H.323 Traffic Characterization
Test Plan Draft

Paul Schopis, pschopis@itecohio.org

I. Introduction

Recent attempts at providing Quality of Service in the Internet2 community have focused primarily on Expedited Forwarding as described in RFC2598[1] and several draft documents in various stages of the IETF approval process[2,3]. These documents and the network behaviors described in them make certain assumptions about an application's network requirements. Additionally, the EF Per Hop Behavior is described in strictly relativistic terms. For example, the simple expression D(i)-E(i)<=S*R/MTU is used to describe a PHB [2]. D(i) is the departure rate and E(i) is the ingress rate. Stated simply, no queuing will occur as long as the departure rate is greater than the arrival rate. This expression appears to be a variation on Little's formula [4] that simply states as long as the offered load is not greater than a system's capacity to process arrival traffic no queuing delay will occur.

Which queuing strategy is appropriate for a given application is not well understood. For example, tests have shown that there is a linear relationship between aggregate traffic load and burstiness when a strict priority queue is used to reduce latency[5]. Attributes such as this may be detrimental to real time video if loads are sufficiently large.

H.323 video is an interesting protocol in that it was developed for a Best Effort network and exhibits certain properties such as video channel transmission rate back-off when less data is needed for accurate video updates. In order to enable a QoS strategy that truly helps H.323 in both performance and reliability ITEC-Ohio and ITEC-NC are collaborating to perform H.323 traffic characterization tests as a first step in developing a QoS strategy for the Megaconference 3.

II. Definitions

For purposes of clarity we will adopt the following definitions from the ITU Y.1541 draft Standard.
IPTD - ip packet transfer delay commonly referred to as latency.
IPDV- ip packet delay variation.
IPLR – ip packet loss ratio.
IPER – ip packet error ratio.
IPPT – ip packet through put.
IPOT – Octet based IP packet throughput.

III. Bounds

The Y.1541[6] provides the following bounds for what it defines as Class 0 service. Class 0 service approximates the IETF EF definition but in hard bounds as opposed to relative treatment. Although we are going to test for H.323 video bounds via network emulation we are aware of the following draft recommendations.
IPTD – Upper bound on the mean IPTD, 150ms.

IPDV - Defined as IPTD $_{\text{Max}}$ - IPTD $_{\text{Min}}$ = IPDV in Appendix II of Y.1541[6]. This intuitively seems sufficient since we are mostly interested in the width of the jitter window. The defined bound is 50ms.
IPLR – Upper bound on packet loss probability, $1x10^{-3}$. Derived for layer 3 from known layer 2 (ATM) bounds.
IPER – Upper bound, $1x10^{-4}$.

The tests will be of two types. First, we will employ network emulation tools to calculate performance bounds on H.323 clients and MCUs. The tests will include cascaded MCUs under load to simulate as nearly as possible the MCU topology that will be required for Megaconference 3. We desire to find the performance bounds for latency, loss and jitter. As a starting point we will use the values for the appropriate metrics from ITU's Y.1541 draft standard. Secondly, we wish to capture the traffic streams and be able to reproduce them at will in the lab so we may experiment with various queuing strategies and other QoS mechanisms. It is our hypothesis that we must understand the characteristics of H.323 video network traffic to improve our probability of success.

Secondly, we will characterize the H.323 traffic at the network injection point particularly in regards to IPDV and burstiness on both clients and MCUs.

IV. Bounds Test Equipment Verification.

A: Baseline test of software and hardware configuration.

A1. Test Objective:
Verify test mechanisms for bounds test.

A2. Entry Criteria:
None.

A3. Test Setup:

1. Configure Linux platform with Nistnet software for network performance emulation.
2. Connect each Ethernet port on Linux platform to a Smartbits e.g. one as a transmit and one as a receive port. See figure one.

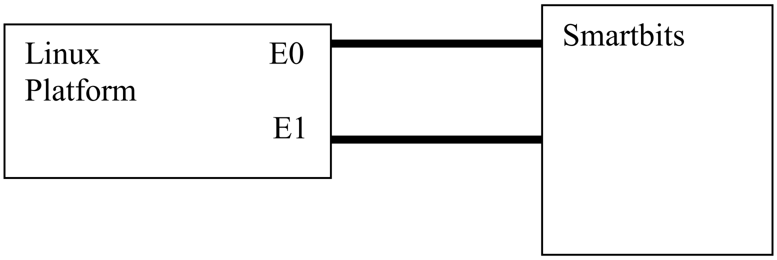


Figure 1.

