# Performance Analysis of NEMO augmented with MPTCP

Pratibha Mitharwal, Christophe Lohr, Annie Gravey

# Performance Analysis of NEMO augmented with MPTCP

Pratibha Mitharwal
Institute Mines-Telecom;
UMR CNRS 6074 IRISA
Technopole Brest Iroise CS 83818;
29238 Brest Cedex 3 France
pratibha.mitharwal@telecom-bretagne.eu

Christophe Lohr
Institute Mines-Telecom;
UMR CNRS 6285 Lab-STICC
Technopole Brest Iroise CS 83818;
29238 Brest Cedex 3 France
christophe.lohr@telecom-bretagne.eu

Annie Gravey
Institute Mines-Telecom;
UMR CNRS 6074 IRISA
Technopole Brest Iroise CS 83818;
29238 Brest Cedex 3 France
annie.gravey@telecom-bretagne.eu

***Abstract*—The present paper focuses on the quantitative and qualitative performance analysis of Network Mobility (NEMO). It compares the original NEMO architecture with an architecture in which NEMO is enhanced with Multi-Path TCP (MPTCP). This quantitative and qualitative analysis is done by plotting theoretical and experimental results using a local testbed implementation. It is shown that the novel combination of NEMO and MPTCP performs significantly better, in terms of routing efficiency and of throughput, compared to a classical implementation of NEMO.**

## I. Introduction

With the evolution of technology, users are mobile and have multiple available interfaces in order to take advantage of connectivity "anywhere and everywhere". This connectivity increases requirements in terms of mobility and multi-homing support. Supporting mobility consists in providing smooth handover when the user changes its attachment to the network, and thus has to connect with different IP addresses. Supporting multi-homing consists in allowing the user to simultaneously make use of multiple interfaces within its equipment. The simultaneous use of all the available network interfaces can indeed improve throughput, allow load balancing and add resiliency to the system. Therefore, supporting both mobility and multi-homing can be beneficial in improving available throughput, in facilitating handover and in increasing resiliency.

There are several proposals to solve mobility and multi-homing. Mobility and multi-homing for networks is supported by NEMO [1], [2], Locator/Identifier Separation protocol (LISP) [3], Identifier Locator Networking protocol (ILNP) [4] etc. On the other hand, mobility and multi-homing for a single host is supported by Mobile IP [5], Multi-Path TCP (MPTCP) [6], Host Identity Protocol [7]. All of the existing solutions for mobility and multi-homing provide location management of the host/network with some pros and cons regarding deployment, infrastructure changes, handover delay, throughput, tunneling overhead, etc.

The approaches followed to support mobility and multi-homing can be classified based on the layer on which they are implemented. Network layer approaches to solve mobility and multi-homing easily provide location support but require traffic related information from the transport layer in order to provide a better mobility support. On the other hand, transport layer approaches provide a better multi-homing support due to the fact that they can easily access to round-trip time or congestion information but require a collaboration with the network layer in order to provide a low-cost location management instead of deploying a cumbersome rendez-vous mechanism. Therefore, it is quite appropriate to assess how network layer and transport layer approaches can be blended.

In a previous paper [8], we proposed a combination of network and transport layer approach by combining NEMO and MPTCP. NEMO provides location management for the Mobile Networks (MNs) and MPTCP enables Mobile Network Nodes (MNNs), i.e. hosts inside the MN, to participate in multi-homing related decision making. This novel combination of NEMO and MPTCP is expected to provide a better mobility support with improved multi-homing support with respect to throughput, cost and load balancing for MNs.

In the present paper, a quantitative analysis is performed to assess the effectiveness of the proposed approach over existing approaches. A testbed architecture has been set up to carry out a theoretical and experimental study in order to quantify the respective behaviours of NEMO and NEMO with MPTCP. The performance of the novel combination of NEMO and MPTCP is evaluated and compared with that of classical NEMO in different network scenarios. In order to evaluate the performance of the different approaches, we consider various performance indicators such as throughput/goodput and delay. Both throughput (Mbits/sec) and round-trip time (ms) are measured experimentally while transmission delay has been calculated theoretically.

Section II presents a background study of our proposal while Section III presents the testbed architecture and the implementations of NEMO and MPTCP. Section IV presents the network scenarios considered for measuring the throughput and round-trip time. In Section V theoretical and experimental measurement results are presented, followed by a conclusion in Section VI.

