

# Research Statement

Cristian Lumezanu

I am interested in designing, building, and measuring networked systems that tolerate abuse, misbehavior, and failures.

As the Internet grows more complex and diverse, networked systems and protocols are increasingly brittle: they experience performance degradation or fail altogether under adverse conditions such as security attacks or network outages. My research has resulted in systems and protocols that perform well and tolerate problems in all aspects of a networked system: explicit misbehavior, such as selfishness or malice, by *the users* of the system [8], failures in *the network* substrate that relays messages between users [4, 9, 7], and bugs and inefficiencies in *the protocol* that defines the rules of communication between users [10, 11].

Designing systems for efficiency and robustness is difficult because one has to account for the complex interactions between the potentially untrusted components of the Internet. My approach combines two complementary and closely related methods: network measurement and system building. Measurement reveals the complexities of the Internet, helps predict the consequences of failures, and ultimately drives the design of efficient and secure systems. Building and deploying the system demonstrates the feasibility and robustness of its design. As an example, in my work on routing overlay networks, I started with a large scale measurement study that revealed opportunities to improve latency between users in the Internet, while protecting them from freeloaders and misbehavers. This led to the design of the PeerWise system, the first routing overlay network that is both efficient and resilient to explicit user misbehavior.

Below, I describe three of my strongest research contributions: PeerWise, an incentive-based routing overlay network for latency reduction that is robust to user misbehavior; Informed marking, a mechanism to save bandwidth in the presence of transmission failures in cellular wireless networks; and Pistachio, a static-analysis tool to detect bugs in network protocol implementations.

## Providing efficiency and robustness to misbehavior in routing overlay networks

Internet users have competing goals. Some behave selfishly, even maliciously, in pursuit of their own interests, thereby affecting the performance of honest users. In my dissertation, I explored how to build systems that are both efficient and robust to explicit user misbehavior in the context of cooperative overlay routing. Routing overlay networks improve the performance of packet delivery by delegating the task of selecting paths to users, rather than ISPs, who forward traffic for each other along virtual topologies. Because forwarding is expensive, users can free-ride by joining and benefiting from the system without providing forwarding service for others in return. I designed and built PeerWise [8], a routing overlay network that improves communication latency between end-users while encouraging cooperation and discouraging misbehavior.

To provide performance, I use triangle inequality violations (TIVs) in the Internet to predict shorter-than-default alternate paths (or detours). To better understand TIVs and their potential, I collected and analyzed latency measurements between thousands of Internet users. I showed that TIVs are prevalent, lasting features of the Internet and offer significant latency reduction [2, 3]. However, a system that measures all possible latencies to discover TIVs would not be scalable and would perform poorly. My insight was to use network coordinate systems to scalably detect TIVs [6, 5]. A network coordinate system associates nodes in the Internet with points in a metric space such that the distance between points estimates the real latency between nodes. Since TIVs are not allowed in metric spaces by definition, estimated distances between nodes in a TIV should have high errors. My measurements confirmed that, indeed, coordinate errors predict when a pair of nodes is part of a TIV.

To provide robustness to misbehavior, I use mutual advantage as a fundamental design principle: overlay edges

exist only between nodes that can help each other find detours [5]. Such an incentive mechanism discourages freeloaders and malicious users to participate, because they would have to provide at least as much service as they receive. At the same time, mutual advantage encourages honest users to join, because it ensures that they can benefit from the system [1]. However, before introducing mutual advantage into the design of routing overlays, we must ensure that it would not severely limit the number and performance of detours that a node can find. Using measurements collected from a globally-distributed set of users, I showed that, even with mutual advantage, the number of destinations reachable and the quality of detours do not drop significantly.

Translating the design of PeerWise into a real, working system required a number of mechanisms, devised after analyzing many measurement results and taking several trips back and forth to the drawing board. One challenge was how to use PeerWise to scalably discover detours to non-participating nodes, such as web servers, that do not maintain network coordinates. With colleagues, I implemented a virtual coordinate system through which a PeerWise node can become responsible and compute coordinates for any host in the Internet. We also proposed and evaluated several policies that nodes can use to predict quickly the best mutually-advantageous detours to any destination. I deployed PeerWise on PlanetLab and showed that nodes quickly find detours to popular destinations, that these detours are stable and that they offer significant latency reductions. I then confirmed that user-level applications such as web transfers can benefit from the network-level detours of PeerWise.