A4. Test Execution: Basic performance

1. Verify Linux platform is routing correctly via route command.
2. Send traffic at 5.38 Load (~384k data bits) of 10M ethernet interface.
3. Set packet size to 128 byte packets.

A5. Expected result:
It is expected all packets will be routed correctly.

A6. Repeat test with following values for load, 15% ,50%, 75-80% , 90-100%. Repeat test with the following values for packet sizes 1000 and 1500 bytes.
Note: The tester must use some discretion in choosing value pairs for this test. The intention is to find the failure modes for the test platform.

A7. Test Execution: Nistnet reliability

1. Configure Nistnet to drop 5% of traffic.
2. Send traffic from the Smartbits into E0 and route out E1 to Smartbits.

A8. Expected Result:
It is expected that the Nistnet will drop the specified traffic.

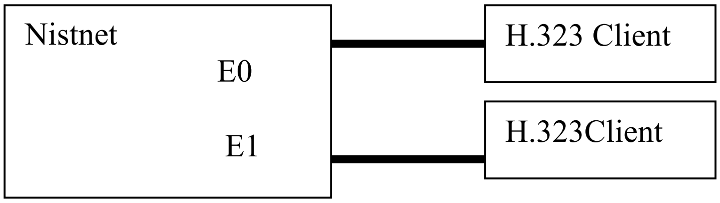
A9. Repeat test using drop of 1%, 0.1%, 0.01% etc.

A10. Repeat testing procedure for latency using 50ms, 100ms, and 500ms delay.
Repeat testing procedure for jitter using 100ms delay and 11.5 sigma with no correlation and work downward to find approximately 10ms steps in the jitter window .

Note: If tests show a significant delta between expected and received results the tester should collect enough data points to try and construct a functional relationship between metrics set and observed.

B: Point to point H.323 client bounds test.

B1: Test objective:
Test performance bounds for Loss, Latency and Jitter.
B2: Entry criteria:
Test A has been performed and there is at least a consistent functional basis for data interpretation.
B3: Test set up:
Connect two H.323 clients back to back through the network emulation device (Nistnet).
See figure two.



B3. Test Execution:

1. Configure Nistnet platform to increase latency by 10ms.
2. Establish call between Clients at 768Kbps.
3. Observe call quality
4. Repeat calls with increasing latency every iteration by 10ms until call quality deteriorates.

B4. Expected results:
Call quality will start to become unusable around 200-400ms range.

B5. Test Execution:

1. Configure Nistnet platform to increase drop starting at 0.01% to 2.25%.
   The following data points for drop will be tested, 0.01%, 0.1%, 0.25%, 0.5%, 0.75%, 1%, 1.25%, 1.5%, 1.75%, 2.0%, 2.25%.
2. Establish call between Clients at 768Kbps.
3. Observe call quality.

B6. Expected Results:
Packet loss will effect quality ~10^{-4}

B7. Test Execution:

1. Configure Nistnet platform to increase jitter by 10ms.
2. Establish call between Clients at 768Kbps.
5. Observe call quality
6. Repeat calls with increasing jitter every iteration by 10ms until call quality deteriorates.

B6. Expected Results:
Jitter will effect quality when it is greater than 50ms.

C: Point to point H.323 client to MCU bounds test.

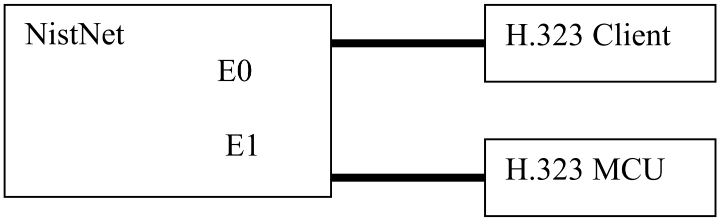


Figure Three.

