Ubiquitous Sentient Network: Measurement-Based Network Operating System

Nageswara S. V. Rao
Center for Engineering Science Advanced Research
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6355
email; raons@ornl.gov

Abstract

The rapid advances and proliferation of network technologies hold a promise for bringing extensive networking capability and functionality to the common user. This promise is only partially fulfilled at present due to the lack of support from the underlying network as well as the sheer lack of tools that optimally exploit the underlying network technologies and devices. Akin to the operating system of a conventional computer, we propose a network operating system that provides the end users a ubiquitous network capable of providing optimized performance. There are two components of such operating system: (a) network connection, modeling and optimization sub-kernel that performs all the networking operations, and (b) device sub-kernel that enables plug-and-play of networking devices. Such network capability requires significant advances in the basic functionality of the network nodes, both internal ones such as switches and routers as well as the end nodes such as workstations, laptops and palmtops. We propose a measurement-based paradigm for the operating system where the network nodes are endowed with measurement and control infrastructures that enable accurate state estimation and optimization needed to perform the tasks. This capability requires overcoming significant technical and non-technical challenges, but can provide ubiquitous networking capability to the user in unprecedented ways.

1. Introduction

There has been an explosive growth in the network devices and services currently available or under development, and their impact is expected to be on several fronts. To a common user devices such as Personal Digital Assistants (PDAs) and cellular phones are becoming more "connectible" to networks, thereby holding a potential for anywhere anytime connectivity without wires. Several services such as Multi Protocol Label Switching (MPLS), Differentiated Services (DiffServ) and Integrated Services (IntServ) [2,3], hold a promise to provide Quality of Service (QoS) unprecedented in the current operational networks. The network components and technologies are expected to provide faster routers and switches as well as much higher bandwidth links. This myriad of developments holds a promise for bringing an extensive networking functionality to the common user.

The image of a user connecting to the Internet from a wireless end device without any restrictions, controlling a remote process, performing on-demand data transfers, and instantly accessing web services, looks promising now more than ever. The capabilities of Ubiquitous Computing (UC) and Electronic Persistence Presence (EPP) are now considered to be within the realms of practical realization [1]. The present reality, however, is a different story. This promise is only partially fulfilled at present due to the variability in network devices, services, infrastructure, and operating systems, and the sheer lack of tools that optimally exploit the underlying network capabilities; the onus of achieving the required network performance currently lies on the user.

1. On-demand data transfers, especially large data volumes, are very poorly supported in the current wide-area network, especially the Internet. Only users of very few dedicated networks might have an access to dedicated networks for these applications. Due to the best effort policies currently used, it is quite possible for a user to experience a very small throughput because the routes for the message are congested, while the network elsewhere stays devoid of traffic.
2. Controlling a remote process in real-time over the Internet is possible only in fairly limited cases, such as processes with very slow dynamics. Indeed, the unstructured delays of datagrams over the Internet could very easily destabilize the control loop of a remote process with fast dynamics.
3. Web traffic from a user to servers can be quite unpredictable, especially for the variety of services that are expected to the available in future.
4. The connectivity of an end device such as a palm top or web-enabled cell phone is possible only through commercial infrastructures. The entire functionality of the Internet is not available to the users of such devices.

The present networking technologies, services, tools and infrastructure are not adequate to address the above problems and many others. Indeed, the present evolutionary plans coupled with short-term agendas adopted by industry and other agencies will only result in less than adequate solutions to these problems.

To address these issues, the network must be *sentient* in that it must be aware of its present state and must automatically provide optimized performance to the user. For example, the network must be able to utilize the unused parts of the network to provide multiple paths for high throughput data transfers. There are two basic properties needed for the network to be sentient: *observability* and *controllability*. These are classical notions from control theory but their implementations in wide-area networks pose challenges beyond the current control theory and networking. Since network's state is not a priori fixed, for example due to mobile nodes and other traffic transients, the state can only be inferred by observing the network. Observability enables us to accurately estimate the state of network to provide best service to the user, and controllability enables us to implement best strategies over the operational networks. Both these properties are very limited in the current networks, thereby limiting the performance that can be offered to the end users. Consequently, automated tools that exploit these properties (if they were available) are extremely limited.

