NM-WG Measurement Schemas

Global Grid Forum
Network Measurements Working Group

Dan Gunter, Martin Swany, Jason Zurawski

Overview

- NM-WG focused on standardizing schemas for exchanging network measurements
- Published one document (“characteristics”): set of standard names for measurements
- Two sets of schemas out there:
  - v.1::Monolithic, straight mapping of “characteristics” doc
  - v.2::Framework, adding in from “characteristics” piecemeal

Where are we now?

- v.2 base schema has been around for a while
- Just got an iperf, ping, traceroute
- Need to discuss these
- Need implementor feedback
- Working on developer's guide to smooth transition and help with Web Svcs details
  - *not yet published, but v.0.1 is available*

v2 Schemas Recap

- Request:
  - metadata:
    - characteristic (e.g. http://www.ggf.org/nmwg/iperf)
    - subject (e.g. host pair)
    - parameters (e.g. TTL)
- Response:
  - metadata: <same as above>
  - data: <depends on characteristic>

Implementation status/demo

- Jason Zurawski will demonstrate a sample client..

Schemata

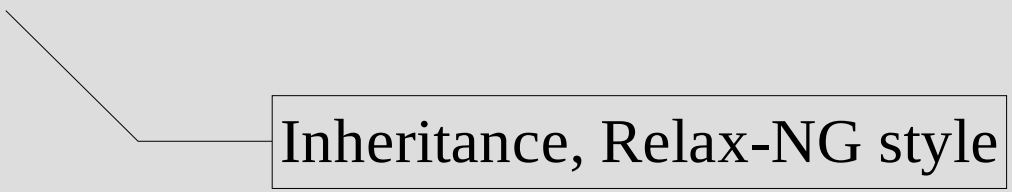
Terminology & Namespaces

- *Characteristic*
  - NMWG documented, blessed network metric
- *Alien characteristic*
  - Network metric not blessed by NMWG
- *Tool*: iperf, ping, etc.
- *Event type* = mapping to XML namespace
  - characteristic delay.rtt
    - (http:)//www.ggf.org/nmwg/characteristic/delay/rtt/
  - alien characteristic is.fine
    - (http:)//www.ggf.org/nmwg/alien/is/fine/
  - tool iperf
    - (http:)//www.ggf.org/nmwg/tool/iperf/

Preamble

- schema language: Relax-NG compact syntax
- assume header for event type “X”:

```
default namespace="http://www.ggf.org/nmwg/X/"
namespace nmwg="http://www.ggf.org/nmwg/"
include "nmbase.rnc"
include "topology.rnc"
Metadata |= XMetadata
Results |= XResults
```



Ping Schema (1)

```
PingMetadata =
   element metadata {
      attribute nmwg:id    { Identifier },
      element   subject    { PingSubject },
      element   parameters { PingParameters }
   }
PingSubject =
   Subject,
   ( nmwg:HostPair | nmwg:HostPairQuery )
PingParameters =
   element pcktsz { xsd:int }?,
   element cnt    { xsd:int }?,
   element itrvl  { xsd:int }?,
   element ttl    { xsd:int }?,
   element tmout  { xsd:int }?
   # etc..
```

Ping Schema (2)

```
PingResults =
   element results {
      element probe {
         attribute num     { xsd:int   },
         element   bytes   { xsd:int   },
         element   seqnum  { xsd:int   },
         element   ttl     { xsd:int   },
         element   value   { xsd:float }
      }*
   }
```

Ping request example

```
<delayroundtrip:request xmlns:delayroundtrip="http://www.ggf.org/nmwg/delay/roundtrip/" xmlns
   <delayroundtrip:metadata id="mdid1">
     <delayroundtrip:subject id="mdid1_S">
       <delayroundtrip:hostPair>
         <delayroundtrip:src type="hostname">dreadnought.cis.udel.edu</delayroundtrip:src>
         <delayroundtrip:dst type="hostname">otc2.psu.edu</delayroundtrip:dst>
       </delayroundtrip:hostPair>
     </delayroundtrip:subject>
     <delayroundtrip:parameters id="mdid1_P">
       <delayroundtrip:meta_id />
       <delayroundtrip:packetsize />
       <delayroundtrip:numbytes />
       <delayroundtrip:arguments />
       <delayroundtrip:units />
       <delayroundtrip:data_id />
       <delayroundtrip:time_type />
       <delayroundtrip:seq_num />
       <delayroundtrip:query_num />
       <delayroundtrip:value />
       <delayroundtrip:hop />
     </delayroundtrip:parameters>
   </delayroundtrip:metadata>
</delayroundtrip:request>
```

Ping response example

```
<delayroundtrip:request xmlns:delayroundtrip="http://www.ggf.org/nmwg/delay/roundtrip/" xmlns
   <delayroundtrip:metadata id="998877">
     <delayroundtrip:subject id="987">
       <delayroundtrip:hostPair>
         <delayroundtrip:src type="hostname">dreadnought.cis.udel.edu</delayroundtrip:src>
         <delayroundtrip:dst type="hostname">alderaan.cse.psu.edu</delayroundtrip:dst>
       </delayroundtrip:hostPair>
     </delayroundtrip:subject>
     <delayroundtrip:parameters id="789">
       <delayroundtrip:packetsize>56</delayroundtrip:packetsize>
       <delayroundtrip:arguments>-c 1</delayroundtrip:arguments>
       <delayroundtrip:units>ms</delayroundtrip:units>
     </delayroundtrip:parameters>
   </delayroundtrip:metadata>
   <delayroundtrip:data id="45451819" metadataId="998877">
     <delayroundtrip:time type="unix" value="1107492095">
       <delayroundtrip:datum seqnum="0" querynum="1" numbytes="64" value="19.1" hop="" />
       <delayroundtrip:datum seqnum="1" querynum="1" numbytes="64" value="19.2" hop="" />
     </delayroundtrip:time>
   </delayroundtrip:data>
</delayroundtrip:request>
```