C1. Test objective:
Test performance bounds for latency between an MCU and client. Entry criteria is tests A and B have been completed.

C2. Entry Criteria:
Tests A and B have successfully been completed.
C3. Test set up:
Connect an H.323 client back to back to an MCU through the network emulation device (Nistnet).

C4. Test Execution:
1. Configure Nistnet platform to increase latency by 10ms.
2. Establish call between Clients at 768Kbps.
3. Observe call quality
4. Repeat calls with increasing latency every iteration by 10ms until call quality deteriorates.

C.5 Expected Results:
Call quality will start to become unusable around 200-400ms range.

C6. Test objective:
Test performance bounds for loss and between an MCU and client. The entry criteria is tests A, B and C1 have been completed.

C7. Test set up:
Connect an H.323 client back to back to an MCU through the network emulation device (Nistnet).

C8. Test Execution:
1. Configure Nistnet platform to increase drop starting at 0.01% to 2.25%.
   The following data points for drop will be tested, 0.01%, 0.1%, 0.5%, 0.75%, 1%, 1.25%, 1.5%, 1.75%, 2.0%, 2.25%.
2. Establish call between Clients at 768Kbps.
4. Observe call quality.

C.9 Expected Results:
Packet loss will effect quality ~10^{-4}

C10. Test objective:
Test performance bounds for jitter and between an MCU and client.

C11. Test set up:
Connect an H.323 client back to back to an MCU through the network emulation device (Nistnet).

C12. Test Execution:
1. Configure Nistnet platform to increase jitter by 10ms.
2. Establish call between Clients at 768Kbps.
5. Observe call quality

Repeat calls with increasing jitter every iteration by 10ms until call quality deteriorates.

C13. Expected Results:
Jitter will effect quality when it is greater than 50ms.

D: Switch Propagation Delay Test

D1. Test Objective:
The purpose of this test is to determine the propagation delay introduced by the switch that will be used for the cascading of MCUs in subsequent tests.

D2. Entry Criteria.
Test A has been completed and the Nistnet Emulator is functioning acceptably.

D3. Test Set Up.
Plug two ports of a Cisco 2924XL into the Smartbits into 10/100 ports.
See figure four.

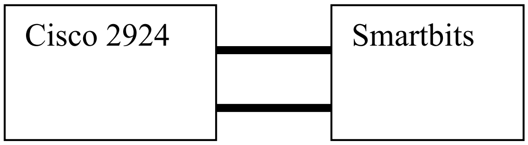


Figure Four.

D4. Test Execution.

1. Configure ports on the 2924XL and the ports on the Smartbits to match the transmit and receive speeds and duplex settings of the MCUs that will be used in test E.
2. Pass traffic at approximately 384Kbps. From one port to the other on the Smartbits.
3. Note delay in traffic.

D5. Expected results.
It is expected that the total propagation delay will be 10ms or less.

E: One Way Network Failure Via MCU Test

E1. Objective:
The purpose of this test is to see the effects of latency using an MCU for one way network problems.

E2. Entry Criteria:
All tests A through D have been completed.

E3. Test Set Up.
1. Plug one client directly into the Network Emulation device.
2. Plug the Cisco 2924XL into the second port of the Nistnet device.
Note: each Nistnet port is a different subnet.

3. Plug the MCU and the other client into the Cisco Switch.
4. Configure a common conference on the MCU and register it with the on board GateKeeper.
4. Register both clients with the GateKeeper.