## Improving bandwidth under high packet loss in cellular wireless networks

The network substrate that transfers messages between users in the Internet may also lead to performance degradation. Consider cellular wireless networks, an increasingly ubiquitous medium for Internet access. The growing subscriber base and rich applications strain the capacity of both the cellular wireless links and the wired backhaul network that connects cell towers to the Internet. There is a growing mismatch between the goals and requirements of users, who send more and more traffic, and the characteristics of cellular networks, which are bandwidth constrained and are likely to remain so for several years. This mismatch leads to network failures: significant congestion and transmission errors occur in the wireless network which degrade user performance and experience.

I developed a system that uses network-level redundancy elimination (RE) to reduce traffic volume on the bandwidth-constrained cellular wireless links. RE avoids the retransmission of repeated sequences of bytes on a network path by deploying a cache at each end of the path. For each packet traversing the path, the ingress node finds common sequences of bytes within packets that were previously sent (and stored in its cache) and replaces them with fixed-size pointers to the cache. At the egress, packets are decoded by replacing encoded content with data from the cache. I started by studying the feasibility of RE in cellular networks using packet traces collected in two North American and one European wireless service providers. My experiments showed that cellular traffic exhibits high-redundancy: an RE-based system would save 8-65% of the available bandwidth.

Yet, deploying RE in cellular networks is not straightforward: the high loss rates due to congestion and transmission errors can cause the two caches to become unsynchronized and hinder the decoding of repeated content. Even worse, packet loss can compound when used with RE: missing packets prevent receivers from being able to decode later packets, which are then dropped. Using both simulations on the three data traces and live experiments in a North American 3G network, I demonstrated that the effect of loss on RE is significant: as the network packet loss rate increases, the bandwidth savings from RE decrease; simultaneously the rate of packets that cannot be decoded, and thus are dropped, also increases.

To detect and recover from packet loss, I augmented RE with informed marking, a mechanism where each receiver signals the sender whenever it cannot decode a packet due to a missing packet from its cache. The receiver sends a control packet with the hash of the missing packet and the sender blacklists the corresponding packet in its own cache; in future encodings, the sender ignores repeated content matched to any blacklisted packet. Informed marking reduces the number of packets that cannot be decoded because they depend on a lost packet. Also, unlike other potential loss recovery techniques such as retransmission, it is flexible because it does not introduce feedback overhead when there is no loss or when lost packets are not used in the encoding. With informed marking, mobile users are able to preserve more than 60% of the bandwidth savings from RE even when packet loss rates are high.

## Detecting security flaws in network protocol implementations

Communication protocols are increasingly designed to provide security against attacks and robustness against network glitches. There has been a significant body of research in scrutinizing abstract protocols and proving that they meet certain reliability and safety requirements. These abstract protocols, however, are ultimately implemented in software, and an incorrect implementation could lead to vulnerabilities even in the most-heavily studied and understood protocol.

With collaborators, I designed Pistachio [10], a static analysis tool that checks that the implementation of communication protocols matches their specification. Pistachio starts from a detailed protocol specification and is able to check properties that generic tools such as buffer overflow detectors do not look for. It is also very fast, enabling it to be deployed regularly during the development cycle. While not an online tool, Pistachio can discover serious security holes, thereby making protocols more secure and efficient for their users.

The input to Pistachio is the C source code implementing the protocol and a rule-based description of its behavior, derived from the specification. Each rule describes what should happen in each round of communication. Pistachio performs symbolic execution, based on abstract interpretation, to simulate the execution of the protocol source code. Using a fully automatic theorem prover, it checks that each rule is satisfied on all valid program paths. We applied Pistachio to three protocols, LSH, OpenSSH, and RCP and discovered a wide variety of known bugs (missing only 5%), including security vulnerabilities, as well as two new, undiscovered bugs.

## Future work

In the future, I will continue to explore new perspectives into measuring and building networked systems and protocols that tolerate misbehavior, abuse, and failures. In particular, I intend to perform research along the following three directions. First, I will study how online social networks can help detect the spread of anomalous network events such as malware or denial-of-service attacks. Second, I intend to explore robustness to performance degradation in the emerging area of cloud computing by proposing protocols that give cloud clients and services more control over the traffic between them. Finally, I will explore network-level mechanisms to avoid censorship and abuse in the Internet. Below, I present more details about each specific project.

