement consistency depends on the probing strategy and the local network conditions. For these reasons, we propose two models for link delay and one model for loss, and evaluate them with respect to varying measurement conditions in combination with different probing strategies. The statistical models are augmented with an adaptive mechanism based on overlapping estimators, which enables direct localization of performance degradations. Moreover, we investigate the algorithm performance with respect to detection and localization of changes to individual links on the monitored path. The method has been evaluated on synthetically generated measurements under different simulated measurement conditions and path lengths. The results indicate that satisfactory detection performance can be achieved in over 80% for all models applied to a path of 10 hops. We see that the performance can be further improved by measuring shorter paths and by applying a probing strategy that can maintain measurement consistency during the probing session.

Overall contribution: A decentralized and adaptive network tomography algorithm for direct localization of performance degradations in estimated link metrics obtained from successive, incremental measurements. The effect of different measurement conditions is evaluated using two models for link delay and one model for link loss. Compared to previous work in the area that assume consistent measurements, we investigate the practical effects of different probing strategies under varying measurement conditions.

My contribution: I wrote the paper under the supervision of Daniel Gillblad. I implemented the algorithm, and performed the evaluation experiments and analysis of the method. The second delay model for non-stationary measurement conditions and the loss model were both added by me. The overlapping parameter estimates used for all models in the paper are based on Paper IV and VII.

4.2 Other Publications by the Author

- W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, and C. Meirosu. "Research Directions in Network Service Chaining". In IEEE Future Networks and Services (SDN4FNS), Trento, Italy, 11-13 November, 2013.
- P. Murray, A. Sefidcon, R. Steinert, V. Fusenig and J. Carapinha. "Cloud Networking: An Infrastructure Service Architecture for the Wide Area". In Future Network and Mobile Summit (FUNEMS), Berlin, Germany, 4-6 July, 2012.

- B. Bjurling, R. Steinert, and D. Gillblad. "Translation of Probabilistic QoS in Hierarchical and Decentralized Settings". In the 13th IEEE Asia-Pacific Network Operations and Management Symposium (APNOMS), Taipei, Taiwan, 21-23 September, 2011.
- M. Bohlin, K. Doganay, P. Kreuger, R. Steinert and M. Wärja. "Searching for Gas Turbine Maintenance Schedules". AI Magazine, vol. 31, no. 1, pages 21-36, 2010.
- M. Bohlin, K. Doganay, P. Kreuger, R. Steinert and M. Wärja. "A Tool for Gas Turbine Maintenance Scheduling". Innovative Applications of Artificial Intelligence (IAAI), pages 9-16, Pasadena, CA, USA, 14-16, July 2009.
- B. Levin, A. Holst, M. Bohlin, R. Steinert and M. Aronsson. "Dynamic Maintenance". In Proceedings of the 21st International Congress and Exhibition On Condition Monitoring and Diagnostic Engineering Management (COMADEM), Prague, Czech Republic, 11-13 June, 2008.
- M. Bohlin, M. Forsgren, A. Holst, B. Levin, M. Aronsson and R. Steinert. "Reducing Vehicle Maintenance Using Condition Monitoring and Dynamic Planning. The 4th IET International Conference on Railway Condition Monitoring (RCM), Derby, UK, 18-20 June, 2008.
- R. Steinert, M. Rehn and A. Lansner. "Recognition of Handwritten Digits Using Sparse Codes Generated by Local Feature Extraction Methods". European Symposium on Artificial Neural Networks (ESANN), pages 161-166, Bruges, Belgium, 26-28 April, 2006.
- R. Steinert and D. Gillblad. "Distributed Detection of Latency Shifts in Networks". Technical report, T2009:12, SICS, Kista, Sweden, 2009.
- R. Steinert and D. Gillblad. "An Initial Approach to Distributed Adaptive Fault-Handling in Networked Systems". Technical report, T2009:07, SICS, Kista, Sweden, 2009.
- D. Gillblad, A. Holst and R. Steinert. "Fault-Tolerant Incremental Diagnosis with Limited Historical Data". Technical report, T2006:17, SICS, Kista, Sweden, 2006.
- R. Steinert. "Pattern Recognition of Sparse Codes Generated by Local Feature Extraction Methods". Master's thesis, TRITA-CSC-E 2006:155, Royal Institute of Technology, 2006.