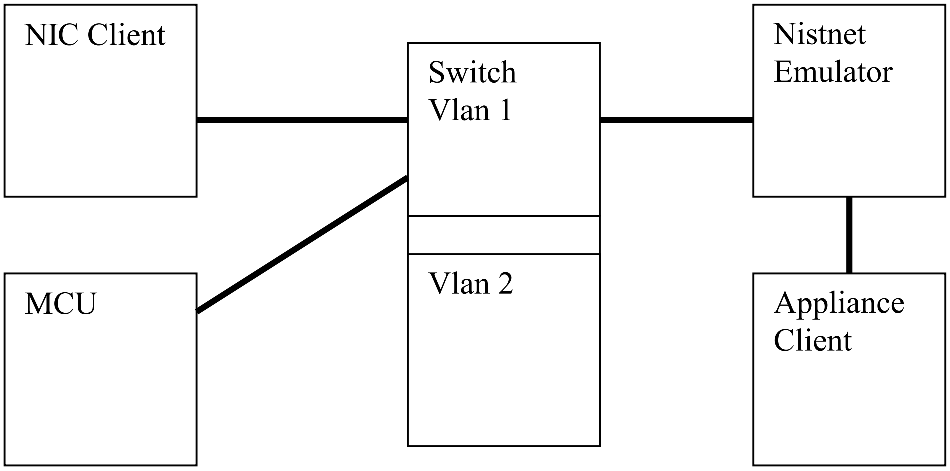


Figure 5.

E3. Test Execution.
1. Configure Nistnet platform to increase latency by 10ms.
2. Establish call between Clients Through MCU at 768Kbps.
3. Observe call quality
4.Repeat calls with increasing latency every iteration by 10ms until call quality deteriorates.

E4.Expected Results.
Call quality will start to become unusable around 200-400ms range.

E5. Objective:
The purpose of this test is to see the effects of loss using an MCU for one way network problems. Secondly, since the errors are introduced between the Appliance and the MCU and the NIC is on the same subnet as the MCU. We wish to evaluate the MCU's role as a replay buffer.

E6. Test Set Up.
Test set up is the same as section E2.

E7. Expected Results:
Packet loss will effect quality ~10^{-4}

E8. Test Execution.

1. Configure Nistnet platform to increase drop starting at 0.01% to 2.25%.
   The following data points for drop will be tested, 0.01%, 0.1%, 0.5%, 0.75%, 1%, 1.25%, 1.5%, 1.75%, 2.0%, 2.25%.
2. Establish call between Clients via MCU at 768Kbps.
3. Observe call quality.

E9. Objective:
The purpose of this test is to see the effects of jitter using an MCU for one-way network problems.

E10. Test Set Up.
Test set up is the same as section E2.

E11. Test Execution.
1. Configure Nistnet platform to increase jitter by 10ms.
2. Establish call between Clients at 768Kbps.
3.Observe call quality
4.Repeat calls with increasing jitter every iteration by 10ms until call quality deteriorates.

E12. Expected Results:
Jitter will effect quality when it is greater than 50ms.

F: Two Way Network Failure Via MCU Test

F1. Objective.
The objective of is to examine the effects of two way latency network problems when the clients are connected via an MCU.

F2. Entry Criteria:
Tests A through E have been completed.

F3. Test Set up.
It is the same as test E2.

F4. Test Execution.
1. Configure Nistnet platform to increase latency by 10ms.
2. Establish call between Clients Through MCU at 768Kbps.
3. Observe call quality
4.Repeat calls with increasing latency every iteration by 10ms until call quality deteriorates.

F5. Expected Results:
Call quality will start to become unusable around 200-400 ms range.

F6. Objective:
The purpose of this test is to see the effects of loss using an MCU with two way network problems.

F7. Test Set Up.
Test set up is the same as section E2.

F8. Test Execution.
1. Configure Nistnet platform to increase drop starting at 0.01% to 2.25%.
   The following data points for drop will be tested, 0.01%, 0.1%, 0.5%, 0.75%, 1%, 1.5%, 1.75%, 2.0%, 2.25%.
2. Establish call between Clients via MCU at 768Kbps.
3. Observe call quality.

F9. Expected Results:
Packet loss will effect quality at ~10^{-4}

F10. Objective:
The purpose of this test is to see the effects of jitter using an MCU for two- way network problems.

F11. Test Set Up.
Test set up is the same as section E2.

F12. Test Execution.
1. Configure Nistnet platform to increase jitter by 10ms.
2. Establish call between Clients at 768Kbps.
3.Observe call quality
4.Repeat calls with increasing jitter every iteration by 10ms until call quality deteriorates.

F13. Expected Results:
Jitter will effect quality when it is greater than 50ms.

G: One-Way Cascaded MCU test.

G1. Objective:
The purpose of this test is to see the effects of latency using a cascaded MCU for one way network problems.

