What’s So Hard About Bandwidth Estimation?

• Aside from the mathematical modeling …

• Bandwidth Estimation techniques stress the capabilities of monitoring platforms:
  – They require accurate collection and accurate emission of packets
    • Accurate collection can be achieved via kernel instrumentation and GPS clocking
    • Accurate emission requires tighter control on generation
  – More so than basic delay/loss/jitter measurements, they require feedback control between the collecting and emitting process to adapt algorithmic behavior.
  – It is also unclear that a single bandwidth estimator will be sufficient for all link speeds, all applications to be measured, and all types of cross traffic.
    • Available bandwidth of a link vs. available bandwidth for a particular traffic class (application) on a link?
    • Do we need the ability to select / switch between multiple techniques as part of an estimation process?
Telcordia Internet Monitoring Platform (IMP)

- Improving the accuracy of the measurement process
  - Better than busy-wait, but without specialized hardware
  - Low cost, easily deployed infrastructure
- Simplify the introduction of new measurement techniques
  - Common plumbing and control
  - Dynamic emission of packets
  - No need to install custom emitters across the network
- Provide a measurement platform for network monitoring systems
  - Accurate measurements for operational support
    - Management of Quality of Service
    - Identification of service-affecting conditions
  - Service Level Agreement support
    - Delay, loss, delay variance and availability are basic requirements
  - Validation of Service Level Agreement claims
    - How does an enterprise know if their agreement is being met?

Telcordia IMP: Internet Monitoring Platform

- A Software Platform to Support Network Traffic Measurements
  - Active measurements of network characteristics
    - One-way end-to-end delay, delay variance, loss, reordering, available bandwidth
  - Probe description language
    - Frequency, spacing, size, contents, (e.g., a customized VoIP probe)
    - SNMP MIB and Command Line Interface Data directly from NEs
    - Extremely accurate time stamps (GPS-based) and probe generation
    - Commercial Off-The-Shelf components + custom software
    - Graphing, reporting, alerting, anomaly detection (wavelet, change point)
**Test Bed Environment**

- Multiple Cross Traffic Generators
- Multiple Bandwidth Estimators

**Techniques for Accurate Packet Emission**

- Packet emission can be supported at several layers
  - Application Software Based
    - Busy waits and program loops
    - Easiest to deal with, but subject to high variance
  - Kernel Interrupt Driven
    - Significantly greater accuracy
    - Can be difficult to use
      - Unless a platform provides the details
    - Still interruptible
  - Exploitation of Network Interface Card characteristics
    - Even greater accuracy for higher speeds
    - No interrupts
    - Requires per-NIC driver – difficult to use
  - Modification of NIC firmware
  - Dedicated Hardware
    - More costly to deploy
    - But, you get what you pay for
Kernel Interrupt Driven Packet Generation

- Common hardware – custom software
  - Real-time clock with 122 µsec granularity
  - Network interface card
  - No impact on standard functions
- Design components
  - RTC interrupt-driven state machine used to schedule IMP packet transmission
  - IMP packets bypass the kernel protocol stack
  - IMP packets formatted via an array of P-Spec Packet Descriptors copied into kernel space (/proc).
- 10 and 100 Mbit links

Single Packet Train
IMP: Measurement Plots

- Send 12 trains of 100 packets through network
- Simple Java implementation vs. kernel support
  - (Most code the same)

NIC FIFO Queue Driven Packet Generation

- Common hardware – custom driver
  - Internal NIC clock with 800 nSec cycle time access to packet data
  - Dedicated NIC, separate data path from IP stack packets
  - Still based on IMP packets formatted via an array of P-Spec Packet Descriptors
- Gbit links
Packet Probe Profile Objectives

- Should be independent of the packet generation technique
  - Whether busy wait, interrupt, NIC, dedicated HW
- Allow the receiver to control the behavior of the packet sender.
- Improve the expressiveness of the measurement request.
  - Short but descriptive specifications
- Allow a single packet emitter to generate multiple forms of packet probe profiles.
  - Minimizes the deployment / update problem
- A programmable specification, but without the overhead and variance caused by program execution.
- The mechanism should be small and efficient enough to be kernel resident.

P-SPEC Language

<table>
<thead>
<tr>
<th>Packet variables</th>
<th>p0 - pN</th>
<th>Sets the values of the first N words of a packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment operators</td>
<td>=, *=, etc.</td>
<td>Binary operators for generating simple progressions such as sequence numbers or varying delays or sizes</td>
</tr>
<tr>
<td>Synchronization commands</td>
<td>wait, sync, delay.</td>
<td>Creates a synchronization point in the stream for communicating back to the measurement collector</td>
</tr>
<tr>
<td>Synchronization variables</td>
<td>arg0 - argN</td>
<td>Control arguments set by &quot;start&quot; or &quot;continue&quot; commands from the collector</td>
</tr>
<tr>
<td>Packet descriptor variables</td>
<td>size, gap, tos,</td>
<td>Define packet constraints such as size and gap before next packet</td>
</tr>
<tr>
<td>Packet generator</td>
<td>packet</td>
<td>Forces a Packet Descriptor to be generated with the current set of values for content, size, etc.</td>
</tr>
<tr>
<td>Looping control</td>
<td>N( ... )</td>
<td>Repeat internal instructions N times, for generating repeated subsequences of packets</td>
</tr>
<tr>
<td>General purpose variables</td>
<td>a - z</td>
<td>General purpose variables for processing, but which have no special meaning to the packet generator.</td>
</tr>
</tbody>
</table>

- The Packet statement does not emit a packet, but creates a packet descriptor for later use by the kernel emitter.
- Synchronization statements are used to flush accumulated packet descriptors and possibly await further control information.
P-SPEC – Specification Language for Packet Sequences

- P-Spec gives control over:
  - Packet content
  - Packet size
  - Inter-packet gap
  - Inter-packet-group delay
  - Synchronization data sent back to local requester
  - Control arguments sent from local requester to remote emitter

- Example P-Spec for an Adaptive Dispersion Technique packet sequence (pathload):

  The probe sends fleets of packets, with the packet size, inter packet gap and inter burst delay adjusted by the collecting process according to the analysis algorithm.

```plaintext
fleets_per_burst=3
packets_per_fleet=20
burst=3

*: (  
  WAIT(fleet++)  
  gap=arg0  
  size=arg1  
  delay=arg2  
  p1=0  
  p3=packets_per_fleet  
  fleets_per_burst: (  
    p0=burst  
    p2=0  
    packets_per_fleet: (  
      PACKET()  
      p2+=1  
    )  
    p1+=1  
  )  
  p1+=1  
  DELAY(delay)  
  inter-fleet delay  
)
```

Platform Architecture

- Descriptors:
  - Packet Emitter
  - P-spec Interpreter
  - Remote Measurement Controller
  - P-Spec Compiler
  - Enhanced OS Driver
  - Packet Collector
  - Local Measurement Controller

- Only functions requiring changes for a new measurement technique:
  - Reporting
  - Anomaly Detection
  - Graphing
  - Storage

Reporting, Anomaly Detection, Graphing

Copyright © 2003, Telcordia Technologies, Inc. All Rights Reserved.