Declarative, Distributed Configuration

Distributed SDN Workshop, July 15, 2014, Paris

Applied Communication Sciences

Sanjai Narain
Dana Chee
Chung-Min Chen
Brian Coan
Ben Falchuk
Samuel Gordon
Jonathan Kirsch
Siun-Chuon Mau
Aditya Naidu
Simon Tsang

Princeton University

Sharad Malik
Shuyuan Zhang

This work has been funded by Air Force Research Laboratory under contract FA8750-13-C-0030. "Approved for Public Release; Distribution Unlimited : 88ABW-2014 1877 22 April 2014"

Relationship With SDN

SDN Motivations

- Planning network as a whole is hard
- Configuration is hard
- Deploying new protocols e.g., for virtualization, is hard
- Control and data planes are merged

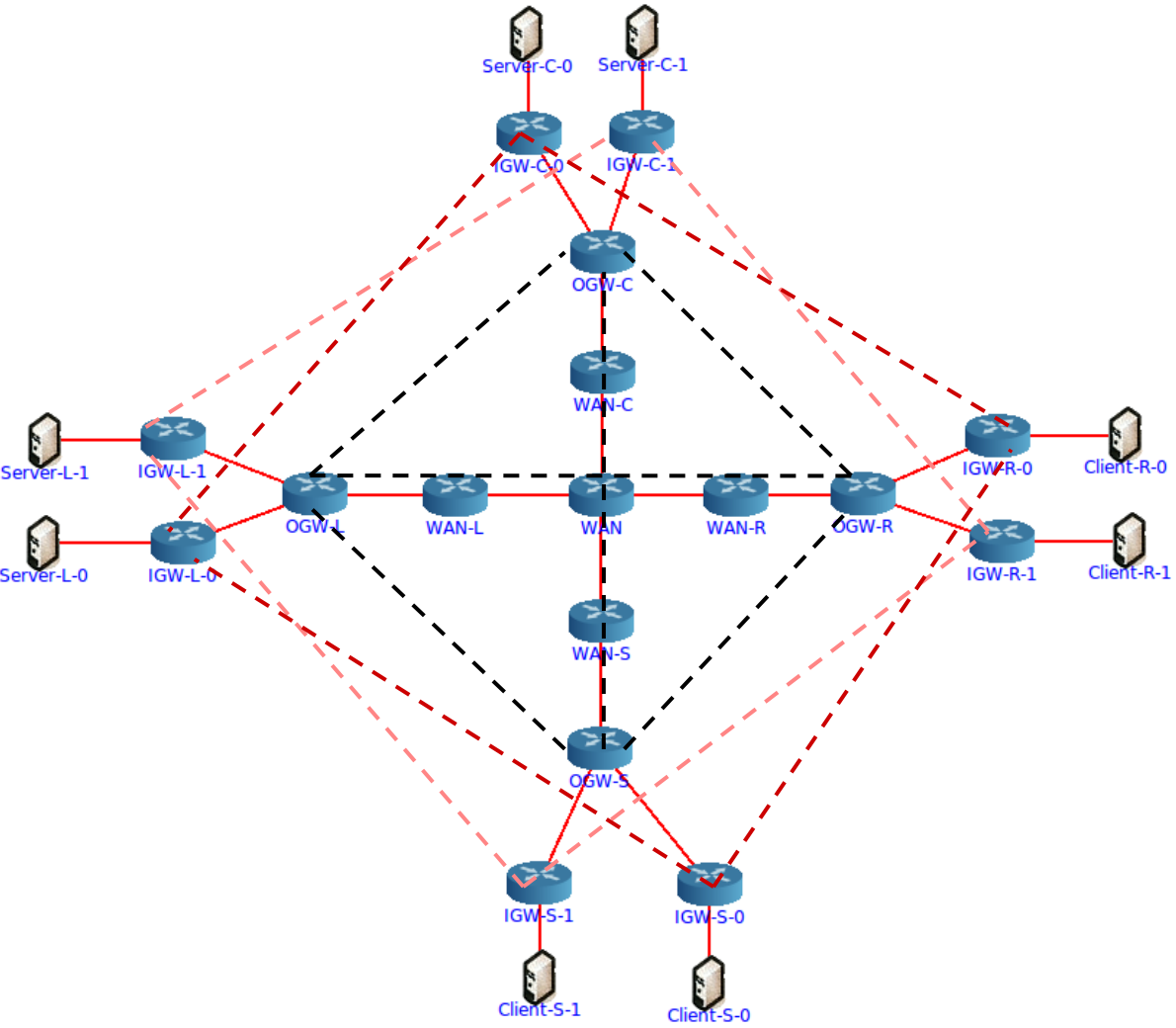
SDN Solution

- Use utterly simple switches that only do routing and access control
- Program all control functionality in logically centralized controller
- Make controller interact with data plane over out-of-band network

But

- Can we solve the problems that SDN is motivated by, with existing network devices and protocols?

