Towards Systematic Roadmaps for Networked Systems[*]

Bin Liu[*], Hsunwei Hsiung[‡], Da Cheng[‡], Ramesh Govindan[*], Sandeep Gupta[‡]
[*]Computer Science Department, [‡]Electrical Engineering Department
University of Southern California, Los Angeles, CA, USA
{binliu, hsunweih, dacheng, ramesh, sandeep}@usc.edu

ABSTRACT

Networked systems have benefited from unprecedented growth in hardware capabilities, but, as we move closer to the end of the Moore's law era, future networked systems are likely to be more constrained by hardware capabilities than they have been in the past. We take the position that the networking community should, in response to this development, proactively and systematically develop *networking roadmaps*, which attempt to predict how trends in hardware capabilities will impact networked systems. In this paper, we discuss a possible methodology for developing networking roadmaps, and present two case studies that illustrate the methodology and reveal how increasing hardware unreliability can affect the performance of routing and transport protocols.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols; B.0 [**Hardware**]: General

General Terms

Design, Documentation

1. INTRODUCTION

Networked systems have benefited from unprecedented growth in hardware capabilities. High speed switching fabrics, data centers, networked sensing, and wireless and mobile computing (to name a few) would not have been possible without improvements in the design of underlying circuits and devices, in terms of speed, reliability, power, cost, etc. Because the evolution of networked systems is strongly determined by hardware advances, it is instructive to examine projected trends in hardware.

The hardware community invests heavily in the development of its roadmaps, of which the ITRS roadmap for chips [3] is the best known (Section 2). These *semiconductor roadmaps* project future directions for chips and systems in terms of a wide range of important metrics, particularly computational and storage capacities, performance, power, cost, yield, reliability (lifetime), and resilience (to internal and external noise).

Recent versions of semiconductor roadmaps show that *technology improvements are generally slowing down*. In particular, improvements are continuing in some dimensions (e.g., cost-per-transistor), slowing in others (e.g., speed), and recessing in others (e.g., power, reliability, and resilience). This slowdown will continue as we move close to the end of Moore's law era and beyond. Due to this change in technology trends, we expect future hardware capabilities to more strongly constrain the development of networking software than the networking community has been accustomed to, since application requirements will start to exceed the capabilities of the underlying hardware systems.

Given these trends, we take the position that the networking community should devote some of its resources to systematically develop a *roadmap for networked systems*. Such a roadmap would (Section 3): a) *project* how future application demands or other considerations would drive system *requirements* in one or more dimensions (e.g., data center applications driving requirements for switch traffic speeds, server processing capabilities, the degree of parallelism required, the availability of servers and networking components, or overall system power); b) *match* these requirements with hardware roadmaps to understand what constraints future hardware will impose on networked systems.

These projected constraints will inform research efforts to develop alternative hardware and software tech-

[*]This material is based upon work supported by the National Science Foundation under Grant No. CNS-1117049. Any opinions, findings and conclusions or recomendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

Hotnets '12, October 29–30, 2012, Seattle, WA, USA.

niques to meet application requirements. For example, if hardware roadmaps project increasing memory unreliability, the corresponding networking roadmap may be able to determine *which* networked systems are affected, *when*, *to what extent*, and *at what levels of unreliability* (Section 4). This roadmap can be used to determine which failure masking technique (e.g., via redundancy in space, through coding, or in time, through retransmissions) would be necessary and when. This examination would also be able to estimate the storage, network or computation *cost* of these techniques.

While the networking community has reacted to developments in hardware capabilities (Section 5) a networking roadmap requires a more proactive and systematic look at the "cliffs" we are likely to face under different projections of the future of hardware components. A research agenda for a networking roadmap should answer the following questions: What is a networking roadmap? How can we systematically develop a networking roadmap? How (if at all possible) can we build systems that evolve according to roadmap projections? In this paper, we take a *first* step towards addressing some of these questions.

2. SEMICONDUCTOR ROADMAPS

In the early years of integrated circuits, Gordon Moore captured [11] the trends in, and projected, the rate of growth of the number of transistors in a chip and the corresponding decrease in price per transistor. Because these have proven to have predictive power, projections of future trends (or *roadmaps*) have since been a critical activity in the semiconductor sector.