We propose a measurement-based paradigm for the next generation networks. These networks support measurement infrastructure that provides observability, and instrumentation that provides the required controllability. These capabilities can be realized as (significant) enhancements to the existing network technologies and can be gradually phased in (as opposed to revamping the whole infrastructure). But the resultant performance improvements could be well beyond the current levels. The proposed system, however, faces significant challenges in terms of network state estimation and optimization, and device integration technologies. Equally important are the non-technical challenges involved in motivating the device vendors and service providers to produce compliant and/or conducive devices and services to support the proposed system. The promised capabilities of the proposed system could, at least in part, help overcome these challenges.

We propose a network operating system that supports basic measurement and control infrastructures in Section 2; we also identify various functionalities needed for such system both from user and network points of view. We compare this system to identify the advances needed over the existing technologies in Section 3. The technical and non-technical challenges needed for the design and implementation of the proposed system are discussed in Section 4. We make a case that the measurement-based paradigm is likely to provide the required functionality in Section 5. Various technical components needed for an implementation of the proposed system are described in Section 6. Some scenarios that will benefit from the proposed system are discussed in Section 7.

2. Operating System for Networks

Akin to the conventional operating system of a computer, we propose a Network Operating System (NOS) that does all the "underlying tasks" and provides an easy high-level interface to the underlying network capability. Various functionalities such as QoS will be simply provided to the user as opposed to the current state in which user is challenged to figure out ways to ensure the performance. For example, user just plugs-in a new high bandwidth network card, then it is automatically integrated into the network such that its performance is optimized for the user without further intervention. For example, when a message is to be sent to a destination, the best path or possibly a multiple-path will be computed and the message will be sent in an optimal manner. The propose operating system quite simply relieves the user from being responsible for all networking tasks needed to get best out of the network.

There are two different types of devices that run the proposed operating system. The *end nodes* provide the network access to the end users; these nodes include workstations, personal desktop and laptops, palm pilots, and PDAs which could connect to the Internet via various devices including network cards, cable modems, wireless cards, and cell phone connections. The end nodes utilize the network to receive as well as to provide services. The *internal nodes* correspond to the network components such as gateways, routers, switches, hubs, network servers, and firewalls. While the distinction between end and internal nodes is not sharp in certain cases such as domain name servers that also act as hosts, it is instructive in understanding the role of network users and providers. The proposed operating system is different for these two types of nodes in terms how tasks are generated and executed.

The overall idea is that NOS runs as a collection of *daemons* on top of the native operating systems of the nodes. The end nodes receive the communication tasks from the users, which will be appropriately dispatched onto the network in the best possible way. Furthermore, the end nodes also receive information requests from other nodes, which will also be handled by NOS. For the internal nodes, the tasks mostly arrive over the network, which include routing and data requests. In a nutshell, NOS acts a buffer between the end user and the network doing all the networking tasks. For the internal nodes NOS provides a layer that will support the required measurement and control functionality needed from the network. In implementations, NOS is auto-configured and runs as a minimal set of daemon processes.

The desired features of such operating system include the following:

1. **Automatic Integration**: The system must provide for automatic assimilation of network devices whether they are end nodes, internal nodes or various devices of the nodes. This is particularly important for the end nodes since they are likely to be used by non-experts of networking technologies. For example, the entire network

capability must be available to the user as soon as he/she gets connected to the network via a wireless link. The devices that enable the connection to the networks must be simply plug-and-play no matter what the host's native operating system is. The NOS running on top of the native operating system will perform all the necessary tasks to integrate various devices into the nodes as well as integrating the nodes into the network.

