Network flow analysis at SCinet

or

“Network flow analysis at ~~880Gb/s~~”

1.2Tb/s

Eric Dull

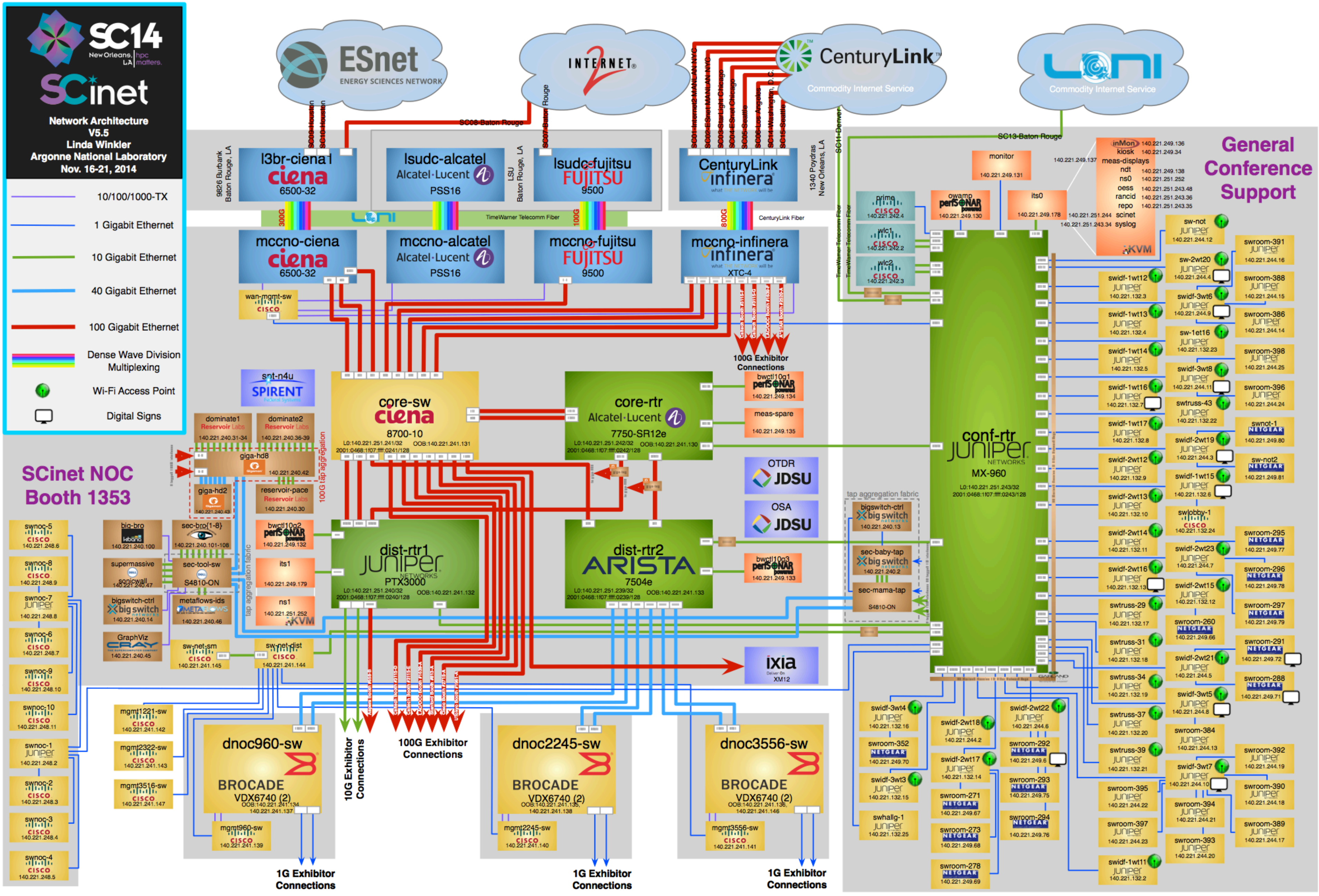
Steven P. Reinhardt

Agenda

- What is SCinet
- What analytic questions were we answering
- How we applied graphs to answer these questions
- Places to start exploring graphs

What is SCinet

- /17 publicly routed network
- Network supporting the SC technical conference and exhibit hall
- 10,972 devices on the network
- 1.2 Tb/s onto the show floor
- 296 Gb/s under BRO observation
- Set-up to teardown – about 10 days
- Rebuilt/reused every year



SC14 New Orleans, LA | hpc matters.

SCinet

Network Architecture
V5.5
Linda Winkler
Argonne National Laboratory
Nov. 16-21, 2014

- 10/100/1000-TX
- 1 Gigabit Ethernet
- 10 Gigabit Ethernet
- 40 Gigabit Ethernet
- 100 Gigabit Ethernet
- Dense Wave Division Multiplexing
- Wi-Fi Access Point
- Digital Signs

ESnet ENERGY SCIENCES NETWORK

INTERNET2

CenturyLink Commodity Internet Service

LONI Commodity Internet Service

SC09-Houston
SC10-Houston
SC08-Baton Rouge
SC07-Baton Rouge
SC01-Internet2 MANLAN NYC
SC02-ESnet MANLAN NYC
SC03-StarLight Chicago
SC04-ESnet Chicago
SC05-Seattle
SC06-Los Angeles
SC14-Washington, D.C.
SC15-Seattle
SC11-Denver
SC13-Baton Rouge

9826 Burbank
Baton Rouge, LA

l3br-ciena1 ciena 6500-32

lsudc-alcatel Alcatel·Lucent PSS16

LSU
Baton Rouge, LA

lsudc-fujitsu FUJITSU 9500

CenturyLink infinera what THE NETWORK will be

1340 Poydras
New Orleans, LA

300G LONI TimeWarner Telecomm Fiber 100G

800G CenturyLink Fiber

TimeWarner Telecomm Fiber
TimeWarner Telecomm Fiber

mccno-ciena ciena 6500-32

mccno-alcatel Alcatel·Lucent PSS16

mccno-fujitsu FUJITSU 9500

mccno-infinera infinera what THE NETWORK will be XTC-4

wan-mgmt-sw CISCO

Caltech Booth #3115-F
Caltech Booth #3115-G
LAC/OCC Booth #1639-F
A*STAR Booth #2520-A

100G Exhibitor Connections

spt-n4u SPIRENT Federal Systems

core-sw ciena 8700-10
L0:140.221.251.241/32 OOB:140.221.241.131
2001:0468:1f07:ffff:0241/128

core-rtr Alcatel·Lucent 7750-SR12e
L0:140.221.251.242/32 OOB:140.221.241.130
2001:0468:1f07:ffff:0242/128

bwctl10g1 perfSONAR powered 140.221.249.134

meas-spare 140.221.249.135

OTDR JDSU

OSA JDSU

dominate1 Reservoir Labs 140.221.240.31-34

dominate2 Reservoir Labs 140.221.240.36-39

giga-hd8 Gigamon 140.221.240.42

giga-hd2 Gigamon 140.221.240.43

100G tap aggregation

reservoir-pace Reservoir Labs 140.221.240.30

SCinet NOC
Booth 1353

big-bro kibana 140.221.240.100

sec-bro{1-8} 140.221.240.101-108

supermassive DELL sonicwall 140.221.240.47

sec-tool-sw DELL S4810-ON

bigswitch-ctrl big switch networks 140.221.240.14

metaflows-ids METAFLOWS 140.221.240.46

GraphViz CRAY THE SUPERCOMPUTER COMPANY 140.221.240.45

tap aggregation fabric

bwctl10g2 perfSONAR powered 140.221.249.132

its1 140.221.249.179

ns1 140.221.251.252 KVM

sw-net-sm CISCO 140.221.241.145

sw-net-dist CISCO 140.221.241.144

dist-rtr1 JUNIPER NETWORKS PTX3000
L0:140.221.251.240/32 OOB:140.221.241.132
2001:0468:1f07:ffff:0240/128

dist-rtr2 ARISTA 7504e
L0:140.221.251.239/32 OOB:140.221.241.133
2001:0468:1f07:ffff:0239/128

bwctl10g3 perfSONAR powered 140.221.249.133

ixia XM12

swnoc-5 CISCO 140.221.248.6
swnoc-8 CISCO 140.221.248.9
swnoc-7 JUNIPER 140.221.248.8
swnoc-6 CISCO 140.221.248.7
swnoc-9 CISCO 140.221.248.10
swnoc-10 CISCO 140.221.248.11
swnoc-1 JUNIPER 140.221.248.2
swnoc-2 CISCO 140.221.248.3
swnoc-3 CISCO 140.221.248.4
swnoc-4 CISCO 140.221.248.5

mgmt1221-sw CISCO 140.221.241.142
mgmt2322-sw CISCO 140.221.241.143
mgmt3516-sw CISCO 140.221.241.147

dnoc960-sw BROCADE VDX6740 (2)
OOB:140.221.241.134
140.221.241.137

mgmt960-sw CISCO 140.221.241.139

1G Exhibitor Connections

10G Exhibitor Connections

Aspera Booth #658-B
Caltech Booth #3115-D
Caltech Booth #3115-E
LAC/OCC Booth #1639-A
Stanford Booth #813-A
Cisco Booth #2715-A
UTokyo Booth #1951-A

