# The Network is (Unfortunately) Not Yet the Computer

R. Guérin (guerin@wustl.edu)

Ever since Sun Microsystems™ coined the sentence *”the network is the computer”*, the tight dependencies that exist between networking and computing have been apparent. The emergence of cloud computing has, if anything, made those dependencies stronger and more complex. However, in spite of this awareness, computing and networking are today not as tightly integrated as one may wish. This in turn creates many challenges if one is to realize the vision of a smart connected city, where computation and communication are seamlessly inter-woven.

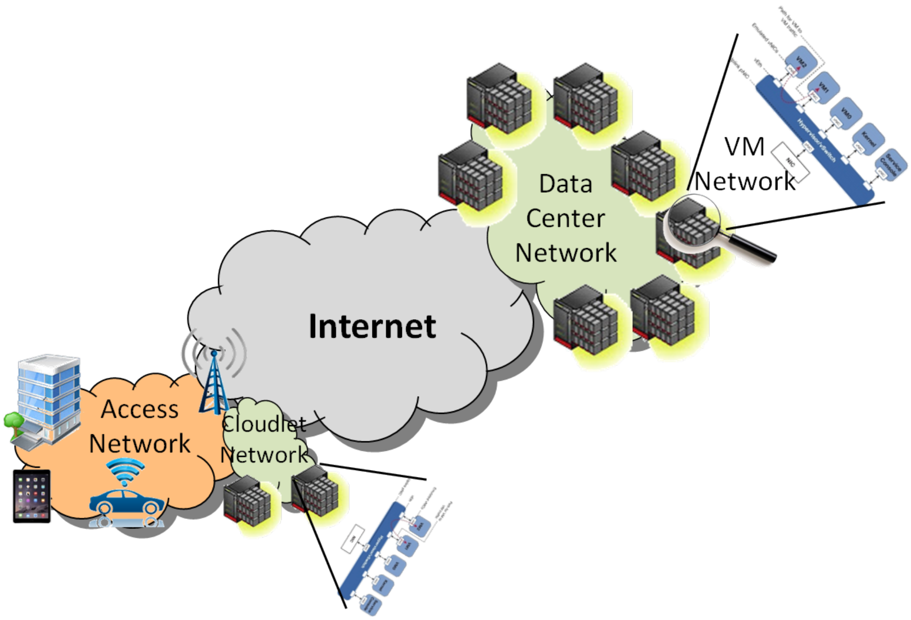

Figure 1: Smart city network components.

As Fig. 1 illustrates, one of the main difficulties in achieving a unified computing and communication environment is the growing diversity of what constitutes the ”network,” its different pieces, and their role in achieving computations. In particular, a typical network will include an access network, often using wireless technologies. This access network will possess some (limited) computing capabilities of its own, *e.g.,* in the form of cloudlets, as the need to better distribute computing and data loads has increasingly led to moving computing to the edges. Access networks are in turn connected to the Internet, either directly or through some metropolitan area network (not shown), which provides connectivity to massive (cloud) computing facilities that offer flexible and cost-effective computing solutions to a wide range of users. Those facilities have their own internal networks with characteristics that differ significantly from those of both the Internet and access networks. Finally, the reach of networks also extends in non-trivial ways into servers, where virtual machines coexist on the same hardware and share network resources (network interface cards). Ensuring consistency, let alone collaboration, between these many different computing and networking components is no easy task and certainly not a reality today.

Specifically, consider a scenario focused on the communication and computing needs of vehicles across a city. Target applications could, among many possible choices, include traffic control, vehicle maintenance, and applications running on the mobile devices of the vehicle's occupants. The computing needs of each one of those applications vary greatly in terms of volume and latency requirements.

Vehicle tracking will usually involve a large number of small messages with relatively tight latency constraints, and computing partially delegated to edge cloudlets, *e.g.,* for real-time traffic control, with some less delay-sensitive background tasks relegated to backend facilities. Handling this traffic introduces significant real-time communication and computing requirements between vehicles and edge cloudlets. In addition, as vehicles migrate from an access network to another, real-time VM migration and/or transfer of state information is typically required, though the resulting data volume are usually low.

The communication and computing needs of vehicle maintenance applications are in turn quite different, primarily involving infrequent batch uploads to a backend data center.

Finally, applications running on the mobile devices of vehicles' occupants can themselves span a wide range of configurations with a variety of latency requirements and data volumes. For example, video streaming will typically involve high volume communication from edge servers, but with relatively loose latency constraints (because of buffering) except for the occasional control traffic. In contrast, interactive games usually have tight timing constraints together with relatively large volumes of data. In addition, both video streaming and gaming have relatively large data migration needs as vehicles move from one access network to another.

In general, scenarios such those outlined above require close coordination in the allocation of computing and networking resources across a diverse set of networks and computing devices. There are a number of challenges that arise in realizing such coordination, but two important components in making progress towards such a goal include:

1. A lightweight low-latency messaging system that can help coordinate state updates across devices and compute facilities, so as to facilitate the appearance of a seamless environment across an entire city infrastructure;
2. Mechanisms for fully extending the network into the virtual environment where most computing resources nowadays reside. In other words, ensuring that the last ”network leg” is accounted for when provisioning systems for different workloads.

Developing systems capable of meeting both of those needs represents important steps towards buidling a smart, connected city. We are currently involved in two projects that are investigating possible solutions. The first is in collaboration with an equipment vendor, and is focused on the development of a lightweight, real-time messaging middleware system that can support the exchange and coordination of state information across a city-wide wireless network and its edge and backend computational facilities. The second project is an NSF-funded project that is investigating the extension of service differentiation capabilities into the neworking stack that supports the communication needs of virtual machines (VMs) in the Xen environment. The goal is to ensure that different throughput and latency requirements can be efficiently met even when multiple VMs share common physical networking resources. Both projects rely on small local testbeds to emulate communcation between mobile/edge devices and the computing facilities (VMs) they need to access. Those testbeds are adequate for proof-of-concept and certain types of performance stress tests, but do not allow full ”at-scale” testing. Testbeds that can allow the emulation of at-scale environments would, therefore, be very beneficial.