2. **Measurement Infrastructure Support**: The network nodes must support measurement functionality so that nodes needing the data must be able to query the other nodes. When a NOS node is connected to the network, it will communicate with other nodes to get a good state estimate of the network. The end nodes receive tasks from users, which they will dispatch onto the network based on the state of the network. The measurements collected at the nodes must be sufficient enough to enable state estimation of the required parts of the network. This is a significant departure from the current state of affairs, especially for the internal nodes such as routers. In the current Internet, ICMP (Internet Control Message Protocol) mechanisms might be viewed as providing primitive measurements; tools such as ping and traceroute provide measurements about the connectivity and availability. Here we propose a whole host of measurements including delays, connectivity, jitter, effective bandwidth, and flow rates needed for state estimation.
3. **Network State Estimation and Optimal Paths**: For the tasks at hand, NOS computes the best strategy based on the current network state in a manner transparent to the end user. This part involves examining the dynamic state of the network and the task requirements. For example, for a real-time control application, it is important to open a connection that provides the adequate stability - bandwidth may not be as important as the controlled jitter in this case. On the other hand, for remote execution of simulations, the end-to-end delay might be the primary criterion. For video conferencing both bandwidth and jitter are important, and perhaps multiple paths might be needed to pull together enough bandwidth. NOS will have the optimizing daemons that perform the state estimation and path optimization.
4. **Path Control and Traffic Engineering**: When a user task is to be performed, the operating system uses the state to identify the best option for the user task, and launches it on the network. The computed optimal paths must be realized over the network and the transmissions must be appropriately carried out. For example, if multiple paths are needed to accumulate enough bandwidth for a data transmission, the routers must allow for such multiple path implementation. Also if certain timing constraints are to be honored for transmissions, for example the closed loop control of remote instruments, the nodes must provide the required support to realize the paths and traffic profiles.
5. **Graceful Auto-Configuration**: NOS daemons of a node are started in a very minimal configuration at a node, especially at the end nodes, and the rest of the system is bootstrapped on-demand. The required daemons will be loaded as needed from the host or downloaded from the network once the connection is made. There must be NOS repositories available on the network to provide the services for bootstrapping the nodes when needed.

In the above list, Items 2-4 constitute the observability and controllability capabilities described in the introduction. The Items 1 and 5 are more related to operational efficiency and easy of use, but are equally important.

The impact of designing the NOS based on measurement and control infrastructures could be quite far reaching. This method does not call for a totally new network design but only for enhancements of the existing systems - the current routers and end nodes can be used as they are with some NOS daemons added to them. In a mixed network NOS nodes will do the best possible by utilizing the others to a maximum extent. During the transition, NOS nodes will make use of whatever support available from the other NOS nodes. The situation would be different for methods that require the entire network to be upgraded. NOS nodes can coexist with the legacy nodes during the deployment stage or later, and other non-NOS sub-networks can also added at any time, but the overall performance depends on how much of the network consists of NOS-enabled nodes. The development of individual network technologies is not constrained in any way by the proposed system – they can continue their own paths.

3. Relation to Existing Networking Technologies

The proposed system is a next step, albeit more of a quantum leap, in several component technologies. The exchange of measurements between routers is an extension of the Boarder Gateway Protocol (BGP) currently used between the routers. The proposed extension, however, is significant in several ways. First, the data itself is more detailed than simple connectivity information used in BGP in that it enables the estimation of a number of dynamics variables such as available bandwidth, delay statistics, and traffic flow variables. Second, the measurements at internal nodes can be queried by the end nodes, which is a departure from the current norm wherein the Internet Service Providers (ISPs) share very little measurements with the end users. While ping and traceroute provide limited measurements from the end-nodes, more detailed information is needed in the proposed system.

In terms of control instrumentation, our proposal calls for certain levels of source-based control in sending the network data. At present, IPV4 allows for the specification of paths on datagrams, but unfortunately such paths are mostly ignored by the routers; sometimes such packets are simply dropped by firewalls. Services such as

DiffServ, and MLPS allow for limited source-based control; for example one can specify a priority service for data. The NOS proposal calls for more flexibility in sending the data. In particular, strict honoring of paths specified by the end nodes is critical for implementing multiple paths and also for rerouting the traffic away from congested areas.