100G Exhibitor Connections

dnoc2245-sw BROCADE VDX6740 (2)
OOB:140.221.241.135,
140.221.241.138

mgmt2245-sw CISCO 140.221.241.140

1G Exhibitor Connections

dnoc3556-sw BROCADE VDX6740 (2)
OOB:140.221.241.136,
140.221.241.146

mgmt3556-sw CISCO 140.221.241.141

1G Exhibitor Connections

prime CISCO 140.221.242.4
wlc1 CISCO 140.221.242.2
wlc2 CISCO 140.221.242.3

owamp perfSONAR powered 140.221.249.130

monitor 140.221.249.131

its0 140.221.249.178

	inMon	140.221.249.136
	kiosk	140.221.249.34
140.221.249.137	meas-displays	
	ndt	140.221.249.138
	ns0	140.221.251.252
	oess	140.221.251.243.48
	rancid	140.221.251.243.36
	repo	140.221.251.243.35
140.221.251.244	scinet	
140.221.251.243.34	syslog	

KVM

General Conference Support

conf-rtr JUNIPER NETWORKS MX-960
L0:140.221.251.243/32
2001:0468:1f07:ffff:0243/128

tap aggregation fabric

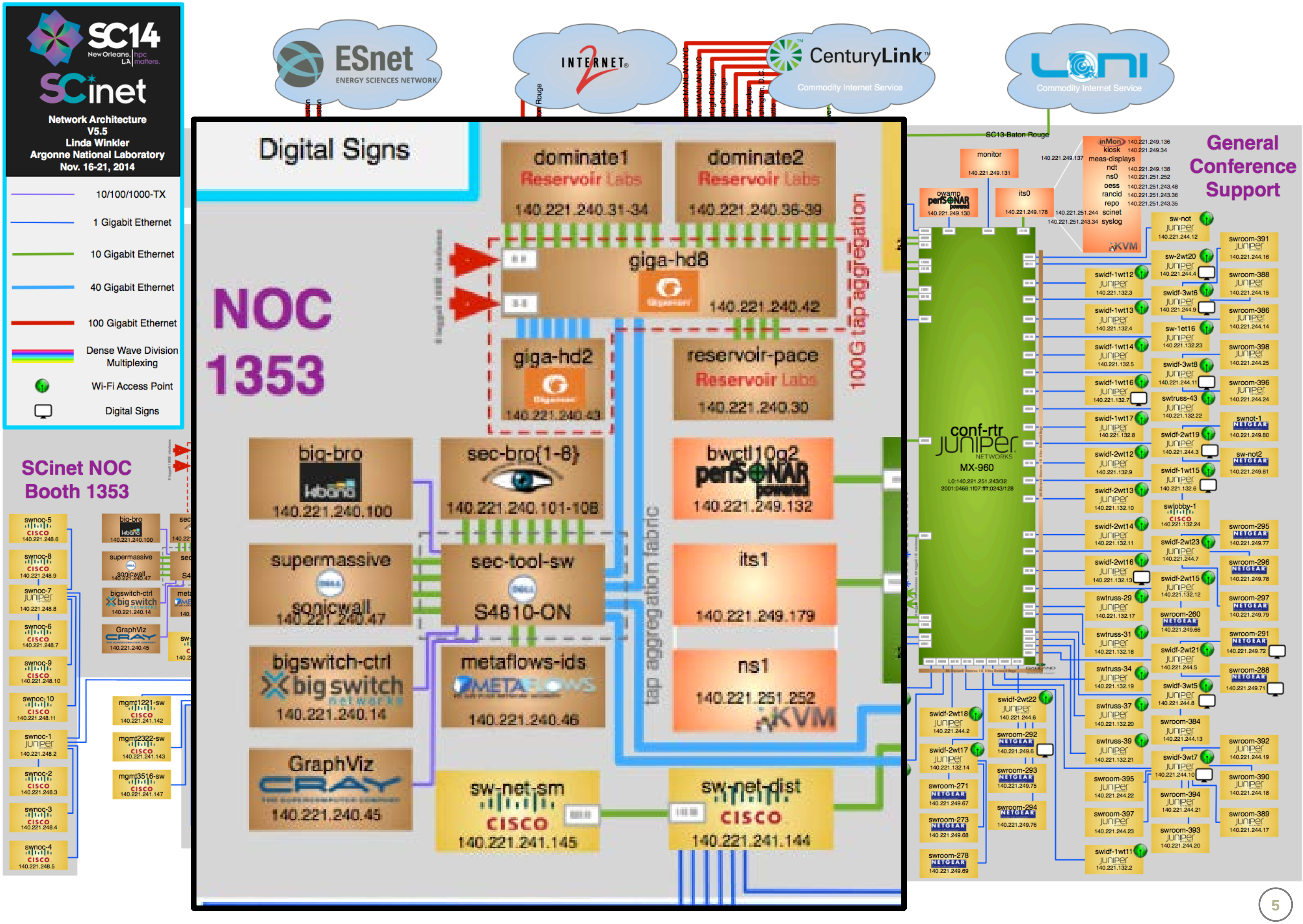
bigswitch-ctrl big switch networks 140.221.240.13

sec-baby-tap big switch networks 140.221.240.2

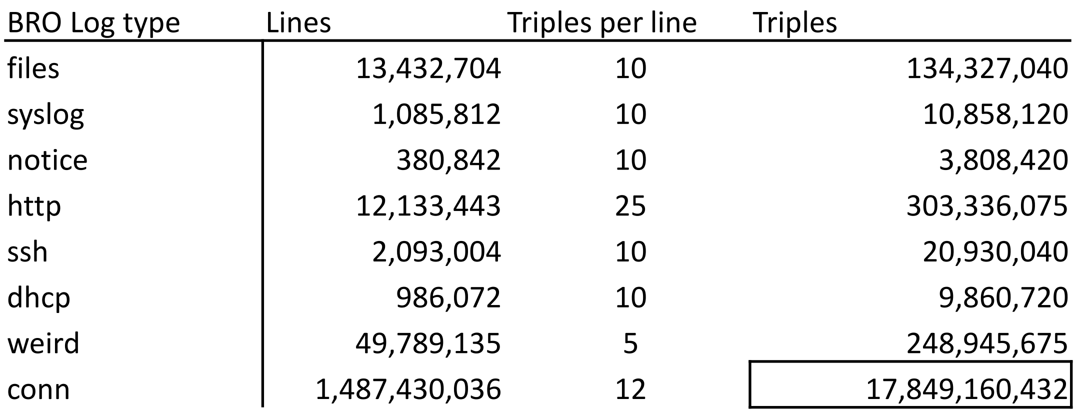
sec-mama-tap S4810-ON

GARLAND

sw-not JUNIPER 140.221.244.12
swroom-391 JUNIPER 140.221.244.16
sw-2wt20 JUNIPER 140.221.244.4
swidf-1wt12 JUNIPER 140.221.132.3
swroom-388 JUNIPER 140.221.244.15
swidf-3wt6 JUNIPER 140.221.244.9
swidf-1wt13 JUNIPER 140.221.132.4
swroom-386 JUNIPER 140.221.244.14
sw-1et16 JUNIPER 140.221.132.23
swidf-1wt14 JUNIPER 140.221.132.5
swroom-398 JUNIPER 140.221.244.25
swidf-3wt8 JUNIPER 140.221.244.11
swidf-1wt16 JUNIPER 140.221.132.7
swroom-396 JUNIPER 140.221.244.24
swtruss-43 JUNIPER 140.221.132.22
swidf-1wt17 JUNIPER 140.221.132.8
swnot-1 NETGEAR 140.221.249.80
swidf-2wt19 JUNIPER 140.221.244.3
swidf-2wt12 JUNIPER 140.221.132.9
sw-not2 NETGEAR 140.221.249.81
swidf-1wt15 JUNIPER 140.221.132.6
swidf-2wt13 JUNIPER 140.221.132.10
swlobby-1 CISCO 140.221.132.24
swidf-2wt14 JUNIPER 140.221.132.11
swroom-295 NETGEAR 140.221.249.77
swidf-2wt23 JUNIPER 140.221.244.7
swidf-2wt16 JUNIPER 140.221.132.13
swroom-296 NETGEAR 140.221.249.78
swidf-2wt15 JUNIPER 140.221.132.12
swtruss-29 JUNIPER 140.221.132.17
swroom-297 NETGEAR 140.221.249.79
swroom-260 NETGEAR 140.221.249.66
swtruss-31 JUNIPER 140.221.132.18
swroom-291 NETGEAR 140.221.249.72
swidf-2wt21 JUNIPER 140.221.244.5
swtruss-34 JUNIPER 140.221.132.19
swroom-288 NETGEAR 140.221.249.71
swidf-3wt5 JUNIPER 140.221.244.8
swidf-3wt4 JUNIPER 140.221.132.16
swidf-2wt22 JUNIPER 140.221.244.6
swidf-2wt18 JUNIPER 140.221.244.2
swtruss-37 JUNIPER 140.221.132.20
swroom-384 JUNIPER 140.221.244.13
swroom-352 NETGEAR 140.221.249.70
swroom-292 NETGEAR 140.221.249.6
swidf-2wt17 JUNIPER 140.221.132.14
swtruss-39 JUNIPER 140.221.132.21
swroom-392 JUNIPER 140.221.244.19
swidf-3wt7 JUNIPER 140.221.244.10
swidf-3wt3 JUNIPER 140.221.132.15
swroom-293 NETGEAR 140.221.249.75
swroom-271 NETGEAR 140.221.249.67
swroom-395 JUNIPER 140.221.244.22
swroom-390 JUNIPER 140.221.244.18
swroom-394 JUNIPER 140.221.244.21
swhallg-1 JUNIPER 140.221.132.25
swroom-294 NETGEAR 140.221.249.76
swroom-273 NETGEAR 140.221.249.68
swroom-397 JUNIPER 140.221.244.23
swroom-389 JUNIPER 140.221.244.17
swroom-393 JUNIPER 140.221.244.20
swidf-1wt11 JUNIPER 140.221.132.2
swroom-278 NETGEAR 140.221.249.69


