

Unified Network Information Services (UNIS) allow users to discover network services and capabilities

Marcos Portnoi, Martin Swany Computer and Information Sciences, University of Delaware, Newark, Delaware

UNIS is an Information Services Plane

- Distributed software architectures use the Information Services plane to discover "meta" information within the network.
- This information services plane facilitates discovery of network topology, location, and capabilities of network services.
- It is used in:
- Performance measurement infrastructures (perfSONAR);
- Dynamic circuit networks (ESnet SDN, Internet2 DCN, ION, GÉANT AutoBAHN);
- Experimental infrastructures (GENI).
- . Information Services Working Group (IS-WG) targets defining functionalities of the information services plane, and driving design and development.

Lookup Service conveys a distributed directory for services

- The Lookup Service (LS) within UNIS is a distributed directory, composed of levels.
- Local directories (hLS): point to local services (measurement tools, archives).
- Global directories (gLS) of local directories (all gLSs are synchronized).
- The hLS accepts registrations from services.
- hLSs combine registration information into single *summaries*.
- hLSs sends *summarized* data to gLSs.
- gLSs share information among other gLSs, and offer complete coverage.
- Clients consult hLS/gLS to discover services.
- · Clients and services will use well-known API for communication.

Service registration





Find the closest **Measurement Point**

- In a network measurement infrastructure (e.g., perfSONAR);
- · Measurement Points (MPs) are devices responsible for running tools to collect measurement data.
- One or more MPs are activated to conduct measurement.
- The MPs might lie totally *inside* the path between end points, or *outside* the path.
- Appropriate located MPs must be chosen to obtain "good" measurements;
- typically, the MP closer to the end point in topology.
- The Related MP service within UNIS facilitates discovery of MPs and their location in topology.





Normalizing diverse

topology schemas

	Location Network Object Relation name: <string> id: <uri> Service Node Port I I</uri></string>	Lifetime
class NodeProperties (models.Model):		Path
<pre>parent = models.ForeignKey(Node, related_name='properti</pre>	.es_bag')	0
class Meta:		Topology
app_label = 'topology'		
@classmethod		
def parse xml(cls, element, parent):		Domain
return parse_xml_unis_properties(cls, element, pact	ass Lifetime(models.Model):	type:
	<pre>start = models.DateTimeField()</pre>	
class PortAddresses (models.Model) :	end = models.DateTimeField()	Link
<pre>port = models.ForeignKey('Port')</pre>		
address = models.ForeignKey(Address, unique=True)	class Meta:	
class Port (NetworkObject) .	<pre>app_label = 'topology'</pre>	Network
capacity = models.CharField(max length=255, null=True		Note: Address
# This solves the reuse of Address without having to	defunicode(self):	aggregation
addresses = models.ManyToManyField(Address, through=	return self.begin + ' -> ' + self.end	
	Gclassmethod	
erass meed:	def parse xml(cls, element, parent):	
abb_raper - cohorodi	assert_xml_element(element, BASE_NAMESPACE, 'lifetime')	
# TODO: These methods should also check for relations		
<pre>def get_links(self):</pre>	# XXX: we only accept unix type for time elements, and ms for duration	
return Link.objects.filter(parent=self)	lifetime = Lifetime()	
@classmethod	<pre>start element = get unique xml child(element, BASE NAMESPACE, 'start')</pre>	
def parse xml(cls, element, parent);	if start element == None:	

- Distinct infrastructures may have their own schemas that represent topology elements.
 - TeraPaths
 - GENI's RSpec
 - perfSONAR's UNIS
- Our Periscope visualization tool is capable of normalizing distinct topology schemas into the UNIS schema.
- In perfSONAR, schemas based on NML-WG (Network Mark-up Language Working Group) definitions are used to specify XML messages format to interact with perfSONAR services.
- . The Django MVC is used to represent UNIS and measurement data within Periscope.
- Topological elements, from links and ports to domains and networks, are cleanly modeled and mapped to measurement data.

def parse_xml(cls, element, parent):

assert_xml_element(element, BASE_NAMESPACE, 'port check_valid_parent(parent, valid_classes=(Topolog error_msg="Port must have Topol

port = super(Port, cls).parse_xml(element, parent

if start_element == None:

raise UNISXMLParsingException, \

"lifetime element must contain start element!" if not start_element.hasAttribute('type'):

raise UNISXMLParsingException, \

"lifetime:start element must have type attribute!"

if start element.getAttribute('type') != "unix": raise UNISXMLParsingException, \

"lifetime:start element has unknown type (must be unix)" if not isinstance(start_element.firstChild, dom.Text):

raise UNISXMLParsingException, \ ...

• Periscope may also translate general network models to UNIS.

Periscope Model