Research Statement

Ramana Kompella

I am primarily interested in the design, implementation and analysis of software and hardware to address the complex management tasks in large-scale networks. Broadly speaking, my research involves a combination of algorithms, statistical inference, and router architecture applied to solving problems related to measurement, security, and fault isolation in networks. My research has also been influenced by my prior industrial experience designing a packet classification architecture (at SwitchOn) and implementing a hardware TCP offload engine (at Chelsio). Next, I describe the specific research problems I have worked on during my graduate studies.

Dissertation work

Despite its tremendous success, the Internet remains fragile with frequent outages and service disruptions. It is estimated that the current reliability (in terms of availability) of the Internet is 94-99% (1-2 nines), considerably lower than the 99.999% (5-nines) reliability of the telephone network. Telephone networks are also simpler to manage; it requires less than 10 technical personnel to maintain a telephone facility compared to a few hundred of them required to manage an IP Point-of-Presence (PoP). Therefore, as we approach the next-generation Internet, I believe all stake-holders—researchers, providers and equipment vendors—are shifting focus from just high performance to include high reliability and manageability as well. My past contributions as well as future interests lie in this exciting area of networking research.

There are many challenges in transforming networks to make them more reliable and manageable. To begin with, backbone networks are extremely complex with a large number of network elements with complex cross-layer interactions (e.g., dependencies between SONET and IP). Further, constant evolution in protocols, hardware, applications and features makes these networks a moving target. At any given time, a real ISP operator may be simultaneously dealing with replacing failed optical amplifiers, upgrading router line cards, and installing software patches. In addition, with 40 Gbps links and millions of concurrent TCP flows, it is difficult to continuously measure and monitor networks.

In order to cope with these challenges, many providers resort to the general paradigm of measure (observe performance degradations), model (perform “what-if” analysis) and control (apply specific changes) through a management plane outside the network. In order for such a feedback loop to function effectively, several components are required. I now describe two such components that I have worked on—automated fault management and scalable network-wide measurement and monitoring.

Fault localization: Faults are detected in the network using a combination of software and hardware self-monitoring mechanisms (e.g., OSPF Hello messages) that are inbuilt into network elements. In addition, external end-to-end probes are also used to monitor performance degradations. On detection of a fault, the network elements often temporarily fix the problem by re-routing traffic through other functional paths. While re-routing is useful in the short run, the fault must still be repaired permanently because the network is operating with reduced capacity. The repair procedure involves first localizing the fault (that takes several minutes to a day), second determining the root-cause (few days), and finally fixing the problem permanently.

Unfortunately, determining the root-cause and repair are necessarily manual procedures. However, automated fault localization alleviates the burden on operators to manually isolate the failure. For example, consider a typical IP link with more than 100 components; manually localizing the failed component is labor-intensive. Further, in certain “silent-failures”, the network elements fail to automatically self-heal, thus causing connectivity-loss to many customers. In such cases, fast localization is in the critical path of the traffic recovery process; fast localization reduces customer downtime significantly.

In order to perform automated fault localization, I have developed a risk-modeling methodology along with collaborators at AT&T Research. At a high-level, the risk-modeling approach involves the creation of a dependency relationship between monitored symptoms and potential root-causes. For example, a single fiber-cut can simultaneously fail all the IP links that are carried over the fiber; the fiber therefore represents a shared-risk and the IP link failures represent the observable symptoms. While a single fiber cut is easy to localize, the problem is more challenging when there are simultaneous failures (which can be explained by multiple hypotheses) and noise (which can lead to spurious and lost symptoms).

We use Occam’s razor to reduce the fault localization problem to obtaining a minimum cardinality hypothesis that explains all symptoms. In turn, we find such a hypothesis by identifying a minimum set-cover in a bipartite dependency graph, for which we use a variant of the greedy approximation algorithm (since finding set-cover is NP-hard).
I have designed and built two different systems—link- and path- fault localization systems—based on our
risk-modeling methodology. Both these systems have been deployed (currently used by ATT researchers, but
in transition to operational deployment) in a tier-I backbone network for more than a year. The link-fault
localization system diagnoses IP link failures caused by optical component failures [NSDI’05]. In our experience
with more than 3000 real failures, we have observed that the system narrowed down the set of suspects to
less than 5% of the total components for more than 40% of real-failures observed.