SC14
New Orleans, LA
hpc matters.
SCinet
Network Architecture
V5.5
Linda Winkler
Argonne National Laboratory
Nov. 16-21, 2014
10/100/1000-TX
1 Gigabit Ethernet
10 Gigabit Ethernet
40 Gigabit Ethernet
100 Gigabit Ethernet
Dense Wave Division Multiplexing
Wi-Fi Access Point
Digital Signs
ESnet
ENERGY SCIENCES NETWORK
INTERNET2
CenturyLink
Commodity Internet Service
LONI
Commodity Internet Service
Digital Signs
NOC
1353
dominate1
Reservoir Labs
140.221.240.31-34
dominate2
Reservoir Labs
140.221.240.36-39
giga-hd8
Gigamon
140.221.240.42
giga-hd2
Gigamon
140.221.240.43
reservoir-pace
Reservoir Labs
140.221.240.30
100G tap aggregation
big-bro
kibana
140.221.240.100
sec-bro{1-8}
140.221.240.101-108
bwctl10g2
perfSONAR powered
140.221.249.132
supermassive
DELL
sonicwall
140.221.240.47
sec-tool-sw
DELL
S4810-ON
its1
140.221.249.179
tap aggregation fabric
bigswitch-ctrl
big switch networks
140.221.240.14
metaflows-ids
METAFLOWS
140.221.240.46
ns1
140.221.251.252
KVM
GraphViz
CRAY
THE SUPERCOMPUTER COMPANY
140.221.240.45
sw-net-sm
CISCO
140.221.241.145
sw-net-dist
CISCO
140.221.241.144
SC13-Baton Rouge
monitor
140.221.249.131
owamp
perfSONAR powered
140.221.249.130
its0
140.221.249.178
inMon 140.221.249.136
kiosk 140.221.249.34
140.221.249.137 meas-displays
ndt 140.221.249.138
ns0 140.221.251.252
oess 140.221.251.243.48
rancid 140.221.251.243.36
repo 140.221.251.243.35
140.221.251.244 scinet
140.221.251.243.34 syslog
KVM
General Conference Support
conf-rtr
JUNIPER NETWORKS
MX-960
L0:140.221.251.243/32
2001:0468:1f07:ffff:0243/128
GARLAND
sw-not JUNIPER 140.221.244.12
sw-2wt20 JUNIPER 140.221.244.4
swidf-1wt12 JUNIPER 140.221.132.3
swidf-3wt6 JUNIPER 140.221.244.9
swidf-1wt13 JUNIPER 140.221.132.4
sw-1et16 JUNIPER 140.221.132.23
swidf-1wt14 JUNIPER 140.221.132.5
swidf-3wt8 JUNIPER 140.221.244.11
swidf-1wt16 JUNIPER 140.221.132.7
swtruss-43 JUNIPER 140.221.132.22
swidf-1wt17 JUNIPER 140.221.132.8
swidf-2wt19 JUNIPER 140.221.244.3
swidf-2wt12 JUNIPER 140.221.132.9
swidf-1wt15 JUNIPER 140.221.132.6
swidf-2wt13 JUNIPER 140.221.132.10
swlobby-1 CISCO 140.221.132.24
swidf-2wt14 JUNIPER 140.221.132.11
swidf-2wt23 JUNIPER 140.221.244.7
swidf-2wt16 JUNIPER 140.221.132.13
swidf-2wt15 JUNIPER 140.221.132.12
swtruss-29 JUNIPER 140.221.132.17
swroom-260 NETGEAR 140.221.249.66
swtruss-31 JUNIPER 140.221.132.18
swidf-2wt21 JUNIPER 140.221.244.5
swtruss-34 JUNIPER 140.221.132.19
swidf-3wt5 JUNIPER 140.221.244.8
swtruss-37 JUNIPER 140.221.132.20
swroom-384 JUNIPER 140.221.244.13
swtruss-39 JUNIPER 140.221.132.21
swidf-3wt7 JUNIPER 140.221.244.10
swroom-395 JUNIPER 140.221.244.22
swroom-394 JUNIPER 140.221.244.21
swroom-397 JUNIPER 140.221.244.23
swroom-393 JUNIPER 140.221.244.20
swidf-1wt11 JUNIPER 140.221.132.2
swroom-391 JUNIPER 140.221.244.16
swroom-388 JUNIPER 140.221.244.15
swroom-386 JUNIPER 140.221.244.14
swroom-398 JUNIPER 140.221.244.25
swroom-396 JUNIPER 140.221.244.24
swnot-1 NETGEAR 140.221.249.80
sw-not2 NETGEAR 140.221.249.81
swroom-295 NETGEAR 140.221.249.77
swroom-296 NETGEAR 140.221.249.78
swroom-297 NETGEAR 140.221.249.79
swroom-291 NETGEAR 140.221.249.72
swroom-288 NETGEAR 140.221.249.71
swroom-392 JUNIPER 140.221.244.19
swroom-390 JUNIPER 140.221.244.18
swroom-389 JUNIPER 140.221.244.17
swidf-2wt18 JUNIPER 140.221.244.2
swidf-2wt22 JUNIPER 140.221.244.6
swidf-2wt17 JUNIPER 140.221.132.14
swroom-292 NETGEAR 140.221.249.6
swroom-271 NETGEAR 140.221.249.67
swroom-293 NETGEAR 140.221.249.75
swroom-273 NETGEAR 140.221.249.68
swroom-294 NETGEAR 140.221.249.76
swroom-278 NETGEAR 140.221.249.69
SCinet NOC
Booth 1353
swnoc-5 CISCO 140.221.248.6
swnoc-8 CISCO 140.221.248.9
swnoc-7 JUNIPER 140.221.248.8
swnoc-6 CISCO 140.221.248.7
swnoc-9 CISCO 140.221.248.10
swnoc-10 CISCO 140.221.248.11
swnoc-1 JUNIPER 140.221.248.2
swnoc-2 CISCO 140.221.248.3
swnoc-3 CISCO 140.221.248.4
swnoc-4 CISCO 140.221.248.5
big-bro kibana 140.221.240.100
supermassive sonicwall 140.221.240.47
bigswitch-ctrl big switch networks 140.221.240.14
GraphViz CRAY 140.221.240.45
mgmt1221-sw CISCO 140.221.241.142
mgmt2322-sw CISCO 140.221.241.143
mgmt3516-sw CISCO 140.221.241.147
Rouge
Light Chicago
Washington, D.C.