## II. Background Study

NEMO [1], [2] has been designed to provide mobility management for MNs by enabling a MN to roam anywhere and receive traffic during roaming. This is realised with the help of a stationary anchor point in the MN's home network i.e., the Home Agent (HA). The Mobile Router (MR) inside the MN informs the HA of its current network attachment point, i.e. Care-Of-Address (CoA), by sending binding updates to the HA whenever it attaches to a new foreign (visited) network. The HA intercepts all the incoming packets for the MN, encapsulates these packets, and forwards them towards the MN's CoA. Upon reception, the MR decapsulates these packets and routes them inside the MN. For outbound traffic, the MR encapsulates the packets with the HA's address; the HA then decapsulates the received packets and routes them inside the Internet. This tunneling between the HA and the MR makes the MNN and the Communicating Node (CN) unaware of the mobility of the MN.

MPTCP [6] enables any host to use multiple available data paths simultaneously for any given session. MPTCP is backward compatible with TCP and therefore does not require any modification in existing network infrastructures. An MPTCP compliant host initiates an MPTCP session as a TCP session with a SYN flag carrying the additional option MP_CAPABLE. If the communicating host is also MPTCP compliant, it responds with the MP_CAPABLE option in the SYN-ACK. Once the TCP connection is established, the hosts can exchange all available addresses and initiate other MPTCP sub-flows corresponding to these addresses.

The combination of NEMO and MPTCP, as proposed in [8], requires no major modifications in operating NEMO or MPTCP. Two minor changes are proposed for NEMO, in order to make the MNNs aware of mobility. The first change is that the MR needs to advertise the current network prefixes or care-of-prefixes to the MNNs. After receiving the care-of-prefixes MNNs can configure new IP addresses to their interfaces with the help of IP stateless autoconfiguration [9]. The second change is that the MR should be able to route the packets with the IP address with care-of-prefix towards Internet and tunnel the packets with the IP address containing the home network prefix towards the HA.

Since NEMO is used for location management, the inbound traffic has to pass through the HA. The CN is only aware of the MNN home address when sending a connection establishment request. The HA receives this packet and tunnels it towards the MR at its current network attachment point. Upon reception, the MNN generates a SYN-ACK with the MP_CAPABLE option. If the CN is also MPTCP compliant, both nodes can establish an MPTCP connection. Once the connection is established for a given session, other subflows can be added to this session using available interfaces. The subflow with the MNN home IP address can be put as a "backup" path with the help of the MP_PRIO option. The backup path is used only when none of the other network interfaces are available. The MR is capable of forwarding the packets with the IP addresses containing care-of-prefix towards Internet. Therefore, after the MPTCP connection establishment, the tunneled path is no more used. However, NEMOs tunnels can still be used in case of some non-friendly networks features such as NAT, firewall, etc.

When, during mobility, the MR looses a connection and attaches to a new network interface, it advertises the newly acquired network interface prefix towards the MNNs. Using this new prefix, the MNNs can configure an IP address. This acquired IP address can then be communicated to the CN using the ADD_ADDR option and the unavailable IP address can be removed using REMOVE_ADDR option.

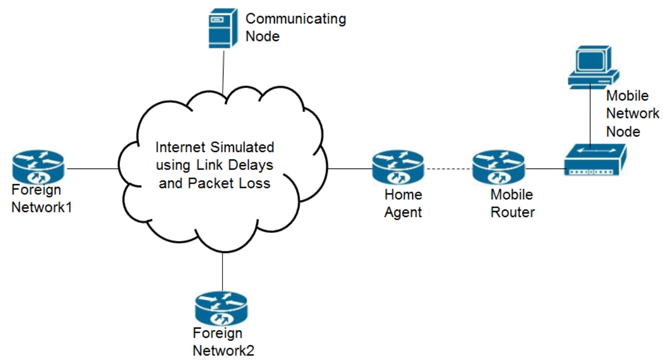


Fig. 1. Testbed implementation architecture

In the present proposal, the tunnel build by NEMO is only used for initiating communications from the nodes in the rest of the world with the MNNs. For outbound traffic, MNNs can use their CoA to establish an MPTCP connection. This is the only difference between outbound and inbound traffic signaling. Once the connection is established, IP addresses can be added or removed using MPTCP options; inbound and outbound traffics take the same route.