**Using social media to detect anomalous network events.** The delivery infrastructure of anomalous network events such as spam, malware, or denial-of-service attacks has evolved dramatically in the past years. While reducing software vulnerabilities can help, we must explore new technologies and approaches to minimize the damage caused by such events. Social network applications, such as Twitter and Facebook, combine the view of millions of users from all corners of the Internet. Because they exceed the coverage of even the best monitoring systems, they can become an effective tool in our fight against security threats. I intend to explore how social networks can help detect and eradicate anomalous network events. I have already begun studying how the publishing and social behavior helps separate spammers from legitimate users in Twitter.

**Application-specific traffic control for cloud providers.** In recent years, many popular applications and services, such as email, photo sharing, or multiplayer games have migrated to the “cloud”, where they run from many geographically distributed data centers. Although these applications have various performance requirements (e.g., online games benefit from low-latency paths while video transfers prefer high-bandwidth paths), cloud operators do not have a way to account for these requirements when controlling traffic to/from their clients. I intend to study methods that enable cloud providers to offer a more robust, application-specific traffic control, that improves the performance of their tenant services. In particular, I will focus on building a performance inference service that harnesses the dual view of an application (from the cloud provider’s data centers) and of its users (from a set of distributed vantage points in the Internet) to build performance profiles for multiple paths to each data center. Further, I will develop an optimization-based architecture to enable providers to choose the best path with respect to the performance requirements of each service and client.

**Route avoidance as an anti-censorship mechanism.** ISPs, enterprises, even governments have been increasingly interfering with end-to-end user traffic that they help carry. For example, ISPs such as Comcast and Bell Canada throttle peer-to-peer traffic and countries like China and Australia enforce strict Internet censorship laws. Such misbehavior emphasizes the usefulness of a route avoidance mechanism, which would allow users to avoid certain parts of the Internet. I plan to study the effects of introducing route avoidance as a user policy in the Internet and build a routing overlay system that implements it. Users will be able to avoid regions of the Internet by smartly forwarding their traffic through other nodes in the overlay.

## References

- [1] D. Levin, R. Baden, C. Lumezanu, N. Spring, and B. Bhattacharjee. Motivating participation in Internet routing overlays. In *ACM Sigcomm Workshop on the Economics of Networks, Systems, and Computation (NetEcon)*, 2008.
- [2] C. Lumezanu, R. Baden, N. Spring, and B. Bhattacharjee. Triangle inequality and routing policy violations in the internet. In *Passive and Active Measurement Conference (PAM)*, 2009.
- [3] C. Lumezanu, R. Baden, N. Spring, and B. Bhattacharjee. Triangle inequality variations in the Internet. In *ACM Sigcomm Internet Measurement Conference (IMC)*, 2009.
- [4] C. Lumezanu, K. Guo, N. Spring, and B. Bhattacharjee. The effect of packet loss on redundancy elimination in cellular wireless networks. In *ACM Sigcomm Internet Measurement Conference (IMC)*, 2010.
- [5] C. Lumezanu, D. Levin, and N. Spring. PeerWise discovery and negotiation of faster paths. In *ACM Sigcomm Workshop on Hot Topics in Networking (HotNets)*, 2007.
- [6] C. Lumezanu and N. Spring. Measurement manipulation and space selection in network coordinates. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2008.
- [7] C. Lumezanu, N. Spring, and B. Bhattacharjee. Decentralized message ordering for publish/subscribe systems. In *ACM/IFIP/Usenix International Middleware Conference*, 2006.
- [8] C. Lumezanu, R. Baden, D. Levin, B. Bhattacharjee, and N. Spring. Symbiotic relationships in Internet routing overlays. In *Usenix Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.
- [9] S. Sundaresan, C. Lumezanu, N. Feamster, and P. François. Autonomous traffic engineering with self configuring topologies. In *ACM Sigcomm Poster Session*, 2010.
- [10] O. Udrea, C. Lumezanu, and J. S. Foster. Rule-based static analysis of network protocol implementations. In *Usenix Security Symposium*, 2006.
- [11] V. Valancius, C. Lumezanu, N. Feamster, R. Johari, and V. Vazirani. Microcontracts for Internet connectivity. Tech. rep., Georgia Tech GT-CS-10-17, 2010.