ADC: A Science of Configuration To Bridge Gap Between Conceptualization and Implementation



How we conceptualize (separated resilient VPNs)

Specification
Synthesis
Diagnosis
Repair
Visualization
Verification
MTD
Emulation
Distribution
Design Intent

Months → minutes

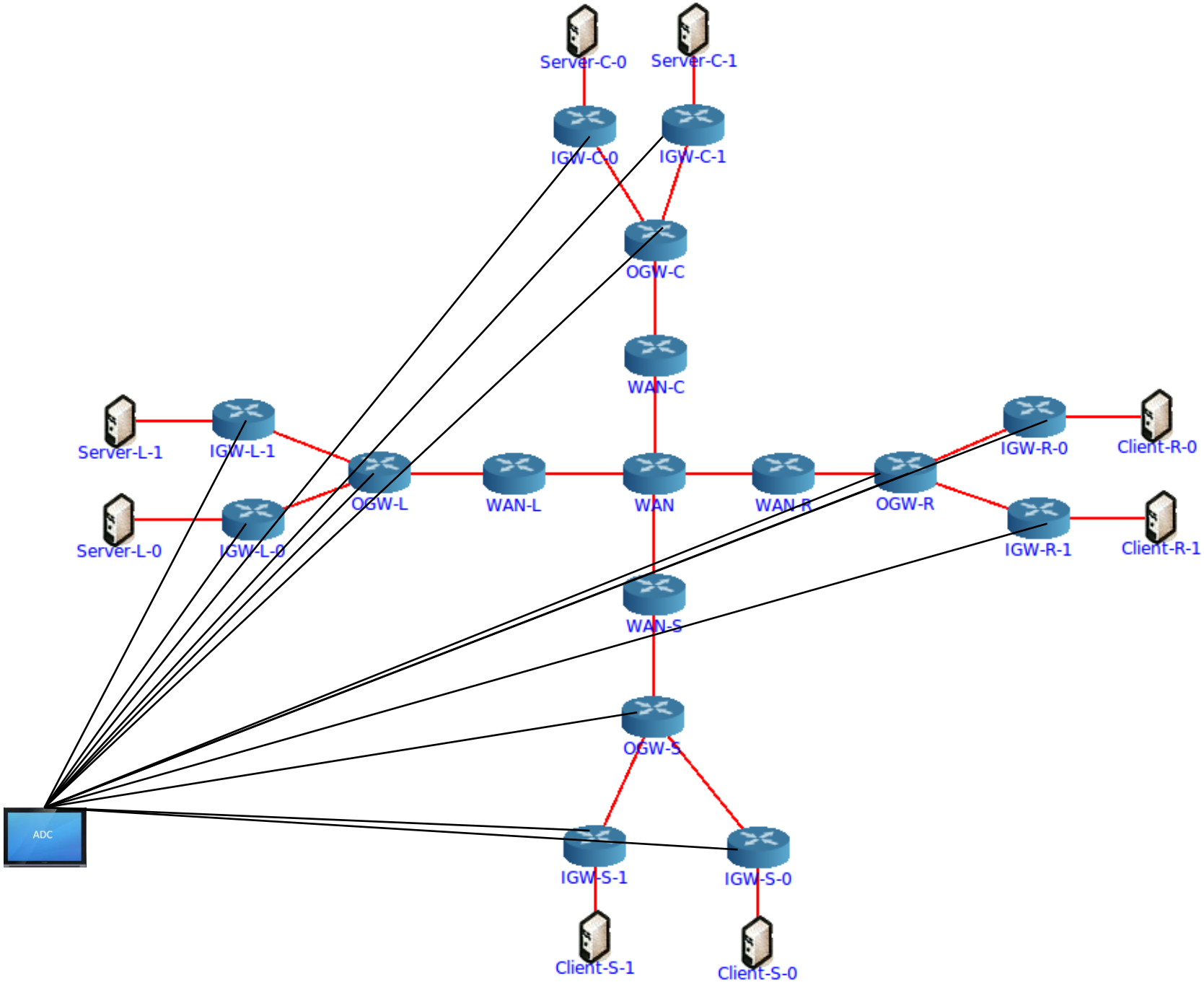
Model and solve dependencies between configuration variables with SMT solvers

Make existing protocols and their compositions do all the work for us

Partial configuration for IGW-C-S.cfg

```
crypto isakmp policy 10
  authentication pre-share
crypto ipsec transform-set esp-des-esp-md5-hmac esp-des esp-md5-hmac
  mode tunnel
crypto isakmp key 1234567890 address 2.1.0.33
crypto map on_eth0 10 ipsec-isakmp
  set peer 2.1.0.33
  set transform-set esp-des-esp-md5-hmac
  match address gre_ipsec_IGW-C-S_eth0_IGW-L-S_eth0

interface Tunnel0
  no shutdown
  ip address 192.168.21.1 255.255.255.252
  tunnel source 1.1.0.2
  tunnel destination 2.1.0.33

interface eth0
  no shutdown
  ip address 1.1.0.2 255.255.255.0
  crypto map on_eth0

router ospf 1
  network 192.168.21.1 0.0.0.0 area 0
  network 192.168.21.9 0.0.0.0 area 0
  network 1.1.100.3 0.0.0.0 area 0

ip route 0.0.0.0 0.0.0.0 1.1.0.3

ip access-list extended gre_ipsec_IGW-C-S_eth0_IGW-L-S_eth0
  permit gre 1.1.0.2 0.0.0.0 2.1.0.33 0.0.0.0
```