Chapter 5

Concluding Remarks

The topic of this thesis is probabilistic fault management approaches following the design goals of scalability, adaptability and reliability defined within the PNM-paradigm. The presented set of probabilistic fault management algorithms implements memory-efficient and reusable probabilistic models for multiple purposes such as fault detection, performance monitoring, and self-adjustment in the algorithm behavior. The algorithms can operate in a decentralized or distributed manner, and can adapt models and algorithm behavior to observed data and probabilistic configuration parameters.

In general, the algorithms are representative examples of the decentralized PNM-paradigm and contribute to many of the different functions expected from most fault management systems. However, the development toward a complete probabilistic fault management system, that fully implements the PNM-concepts and fulfills the requirements on self-management and high-level configurability in large-scale networked systems, is not trivial and requires further scientific work.

Important areas to further investigate relate to the dynamics and modeling of the network behavior and state. With continuously changing network conditions, in terms of hardware, software, and varying user behavior, it is necessary to carefully study the statistical properties of each physical and logical aspect of the network, on both local and network-wide levels. Dependency relationships between the observed network behavior and the network conditions (e.g. available equipment, tools, usage, etc.) need to be well understood for accurate modeling of network states. In addition, resource-efficient and scalable methods for passive and active monitoring that can increase the observability of the network are essential to any type of modeling. In this respect, currently available monitoring systems or tools are limited for different reasons - they may not scale (Asgari et al. 2002) or may only be used under certain measurement conditions in order to provide sufficiently reliable results (Denby et al. 2007).

The PNM-paradigm offers, among other things, simplified configurability and performance predictability - two features that are necessary for efficient manage-

ment of large networked systems. Simplified configuration requires further investigation on how high-level goals or objectives should be expressed for easier understanding of the resulting algorithm performance, as more complex probabilistic expressions quickly may become unintuitive. Increased performance predictability can be achieved by the use of management algorithms operating more autonomously, but with the effect of less strict control over various network aspects. There are, for example, management algorithms that self-adjust the sampling behavior relative to specifications on model accuracy and information uncertainty (e.g. Cantieni et al. 2006), but little is known of any potential side-effects on the network state when several self-adjusting algorithms are operating in the same system. Hence, another problem area that needs to be investigated relates to the control of probabilistic fault management algorithms that autonomously perform management actions side-by-side.

Finally - the amount of data generated by the network devices (from e.g. logs, measurements, audit data, etc.) continues to increase, but in practice relatively little is analyzed and used as support to network management decisions and functions. Centralized approaches (based on distributed computation techniques) can be used for analyzing large amounts of data relatively fast - the limitations are related mainly to timing, but also to the consumption of network resources when data is streamed for the purpose of centralized analysis. Distributed real-time monitoring approaches based on probabilistic management and machine learning (with inspiration from big data analytics methods), could in this respect provide more scalable means to fully exploit monitoring data, which would give better support to efficient and proactive management decisions within short time-scales.

The PNM-concepts and the requirements on probabilistic fault management approaches have gradually matured throughout the work of this thesis, and will continue to be refined with the development of different network technologies and associated management challenges. The concepts are not restricted only to fault management as they are applicable to other areas of network management, such as resource management. Numerous collaborations[1] between the telecom industry and the network research community, confirm the increasing interest in adopting probabilistic management concepts in practice. This, in combination with the increasing demands on dynamic and flexible network operation (based on virtualized network functions and programmable infrastructures), indicates the beginning of a paradigm shift in how networks are managed and where probabilistic management approaches will play a crucial role.

