***Dependability in a connected world: From the very large to the very small***

*Target audience*

I will target 75% of the talk to be high level and understandable to anyone with a grounding in Computer Science and Engineering without esoteric research details. In 25% of the talk will I touch upon some recent research results.

*Abstract*

Much of the computational infrastructure that we encounter today and that we increasingly rely on for critical applications is provided by a distributed system, be it the air traffic control system or the smart electric grid or the cyber physical systems that embed sentient sensors in our physical spaces. Dependability is the property that the system continues to provide its functionality despite the introduction of faults, either accidental faults (design defects, environmental effects, etc.) or maliciously introduced faults (security attacks, either external or internal). The distributed systems are increasing in scale, both in terms of the number of executing elements and the amount of data that they need to process. For example, algorithms for solving complex genomics problems are requiring a larger and larger number of executing elements, while the amount of genomics data that the algorithms have to process are also increasing rapidly. Another emerging trend in distributed systems is that they are being built out of heterogeneous components – different kinds of computer platforms and different software platforms. These two trends have thrown the challenge to the designers of computer systems of how to ensure their high dependability.

To ensure dependability, we need to address five elements ⎯ detection (tell quickly that there is something wrong), diagnosis (what is the root cause of the failure), containment (how to prevent the failure from propagating through the system), recovery (recover the failed system to an operational state), and in some cases, prediction (of an impending failure, so that proactive mitigation actions can be triggered). The dependability mechanisms must not overly intrude on the application or the execution environment, either in terms of performance impact or in terms of the level of changes that is required from them.

In thistalk, I will describe briefly traditional dependability architectures and in more detail, newer mechanisms that are being devised to handle the challenges mentioned above. I will give a few case studies where dependability mechanisms have achieved great success (financial transactions, embedded medical devices) and some systems studies where a distressing lack of dependability mechanisms have come to the fore. I will then give snapshots of research projects in our lab that are addressing different dependability problems. I will conclude by pointing out lessons in systems design that should apply to the emerging classes of distributed systems.

**Biographical Sketch**

****

Saurabh Bagchi is a Professor in the School of Electrical and Computer Engineering and the Department of Computer Science (by courtesy) at Purdue University in West Lafayette, Indiana. He is an ACM Distinguished Scientist (2013), a Senior Member of IEEE (2007) and of ACM (2009), a Distinguished Speaker for ACM (2012), an IMPACT Faculty Fellow at Purdue (2013-14), and the founding Director of a university-wide resiliency center called CRISP. He is the recipient of a Google Faculty Award in 2015 and the AT&T Labs VURI Award in 2016. He has also been a Visiting Scientist at IBM Research since 2011.

Saurabh's research interest is in distributed systems and dependable computing. He is proudest of the 15 PhD students who have graduated from his research group and are in various stages of building wonderful careers in industry or academia. In his group, he and his students have far too much fun building real systems and transitioning them to practice. Saurabh received his MS and PhD degrees from the University of Illinois, Urbana-Champaign (Computer Science) and his B.Tech. degree from IIT Kharagpur (Computer Science and Engineering).

**URL: https://engineering.purdue.edu/~sbagchi/**