G2. Test Set Up.
1. Plug each client directly into the Cisco 2924XL
2. Create two separate VLANs so that each client is on a separate subnet.
2. Plug each VLAN of the Cisco 2924XL into the ports of the Nistnet device.
Note: Each Nistnet port is a different subnet.
3. Plug the MCUs into the Cisco 2924XL so that the each client and MCU pair are on different subnets and must be routed via the Nistnet router.
4. Configure a common conference on one MCU and register it with the on board GateKeeper. Register the other MCU to the same GateKeeper
5. Register both clients with the GateKeeper.

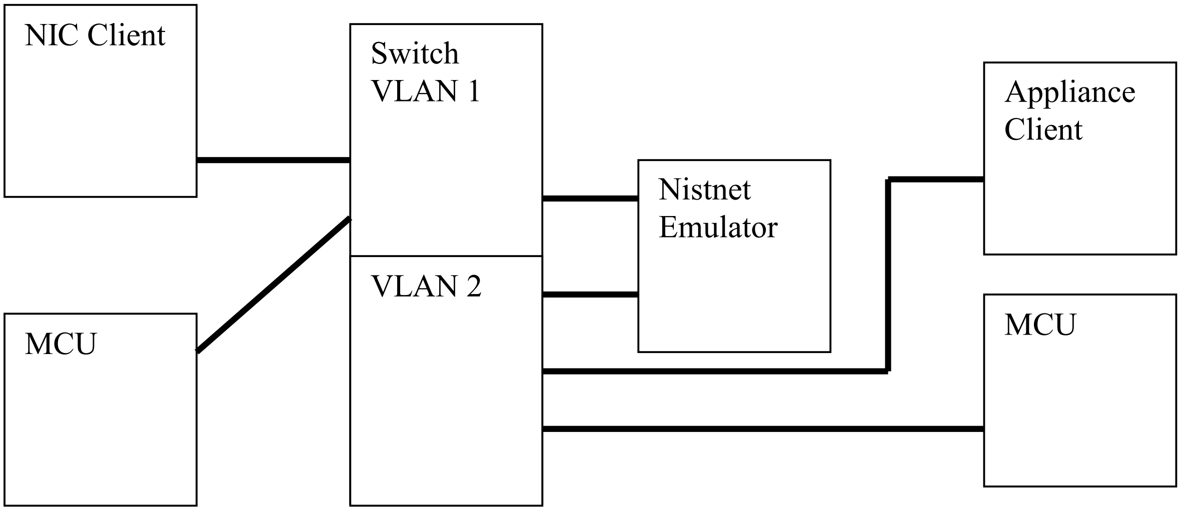


Figure 6.

G3. Test Execution.
1. Configure Nistnet platform to increase latency by 10ms.
2. Establish call between Clients Through MCU at 768Kbps.
3. Observe call quality
4.Repeat calls with increasing latency every iteration by 10ms until call quality deteriorates.

G4. Expected Results.
Call quality will start to become unusable around 200-400ms range.

G5. Objective:
The purpose of this test is to see the effects of loss using an MCU for two way network problems. Secondly, since the errors are introduced between the Appliance and the MCU and the NIC are on the same subnet as the MCU. We wish to evaluate the MCU's role as a replay buffer.

G6. Test Set Up.
Test set up is the same as section E2.

G7. Test Execution.
1. Configure Nistnet platform to increase drop starting at 0.01% to 2.25%.
   The following data points for drop will be tested, 0.01%, 0.1%, 0.5%, 0.75%, 1%, 1.25%, 1.5%, 1.75%, 2.0%, 2.25%.
2. Establish call between Clients via MCU at 768Kbps.
3. Observe call quality.

G8. Expected Results:
Call quality will deteriorate at ~ 10^{-4}.

G9. Objective:
The purpose of this test is to see the effects of jitter using an MCU for one-way network problems.

G10. Test Set Up.
Test set up is the same as section E2.

G11. Test Execution.
1. Configure Nistnet platform to increase jitter by 10ms.
2. Establish call between Clients at 768Kbps.
3.Observe call quality
4.Repeat calls with increasing jitter every iteration by 10ms until call quality deteriorates.

G12. Expected Results:
Jitter will effect quality when it is greater than 50ms.

H: One-Way Cascaded MCU test.