The proposed approach thus improves routing, reduces the use of tunnels, potentially improves load balancing and throughput. These improvements have been brought to light with the help of a local testbed implementation, and the results of the experiments that were carried out are reported in the following sections.

## III. Testbed Architecture

The local testbed architecture involves a MN, a home network, two foreign networks and a CN. The MN consists of a MR and a MNN. The home network implements a HA for managing the mobility of the MN. The two visited/foreign networks are used for illustrating different mobility scenarios in which the MN moves from one foreign network to another. The communicating (distant) node is assumed to be somewhere inside the Internet.

In order to locally replicate a real networking scenario, delay and packet loss are added on the outgoing links to the HA, and to the foreign networks. In the local test bed, 10ms of delay and 5% of packet loss has been added. Those values have been chosen as quite typical of existing data paths.

The local testbed is a cluster of generic-purpose Linux PCs linked with Ethernet cables and switches as shown in Fig. 1.

- Delays are added using the netem tool in Linux [10].
- The NEMO implementation for Linux is provided by the UMIP working group [11]. NEMO is installed on the MR and on the HA. This particular implementation uses IPv6 addresses.
- The latest available version v0.90 of MPTCP available for Linux provided by University Catholique de Louvain. [12] is installed on the MNN and on the CN.

## IV. Measurement Scenarios

This section presents the network scenarios simulated on the local testbed. These scenarios are used to demonstrate how our proposed solution, augmenting NEMO with MPTCP, improves the performance delivered by NEMO only for the communications between a MNN and the Internet.

The three following network settings are used in following subsections:
(i) Network Settings 0 (NS0): the MN is directly attached to its home network (HN) via the HA;
(ii) Network Settings 1 (NS1): the MN is attached to a foreign network 1 (FN1) close to the CN; the MR is connected to AR(FN1).
(iii) Network Settings 2 (NS2): the MN is attached to a foreign network 2 (FN2), distant from the CN: the MR is connected to AR(FN2).

The network settings NS1 and NS2 differ only in terms of distance to the CN: FN2 is farther from CN than FN1. The different routing paths taken based on NEMO and NEMO augmented with MPTCP are shown in Fig. 2.

Simulating the attachment of the MN to different networks is simply implemented by modifying the wiring within the testbed: the Ethernet wire is unplugged manually from one link switch and then plugged manually to the switch connected to another link. Whenever such an operation is performed, the MR acquires a new IP address from the newly connected network and the CNs (MNN and CNs) restart their connection. Measurements can then be performed to assess the performance of the connection.

Measurements are performed for 5 different scenarios:

- The MN is attached to its home network (NS0);
- The MN is attached to FN1 and "classical" NEMO is used (NS1C); the traffic is routed through the tunnel between HA and MR via AR(FN1);
- The MN is attached to FN1 and NEMO augmented with MPTCP is used (NS1P); the traffic is directly routed between MR and CN via AR(FN1);
- The MN is attached to FN2 and "classical" NEMO is used (NS2C); the traffic is routed through the tunnel between HA and MR via AR(FN2);
- The MN is attached to FN2 and NEMO augmented with MPTCP is used (NS2P); the traffic is directly routed between MR and CN via AR(FN2).

## V. Performance comparison of classical NEMO vs NEMO augmented with MPTCP

This section evaluates how augmenting NEMO with MPTCP impacts the performance delivered to connections between a MNN and a CN.

Theoretical assessments are performed, in order to assess the gain in terms of unidirectional delay, round trip time (RTT) and throughput.

Measurements carried out on the testbed between the MNN and the CN complete and confirm the theoretical computations.

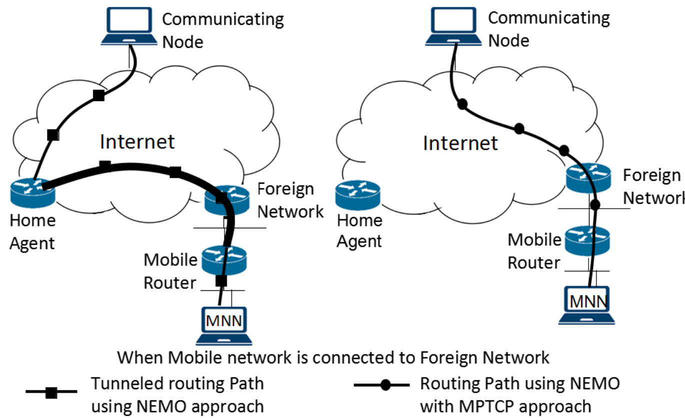


Fig. 2. Routing Path between CN and MR using Classical NEMO vs NEMO augmented with MPTCP

Throughput measures are obtained using netperfmeter [13] command, while RTT measures are obtained using ping6.