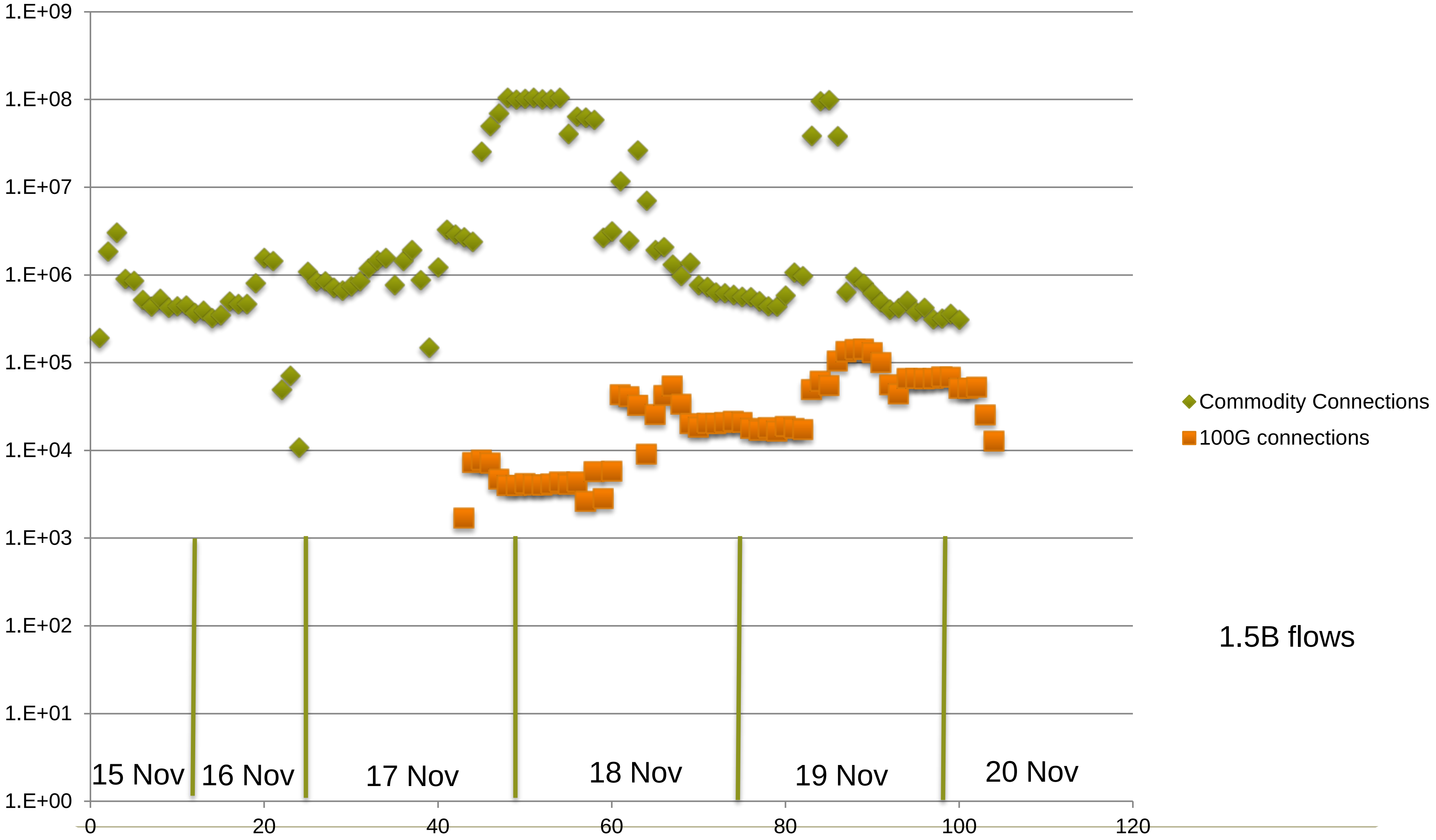
Total generated triples



BRO Log type	Lines	Triples per line	Triples
files	13,432,704	10	134,327,040
syslog	1,085,812	10	10,858,120
notice	380,842	10	3,808,420
http	12,133,443	25	303,336,075
ssh	2,093,004	10	20,930,040
dhcp	986,072	10	9,860,720
weird	49,789,135	5	248,945,675
conn	1,487,430,036	12	17,849,160,432

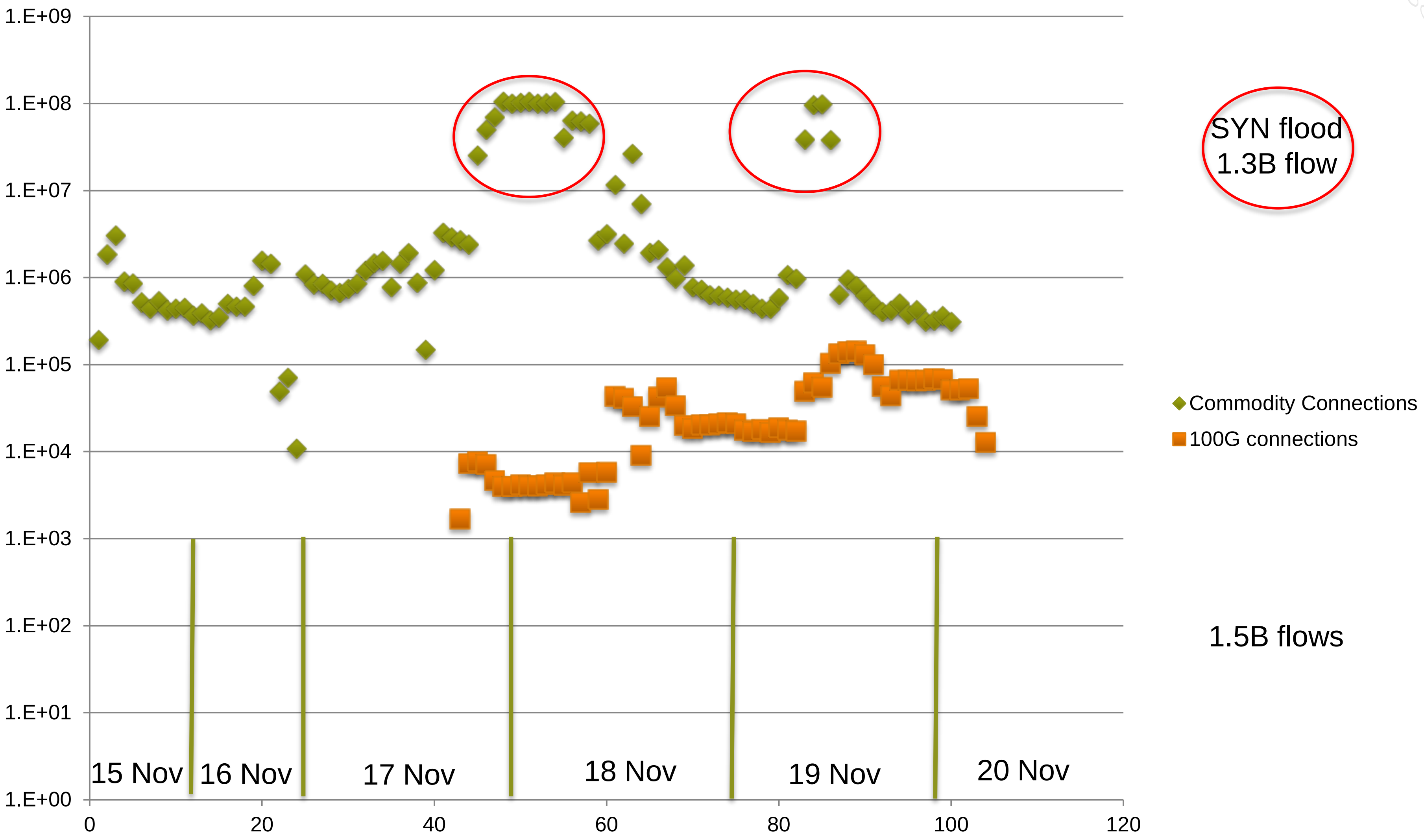
- **RDF generated on Discover using Python scripts written at SC13**
- **Used the OCOG netflow RDF format for the first time in analysis!**

Flow counts



1.5B flows

Flow counts



SYN flood
1.3B flow

1.5B flows

Analytic charge

- Find outbound scanning or attacking
- Help identify groups of infected systems from C2 and download activities
- “Perform the next hop” of analysis. Use graphs to ease automated analysis
- Find new DNS and DHCP servers as they appear on the network