[1] See for example:
4WARD http://www.4ward-project.eu/;
SAIL http://www.sail-project.eu/;
UNIVERSELF http://www.univerself-project.eu;
ECODE http://www.ecode-project.eu;
UNIFY http://www.fp7-unify.eu/.

Bibliography

D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. *ACM SIGOPS Oper. Syst. Rev.*, 35(5):131–145, 2001.

J. Andrews. Seven Ways that HetNets Are a Cellular Paradigm Shift. *IEEE Communications Magazine*, 51(3):136–144, 2013.

K. Appleby, G. Goldszmidt, and M. Steinder. Yemanja - A Layered Event Correlation Engine for Multi-Domain Server Farms. In *2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings*, pages 329–344, 2001.

M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A View of Cloud Computing. *Communications of the ACM*, 53(4):50–58, 2010.

A. Asgari, P. Trimintzios, M. Irons, G. Pavlou, R. Egan, and S. V. Den Berghe. A Scalable Real-Time Monitoring System for Supporting Traffic Engineering. In *2002 IEEE Workshop on IP Operations and Management*, pages 202–207. IEEE, 2002.

L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A Survey. *Computer Networks*, 54(15):2787–2805, 2010.

R. Badonnel, R. State, and O. Festor. Probabilistic Management of Ad-Hoc Networks. In *10th IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pages 339–350, 2006.

P. Barford, J. Kline, D. Plonka, and A. Ron. A Signal Analysis of Network Traffic Anomalies. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment (IMW)*, pages 71–82, 2002.

P. Barford, N. Duffield, A. Ron, and J. Sommers. Network Performance Anomaly Detection and Localization. In *Proceedings IEEE INFOCOM 2009*, pages 1377–1385, 2009.

M. Basseville and I. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Information and System Sciences Series. Prentice-Hall, Englewood Cliffs, N.J., 1993.

K. Bhaduri and A. N. Srivastava. A Local Scalable Distributed Expectation Maximization Algorithm for Large Peer-to-Peer Networks. In *ICDM '09. Ninth IEEE International Conference on Data Mining*, pages 31–40, 2009.

A. Bifet and R. Gavaldà. Learning from Time-Changing Data with Adaptive Windowing. In *SIAM International Conference on Data Mining*, pages 443–448, 2007.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2nd edition, 2007.

J. Bolot. Characterizing End-to-End Packet Delay and Loss in the Internet. *Journal of High Speed Networks*, 2(3):305–323, 1993.

M. Brunner, D. Dudkowski, C. Mingardi, and G. Nunzi. Probabilistic Decentralized Network Management. In *2009 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 25–32, 2009.

H. Burch and C. Chase. Monitoring Link Delays With One Measurement Host. *ACM SIGMETRICS Performance Evaluation Review*, 33(3):10–17, 2005.

R. Cáceres, N. G. Duffield, J. Horowitz, and D. F. Towsley. Multicast-Based Inference of Network-Internal Loss Characteristics. *IEEE Transactions on Information Theory*, 45(7):2462–2480, 1999.

G. R. Cantieni, G. Iannaccone, C. Barakat, C. Diot, and P. Thiran. Reformulating the Monitor Placement Problem: Optimal Network-Wide Sampling. In *2006 40th Annual Conference on Information Sciences and Systems (CISS)*, pages 1725–1731, 2006.

S. Chen and P. Gopalakrishnan. Speaker, Environment, and Channel Change Detection and Clustering Via the Bayesian Information Criterion. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 127–132, 1998.

H. K. Choi and J. O. Limb. A Behavioral Model of Web Traffic. In *Proceedings Seventh International Conference on Network Protocols (ICNP)*, pages 327–334, 1999.

N. M. K. Chowdhury and R. Boutaba. Network Virtualization: State of the Art and Research Challenges. *IEEE Communications Magazine*, 47(7):20–26, 2009.