### A. Delay Performance

Delay performance is first theoretically assessed in subsection V-A1 and testbed measures are reported in subsection V-A2.

*1) Qualitative Analysis:* The unidirectional delay $DelayTime$ between MNN and CN is the sum of processing delay and propagation delay ($PathDelay$) as shown in the following equation.

$$DelayTime = ProcessingDelay + PathDelay \quad (1)$$

With NEMO, the MR sends the packets through a tunnel between MR and HA. Therefore, the traffic between the MNN and the CN also follows this route. The $PathDelay$ between MNN and CN is derived from Fig. 2, as follows:

$$MNN \rightarrow MR \rightarrow AR(FN) \rightarrow Internet \rightarrow HA \rightarrow Internet \rightarrow CN \quad (2)$$

When NEMO is augmented by MPTCP the route between MNN and CN is more direct. The $PathDelay$ between MNN and CN is in that case is also derived from Fig. 2:

$$MNN \rightarrow MR \rightarrow AR(FN) \rightarrow Internet \rightarrow CN \quad (3)$$

The respective unidirectional delays for NEMO ($Delay_{classical}$) and for NEMO augmented by MPTCP ($Delay_{proposed}$) are derived from equations (1), (2) and (3) as follows:

$$Delay_{classical} = ProcessingDelay@_{(\text{MNN,MR,AR(FN),Internet,HA,Internet,CN})} + PathDelay_{between\_(\text{MNN,MR,AR(FN),Internet,HA,Internet,CN})} + TunnelDelay@_{(\text{MR,HA})} \quad (4)$$

where $TunnelDelay@_{(\text{MR,HA})}$ is the time taken for encapsulating and decapsulating packets.

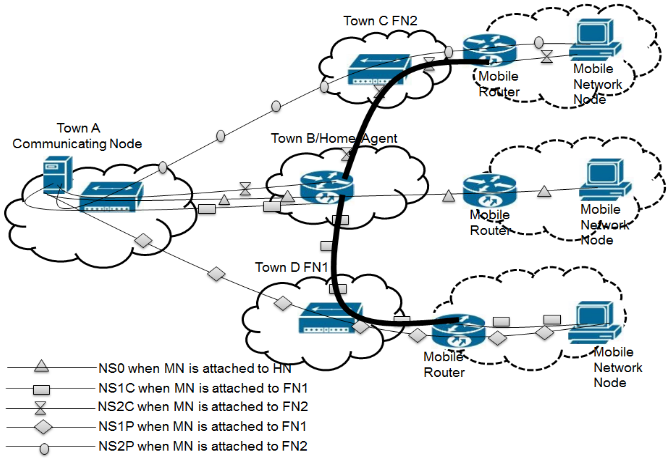


Fig. 3. Network settings for comparing RTT values

$$Delay_{proposed} = ProcessingDelay@_{(\text{MNN,MR,AR(FN),INTERNET,CN})} + PathDelay_{between\_(\text{MNN,MR,AR(FN),INTERNET,CN})} \quad (5)$$

Comparing equations (4) and (5), it appears that augmenting NEMO with MPTCP potentially reduces the unidirectional delay between MNN and CN as the tunnel between MR and HA is avoided; this limits not only the transmission delay as the route is shorter but also the processing delay as there is no need to encapsulate packets between MR and HA.

This qualitative analysis is confirmed below, by measuring the RTT on the testbed.

*2) Testbed measurements:* The improvement in terms of transmission delay can be assessed by measuring RTT for the network scenarios. Although the route followed by the packets is not same in both the directions yet the measurement of round-trip time can approximately be used to confirm the results of the qualitative analysis in section V-A1.

