New Problems in Dependability: Quantum Computing

Cristina Nita-Rotaru
Khoury College of Computer Science

25 years ago

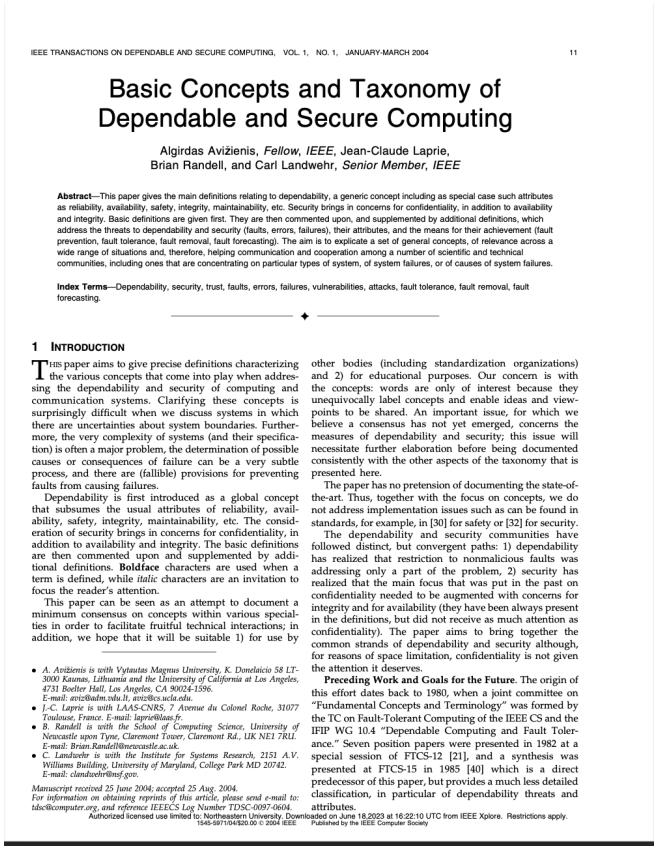
- Study distributed systems
- Learned about dependability of distributed systems
- PhD at the intersection of fault-tolerance and security

“High-Performance Secure Group Communication”

… in 1998

- **At Johns Hopkins University**
  - There was no faculty working in security
  - There were no classes in security

- **In the research community**
  - **DSN did not exist:** It will become DSN in 2000 (FTCS-30 and DCCA-8) with two sub-conferences DCCS and PDS
  - **A few security conferences**: NDSS was a small and young conference; about 50 papers was all I needed to read to get up to speed
  - **The relation between security and dependability was not yet articulated:** IEEE Transactions on Dependable and Secure Computing did not exist, first issue will appear in 2004



Basic Concepts and Taxonomy of Dependable and Secure Computing

Algirdas Avižienis, *Fellow, IEEE*, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, *Senior Member, IEEE*

Abstract—This paper gives the main definitions relating to dependability, a generic concept including as special case such attributes as reliability, availability, safety, integrity, maintainability, etc. Security brings in concerns for confidentiality, in addition to availability and integrity. Basic definitions are given first. They are then commented upon, and supplemented by additional definitions, which address the threats to dependability and security (faults, errors, failures), their attributes, and the means for their achievement (fault prevention, fault tolerance, fault removal, fault forecasting). The aim is to explicate a set of general concepts, of relevance across a wide range of situations and, therefore, helping communication and cooperation among a number of scientific and technical communities, including ones that are concentrating on particular types of system, of system failures, or of causes of system failures.

Index Terms—Dependability, security, trust, faults, errors, failures, vulnerabilities, attacks, fault tolerance, fault removal, fault forecasting.

1 Introduction

This paper aims to give precise definitions characterizing the various concepts that come into play when addressing the dependability and security of computing and communication systems. Clarifying these concepts is surprisingly difficult when we discuss systems in which there are uncertainties about system boundaries. Furthermore, the very complexity of systems (and their specification) is often a major problem, the determination of possible causes or consequences of failure can be a very subtle process, and there are (fallible) provisions for preventing faults from causing failures.