H1. Objective:
The purpose of this test is to see the effects of latency using a cascaded MCU for two-way network problems.

H2. Test Set Up.
3. Plug each client directly into the Cisco 2924XL
4. Create two separate VLANs so that each client is on a separate subnet.
2. Plug each VLAN of the Cisco 2924XL into the ports of the Nistnet device.
Note: each Nistnet port is a different subnet.
3. Plug the MCUs into the Cisco 2924XL so that the each client and MCU pair are on different subnets and must be routed via the Nistnet router.
4. Configure a common conference on one MCU and register it with the on board GateKeeper. Register the other MCU to the same GateKeeper
5. Register both clients with the GateKeeper.

H3. Test Execution.
1. Configure Nistnet platform to increase latency by 10ms.
2. Establish call between Clients Through MCU at 768Kbps.
3. Observe call quality
4.Repeat calls with increasing latency every iteration by 10ms until call quality deteriorates.

H4. Test Results.
Call quality will start to become unusable around 200-400ms range.

H5. Objective:
The purpose of this test is to see the effects of loss using an MCU for two-way network problems. Secondly, since the errors are introduced between the Appliance and the MCU

and the NIC is on the same subnet as the MCU. We wish to evaluate the MCU's role as a replay buffer.

H6. Test Set Up.
Test set up is the same as section E2.

H7. Test Execution.
1. Configure Nistnet platform to increase drop starting at 0.01% to 2.25%.
   The following data points for drop will be tested, 0.01%, 0.1%, 0.5%, 0.75%, 1%, 1.25%, 1.5%, 1.75%, 2.0%, 2.25%.
2. Establish call between Clients via MCU at 768Kbps.
3. Observe call quality.

H8. Expected Results.
Call quality will deteriorate at ~ 10^{-4}.

H9. Objective:
The purpose of this test is to see the effects of jitter using an MCU for two-way network problems.

H10. Test Set Up.
Test set up is the same as section E2.

H11. Test Execution.
1. Configure Nistnet platform to increase jitter by 10ms.
2. Establish call between Clients at 768Kbps.
3.Observe call quality
4.Repeat calls with increasing jitter every iteration by 10ms until call quality deteriorates.

H12. Expected Results:
Jitter will effect quality when it is greater than 50ms.

I: Traffic characterization test.

I1. Test Objective.
Characterize traffic in terms of IPDV and burstiness.

I2. Entry Criteria.
All previous tests have been successfully completed.

I3. Test Setup.
1. Connect two H.323 clients, one MCU and Smartbits to Cisco Switch.

I4. Test execution.
1. Set switch to snoop one H.323 client and pass that to the Smartbits.

2. Set Smartbits to trigger on source address of snooped client and additional criteria in item #4.
3. Connect both clients via call.
4. Repeat call three times each time changing the secondary trigger for the
    a. Control Channel (TCP stream)
    b. Video channel (Port number)
    c. Audio channel (Port number)
5. Observe IPDV and burstiness.
6. Repeat all steps for MCU.

I5. Expected Results.
No expectation except the traffic should be no worse than the bounds established by other tests.

Bibliography

[1] V. Jacobson, K. Nichols, and K. Poduri. RFC 2598: An expedited forwarding PBH, June 1999.
[2] G. Armitage, A. Casati, J. Crowcroft, J. Halpern, B. Kumar, J. Schnizlein. <draft-ietf-diffserv-efresolve-00.txt>: A revised expression of the Expedited Forwarding PHB. November 2000.
[3] V. Jacobson, K. Nichols, and K. Poduri. <draft-ietf-diffserv-pdb-vw-00.txt>: The 'Virtual Wire' Per-Domain Behavior. July 2000.
[4] Donald Gross and Carl M. Harris. Fundamentals of Queuing Theory. Third Edition 1998. pp. 10-13.John Wiley & Sons, New York.
[5] Tizana Ferrari, End-to-end performance analysis with traffic aggregation. Computer Networks, Volume 34 Number 6, December 2000. pp 905-914. Elesevier, Amsterdam.
[6] ITU Study Group 13, Revised draft Recommendation Y. 1541 'Internet protocol communication service - IP Performance and Availability Objectives and Allocations', November 2000.