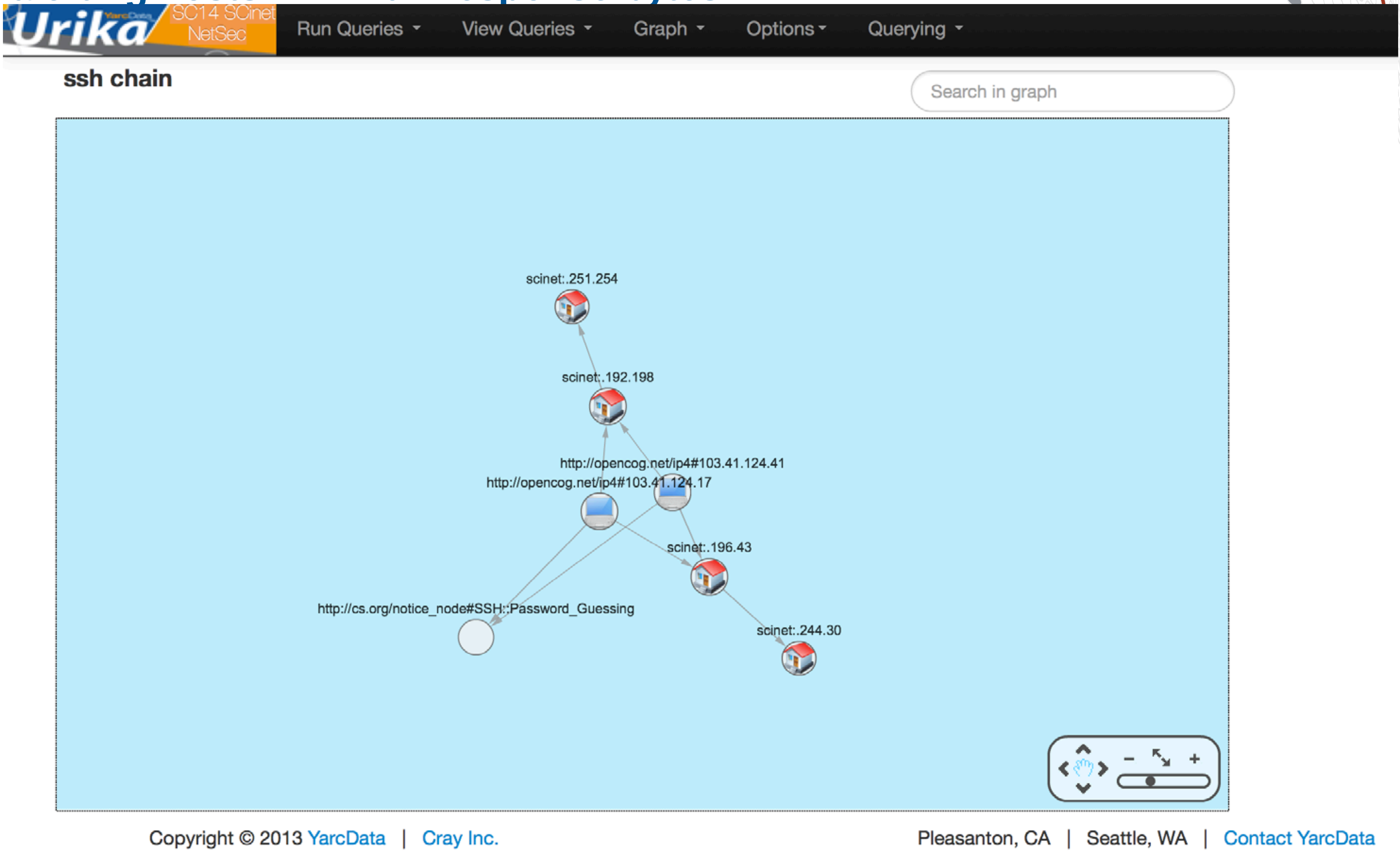
Applicable graph operations

- **Search – “Find SSH connection networks”**
  - IP address based search
  - Port and volume based search
  - IDS alert based search
  - 1,2, or 3 hop

- **Jaccard Scoring – “which is the likely C2 channel for IPs downloading from this port?”**
- **Betweeness Centrality – “Which IP address in this network should be considered first when cleaning up an infection network?”**

Search example: SSH chain

alerting hosts– X > 10K response bytes



ssh chain

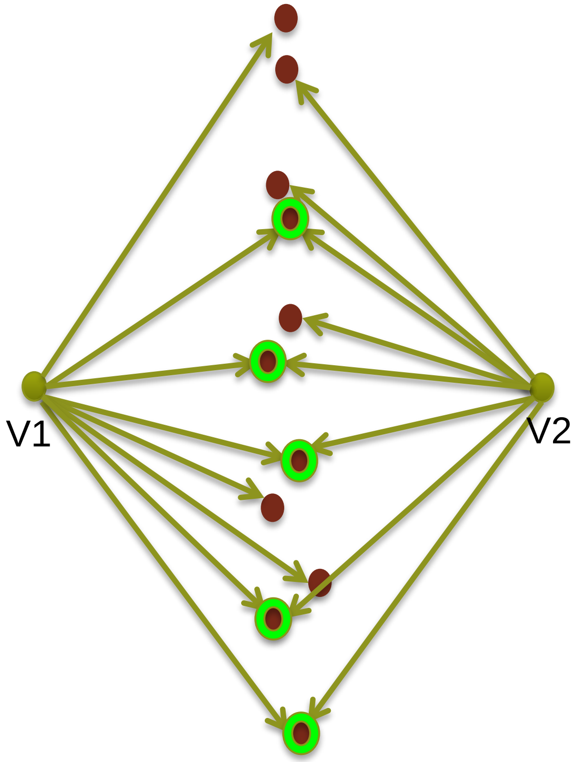
Pleasanton, CA | Seattle, WA | Contact YarcData

Search example: SSH connection chain SPARQL query

```
CONSTRUCT{
  ?ap_addr <http://cs.org/p/hasNoticeNote>  <http://cs.org/notice_node#SSH::Password_Guessing>.
  ?ap_addr <urn:p/hasSSH> ?internal_addr.
  ?internal_addr <urn:p/hasSSH> ?a_addr.
}
{
 SELECT distinct ?internal_addr ?ap_addr ?a_addr
 WHERE
 {
  ?uid4 <http://opencog.net/p/destinationAddress> ?a_addr.
  ?uid4 <http://opencog.net/p/sourceAddress> ?internal_addr.
  ?uid4 <http://opencog.net/p/hasProtocol> <http://opencog.net/proto#tcp>.
  ?uid4 <http://opencog.net/p/destinationPort> <http://opencog.net/port#22> .
  ?uid4 <http://cs.org/p/hasRespBytes> ?rbytes1.
  FILTER(?rbytes1 > 10000)
  {
   SELECT distinct ?internal_addr ?ap_addr
   WHERE
   {
    ?uid <http://cs.org/p/hasNoticeNote>  <http://cs.org/notice_node#SSH::Password_Guessing>.
    ?uid <http://cs.org/p/hasNoticeMsg> ?msg.
    ?uid <http://cs.org/p/hasOrigAddr> ?ap_addr.
    ?uid4 <http://opencog.net/p/sourceAddress> ?ap_addr.
    ?uid4 <http://opencog.net/p/destinationAddress> ?internal_addr.
    ?uid4 <http://opencog.net/p/destinationPort> <http://opencog.net/port#22>.
    ?uid4 <http://cs.org/p/hasRespBytes> ?rbytes1.
    FILTER(?rbytes1 > 20900)
   }
   LIMIT 1000
  }
 }
}
```