ITRS Roadmaps. By now, the process of preparing roadmaps has become formalized and these roadmaps have become extensive. For example, the International Technology Roadmap for Semiconductors (ITRS) [3], is sponsored by the largest industry organizations in five leading chip manufacturing regions in the world and is compiled with significant inputs from industry and academic experts. This roadmap covers the entire range of activities in this sector, including research in materials and devices; all aspects of manufacturing, including lithography, metrology, chip assembly and packaging, test equipment, factory integration, and environmental and health; all types of chips and components, including radio frequency, analog, mixed-signal, digital, and micro-electro-mechanical; all aspects of tools and methods, including modeling and simulation, design, and yield enhancement; as well as the primary system drivers, i.e., new applications that challenge the semiconductor industry and fuel its growth. Each chapter of the ITRS roadmap is an extensive catalog of the trends and projections of virtually every important factor that may affect some aspect of the semiconductor sector.

In addition to serving the semiconductor industry, ITRS also projects, for major *users* of chips (computing, consumer electronics, health and wellness, and networking), metrics governing the next generation hardware chips and systems, particularly computational and storage capacities, performance, power, cost, reliability (lifetime), and resilience (to external and internal noise).

Roadmap Example: Static RAMs (SRAMs). SRAMs are central to many aspects of networking, since they may be used to buffer packets or parts thereof, or store control information such as routing tables.

The ITRS roadmap for SRAMs is extensive, so we focus on one part: consumer-relevant parameters for SRAMs. The roadmap covers SRAMs for many types of applications, such as portable devices, game consoles, and computing; for example, within computing, it covers SRAMs for cost-performance (CP) microprocessors (for desktops), and high-performance (HP) microprocessors (for servers). For both these types of microprocessors, the roadmap projects that the amount of SRAM available on a constant die area (140mm^2 for CP, 160mm^2 for HP) will double with each successive technology generation. (Interestingly, the 2011 roadmap does *not* predict Moore's law, i.e., that successive technology generations will become available in a particular time duration, say, every 18 months.)

In addition, the roadmap also provides projections for many other SRAM characteristics of interest to chip designers, computer architects, and those who use processors as building blocks of their hardware or hardware-software system. Tables 1 and 2 summarize the projections for power supply voltage (V_{dd}), power consumption, read and write delays, and read and write failure rates for each generation of CMOS technology. (The total power is estimated using SRAM size, access time (read/write time), and the per-cell static and dynamic power.) Note the increasing SRAM read-write failure rates and increasing power requirements: with increasing technology density, SRAM needs more power due to higher leakage, and becomes more vulnerable to manufacturing process variations.

Table 1 assumes existing circuit technology and standard testing approaches. Circuit and architecture innovations can decrease power consumption. New testing approaches will eliminate a large proportion of chips with SRAMs that are failure-prone due to manufacturing variations and reduce the failure rates for chips sold to customers, at the expense of lower yields and higher costs. To combat this, other techniques (e.g., fault masking in software) may be needed to control costs.

Off-roadmap Excursions and Near-threshold computing (NTC). ITRS roadmaps play an important role in making many crucial decisions, e.g., developing specifications for the next generation computing systems and

Table 1—SRAM Roadmap for Consumer-relevant Parameters [2]

Technology (nm)	65	45	35	25	18	13
Size (MB)	4	8	16	32	64	128
V_{dd} (V)	1/1.1	1	0.9/1	0.7	0.7	0.7
Static power (mW)	$3E-04$	$5E-04$	$1E-03$	$2E-03$	$3E-03$	$5E-03$
Dynamic power (mW/mHz)	$6E-07$	$5E-07$	$4E-07$	$4E-07$	$3E-07$	$2E-07$
Write/Read time (ns)	1.5	1.2	0.8	0.5	0.3	0.3
Total power (W)	22.4	58.67	200	716.8	2048	5803

Table 2—SRAM failure rates [1]

Technology (nm)	65	45	32	22	16	12
Read failure rate	-	$3E-07$	$2E-04$	$1E-02$	$6E-02$	$2E-01$
Write failure rate	-	$3E-03$	$1E-02$	$4E-02$	$1E-01$	$2E-01$

making decisions about R&D investments. Their influence is so widespread, that occasionally some have wondered whether these roadmaps sometimes become a self-fulfilling prophecy and limit innovation! This is not typically true, since the exponential improvements in integrated circuits over five decades have been enabled by many disruptive changes enabled by engineers and scientists pursuing *off-roadmap excursions*. The roadmaps are continually updated as a result of these excursions.