While active networks in theory can provide both observability and controllability needed for NOS, they do not provide a practical solution: it is unrealistic to expect the entire Internet to switch over to an active network. The Internet is likely to be varied with vestiges of old components interoperating with the newer ones – the proposed NOS provides a means to gradually migrate to NOS-based systems.

4. Challenges of NOS Design and Implementation

The resemblance of NOS to a conventional operating system stops at a high functional level. The latter runs on a hardware with entirely known functionality and dynamics, whereas NOS runs on highly distributed hardware with only partially known dynamics. The design and implementation of NOS is replete with technical as well as other challenges such as industrial and government cooperation and support. Obviously, a complete implementation of NOS requires an unprecedented collaboration between various commercial vendors that produce various end and internal nodes, service providers that operate the networks, and perhaps federal agencies that regulate their own networks. The required capabilities of NOS are beyond the short term interests of commercial vendors of internal nodes and service providers, since there seems to be no immediate benefit to them. For this reason a radically new approach is needed.

An operating system for a wide-area network is significantly more complicated than that for a processor or set of processors, for it involves additional tasks unprecedented in the conventional operating systems. First, a wide-area network is highly distributed with widely varying components such as network interfaces, hubs, routers, and switches. The NOS must have a clear idea about the state of the network to obtain optimal performance for the task at hand. But the only information available to NOS at a host is the measurement and state estimators provided by other nodes. The challenge is to identify the needed measurements and skeletal form of state estimation needed for various network tasks. Then the next step is to instrument the nodes to provide the required information. There are limitations as to how much data and processing a host NOS can handle without significantly cutting into the host's other tasks.

Compared to processors, most of the network is unknown, which complicates the design and implementation of NOS at the end nodes. The optimizations needed for path computations and traffic engineering policies are significantly challenging, especially if the entire network state is not completely available or too difficult to estimate. But, based on simple measurements alone significant end-to-end QoS problems could be solved (see next section), which gives us a hope that the proposed system could be possible. Nevertheless, the implementation of various optimization modules will be highly challenging. Dynamics of the network play very important role in the end performance seen by the user. But, most network dynamics are unknown and very hard to predict over small time scales. NOS daemons must have reasonable estimates for the dynamics so that the traffic can be suitably engineered to account for partially known dynamics. This is a significant technical challenge.

The implementation of NOS daemons is very involved considering that they run on top of native operating system, which could be widely varying such as linux, Windows, vxWorks, Palm OS, and Windows CE. Our proposal is to implement NOS at a level higher than TCP/IP stack (or similar communication stack), with suitable hooks to the native operating system kernel.

In addition to the above technical challenges, there are a number of other challenges. The internal nodes are operated by ISPs, who do not disclose the details of their routing policies and are not likely to be forthcoming in providing the required measurements. The commercial vendors that produce internal nodes such as routers, will resist the efforts to incorporate NOS measurement and control modules. Even at the end nodes, it not easy to provide the required hooks needed by the NOS modules into the native operating system kernels, since most of the operating systems are proprietary. Comprehensive efforts by the user community might be needed to effect the required changes. Perhaps the projected benefits and widespread user support of NOS will provide the impetus for the needed cooperation from the industry.

5. Network Measurements are the Key

It is well known that Internet traffic is highly complicated and very difficult to characterize [9]. In particular, the network traffic is self-similar and traditional modeling methods such as Poisson model have been shown to be inadequate [4]. Thus, it is reasonable to assume that an accurate state estimation that includes the estimation of traffic distributions will be impractical, thereby casting a doubt on QoS methods that utilize distribution models. Our contention is that several important end-to-end QoS problems can be solved based on measurements without estimating the underlying distributions. For example, end-to-end delays can be guaranteed to be very close to optimal (possible only under a complete knowledge of distributions) with a high probability [5]. Indeed, there is an intimate relationship between sample-based regression estimation and end-to-end delays [6]. This result is valid no

matter what the distributions are, and the required paths can be computed in polynomial-time based entirely on the delay measurements. We believe several other QoS problems of NOS can also be similarly solved but are technically more challenging. The result in [6] is valid for stationary distributions, and one needs to employ techniques such as martingale/mixing methods to account for the time dynamics. Nevertheless, measurements are the key to estimating the state and providing QoS. From a pragmatic viewpoint, it is easier to collect measurements than to estimate the distributions accurately. However, the required measurement infrastructure must be in place to collect measurements.