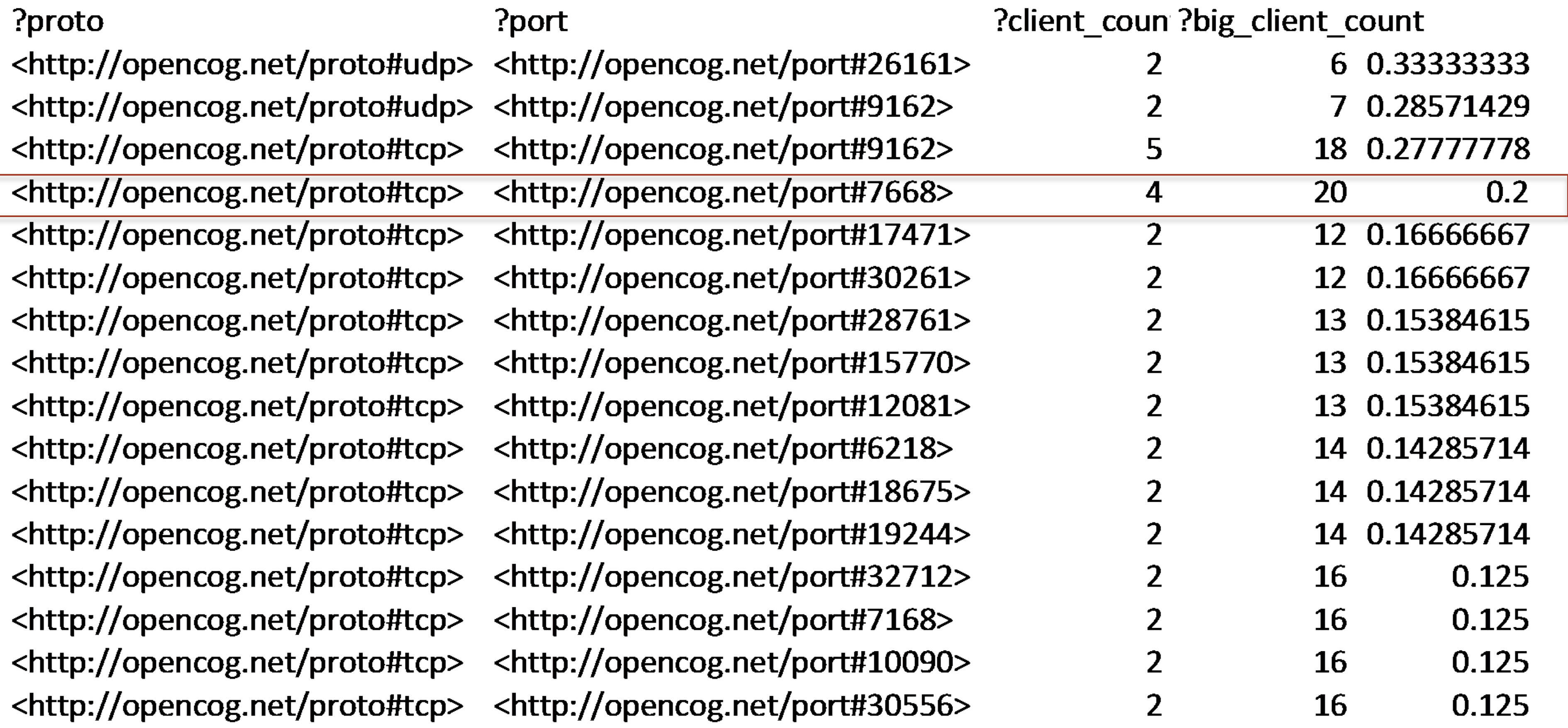
Jaccard example: math and SPARQL implementation

```
SELECT ?proto ?port ?client_count ?big_client_count
WHERE
{
 {
 SELECT ?proto ?port (count(distinct ?ap_addr) as ?big_client_count)
 WHERE
 {
  ?uid3 <http://opencog.net/p/sourceAddress> ?ap_addr.
  ?uid3 <http://opencog.net/p/destinationAddress> ?dest_addr2 .
  ?uid3 <http://opencog.net/p/destinationPort> ?port .
  ?uid3 <http://opencog.net/p/hasProtocol> ?proto .
  ?uid3 <http://cs.org/p/hasRespBytes> ?rbytes2.
 }
 GROUP BY ?proto ?port
 }
 {
 SELECT ?proto ?port (count(distinct ?ap_addr) as ?client_count)
 WHERE
 {
  ?uid3 <http://opencog.net/p/sourceAddress> ?ap_addr.
  ?uid3 <http://opencog.net/p/destinationAddress> ?dest_addr2 .
  ?uid3 <http://opencog.net/p/destinationPort> ?port .
  ?uid3 <http://opencog.net/p/hasProtocol> ?proto .
  ?uid3 <http://cs.org/p/hasRespBytes> ?rbytes2.
  FILTER(?rbytes2 > 0)
  ?uid4 <http://opencog.net/p/sourceAddress> ?ap_addr.
  ?uid4 <http://opencog.net/p/destinationAddress> ?dest_addr .
  ?uid4 <http://opencog.net/p/destinationPort> <http://opencog.net/port#9162>.
  ?uid4 <http://cs.org/p/hasRespBytes> ?rbytes1.
  FILTER(?rbytes1 > 0)
 }
 GROUP BY ?proto ?port
 HAVING (?client_count > 1)
 }
}
ORDER BY DESC(?client_count)
```



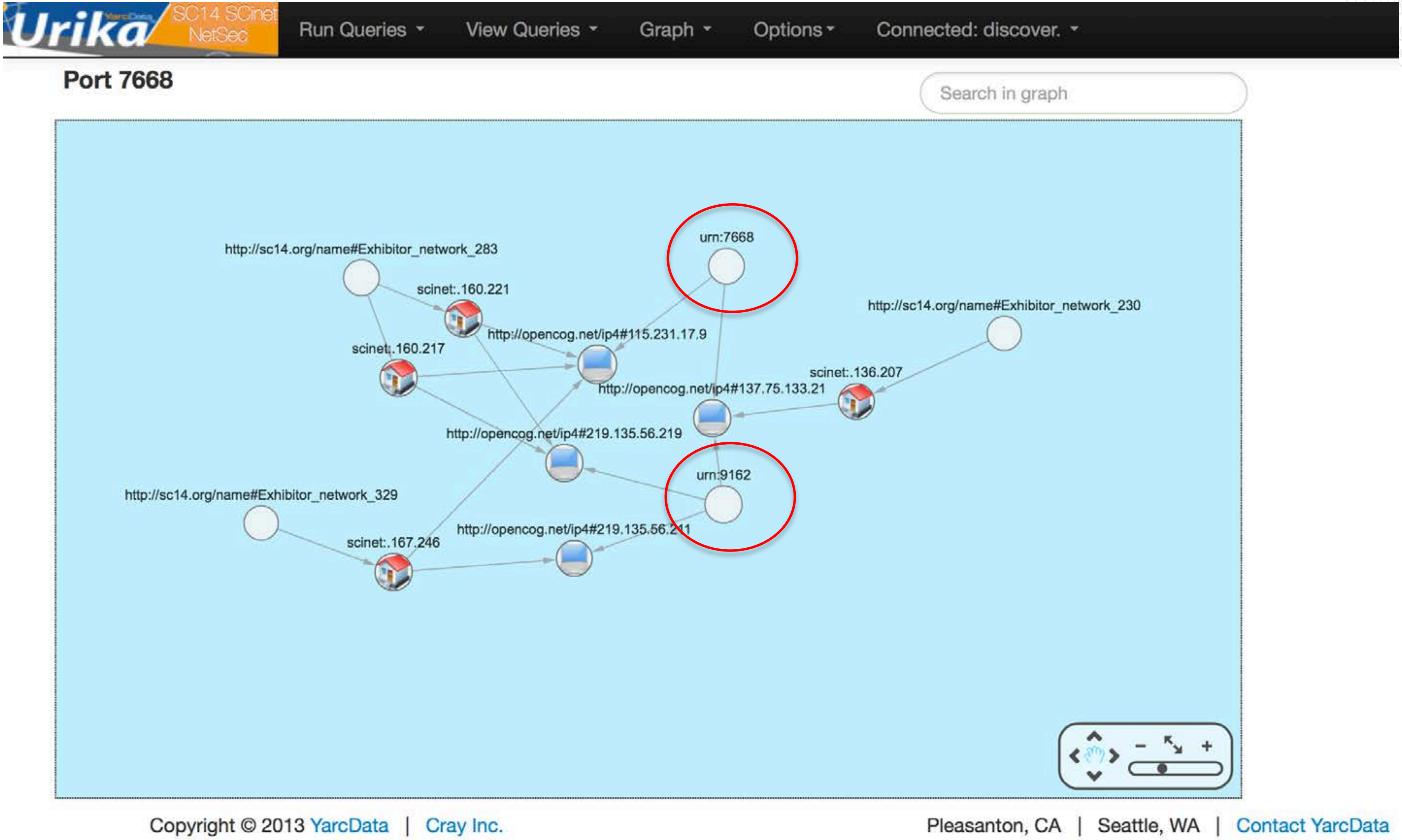
Definition: $|V1 \cap V2| \; / \; |V1 \cup V2|$

Jaccard example: SSH password forced C2 channel candidates



?proto	?port	?client_coun	?big_client_count	
<http://opencog.net/proto#udp>	<http://opencog.net/port#26161>	2	6	0.33333333
<http://opencog.net/proto#udp>	<http://opencog.net/port#9162>	2	7	0.28571429
<http://opencog.net/proto#tcp>	<http://opencog.net/port#9162>	5	18	0.27777778
<http://opencog.net/proto#tcp>	<http://opencog.net/port#7668>	4	20	0.2
<http://opencog.net/proto#tcp>	<http://opencog.net/port#17471>	2	12	0.16666667
<http://opencog.net/proto#tcp>	<http://opencog.net/port#30261>	2	12	0.16666667
<http://opencog.net/proto#tcp>	<http://opencog.net/port#28761>	2	13	0.15384615
<http://opencog.net/proto#tcp>	<http://opencog.net/port#15770>	2	13	0.15384615
<http://opencog.net/proto#tcp>	<http://opencog.net/port#12081>	2	13	0.15384615
<http://opencog.net/proto#tcp>	<http://opencog.net/port#6218>	2	14	0.14285714
<http://opencog.net/proto#tcp>	<http://opencog.net/port#18675>	2	14	0.14285714
<http://opencog.net/proto#tcp>	<http://opencog.net/port#19244>	2	14	0.14285714
<http://opencog.net/proto#tcp>	<http://opencog.net/port#32712>	2	16	0.125
<http://opencog.net/proto#tcp>	<http://opencog.net/port#7168>	2	16	0.125
<http://opencog.net/proto#tcp>	<http://opencog.net/port#10090>	2	16	0.125
<http://opencog.net/proto#tcp>	<http://opencog.net/port#30556>	2	16	0.125

Jaccard example: ports 7668 and 9162 visualization


Urika
YarcData
SC14 SCinet NetSec
Run Queries
View Queries
Graph
Options
Connected: discover.
Search in graph
urn:7668
http://sc14.org/name#Exhibitor_network_283
scinet:.160.221
http://opencog.net/ip4#115.231.17.9
http://sc14.org/name#Exhibitor_network_230
scinet:.160.217
scinet:.136.207
http://opencog.net/ip4#137.75.133.21
http://opencog.net/ip4#219.135.56.219
urn:9162
http://sc14.org/name#Exhibitor_network_329
http://opencog.net/ip4#219.135.56.211
scinet:.167.246