For assessing the performance in terms of RTT, we configure the local testbed in order to simulate the network settings shown in Fig. 3. The CN and the HA are respectively in town A and town B while FN1 and FN2 are respectively in town C and town D. Link delays between CN and respectively HA and AR(FN1) are identical, and smaller than the link delay between CN and AR(FN2). The RTT is measured on the testbed by using ping6 for each scenario.

The RTT is measured on the testbed by using ping6 for the five scenarios described in section IV and is reported in Fig. 4.

It is first seen that the RTT performance observed for a MNN is always worse when the MN is not attached to its home network when classical NEMO is used: this is seen by comparing the RTT values for NS1C and NS2C to the RTT values for NS0. This was to be expected as the classical NEMO path is always longer than the path between MR and CN.

Also, the RTT performance is better for NEMO augmented with MPTCP than for classical NEMO: RTT for NS1P (respectively NS2P) is smaller than for NS1C (respectively NS2C). This confirms the theoretical assessment of section V-A1.

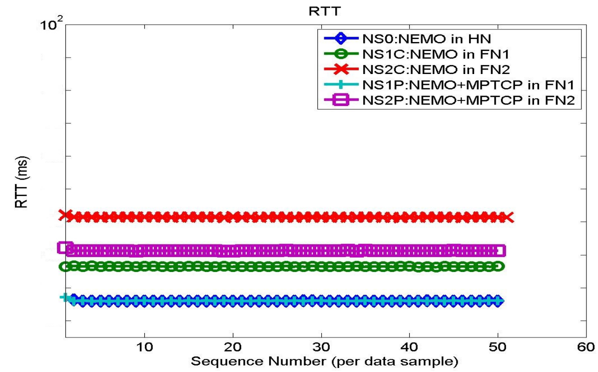


Fig. 4. RTT values: NEMO vs NEMO augmented with MPTCP

Lastly, the RTT performance observed for a MN attached to foreign network when NEMO is augmented with MPTCP could be either better or worse than the RTT performance observed for the MNN attached to its home network (the MR could be closer to the FN than to its Home Network). In Fig. 4, with the considered network settings, the RTT values for NS0 and NS1P are identical as the link delays between CN and respectively HA and AR(FN1) are also identical.

## B. Throughput Gain

Throughput is the rate at which any node can process the data. The gain in the throughput is achieved by avoiding the tunneling packet overhead and by enhanced routing, which are respectively addressed in subsections V-B1 and V-B2.

*1) Packet Overhead:* The bi-directional tunnel in classical NEMO is established by encapsulating packets. The HA encapsulates the mobile network's inbound traffic with the MR's care-of-address, and the MR encapsulates the outbound traffic with the HA's address.

As the size of an IPv6 header is 40 bytes, the maximum packet size is 1460 bytes, which corresponds to an overhead for NEMO at least equal to 2.74% , and possibly larger if the packet size is smaller. There is no overhead for NEMO augmented with MPTCP as no tunnel is used.

*2) Enhanced Routing:* Network throughput is inversely proportion to the round trip time [14]. Therefore, a larger RTT will result in a smaller throughput. In the measures performed on the testbed, the throughput is computed on both the MNN and the CN with the help of netperfmeter [13].

For the measurement of the throughput, the MNN and the CN form a client-server relation where the MNN acts as a client and the CN acts as a server. The throughput is measured in two different settings:

***Netperfmeter-Bidirectional:*** Netperfmeter, measures the throughput by sending packets of different sizes, on longer duration periods. This utility measures the throughput when the client and server both send data to each other. This is the default scenario provided by netperfmeter for bidirectional data exchange in between client and server.

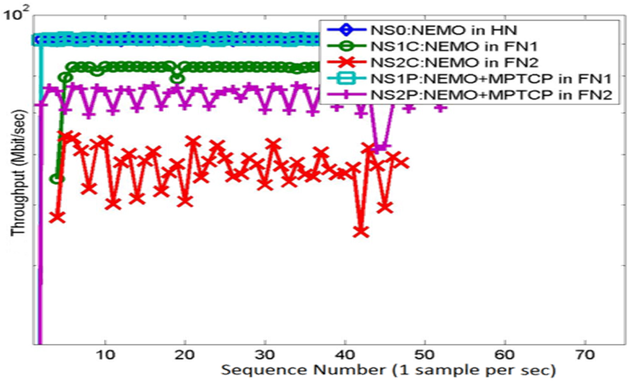


Fig. 5. Throughput Graph

Measures are performed for the five scenarios described in section IV and are reported in Fig. 5.