6. Components of NOS Kernel

There are two components of NOS kernel: (a) network connection, modeling and optimization sub-kernel that performs all operations needed to provide optimized performance to the user, and (b) device sub-kernel that enables plug-and-play of networking devices. Both are important from an operational viewpoint, and require two different types of functionalities and technologies.

(a) Network Modeling and Optimization Kernel: A number of daemons are needed to implement various parts of this sub-kernel. A bootstrapping daemon runs at the nodes at all times, which is the minimal configuration of NOS, especially at the end nodes. Once the node is connected to the network various other daemons will be invoked either by loading them from the host as in the case of workstations and personal computers or downloading them from the network as in the case of wireless devices such as palm tops or cell phones. While such capability exists in very limited cases now, our framework universally provides this functionality. Then measurement, path computation, and communication and dispatch daemons will be loaded as needed.

(b) Device Kernel for End Devices: The device kernel enables plug-and-play integration of various network hardware devices with minimal demands on the user. While this part is not a strictly networking issue, it is very important for the operational effectiveness of the networks. A variety of devices may be used to connect the nodes to the network. For example, ethernet, FDDI, Token ring and ATM cards are the devices used to connect workstations and desktops to the networks. Note that at present attaching such devices to workstations is complicated depending on the device and the host operating system. As a minimum the drivers of these devices must be loaded at present, but often the task is technically involved and time consuming. This is particularly true for newer devices to be installed on older versions of operating systems. Our idea is to provide self-installing drivers with each device so that it will be operational all by itself soon after the device is plugged in. The device driver will automatically connect to the kernel, which will setup the device to be operational.

7. Field Application Scenarios

We now describe application scenarios where the proposed ubiquitous sentient network will have a profound impact. While these scenarios primarily deal with the applications from the U.S. Department of Energy programs, similar applications will arise in several other domains.

Scenario 1: Simulations and visualization using high performance computers: A number of applications require the utilization of massively parallel computers to execute complicated codes in an interactive closed-loop. For example, a climate simulation program might be interactively executed by a number of experts distributed over the network. The simulation itself may be guided by domain experts, who dynamically set the parameters and archive important datasets. This application provides several challenges to the networking technologies. First, the simulation data must be presented to all experts as it is generated. The interactive data from the experts must be very promptly provided to the simulation. In these tasks, it is very important to minimize the end-to-end delay between the communicating processes. Typically visualization data requires high bandwidths, which might require multiple paths, whereas user commands can be transmitted via paths with small bandwidth but with less delay. Here it is important to note that the experts participating in this application typically do not have an extensive knowledge of networking. The proposed NOS daemons running at various locations will do the job with minimal demands on the users.

Scenario 2: Remote explorations by robot teams: In several environmental applications, a team of autonomous mobile robots might be dispatched to map an area by scanning with instruments such as radiation detectors. An example is autonomously building the radiation map of an area suspected of nuclear radiation. The robot team members must cooperate with each other and with remote operators. The operational area in general does not have wireless network infrastructure (namely, a wired backbone of access points), and hence the robots must form an adhoc wireless network, which is interfaced to a network that connects to remote operators. Furthermore, the wireless connectivity is likely to be highly dynamic as the robots move in and out of various rooms of a highly reinforced building. Also there may not be a direct connectivity between several pairs of robots at any given point in time, and hence the other robots must be used as relays. More precisely, the locations and movements of robots must be exploited to deliver the messages. This problem is currently partially solved using adhoc solutions [8], which require

networking experts to setup and operate the network. The proposed NOS system, wherein each robot runs a combination of daemons of end and internal nodes, can be operated by non-experts in networking. The NOS daemons will detect that the connectivity is dynamic and will account for it by adapting to the movements of robots.