Dependability is first introduced as a global concept that subsumes the usual attributes of reliability, availability, safety, integrity, maintainability, etc. The consideration of security brings in concerns for confidentiality, in addition to availability and integrity. The basic definitions are then commented upon and supplemented by additional definitions. **Boldface** characters are used when a term is defined, while *italic* characters are an invitation to focus the reader's attention.

This paper can be seen as an attempt to document a minimum consensus on concepts within various specialties in order to facilitate fruitful technical interactions; in addition, we hope that it will be suitable 1) for use by other bodies (including standardization organizations) and 2) for educational purposes. Our concern is with the concepts: words are only of interest because they unequivocally label concepts and enable ideas and viewpoints to be shared. An important issue, for which we believe a consensus has not yet emerged, concerns the measures of dependability and security; this issue will necessitate further elaboration before being documented consistently with the other aspects of the taxonomy that is presented here.

The paper has no pretension of documenting the state-of-the-art. Thus, together with the focus on concepts, we do not address implementation issues such as can be found in standards, for example, in [30] for safety or [32] for security.

The dependability and security communities have followed distinct, but convergent paths: 1) dependability has realized that restriction to nonmalicious faults was addressing only a part of the problem, 2) security has realized that the main focus that was put in the past on confidentiality needed to be augmented with concerns for integrity and for availability (they have been always present in the definitions, but did not receive as much attention as confidentiality). The paper aims to bring together the common strands of dependability and security although, for reasons of space limitation, confidentiality is not given the attention it deserves.

Preceding Work and Goals for the Future. The origin of this effort dates back to 1980, when a joint committee on "Fundamental Concepts and Terminology" was formed by the TC on Fault-Tolerant Computing of the IEEE CS and the IFIP WG 10.4 "Dependable Computing and Fault Tolerance." Seven position papers were presented in 1982 at a special session of FTCS-12 [21], and a synthesis was presented at FTCS-15 in 1985 [40] which is a direct predecessor of this paper, but provides a much less detailed classification, in particular of dependability threats and attributes.

- *A. Avižienis is with Vytautas Magnus University, K. Donelaicio 58 LT-3000 Kaunas, Lithuania and the University of California at Los Angeles, 4731 Boelter Hall, Los Angeles, CA 90024-1596. E-mail: aviz@adm.vdu.lt, aviz@cs.ucla.edu.*
- *J.-C. Laprie is with LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse, France. E-mail: laprie@laas.fr.*
- *B. Randell is with the School of Computing Science, University of Newcastle upon Tyne, Claremont Tower, Claremont Rd., UK NE1 7RU. E-mail: Brian.Randell@newcastle.ac.uk.*
- *C. Landwehr is with the Institute for Systems Research, 2151 A.V. Williams Building, University of Maryland, College Park MD 20742. E-mail: clandwehr@isr.gov.*

Manuscript received 25 June 2004; accepted 25 Aug. 2004.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-0097-0604.

1545-5971/04/$20.00 © 2004 IEEE Published by the IEEE Computer Society

DSN 2000

Session Titles (both DCCS and PDS)

- Embedded Systems
- Language Support
- Measurement and Assessment
- Mobile Agents
- Analysis and Evaluation
- Theory
- Testing
- Byzantine Faults
- Group Communication
- Fault Injection
- Replication
- High-performance Architecture
- Distributed Systems Models

DSN 2023

Session Titles

- Vehicles
- Memory
- Blockchain and Replication
- Software Security
- Network Security and Privacy
- Machine Learning
- Obfuscation
- Cyberphysical systems
- Virtualization
- Web Security
- Mobile Systems and IOT
- Systems Analysis and Modeling
- Smart Home

Popular topics over the years

- **The eternal sunshine of byzantine-resilience**
  - Including blockchains, private and public distributed ledgers
- **New systems abstractions**
  - Virtualization, containerization, microservices
- **New computing platforms**
  - Public, private, hybrid clouds
  - Fog, edge, smart homes
- **Programmability (and automation) of everything**
  - SD-X
- **Test everything for some functionality**