On the other hand, the path fault localization system is designed to localize silent failures occurring in
an MPLS network. MPLS paths are established via IP routing protocols and IP topology changes trigger
subsequent MPLS path reconfigurations. The symptoms in this system are dropped end-to-end probes that
monitor the state of MPLS paths. The root causes are the IP links where packets are being dropped. Upon
detecting a failure using the periodic probes, our system outputs in less than a minute, the most likely set
of links that can explain the observed symptoms. Thus, the total downtime of the failure effectively reduces
from a few hours to a few minutes.

Our system localized faults with an accuracy (percentage of ground-truth in the hypothesis) greater than
80%, when we correlated about 600 non-silent failures. In general, it is hard to obtain accurate ground-truth,
even for non-silent failures. Therefore, for this correlation, we relied on approximate ground-truth obtained
via OSPF link-state advertisements [Infocom’07]. We have also manually verified three known silent failures
that our system could localize within seconds. One such silent failure was caused by the forwarding component
of a router line-card that failed to dequeue packets until the card was reset. While old problems eventually
disappear because of new router work-arounds, it is the experience of operators that new router bugs continue
to appear. Thus we believe our fault localization system will remain valuable to network operators even when
exact failure modes change.

One of the major challenges involved in working in the area of network management is the need to abstract
and formulate clean and interesting problems without getting caught up in the operational details. With
my ongoing and well-established collaboration with AT&T research, I am hoping to continue to make a real
impact by solving real problems born out of operational need. In this process, I also hope to introduce new
problems to the larger research community.

In addition to fault management, another important component required in the management plane is
measurement. Effective network-wide measurement capability is critical for a wide-variety of management
tasks including capacity planning, billing, accounting, QoS guarantees for service level agreements (SLAs). I
now describe my contributions in this area of research.

**Scalable Measurement and Monitoring:** Fundamentally, there are two commonly used approaches for
measurement—active and passive measurement. Active measurement involves injecting controlled synthetic
traffic to measure metrics of interest while passive measurement requires network element support. Active
measurements are both intrusive (interfere with regular traffic) and unscalable (O(n^2) probes for n end-points).
On the other hand, current passive measurement approaches are either too coarse (SNMP, per-interface) or too
heavy-weight (NetFlow, per-flow). In my research, I have designed scalable passive measurement approaches
that trade-off granularity for scalability, which I explain below.

**DoS detection:** Currently, most solutions for denial-of-service (DoS) attacks are deployed at or closer to the
end-host. I, along with collaborators, argue that attack detection also needs to be performed in the network
to prevent collateral damage. This in turn needs scalable attack detection primitives. I proposed a general
design framework in which I argued that any scalable primitive has to tackle two main issues—behavioral
aliasing and spoofing [ToN’07]. I designed a scalable detection primitive called partial completion filter (PCF)
that can detect a class of protocol attacks where sessions are initiated but not terminated. For two examples
of such attacks—the classic SYN flooding and worm scanning attacks—I also showed how a service provider
can instrument the primitive to perform attack detection in the network.

**Measurement:** With a collaborator, I proposed a scalable measurement technique called flow slicing for
traffic measurement [IMC’05]. Flow slicing is motivated by the fact that existing passive measurement solutions
do not provide a mechanism to independently control the usage of three most important resources in a
router—CPU, memory and reporting bandwidth, thus sacrificing accuracy in the presence of adversarial traffic
conditions. In flow slicing, CPU usage is controlled via packet sampling, memory usage via flow sampling in
order to adapt to various traffic mixes. Multi-factor sampling controls reporting bandwidth by identifying the
right set of flows to export so that accurate unbiased estimates of the required metrics can be computed.