One such recent off-roadmap excursion is an idea called near-threshold-voltage computing (NTC [7]), which is motivated by the increased total power consumption of SRAMs (Table 1) and logic circuits despite lower power supply voltage (V_{dd}) in future technology generations. NTC posits that many components can be operated at supply voltages well-below the 2011 projections, resulting in potentially significant power savings.

3. NETWORKING ROADMAPS

With the projected slowdowns in hardware technology growth, we take the position that the networking community should develop systematic *networking roadmaps* to understand how future generations of hardware will constrain networked systems. A networking roadmap attempts to project the properties (e.g., performance, reliability, availability) of important networking and distributed software subsystems (e.g., TCP, routing, distributed storage) using projections of hardware obtained from semiconductor roadmaps. We envision an analysis methodology to develop a networking roadmap that takes the following generic form:

- *Projecting* application demand to quantify desired critical properties of networked systems;
- *Deriving* the hardware component(s) that may present a barrier to achieving one or more of these critical properties, and using the hardware roadmaps to understand how these components are likely to evolve over time;
- *Roadmapping* the application behavior on the derived hardware systems in order to assess the end-to-end impact of hardware on applications.

This methodology is inspired by the ITRS methodology, but differs in some ways, as discussed below.

There are three important aspects of this analysis methodology. First, many of the steps necessarily involve judgement of experts; for example, quantifying the robustness, or determining the hardware components that present a barrier to robustness require judgement and may not yield unique answers. In this case, the steps above might be repeated for different values of these "parameters", or by selecting best-case and worst-case parameter sets. Second, roadmapping is necessarily approximate given the margin of error in the hardware roadmaps as well as the errors in judgment, so this analysis should focus on understanding qualitative trends and when inflection points are likely to set in; this is discussed in detail below. Third, we expect this process to be iterative: when new networking applications (or application classes) emerge or hardware roadmaps change, this analysis methodology should be re-executed.

Projecting Requirements. The first step in our analysis methodology is to project application needs into requirements for the underlying platform or system that would be used to realize these applications.

There are two challenges in doing this. First, networking is a broad and rapidly evolving field, and the requirements for two different networking applications can vary significantly in one or more dimensions, e.g., data-parallel computations in data centers vs. content delivery in delay-tolerant networks. Thus, any attempt to derive a generic set of networking system requirements will likely degenerate into a least common denominator specification that may not result in a useful networking roadmap. For this reason, we recommend defining *separate roadmaps* for different classes of networks (data centers, the Internet, static wireless meshes, mobile networks, delay-tolerant networks). We discuss below how to distinguish different classes of networks.

The second challenge is to find the right granularity at which to specify system requirements. At one end, a designer may specify a detailed design that describes for example, for a projected set of data center applications, the number of nodes, the precise interconnect, the switching capabilities and link speeds at different tiers, and server processing and storage capabilities. At the other end of the spectrum, one may characterize system requirements using estimates for a single property (e.g., bisection bandwidth, or end-to-end latency).

In general, more detailed specifications can increase the predictive power of roadmaps. However, such specifications are harder to predict correctly, which can result either in inaccurate or infeasible roadmaps. On the other hand, specifying a single requirement (e.g., bisection bandwidth) leaves other potentially important requirements (e.g., end-to-end latency) unspecified, and can also result in optimistic roadmaps.

We now discuss two principles that address this tension. First, to construct a roadmap, a designer specifies requirements by quantifying a small number of desired *system properties*. For example, one might believe that future networked systems may need 5-nines availability, or that (wired) link speeds of 500 Gbps will become common in a few years. As discussed above, these predictions should be tailored to specific kinds of network systems (data centers in our example). The principle guiding the choice of system properties for a given class of networked system is that the designer should select *critical properties*, mainly those that represent fundamental constraints for applications.

Different classes of networked systems are distinguished by different sets of requirements: for example, to a first approximation, availability and latency are primary capabilities in data-center systems, robustness and reachability in Internet-scale systems, energy in wireless sensor networks, or cellular bandwidth usage and energy in mobile computing systems. These different classes also have a common requirement (the scale of the network), but may differ by an order of magnitude in that dimension (e.g., data centers vs. Internet vs. networked sensors). Hence, our prescription for deciding whether a network system qualifies for a separate roadmap is that it should have different sets of critical properties from other networked systems, or, if its set of critical properties matches that of another, should differ in one or more critical properties by an order of magnitude.