In Fig. 5, the curves for NS1P and NS2P can be respectively compared with the curves NS1C and NS2C, considering NS0 as a benchmark. The curves for NS0 and NS1P are overlapping each other. The comparison shows a significant throughput gain by using NEMO augmented with MPTCP. It is also noticed that the ranking of the different scenarios is the same than the one observed in Fig. 4, which was to be expected as the throughput should roughly be inversely proportional to the RTT. By considering the average values for each scenario in all the throughput graphs, the throughput gain can be calculated as $15(\pm 5)\%$; however, since this value is directly related to the delays between the MNN and respectively the HA and the AR(FN), it is highly dependent on the network settings.

All the scenarios present fluctuations in the network throughput. One of the contributing factor in the fluctuation of throughput data could be the current version of MPTCP stack, as the throughput measurements for UDP datagrams are consistent.

## VI. Conclusion and perspective

This paper presents a quantitative and qualitative performance comparison between classical NEMO and NEMO augmented with MPTCP using relevantly chosen scenarios. The novel combination of NEMO and MPTCP, works effectively with the addition of two very small functions in the NEMO functionality on the MR. With these two small functions the MR is able to advertise the acquired care-of-prefix in the foreign network. This IP address with the acquired prefix is then used to route the packets towards the Internet instead of the HA. In the results section V, we can see that NEMO augmented with MPTCP effectively reduces the round trip time, transmission delay and significantly improves the throughput when compared with that of the classical NEMO. This corroborates that the novel combination of NEMO augmented with MPTCP indeed provides better network mobility support than classical NEMO alone.

In future, it would be interesting to study more number of scenarios e.g., when there are many MNNs and CNs which are communicating in parallel. This study could be done using a simulator (e.g. Omnet++ or NetKit). In the local test bed, the hand over performed is by plugging in and unplugging the wires (i.e., hard handover). Using the simulator, the smooth handover can be studied for comparing NEMO and NEMO augmented with MPTCP.

## References


[1] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, “Rfc 3963 network mobility (nemo) basic support protocol,” 2005.

[2] C. J. Bernardos, A. De La Oliva, M. Calderón, D. von Hugo, and H. Kahle, “Nemo: Network mobility. bringing ubiquity to the internet access,” *demonstration at IEEE INFOCOM*, 2006.

[3] D. Farinacci, D. Lewis, D. Meyer, and V. Fuller, “The locator/id separation protocol (lisp),” *RFC 6830*, 2013.

[4] S. Bhatti and R. Atkinson, “Identifier-locator network protocol (ilnp) architectural description,” *RFC6740, April*, 2012.

[5] C. E. Perkins, “Mobile networking through mobile ip,” *Internet Computing, IEEE*, vol. 2, no. 1, pp. 58–69, 1998.

[6] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, “Rfc 6824: Tcp extensions for multipath operation with multiple addresses,” 2013.

[7] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, “Host identity protocol,” *RFC5201, April*, 2008.

[8] P. Mitharwal, C. Lohr, and A. Gravey, “Boosting nemo with multi-path tcp,” in *Network of the Future (NOF), 2015 6th International Conference on the*. IEEE, 2015, pp. 1–7.

[9] T. Narten, S. Thomson, and T. Jinmei, “Ipv6 stateless address autoconfiguration,” *RFC 2462*, 2007.

[10] T. L. Foundation, “Netem,” http://www.linuxfoundation.org/collaborate/workgroups/-networking/netem, Accessed: 2016-03-18.

[11] U. group, “How to setup a nemo basic support testbed with ipsec static keying),” http://umip.org/docs/umip-nemo.html, Accessed: 2016-02-24.

[12] F. D. Sébastien Barré, Christoph Paasch and G. Detal., “Multipath tcp - linux kernel implementation,” http://www.multipath-tcp.org/, Accessed: 2016-03-18.

[13] T. Dreibholz, “Netperfmeter : A network performance metering tool,” http://blog.multipath-tcp.org/blog/html/2015/09/07/netperfmeter.html, Accessed: 2016-02-24.

[14] Wikipedia, “Transmission time,” https://en.wikipedia.org/wiki/Transmission_time, Accessed: 2016-02-24.