A. Coates, A. Hero, R. Nowak, and B. Yu. Internet Tomography. *IEEE Signal Processing Magazine*, 19(3):47–65, 2002.

M. Coates and R. Nowak. Network Loss Inference Using Unicast End-to-End Measurement. In *Proceedings of the ITC Seminar IP Traffic, Modeling and Management*, pages 28:1–9, 2000.

E. Cohen and M. Strauss. Maintaining Time-Decaying Stream Aggregates. *Journal of Algorithms*, 59(1):19–36, 2006.

R. Cohen and A. Landau. Not All At Once! - A Generic Scheme for Estimating the Number of Affected Nodes While Avoiding Feedback Implosion. In *2009 Proceedings IEEE INFOCOM*, pages 2641–2645, 2009.

Í. Cunha, R. Teixeira, N. Feamster, and C. Diot. Measurement Methods for Fast and Accurate Blackhole Identification with Binary Tomography. In *Proceedings of the SIGCOMM Internet Measurement Conference (IMC)*, pages 254–266. ACM, 2009.

E. Damsleth. Conjugate Classes for Gamma Distributions. *Scandinavian Journal of Statistics*, 2(2):80–84, 1975.

T. Dasu, S. Krishnan, D. Lin, S. Venkatasubramanian, and K. Yi. Change (Detection) You Can Believe in: Finding Distributional Shifts in Data Streams. In *Advances in Intelligent Data Analysis VIII*, volume 5772 of *Lecture Notes in Computer Science*, pages 21–34. Springer, 2009.

M. Datar, A. Gionis, P. Indyk, and R. Motwani. Maintaining Stream Statistics Over Sliding Windows. In *Proceedings of the of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 635–644, 2002.

L. Denby, J. M. Landwehr, C. L. Mallows, J. Meloche, J. Tuck, B. Xi, G. Michailidis, and V. N. Nair. Statistical Aspects of the Analysis of Data Networks. *Technometrics*, 49(3):318–334, 2007.

T. Dillon, C. Wu, and E. Chang. Cloud Computing: Issues and Challenges. In *24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pages 27–33, 2010.

N. Duffield and F. Lo Presti. Multicast Inference of Packet Delay Variance at Interior Network Links. In *Proceedings IEEE INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1351–1360, 2000.

N. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley. Network Delay Tomography from End-to-End Unicast Measurements. In *Evolutionary Trends of the Internet*, volume 2170 of *Lecture Notes in Computer Science*, pages 576–595. Springer, 2001.

S. Eguchi and J. Copas. Interpreting Kullback-Leibler Divergence with the Neyman-Pearson Lemma. *Journal of Multivariate Analysis*, 97(9):2034–2040, 2006.

E. Elmroth, J. Tordsson, F. Hernández, A. Ali-Eldin, P. Svärd, M. Sedaghat, and W. Li. Self-Management Challenges for Multi-Cloud Architectures. *Towards a Service-Based Internet*, 6994:38–49, 2011.

M. H. Firooz and S. Roy. Network Tomography via Compressed Sensing. In *2010 IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–5, 2010.

T. Forell, D. Milojicic, and V. Talwar. Cloud Management: Challenges and Opportunities. In *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW)*, pages 881–889, 2011.

C. Fragouli and A. Markopoulou. A Network Coding Approach to Overlay Network Monitoring. In *43rd Annual Allerton Conference on Communication, Control, and Computing*, University of Illinois, Urbana-Champaign, USA, 2005.

S. Fu and C.-Z. Xu. Exploring Event Correlation for Failure Prediction in Coalitions of Clusters. In *Proceedings of the 2007 ACM/IEEE Conference on Supercomputing (SC)*, pages 1–12, 2007.

R. Gaeta, M. Gribaudo, D. Manini, and M. Sereno. On the use of petri nets for the computation of completion time distribution for short tcp transfers. In *Applications and Theory of Petri Nets 2003*, volume 2679 of *Lecture Notes in Computer Science*, pages 181–200. Springer, 2003.