Second, since projection is inherently erroneous, conclusions drawn from network roadmaps should be *qualitative*. That is, in using these projections, the designer should look for trends and changes in trends (*inflection points*). To take a simple example, suppose that a hardware roadmap predicts increasing unreliability of a particular hardware component as a function of semiconductor manufacturing technology. Using our roadmap methodology, the designer may find that TCP's performance degrades slowly until, at a certain technology feature size, the protocol is practically unusable. Our principle suggests that the designer should pay little attention to the slope of the degradation (a *quantitative* concern) but focus on the fact that, at some *inflection point*, the hardware component becomes unreliable enough for the application. Before these inflection points hit, the community should be prepared with mitigation strategies for extending the usability of the networking subsystem (TCP in our example).

Deriving Hardware Components. Given these critical properties, the next step in the process is to generate one or more *canonical hardware configurations*. A canonical hardware configuration instantiates a networked subsystem with the appropriate hardware components (nodes/devices, memories, storage systems, links, controllers etc.) such that the resulting system has (approximately) the desired critical properties.

There may be more than one canonical hardware configuration for a given set of critical properties. This is because, by definition, the critical properties only specify a subset of the properties of the system. For example, if power is not one of the critical properties, different canonical configurations may have different power requirements. Similarly, different canonical configurations may have different *costs* (hardware roadmaps also project components costs); some may be cost-neutral (i.e., have the same inflation-adjusted cost as a similar component costs today), while others may not.

Finally, some sets of critical properties may be *infeasible* since there may not exist a canonical hardware configuration, or all possible canonical configurations may be prohibitively expensive. In this case, the designer: (a) has learned about the infeasibility of a hardware configuration, and (b) can iteratively explore *relaxations* of one or more critical properties to determine feasible canonical configurations. Thus, the designer can identify that critical property whose relaxation can ensure feasible hardware designs, and can explore whether this relaxation can be compensated for in software.

Ideally, generating a canonical hardware configuration should be done by a design tool. Designing such a tool itself is a major intellectual challenge. Hence, we expect this process to be initially manual, involving a collaboration between network system designers and hardware architects, until enough experience has been gained to devise a design tool.

Roadmapping. The final step in the process is *roadmapping*: understanding how application behavior evolves over time, and when inflection points may develop in some aspect of application performance. Mechanistically, roadmapping proceeds as follows. A designer first selects a class of networked systems, and, within that class, a software "application" whose behavior is to be studied. We use the term application loosely: from a hardware perspective, any software subsystem (e.g., reliable transport protocols, consensus sub-systems, key-value stores, etc.) running on top of hardware would qualify. For this choice of application and networked system, the designer projects critical properties for different points in time (e.g., one set for 2015, another for 2018). For each set, she generates, using the design tool, one or more canonical hardware configurations.

Finally, she *evaluates* the application on the canonical hardware configurations to understand how application functionality and performance are affected by different hardware configurations. This step is necessary because, although a canonical hardware configuration may satisfy critical properties, it may be that specific choices of hardware components may affect *end-to-end* performance in unforeseen ways. This evaluation step "closes the loop", and helps approximately quantify application behavior as a function of technology; this output is the *roadmap* for the given application. The roadmap can then be used to identify trends and inflection points.

Roadmapping can either be done using mathematical modeling, or simulation, or some combination thereof. Modeling and simulation using network simulators can provide coarse roadmapping, while a hybrid simulator that integrates circuit simulators with network simulators may provide more precise roadmaps. Developing these simulation and modeling techniques is the research challenge in roadmapping.

Once a roadmap has been developed, and inflection points have been identified, it will be necessary to explore *mitigation* strategies. These strategies counter the adverse effects of projected hardware trends by performing appropriate trade-offs: e.g., increase reliability in hardware or software by fault-masking or replication, or hide latency degradations by caching or prefetching.

4. CASE STUDIES

To illustrate some of the methodology for, and the benefits of, developing networking roadmaps, we discuss two *preliminary* case studies that focus on one major limitation expected of all future technologies: decreasing levels of resilience to internal and external noise.