Emerging/recent developments

- **Machine learning everywhere**
  - ChatGPT and open source similar technologies (Huggingface)
- **New communication advancements**
  - 5G, 6G, nextG – transforming the communication and enabling new applications
- **New applications**
  - Autonomous and connected vehicles, field robots for hazardous environments
  - Virtual reality, mix-reality telepresence
  - Smart-X (healthcare, agriculture, manufacturing, transportation, cities)
- **Human in the loop**
  - Machine-human cooperation
- **New computing paradigm**
  - Quantum computing

Looking in my crystal ball

- **Data**
  - How to collect it, how to store it, how to distribute it
  - How to control access to it, how to use it
- **ML algorithms **correctness****
  - How to ensure robustness, fairness, privacy
- **ML impact on design/understanding of systems**
  - Understand its implications for dependability (and other properties) of systems
- **Human-machine cooperation**
  - Understand dependability of such systems

The immediate future seems to be ``applied'' and ``human-centric''

New problems (?)

- **New different ways in which software is produced**
  - ML-only
  - ML/human cooperation
  - Quantum computing
- **Applications with requirements that we do not understand or for which need new methods/frameworks**
  - How to specify human-machine cooperation (beyond autonomous vehicles, think remote medicine)
  - What will the legal implications be?
- **What about hardware?**
  - Security problems in hardware dominated the last few years (side-channels)
  - With new applications and the quest for performance, these type of problems will continue

Dependability

``the quality of being trustworthy and reliable.''

``In systems engineering, dependability is a measure of a system's availability, reliability, maintainability, and in some cases, other characteristics such as durability, safety and **security**. In real-time computing, dependability is the ability to provide services that can be trusted within a time-period.''

Revisiting the role of dependability

Research questions

- How does dependability relate to goals for these emerging applications? (How does it relate to privacy, fairness, accountability)
- Are there new challenges in dependability in these new emerging applications?
- Do we need new definitions of dependability for new emerging computing paradigms, networks, and applications?
- Do we need new methods to model, measure, test, prove?

This Talk

Answer some of these questions in the context of quantum computing

Propose ideas moving forward to ensure that our community is anchored in fundamental ways in these problems and not at the periphery

Quantum computing

Qubits

- Quantum computers: representations of composite quantum systems
  - Classical computers process information by sequentially flipping digital switches representing 0s and 1s
  - Quantum computers use units called qubits that represent **multiple values simultaneously**
- Qubits
  - Do not need to process information sequentially
  - Can perform calculations significantly faster than bits, which can only do so using discrete values.
- Goal is to entangle as many qubits as possible to increase the processing power (exponentially)

Fault-tolerance

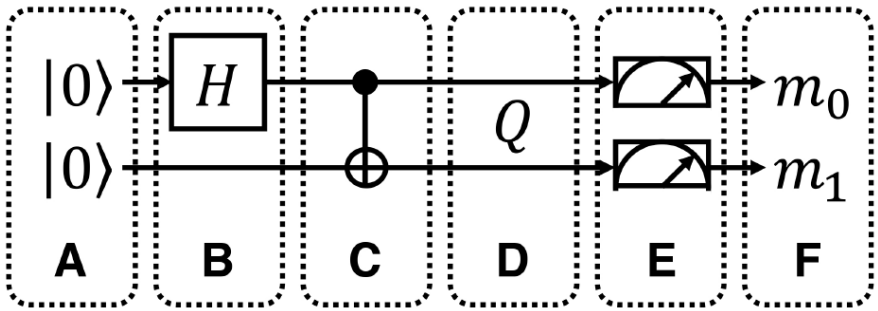
Decoherence

- Decoherence poses significant challenges to the dependability of quantum computers
- Extremely sensitive to environmental disturbances like temperature and dust, and disruptions to any part of a composite system can cascade across the whole system
- The current coherence time – the amount of time a qubit can store memory before succumbing to decoherence – world record is 2ms

A Quantum Computing Program

A Bell state creation quantum program

- A classical state of two qubits (A) is manipulated into a superposition state by a quantum operation (B).
- A controlled-NOT gate (C) induces entanglement between the two qubits to create an entangled state Q, which can no longer be factored into two separate pieces of information (D).
- Measurement of both qubits collapses the quantum state (E). Because the qubits were entangled when they were measured, the measurement results m_0 and m_1 are correlated (F).