A. Gençata and B. Mukherjee. Virtual-Topology Adaptation for WDM Mesh Networks Under Dynamic Traffic. *IEEE/ACM Transactions on Networking*, 11(2): 236–247, 2003.

A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T. Thomas, J. Andrews, P. Xia, H. Jo, H. Dhillon, and T. Novlan. Heterogeneous Cellular Networks: From Theory to Practice. *IEEE Communications Magazine*, 50(6):54–64, 2012.

D. Gillblad. *On Practical Machine Learning and Data Analysis.* PhD thesis, TRITA-CSC-A 2008-11, School of Computer Science and Communication (CSC), Royal Institute of Technology, Stockholm, Sweden, 2008.

P. Gonçalves, R. Shubhabrata, T. Begin, and P. Loiseau. Dynamic Resource Management in Clouds: A Probabilistic Approach. *IEICE Transactions on Communications*, 95(8):2522–2529, 2012.

Z. Guo, G. Jiang, H. Chen, and K. Yoshihira. Tracking Probabilistic Correlation of Monitoring Data for Fault Detection in Complex Systems. In *Proceedings of the 2006 International Conference on Dependable Systems and Networks (DSN)*, pages 259–268, 2006.

A. Gut. *An intermediate course in probability.* Springer, 2nd edition, 2009. ISBN 978-1-4419-0161-3.

H. Hajji. Statistical Analysis of Network Traffic for Adaptive Faults Detection. *IEEE Transactions on Neural Networks*, 16:1053–1063, 2005.

S. Han, E. Chan, R. Cheng, and K.-Y. Lam. A Statistics-Based Sensor Selection Scheme for Continuous Probabilistic Queries in Sensor Networks. *Real-Time Systems*, 35(1):33–58, 2007.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

O. Heinonen and H. Mannila. Attribute-Oriented Induction and Conceptual Clustering. Technical Report C-1996-2, Department of Computer Science, University of Helsinki, Finland, 1996.

A. Hernandez and E. Magafia. One-Way Delay Measurement and Characterization. In *3rd International Conference on Networking and Services (ICNS)*, pages 114–119, 2007.

X. Hoang and J. Hu. An Efficient Hidden Markov Model Training Scheme for Anomaly Intrusion Detection of Server Applications Based on System Calls. In *Proceedings of the 12th IEEE International Conference on Networks (ICON)*, volume 2, pages 470–474, 2004.

J. Hoebeke, G. Holderbeke, I. Moerman, B. Dhoedt, and P. Demeester. Virtual Private Ad Hoc Networking. *Wireless Personal Communications*, 38(1):125–141, 2006.

R. Hofstede, I. Drago, G. Moura, and A. Pras. Carrier Ethernet OAM: An Overview and Comparison to IP OAM. In *Managing the Dynamics of Networks and Services*, volume 6734 of *Lecture Notes in Computer Science*, pages 112–123. Springer, 2011.

A. Holst. *The use of a Bayesian neural network model for classification tasks*. PhD thesis, TRITA-NA-P9708, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, 1997.

R. Q. Hu, Y. Qian, S. Kota, and G. Giambene. HetNets - A New Paradigm for Increasing Cellular Capacity and Coverage. *IEEE Wireless Communications*, 18 (3):8–9, 2011.

Y. Huang, N. Feamster, A. Lakhina, and J. J. Xu. Diagnosing Network Disruptions with Network-Wide Analysis. *ACM SIGMETRICS Performance Evaluation Review*, 35(1):61–72, 2007.

ITU-T Rec. Y.1731. OAM functions and mechanisms for Ethernet based networks, 2008.

J. E. Jackson and G. S. Mudholkar. Control Procedures for Residuals Associated with Principal Component Analysis. *Technometrics*, 21(3):341–349, 1979.