Case Study 1: Impact of SRAM Unreliability on Transport Protocols. As Table 2 shows, SRAM failure rates are projected to increase with CMOS technology *scaling*. SRAM failures result in erroneous values read from or written into memory cells. A primary cause of SRAM failures is the mismatch in the strength of transistors in individual memory cells [12]. CMOS technology scaling aggravates manufacturing process variations, and these *process variations* intensify the mismatch between the fabricated transistors. Transistor mismatches can be detected by testing after fabrication; as process variations increase, lower yields may result (i.e., chips may be erroneous at manufacture), increasing overall memory costs. A secondary cause of failures, which we ignore, is bit-flips during memory operations resulting from alpha particles or cosmic rays [13].

The other trend from the roadmap (Table 1) is that SRAM power usage is predicted to increase dramatically. To counter this, some researchers have proposed to use lower supply voltages from the nominal 1.2V down to "near threshold" voltages (400-500mV) to operate electronic components; these voltages represent the limits of operation of these components. This off-roadmap excursion on Near-threshold Computing (NTC) [6, 7] holds the promise of factors of 2 or 3 reduction in power; while it seems a promising technique for tackling, say, datacenter energy consumption, NTC can also increase *bit failure rates* (BFR) for SRAMs (Figure 1(a)).

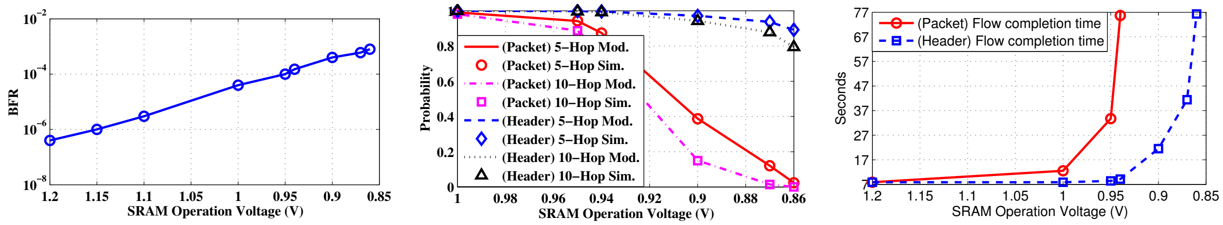
Qualitatively, these trends spell trouble for components such as switches and routers which use SRAMs, either for packet buffers (less frequently, since SRAM is expensive) or for packet headers (in on-chip memory). To understand more precisely how NTC affects networked systems, consider the following roadmapping methodology, modeled on the discussion in Section 3.

Network Class	Data center
Projection	Critical properties are bisection bandwidth and end-to-end packet latency
Derivation	A chain topology of routers with sender at one end, receiver at the other. Two canonical hardware configurations, one where *Whole-Packet*s are buffered in SRAM, another with *Headers-Only* in SRAM.
Roadmapping	Evaluate, analytically and in simulation, end-to-end packet drop rate and TCP flow completion times

We have conducted a complete road-mapping exercise by building an analytical model of end-to-end packet loss rates in this chain topology, and validating it with *ns-2* simulations. Our model takes into account error-correcting capabilities of memories (most SRAMs use ECC), as well as IP header checksums, MAC-layer CRCs, and the TCP checksum. We omit the details of the analytical model for brevity, but the results are intriguing: as Figure 1(b) shows, end-to-end packet delivery probabilities show an inflection point at about 0.94V with the *Whole-Packet* configuration and 0.88V with the *Header-Only* configuration. These voltages are quite far from the CMOS threshold voltage of 0.4V. Well above these voltages, TCP flow completion times (Figure 1(c)) show an inflection (at 1V and 0.94V, respectively). These preliminary results suggest that NTC, without additional software changes, is unlikely to reduce data center power consumption without significantly affecting performance. We emphasize that our conclusions do not indicate that NTC is inherently a bad idea; for example, DRAM reliability is relatively unaffected by NTC, so systems that use DRAMs can exploit the full potential of NTC.

Case Study 2: Low voltage TCAMs. TCAMs are becoming indispensable components of networked systems and will likely see widespread deployment in routers and switches as OpenFlow-style programmable networking becomes common. Interestingly, we discovered that TCAMs can be unreliable at low voltages. Using a methodology discussed in [9], we found that the asymptotic probability of *at least one cell being incorrect* under NTC, even in small TCAM arrays where every cell is written exactly once, can approach 1 at 1V.