Statistical tests on these measurements aid programmers in implementing and debugging quantum programs

Faults in Quantum Programs

Quantum Assertions Using Statistical Tests

- **Classical assertions**: a quantum variable should take on a deterministic (classical) integer value upon measurement
- **Superposition assertions**: a quantum variable in superposition should take on a probabilistic distribution of multiple values upon measurement
- **Entanglement assertions**: two or more quantum variables in an entangled state should take on associated (correlated) values once they are measured

Statistical Assertions for Validating Patterns and Finding Bugs in Quantum Programs Yipeng Huang and Margaret Martonosi, 2019

Benchmarks for Quantum Programs

Challenges

- Low-level approaches to measuring individual gate errors, qubit coherence times, or other hardware-level properties
  - (+) Understand exactly what process the quantum hardware was implementing in the presence of imperfect controls and noise
  - (-) Do not directly capture how the system will perform on real-world applications.
- Synthetic benchmarks that utilize random circuits to measure hardware performance
  - (-) Neither meaningful nor scalable. Typical quantum applications do not generally take the form of random quantum circuits and therefore the quantum volume and are not necessarily representative of useful workloads
- Single application benchmarks that focus on a particular use-case
  - (+) Increased scalability
  - (-) Many different applications are required to reflect the diversity of possible workloads.

Fault-tolerant Architectures for Quantum Computing

Magic-State Distillation

- **Noisy Intermediate-Scale Quantum** (NISQ) computers without error correction
- **Large-scale fault-tolerant** quantum computing machines
  - Use error correction
  - Fault-tolerant operations known as **magic-state distillation**
- Effective magic-state distillation must reconcile
  - Their usefulness for quantum applications
  - Their high overhead in physical area and latency

Magic-State Functional Units: Mapping and Scheduling Multi-Level Distillation Circuits for Fault-Tolerant Quantum Architectures Y. Ding, A. Holmes, A. Javadi-Abhari, D. Franklin , M. Martonosi and F, T. Chong

Quantum Computing

Impact

- **AI/ML:**
  - Quantum technology can process data more rapidly than classical machines, making quantum AI/ML tools more accurate and scalable
- **Communication**:
  - Significantly increase the connectivity and speed of the internet
  - Quantum internet links quantum devices together using entanglement
- **Security**:
  - Will be able to break traditional cryptography such as those based on factoring and DLP
- **Industries impacted**:
  - Automotive, financial, chemical, life sciences

Synergies

- NSF Expeditions: EPiQC: Enabling Practical-scale Quantum Computation

- **NIST Announces First Four Quantum-Resistant Cryptographic Algorithms from its six-year competition**
  - **General encryption,** used when we access secure websites, NIST has selected the CRYSTALS-Kyber algorithm.
  - **Digital signatures,** NIST has selected the three algorithms CRYSTALS-Dilithium, FALCON and SPHINCS+

Lots of open problems (and funding)

- How to model
- How to measure
- How to design fault-tolerant architectures
- How to evaluate/test these architectures

This Talk

Answer some of these questions in the context of quantum computing

Propose ideas moving forward to ensure that our community is anchored in fundamental ways in these problems and not at the periphery

Proposed actions

Revisit and extend the definitions considering the current landscape

Write a new TDSC like article on what dependability means for

- ML
- Autonomous and connected vehicles
- Quantum computing
- Human-in-the-loop systems

Proposed actions

Organize events

- Further looking into the future, start with the working group and make sure that some of this knowledge/efforts make it to dependability conferences
  - Workshop on quantum computing
  - Workshop on human-in-the loop systems
  - Workshop on beyond 6G and applications

Proposed actions

Keynote speakers

- Further looking into the future
  - Quantum computing
  - Legal aspects of autonomous driving
  - Virtual reality
- Revisit topics
  - Remote medical services
  - Remote work, remote education

Proposed actions

Super SG of Conferences in Dependability

- Coordination across all conferences in dependability
- Common action plan
- Leverage common experience
- Leverage fund raising
- Rotate some interesting events/speakers to give opportunities to all communities to participate/learn

Conclusion

- 25 years ago we were looking at the impact of security on dependability

- next 25 years ?
  - We should look not only at the immediate future but also long term and influence the design of new computing paradigms and applications instead of reacting once they already arrived