**Scalable passive measurement:** In ongoing work, we are building a scalable, next-generation, clean-slate
network architecture that makes measurement a first-class entity. We ask the question: how can we architect
routers from the ground up so that they directly provide the measures of interest without requiring the indirect
methods used today. We observe that many stake-holders independently want to measure Internet characteristics for various applications such as overlay route-selection (overlay providers), failure and performance monitoring (ISPs) and characterizing Internet properties (researchers). These large and often interfering sets of independent measurements can make the data unreliable.

Our main idea (under submission to SIGCOMM’07) is to provide measurement as a service by employing scalable passive measurement primitives. We use two observations in our approach. First, end-to-end metrics (e.g., delay) can be composed from individual hop-level metrics (router and link propagation delay). Second, within a router, the number of possible paths a set of packets take is relatively small; the entropy within a router can be categorized into what we call measurement equivalence classes (MEC). Scalability is therefore, achieved by decomposing the path metrics to hop metrics and then further aggregating hop metrics using MECs. Incremental deployability is achieved by interconnecting clouds of compliant routers and disseminating measurement information via measurement-state packets (inspired by link-state packets).

Other Research

Apart from my dissertation work, I have also worked on other problems in both wireline and wireless networks, of which I describe two.

Cooperative scheduling protocol: In a wireless network setting (e.g., 802.11, sensor networks), I have proposed, designed and built a distributed cooperative scheduling protocol that enables multiple wireless nodes to share access to the channel according to a globally consistent scheduling policy [Sensys’03, SIGCOMM E-Wind’05]. In this project, I modified the default 802.11 wireless MAC driver in the kernel to implement different scheduling disciplines that optimize different global objectives.

High-speed packet buffers: In a different project, along with colleagues at Stanford, I devised a high-speed packet buffer architecture that emulates an SRAM based packet buffer using a combination of high-density low-speed DRAMs and a small amount of SRAM [HPSR’01, ToN]. This work later resulted in a company (Nemo Systems Inc.) which was acquired by Cisco in 2005.

A common theme in my research projects is to design and build software and/or hardware solutions for specific problems involving a combination of algorithms, architectures and protocols. I intend to continue actively pursuing such problems in future.

Future Research Agenda

While network reliability and management have been long recognized as important problems, traditionally, both academia as well as industry have not focused too much in this direction. As such, the field is still in its infancy with many open problems that perhaps require re-thinking established architectural principles. I believe my experience with real operational networks along with router architectures provides me with a unique ability to shape this upcoming area of research. I present some example problems and directions in this area that I intend to work on.

One important problem today is cascaded failures. Subtle configuration changes in one router can trigger large-scale cascaded events that eventually may manifest somewhere else. It is difficult today to “trace-back” the root-cause of such manifestations in any systematic way, forcing operators to rely on ad hoc mechanisms that are both slow as well as tedious. One possible solution is to create “debuggable” routers that are equipped with incremental data structures to store configuration changes, which can then be correlated with failure data. Such problems require operational as well as router experience, where I believe my skills are most applicable.

I also plan to leverage my experience in hardware primitives to provide scalable support for many other measurement and monitoring tasks. For example, it is a challenge today to provide native router support to measure and characterize the convergence time after a routing change. A second example involves characterizing new and upcoming applications such as video streaming and VoIP that require application-level monitoring. Packet level monitoring does not suffice because application level performance (e.g., video quality) cannot be inferred from packet level metrics (such as packet loss) alone. Application level network monitoring, which may require parsing packets, is hard to achieve at high speeds in a scalable fashion.

While designing router primitives for specific problems is itself challenging because of issues of speed and scale, an even bigger challenge is to design generic router primitives that are both flexible, programmable, and composable. In summary, I believe that this research agenda offers a rich variety of high-impact research problems, and that my research and industrial experience provides a unique perspective that can hopefully stimulate fresh approaches to these problems.