E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

S. Jha, K. M. Tan, and R. A. Maxion. Markov Chains, Classifiers, and Intrusion Detection. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW)*, pages 206–219, 2001.

G. Jiang and G. Cybenko. Temporal and Spatial Distributed Event Correlation for Network Security. In *Proceedings of the 2004 American Control Conference (ACC)*, volume 2, pages 996–1001, 2004.

K. Julisch. Mining Alarm Clusters to Improve Alarm Handling Efficiency. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC)*, pages 12–21, 2001.

M. Kalman and B. Girod. Modeling the Delays of Successively-Transmitted Internet Packets. In *IEEE International Conference on Multimedia and Expo (ICME)*, volume 3, pages 2015–2018, 2004.

S. Kandula, D. Katabi, and J.-P. Vasseur. Shrink: A Tool for Failure Diagnosis in IP networks. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data (MineNet)*, pages 173–178, 2005.

W. Khreich, E. Granger, R. Sabourin, and A. Miri. Combining Hidden Markov Models for Improved Anomaly Detection. In *2009 IEEE International Conference on Communications (ICC)*, pages 1–6, 2009.

A. Kind, X. Dimitropoulos, S. Denazis, and B. Claise. Advanced Network Monitoring Brings Life to the Awareness Plane. *IEEE Communications Magazine*, 46 (10):140–146, 2008.

R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. Detection and Localization of Network Black Holes. In *Proceedings IEEE INFOCOM 2007. The 26th IEEE International Conference on Computer Communications*, pages 2180–2188, 2007.

K. Konstanteli, T. Cucinotta, K. Psychas, and T. Varvarigou. Admission Control for Elastic Cloud Services. In *Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, pages 41–48, 2012.

B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen. Sketch-based Change Detection: Methods, Evaluation, and Applications. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement (IMC)*, pages 234–247, 2003.

S. Kullback and R. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

P. Kumar. Probability Distributions Conditioned by the Available Information: Gamma Distribution and Moments*. *Computers & Mathematics with Applications*, 52(3-4):289–304, 2006.

R. Kwitt and A. Uhl. Image Similarity Measurement by Kullback-Leibler Divergences Between Complex Wavelet Subband Statistics for Texture Retrieval. In *Proceedings of the 2008 IEEE International Conference on Image Processing (ICIP)*, pages 933–936, 2008.

A. Lakhina, M. Crovella, and C. Diot. Characterization of Network-wide Anomalies in Traffic Flows. In *Proceedings of the of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC)*, pages 201–206, 2004a.

A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-wide Traffic Anomalies. *ACM SIGCOMM Computer Communication Review*, 34(4):219–230, 2004b.

B. Lantz, B. Heller, and N. McKeown. A Network in a Laptop: Rapid Prototyping for Software-Defined Networks. In *Proceedings of the of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 19:1–19:6, 2010.

E. Lawrence, G. Michailidis, V. N. Nair, and B. Xi. Network Tomography: A Review and Recent Developments. *Frontiers in Statistics*, pages 345–364, 2006.

P. M. Lee. *Bayesian Statistics: An Introduction*. Wiley, 3rd edition, 2004. ISBN 047068920X.

X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina. Detection and Identification of Network Anomalies Using Sketch Subspaces. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC)*, pages 147–152, 2006.

J. Lin, J. Zhang, and W. Lin. Multicast-Based Inference of Network-Internal Delay Performance Using the Method of Moments. In *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR)*, volume 2, pages 193–196, 2010.

W. Liu and Y. Yang. Parametric or Nonparametric? A Parametricness Index for Model Selection. *Annals of Statistics*, 39(4):2074–2102, 2011.

Y. Liu, X. Huang, A. An, and G. Promhouse. Clustering Web Surfers With Probabilistic Models in a Real Application. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 761–765, 2004.

J. Lv, X. Li, and T. Li. The New Detection Algorithms for Network Traffic Anomalies. In *Proceedings of the 6th International Conference on Networking (ICN)*, pages 52–57, 2007.