Scenario 3: Remote instrument control: There are several facilities that provide unique experimental capabilities to be utilized by researchers all over the country or the world. With the advent of Internet, remote access to such facilities is becoming more and more possible. While instruments with slow dynamics can be easily controlled using the existing network technologies (for example web instruments), devices with fast dynamics pose challenges. The stability of control loops is very important, which not only depends on the end-to-end delays of various control messages but also on the higher order moments of various delay times. For example, unbounded jitter can render the control loops unstable by feeding the commands to the instruments faster than warranted. At present there are no easy solutions to these problems when the control loops are implemented in wide-area IP networks. Note here that simply minimizing the end-to-end delay of control messages can lead to significant jitter that can in turn lead to instabilities. In this case, the network plus the instrument must be treated as a composite instrument to be controlled by the end user. The challenge is that while the instrument is sufficiently known to devise a suitable controller, the network is not. The proposed NOS provides the framework to utilize the measurements to design adaptive controllers, which also utilize the control daemons at the instrument locations to smooth out the control signals before they are delivered to the instrument. While the actual controllers will be implemented on top of NOS modules, the infrastructure needed for implementing the controls is provided by NOS.

8. Conclusions

We proposed an operating system for the network that provides an optimized performance to the users in a transparent manner. The actual implementation of the proposed system requires successfully overcoming significant technical and non-technical challenges. Technical challenges include development of a number of network optimization and instrumentation technologies. Non-technical challenges include getting the support from component vendors and service providers to produce compliant products. Nevertheless, the proposed system holds a promise for significantly enhancing the network capabilities that will be available to the users with minimum demands on them.

We did not address the very important issue of network security in this paper. Since NOS requires network nodes to provide several kernel services over the network, there is a scope for misuse and attacks. The security issues must be paid attention to from the beginning in the design and implementation rather than as an after thought.

Acknowledgements and Disclaimer

This work is sponsored in part by Defense Advanced Research Projects Agency (DARPA) under MIPR No. K153, and by the Laboratory Director's Research and Development Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U. S. Department of Energy (DOE). The ideas expressed in this paper are solely the personal opinions of the author, and do not necessarily represent the official or unofficial positions of DARPA, ORNL or DOE. Due to the page limitations, several important works and references have not been included here, and no judgment of value of various works is implied here by their inclusion or omission.

References

1. R. J. Aiken, New frontiers for research networks in the 21st century, *The Internet Protocol Journal*, vol. 3, no. 4, 2000, pp. 26-39.
2. B. E. Carpenter and D. D. Kandlur, Diversifying Internet Delivery, *IEEE Spectrum* , Nov. 1999, pp. 57-61.
3. A.Dutta-Roy, The cost of quality in Internet-style networks, *IEEE Spectrum*, September 2000, pp. 57-62.
4. V. Paxson, End-to-End Routing Behavior in the Internet, *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 601-615, October 1997.
5. N. S. V. Rao and S. G. Batsell, "On routing algorithms with end-to-end delay guarantees," in the proceedings of *IC3N: Int. Conf. on Computer Communications and Networks*, pp. 162-167, 1998.
6. N.S.V. Rao, End-to-end delay guarantees in computer networks: Analysis and NetLet implementation, ORNL draft, 2001, http://www.cesar.ornl.gov/~nrao
7. N.S.V. Rao, S. Radhakrishnan, B. Y. Cheol, NetLets: End-to-End Performance Guarantees for the Internet, *Int. Conf. on Networking*, 2000.
8. N. S. V. Rao, S. G. Batsell, J. Jonnasaint, Connectivity-through-time protocol for dynamic wireless networks, ORNL Invention Disclosure, 2000.
9. W. Willinger and V. Paxson, Where mathematics meets the Internet, Notices of American Mathematical Society, September 1998, pp. 961-970.