Iperf (1)

IperfMetadata = element ipf:metadata {

attribute id { Identifier },

element ipf:subject { IperfSubject },

element ipf:parameters { IperfParameters } }

IperfSubject = (HostPair | HostPairQuery)

IperfParameters =

element frmt { xsd:string }?, element transtime { xsd:int }?, element intrvl { xsd:int }?, element windowsz { xsd:float }?, # etc..

Iperf (2)

```
IperfResults =
   element ipf:results {
      element probe {
         attribute num { xsd:int },
         element interval { string },
         element transfer { xsd:float },
         element bandwidth { xsd:float }
       }*
    }
```

Iperf request example

```
<bandwidthavailable:request xmlns:iperf="http://www.ggf.org/nmwg/tools/iperf/"
       xmlns:bandwidthavailable="http://www.ggf.org/nmwg/bandwidth/available/"
       xmlns="http://www.ggf.org/nmwg/" xmlns:nmwg="http://www.ggf.org/nmwg/">
    <bandwidthavailable:metadata id="mid1">
       <bandwidthavailable:subject id="mid1s">
          <bandwidthavailable:hostPair>
             <bandwidthavailable:src type="hostname">
          dreadnought.cis.udel.edu
          </bandwidthavailable:src>
             <bandwidthavailable:dst type="hostname">
          planet2.cs.ucsb.edu
          </bandwidthavailable:dst>
          </bandwidthavailable:hostPair>
       </bandwidthavailable:subject>
       <bandwidthavailable:parameters id="mid1p">
          <bandwidthavailable:port />
          <bandwidthavailable:protocol>TCP</bandwidthavailable:protocol>
       </bandwidthavailable:parameters>
    </bandwidthavailable:metadata>
</bandwidthavailable:request>
```

Iperf Request other Params

```
<bandwidthavailable:windowsize />
<bandwidthavailable:meta_id />
<bandwidthavailable:numbytes />
<bandwidthavailable:arguments />
<bandwidthavailable:units />
<bandwidthavailable:data_id />
<bandwidthavailable:time_type />
<bandwidthavailable:seq_num />
<bandwidthavailable:query_num />
<bandwidthavailable:value />
<bandwidthavailable:hop />
```

Iperf response example

```
<bandwidthavailable:response xmlns:iperf="http://www.ggf.org/nmwg/tools/iperf/" xmlns:bandwidth
   <bandwidthavailable:metadata id="7713496">
      <bandwidthavailable:subject id="9812740">
         <bandwidthavailable:hostPair>
            <bandwidthavailable:src type="hostname">dreadnought.cis.udel.edu</bandwidthavailab
            <bandwidthavailable:dst type="hostname">lager.cis.udel.edu</bandwidthavailable:dst
         </bandwidthavailable:hostPair>
      </bandwidthavailable:subject>
      <bandwidthavailable:parameters id="1957532">
         <bandwidthavailable:port>5001</bandwidthavailable:port>
         <bandwidthavailable:protocol>tcp</bandwidthavailable:protocol>
         <bandwidthavailable:windowsize>49.3 kbyte</bandwidthavailable:windowsize>
         <bandwidthavailable:units>gbits/sec</bandwidthavailable:units>
         <bandwidthavailable:arguments>-t 3 -i 1 -c</bandwidthavailable:arguments>
      </bandwidthavailable:parameters>
   </bandwidthavailable:metadata>
   <bandwidthavailable:data metadataId="7713496" id="14499519">
      <bandwidthavailable:time type="unix" value="1108528753">
         <bandwidthavailable:datum seqnum="0.0-1.0 sec" querynum="1" numbytes="242 mbytes" val
         <bandwidthavailable:datum seqnum="1.0-2.0 sec" querynum="1" numbytes="242 mbytes" val
         <bandwidthavailable:datum seqnum="2.0-3.0 sec" querynum="1" numbytes="231 mbytes" val
         <bandwidthavailable:datum seqnum="0.0-3.0 sec" querynum="1" numbytes="716 mbytes" val
      </bandwidthavailable:time>
   </bandwidthavailable:data>
</bandwidthavailable:response>
```

Next steps for developer's guide

- Finish examples
- Language+toolkit notes
- WS-RF integration plans (feature negotiation, etc.)

Next steps for adoption

- Get rough consensus on ping, iperf, traceroute
- Put these in canonical NM-WG namespace: publish doc in GGF?
- Get 2+ implementations to interoperate
- Iterate, using developer's guide as read/write resource

Using WSRF

- The Web Services Resource Framework specifies referring to stateful resources in the stateless WS environment
- Elements of state are referred to via an EndPoint Reference (EPR)
- The NMWG framework must expose state in a few areas:
  - Subsequent batches of data
  - Ongoing measurements
  - Stored measurement data