A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and Q. Zhao. Towards Automated Performance Diagnosis in a Large IPTV Network. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, pages 231–242, 2009.

Y. Mao, F. Kschischang, B. Li, and S. Pasupathy. A Factor Graph Approach to Link Loss Monitoring in Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications*, 23(4):820–829, 2005.

S. Meng and L. Liu. Enhanced Monitoring-as-a-Service for Effective Cloud Management. *IEEE Transactions on Computers*, 62(9):1705–1720, 2012.

R. Miller. Bayesian Analysis of the Two-Parameter Gamma Distribution. *Technometrics*, 22(1):65–69, 1980.

D. L. Mills. Internet Time Synchronization: The Network Time Protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, 1991.

D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac. Internet of Things: Vision, Applications & Research Challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.

T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. ISBN 0071154671.

A. Mukherjee. On the dynamics and significance of low frequency components of internet load. Technical Report CIS-92-83, University of Pennsylvania, Philadelphia, USA, 1992.

A. Munaretto, M. Fonseca, K. Al Agha, and G. Pujolle. Virtual Time Synchronization for Multimedia Ad Hoc Networks. In *2004 IEEE 60th Vehicular Technology Conference*, volume 4, pages 2587–2590, 2004.

P. Murray, A. Sefidcon, R. Steinert, V. Fusenig, and J. Carapinha. Cloud Networking: An Infrastructure Service Architecture for the Wide Area. In *2012 Future Network and MobileSummit*, pages 1–8, 2012.

J. Neyman and E. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society*, 231:289–337, 1933.

J. Postel. Internet Control Message Protocol. *RFC792, Internet Engineering Task Force (IETF)*, 1981.

B. Pradhan and D. Kundu. Bayes Estimation and Prediction of the Two-Parameter Gamma Distribution. *Journal of Statistical Computation and Simulation*, 81(9): 1187–1198, 2011.

A. Pras, J. Schonwalder, M. Burgess, O. Festor, G. Pérez, R. Stadler, and B. Stiller. Key Research Challenges in Network Management. *IEEE Communications Magazine*, 45(10):104–110, 2007.

A. G. Prieto and R. Stadler. Adaptive Distributed Monitoring With Accuracy Objectives. In *Proceedings of the 2006 ACM SIGCOMM Workshop on Internet Network Management*, pages 65–70, 2006.

R. Rajagopal, X. Nguyen, S. C. Ergen, and P. Varaiya. Distributed Online Simultaneous Fault Detection for Multiple Sensors. In *2008 International Conference on Information Processing in Sensor Networks (IPSN)*, pages 133–144, 2008.

I. Rish, M. Brodie, S. Ma, N. Odintsova, A. Beygelzimer, G. Grabarnik, and K. Hernandez. Adaptive Diagnosis in Distributed Systems. *IEEE Transactions on Neural Networks*, 16(5):1088–1109, 2005.

K. Römer. Time Synchronization in Ad Hoc Networks. In *Proceedings of the of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 173–182, 2001.

S. Roychowdhury and S. Khurshid. A Novel Framework for Locating Software Faults Using Latent Divergences. In *Machine Learning and Knowledge Discovery in Databases*, volume 6913 of *Lecture Notes in Computer Science*, pages 49–64. Springer, 2011.

G. Schwarz. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2): 461–464, 1978.

R. Schweller, A. Gupta, E. Parsons, and Y. Chen. Reversible Sketches for Efficient and Accurate Change Detection Over Network Data Streams. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC)*, pages 207–212, 2004.

M. Shih and A. Hero. Unicast Inference of Network Link Delay Distributions From Edge Measurements. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 6, pages 3421–3424, 2001.

R. Solis, V. Borkar, and P. Kumar. A New Distributed Time Synchronization Protocol for Multihop Wireless Networks. In *2006 45th IEEE Conference on Decision and Control*, pages 2734–2739, 2007.