Implications. We emphasize that these results are preliminary, and merely illustrate the kinds of conclusions one can draw from a systematic attempt to match hardware roadmaps with networked subsystems' end-to-end performance. Many of the inflection points illustrated above can be mitigated using a variety of (not mutually



(a)

(b)

(c)

Figure 1—Effects of SRAM Unreliability on a Transport Protocol: (a) SRAM bit-failure rates. (b) End-to-end error-free packet delivery probabilities for whole-packet and header-only configurations under 5 and 10 hops (theoretical and simulation results). (c) TCP flow completion times for whole-packet and header-only configurations (simulation results).

exclusive) techniques, each of which has different cost implications: more extensive chip testing, new hardware approaches to improve resilience, node-level software fault masking, end-to-end error detection and correction, and so forth. More generally, systematic roadmapping can open up new research directions to address hardware constraints using novel software techniques.

5. RELATED WORK

The networking community has had a good history of recognizing hardware trends and pursuing research agendas to address limitations imposed by hardware. Examples include developing fast packet forwarding algorithms to deal with memory latency [14], re-arranging packet processing rules to match TCAM power management capabilities [10], designing novel data center interconnects to achieve high bisection bandwidth using commodity switches [4], devising software radios to leverage processing power improvements [5], and exploring in-situ sensing using advances in device miniaturization [8]. Our paper makes the case for a concerted effort to use available semiconductor roadmaps to proactively develop networking roadmaps.

The semiconductor sector has been actively roadmapping future technologies periodically for some time [3]. Off-roadmap excursions, like NTC [7], illustrate attempts to balance different dimension of future hardware trends (power and reliability). Our proposed methodology for networking roadmaps is inspired by the methodologies adopted by the semiconductor sector, but differs crucially in its reliance on qualitative assessments.

6. CONCLUSIONS

As we approach the end of an era of Moore's law fueled hardware improvements, the networking community needs to develop systematic roadmaps for networked systems modeled after, and informed by, semiconductor roadmaps. Such an exercise can better direct research resources to relevant networking problems identified by roadmap inflection points, at which innovative solutions may be needed to address constraints imposed by future generations of hardware. Our case studies on the reliability of memory technology illustrate both the methodology of developing roadmaps and the kinds of insights that may be obtained from roadmapping. In the future, we expect to work out details of the roadmapping methodology, develop design tools for the derivation step, and modeling methods for roadmapping.

Acknowledgements. We would like to thank our shepherd Matt Caesar and the referees for their comments.

7. REFERENCES

[1] International technology roadmap for semiconductors 2011 design. http://www.itrs.net/Links/2011ITRS/2011Chapters/2011Design.pdf.

[2] International technology roadmap for semiconductors 2011 system drivers. http://www.itrs.net/Links/2011ITRS/2011Chapters/2011SysDrivers.pdf.

[3] International technology roadmap for semiconductors. 2011.

[4] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *Proc. ACM SIGCOMM*, 2008.

[5] V. Bose. *Design and implementation of software radios using a general purpose processor*. PhD thesis, MIT, 1999.

[6] R. Dreslinski, M. Wieckowski, Blaauw, D. Sylvester, and T. Mudge. Near threshold computing: Overcoming performance degradation from aggressive voltage scaling. In *Proc. ISCA Workshop on Energy-Efficient Design*, 2009.

[7] R. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge. Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits. *Proceedings of the IEEE*, 98(2):253–266, 2010.

[8] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Networking*, 11(1):2–16, 2003.

[9] B. Liu, H. Hsiung, D. Cheng, R. Govindan, and S. Gupta. Towards Systematic Roadmaps for Networked Systems. Technical Report 12-931, University of Southern California, October 2012.

[10] Y. Ma and S. Banerjee. A smart pre-classifer to reduce power consumption of tcams for multi-dimensional packet classification. In *Proc. ACM SIGCOMM*, 2012.

[11] G. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), 1965.

[12] S. Mukhopadhyay, H. Mahmoodi, and K. Roy. Modeling of failure probability and statistical design of sram array for yield enhancement in nanoscaled cmos. *IEEE Tran. Computer-Aided Design of Integrated Circuits and Systems*, 24(12):1859–1880, 2005.

[13] T. Semiconductor. Soft errors in electronic memory-a white paper. 2004.

[14] V. Srinivasan, S. Suri, and G. Varghese. Packet classification using tuple space search. In *Proc. ACM SIGCOMM*, 1999.