Port 7668

Pleasanton, CA | Seattle, WA | Contact YarcData

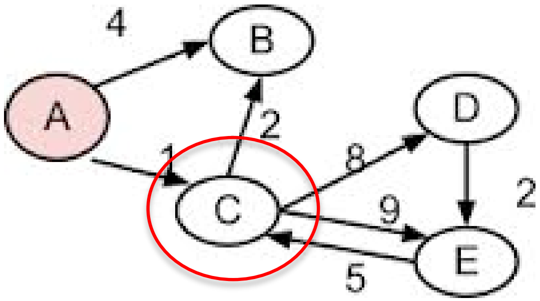
Betweenness example: pseudo-math and SPARQL implementation

```
SELECT ?vertices ?scores
WHERE
{
CONSTRUCT{ #<urn:SSH_forcer> <urn:/p/HasMember> ?src_addr.
?src_addr <urn:p/hasSSH> ?dest_addr.
?dest_addr <urn:p/hasSSH> ?dest_addr2
}
WHERE
{
SELECT distinct ?src_addr ?dest_addr ?dest_addr2
WHERE
{ ?booth2 a <http://sc14.org/class#SCinet_subnet> .
  ?booth2 <http://opencog.net/hasMember> ?dest_addr .
  ?uid3 <http://opencog.net/p/sourceAddress> ?dest_addr .
  ?uid3 <http://opencog.net/p/destinationAddress> ?dest_addr2 .
  ?uid3 <http://opencog.net/p/hasProtocol> <http://opencog.net/proto#tcp>.
  ?uid3 <http://opencog.net/p/destinationPort> <http://opencog.net/port#22> .
  ?uid3 <http://opencog.net/p/start> ?start_time2.
  ?uid3 <http://cs.org/p/hasRespBytes> ?rbytes2.
  FILTER (?rbytes2 > 12000)
  FILTER (?start_time < ?start_time2)
  OPTIONAL

{
SELECT ?src_addr ?dest_addr ?start_time
{ #?src_addr a <http://sc14.org/class#SSHattacker>.
  ?uid <http://cs.org/p/hasNoticeNote>  <http://cs.org/notice_node#SSH::Password_Guessing>.
  ?uid <http://cs.org/p/hasNoticeMsg> ?msg.
  ?uid <http://cs.org/p/hasOrigAddr> ?src_addr.

  ?uid3 <http://opencog.net/p/sourceAddress> ?src_addr .
  ?uid3 <http://opencog.net/p/destinationAddress> ?dest_addr .
  ?uid3 <http://opencog.net/p/hasProtocol> <http://opencog.net/proto#tcp>.
  ?uid3 <http://opencog.net/p/destinationPort> <http://opencog.net/port#22> .
  ?uid3 <http://opencog.net/p/start> ?start_time.
  ?uid3 <http://cs.org/p/hasRespBytes> ?rbytes2.
  FILTER(?rbytes2 > 12000)
}
LIMIT 500
}
}
}INVOKE yd:graphAlgorithm.betweenness_centrality (.5,1)
 PRODUCING ?vertices ?scores
}
ORDER BY DESC(?scores)
```

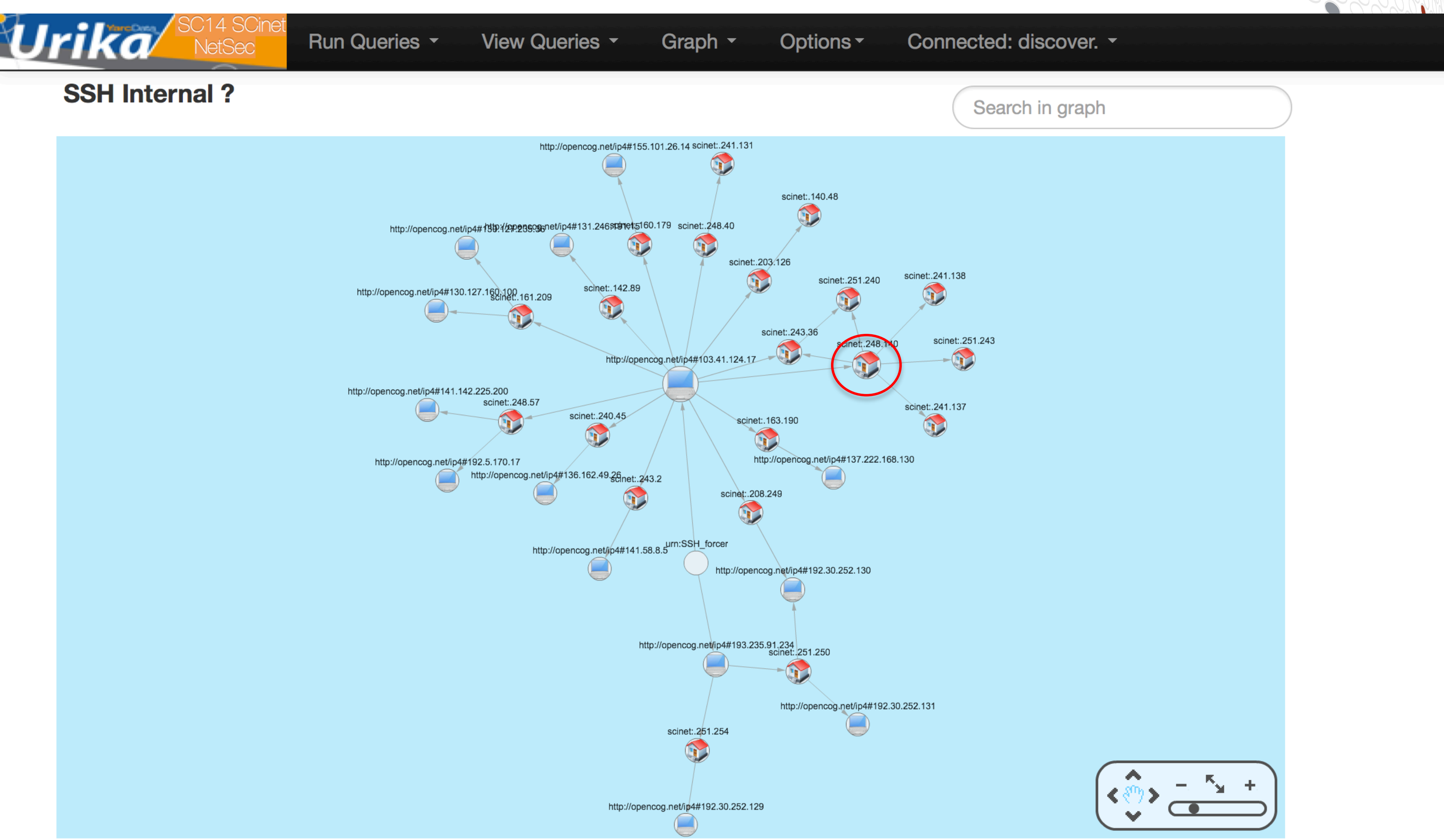
How to compute Betweeness centrality (All-pairs shortest-path)

- From every node, compute the shortest path(s) to every other node
- For every node, count the number of shortest paths that go through it
- For every edge, count the number of shortest paths that go through it
- Divide the shortest path counts by the total number of shortest paths to generate centrality scores

The nodes and edges with the highest centrality scores are most central



Betweenness example: SSH / Internal / ?


Urika
YarcData
SC14 SCinet NetSec
Run Queries
View Queries
Graph
Options
Connected: discover.
SSH Internal ?
Search in graph
http://opencog.net/ip4#155.101.26.14
scinet:.241.131
scinet:.140.48
http://opencog.net/ip4#130.127.235.36
http://opencog.net/ip4#131.246.151.15
scinet:.160.179
scinet:.248.40
scinet:.203.126
scinet:.251.240
scinet:.241.138
http://opencog.net/ip4#130.127.160.100
scinet:.161.209
scinet:.142.89
scinet:.243.36
scinet:.248.140
scinet:.251.243
http://opencog.net/ip4#103.41.124.17
http://opencog.net/ip4#141.142.225.200
scinet:.248.57
scinet:.240.45
scinet:.241.137
scinet:.163.190
http://opencog.net/ip4#192.5.170.17
http://opencog.net/ip4#136.162.49.26
scinet:.243.2
http://opencog.net/ip4#137.222.168.130
scinet:.208.249
http://opencog.net/ip4#141.58.8.5
urn:SSH_forcer
http://opencog.net/ip4#192.30.252.130
http://opencog.net/ip4#193.235.91.234
scinet:.251.250
http://opencog.net/ip4#192.30.252.131
scinet:.251.254
http://opencog.net/ip4#192.30.252.129
−
+