Y. Song, A. D. Keromytis, and S. Stolfo. Spectrogram: A Mixture-of-Markov-Chains Model for Anomaly Detection in Web Traffic. In *Proceedings of the Network and Distributed System Security Symposium*, pages 121–135, 2009.

M. Steinder and A. S. Sethi. End-to-End Service Failure Diagnosis Using Belief Networks. In *2002 IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pages 375–390, 2002.

M. Steinder and A. S. Sethi. Probabilistic Fault Diagnosis in Communication Systems Through Incremental Hypothesis Updating. *Computer Networks*, 45(4): 537–562, 2004.

R. Sterritt. Pulse Monitoring: Extending the Health-Check for the Autonomic GRID. In *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN)*, pages 433–440, 2003.

Y. Tang, E. S. Al-Shaer, and R. Boutaba. Active Integrated Fault Localization in Communication Networks. In *2005 9th IFIP/IEEE International Symposium on Integrated Network Management*, pages 543–556, 2005.

Y. Tang, E. S. Al-Shaer, and B. Zhang. Toward Globally Optimal Event Monitoring & Aggregation For Large-scale Overlay Networks. In *10th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 236–245, 2007.

D. K. Tasoulis and M. N. Vrahatis. Unsupervised Distributed Clustering. In *Parallel and Distributed Computing and Networks*, pages 347–351, 2004.

M. Thottan and C. Ji. Anomaly Detection in IP Networks. *IEEE Transactions on Signal Processing*, 51(8):2191–2204, 2003.

Y. Tsang, M. Coates, and R. Nowak. Passive Network Tomography Using EM Algorithms. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 1469–1472, 2001.

Y. Tsang, M. Coates, and R. Nowak. Network Delay Tomography. *IEEE Transactions on Signal Processing*, 51(8):2125–2136, 2003.

C. Wang, V. Talwar, K. Schwan, and P. Ranganathan. Online Detection of Utility Cloud Anomalies Using Metric Distributions. In *Proceedings of the 2010 IEEE Network Operations and Management Symposium (NOMS)*, pages 96–103, 2010a.

C. Wang, K. Viswanathan, L. Choudur, V. Talwar, W. Satterfield, and K. Schwan. Statistical Techniques for Online Anomaly Detection in Data Centers. In *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 385–392, 2011.

T. Wang, M. Srivatsa, D. Agrawal, and L. Liu. Spatio-Temporal Patterns in Network Events. In *Proceedings of the 6th ACM International Conference on emerging Networking EXperiments and Technologies (Co-NEXT)*, pages 3:1–3:12, 2010b.

M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and Modelling of the Temporal Dependence in Packet Loss. In *Proceedings IEEE INFOCOM'99 Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 345–352, 1999.

K. Yamanishi and Y. Maruyama. Dynamic Syslog Mining for Network Failure Monitoring. In *Proceedings of the of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 499–508, 2005.

Y. Yang, N. Kim, and S. Lam. Transient Behaviors of TCP-Friendly Congestion Control Protocols. In *Proceedings IEEE INFOCOM 2001 Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1716–1725, 2001.

N. Ye. A Markov Chain Model of Temporal Behavior for Anomaly Detection. In *Proceedings of the of the 2000 IEEE Workshop on Information Assurance and Security*, pages 171–174, 2000.

Q. Zhang, L. Cheng, and R. Boutaba. Cloud Computing: State-of-the-art and Research Challenges. *Journal of Internet Services and Applications*, 1(1):7–18, 2010.

Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan. Network Anomography. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC)*, pages 317–330, 2005.

Y. Zhu and L. M. Ni. Probabilistic Approach to Provisioning Guaranteed QoS for Distributed Event Detection. In *Proceedings IEEE INFOCOM 2008. The 27th IEEE International Conference on Computer Communications*, pages 592–600, 2008.