Pleasanton, CA | Seattle, WA | Contact YarcData

Betweenness example: Centrality results

```
?vertices                               ?scores
http://opencog.net/ip4#A.B.248.140      1^^<http://www.w3.org/2001/XMLSchema#double>
<http://opencog.net/ip4#A.B.248.57>     0.5217391304347826^^<http://www.w3.org/2001/XMLSchema#double>
<http://opencog.net/ip4#A.B.161.209>    0.5217391304347826^^<http://www.w3.org/2001/XMLSchema#double>
<http://opencog.net/ip4#A.B.160.131>    0.2608695652173913^^<http://www.w3.org/2001/XMLSchema#double>
<http://opencog.net/ip4#A.B.243.2>      0.2608695652173913^^<http://www.w3.org/2001/XMLSchema#double>
<http://opencog.net/ip4#A.B.203.126>    0.2608695652173913^^<http://www.w3.org/2001/XMLSchema#double>
<http://opencog.net/ip4#A.B.248.40>     0.2608695652173913^^<http://www.w3.org/2001/XMLSchema#double>
<http://opencog.net/ip4#A.B.163.190>    0.2608695652173913^^<http://www.w3.org/2001/XMLSchema#double>
<http://opencog.net/ip4#A.B.240.45>     0.2608695652173913^^<http://www.w3.org/2001/XMLSchema#double>
<http://opencog.net/ip4#A.B.142.81>     0.2608695652173913^^<http://www.w3.org/2001/XMLSchema#double>
<http://opencog.net/ip4#A.B.243.36>     0.04347826086956522^^<http://www.w3.org/2001/XMLSchema#double>
```

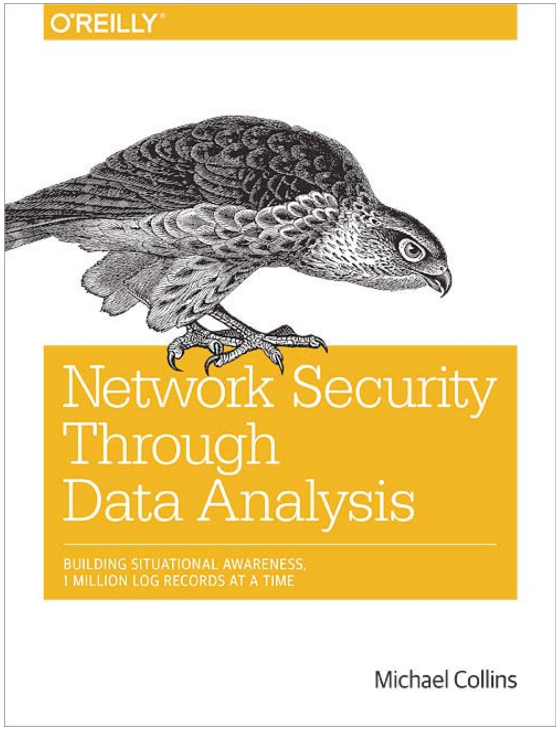
Successes and next steps

- **Successes**
  - Identified outbound scanning behaviors (and SYN floods)
  - Identified candidate external C2 hosts
  - Identified candidate internal infected hosts based on port usage
  - Identified candidate C2 ports using Jaccard scoring
  - Identified the first place to start cleaning up the XX SSH client chain (if we chose to do that. We turned off the network instead)
  - Used Spark Streaming to identify DHCP servers during Wireless network ‘troubles’

- **Next steps**
  - More RDF/BRO parsers (particularly DNS)
  - Improved Python parser to more easily use the multiple cores on the XT5 blades
  - Easier link-chart generation
  - More and more mature Spark Streaming

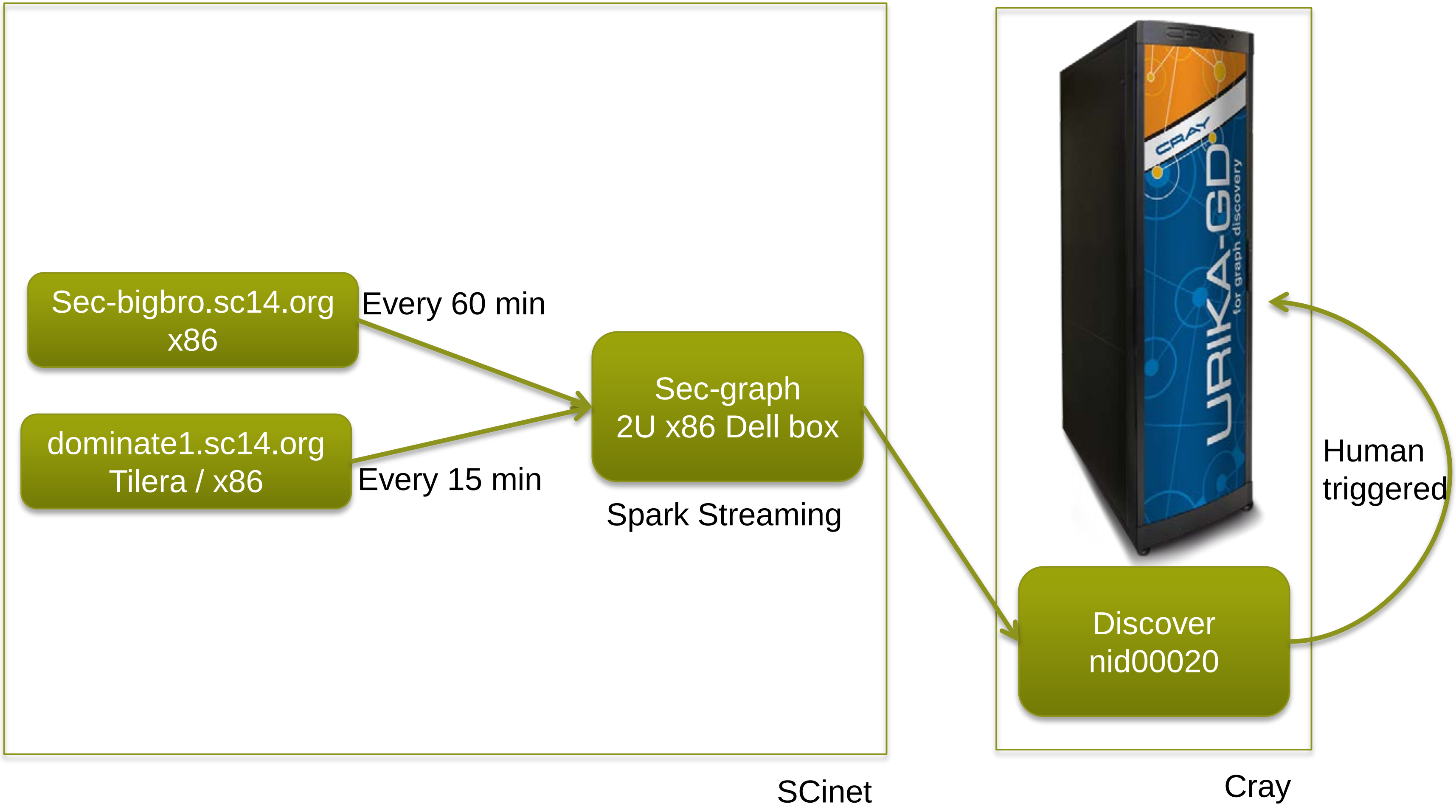
Places to start your own graph analysis journey

- **Literature**
  - Chapter 13 of Network Security through Data Analysis by Michael Collins
  - Mark Newman's publications (http://www-personal.umich.edu/~mejn/pubs.html)
  - Linked by Barabasi
- **Available FOSS tools**
  - Gephi
  - Cytoscape
  - Apache Jena
  - MTGL (https://software.sandia.gov/trac/mtgl)


O'REILLY
Network Security Through Data Analysis
BUILDING SITUATIONAL AWARENESS, 1 MILLION LOG RECORDS AT A TIME
Michael Collins

BACKUP SLIDES

Data processing architecture



Available Algorithms

- **Search / neighborhood identification and extraction**
  - Pattern-matching / subgraph isomorphism: **(Core functionality)**
  - **Cybersecurity application: Context and search, data exfiltration, beaconing, attack identification**
- **Community detection**
  - Modularity: **(Fall2013 release)**
  - Relaxed clique
  - **Cybersecurity application: Botnet detection and server hierarchy mapping**
- **Similarity scoring**
  - Jaccard scoring: **(Example code available)**
  - **Cybersecurity: Infrastructure mapping, botnet detection, client / server mapping**
- **Path finding**
  - Shortest path: **(Summer2014 release),** S-T connectivity: : **(Fall2014 release)**
  - **Cybersecurity application: Identify likely paths for information flow between nodes**
- **Key node / edge identification**
  - Betweenness centrality: **(Summer2014 release)**
  - **Cybersecurity application: find the vulnerable points in network configurations**
- **Anomaly identification and clustering**
  - **Cybersecurity application: Unknown-unknown identification**
  - **Cybersecurity application: BadRank (Summer2014 release): finds likely worst actors by association with known bad actors, a la PageRank**