How we implement



Centralized Management over Out of Band Network

Using CORE Linux Router Emulator

Specify Secure and Resilient VPNs As Composition of GRE, IPSec and OSPF

Global Specification

```
gre ipsec tunnel options 192.168.21.0 30
   IGW-L-0 Tunnel0  eth0
   IGW-C-0 Tunnel0  eth0
   1234567890 esp-des esp-md5-hmac

gre ipsec tunnel options 192.168.21.4 30
   IGW-L-0 Tunnel1  eth0
   IGW-S-0 Tunnel0  eth0
   1234567890 esp-des esp-md5-hmac

gre ipsec tunnel options 192.168.21.8 30
   IGW-R-0 Tunnel0  eth0
   IGW-C-0 Tunnel1  eth0
   1234567890 esp-des esp-md5-hmac

gre ipsec tunnel options 192.168.21.12 30
   IGW-R-0 Tunnel1  eth0
   IGW-S-0 Tunnel1  eth0
   1234567890 esp-des esp-md5-hmac
```

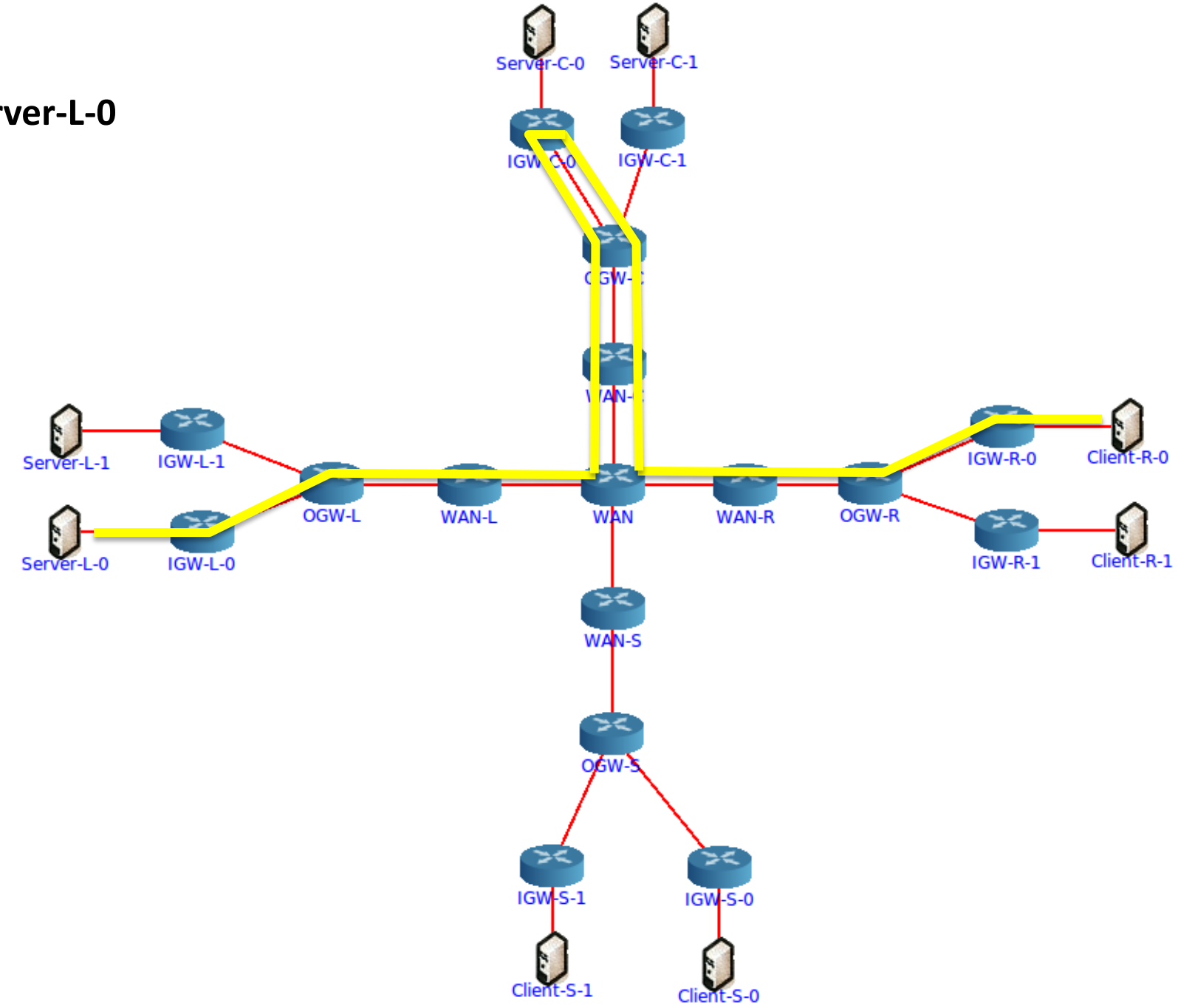
Abstract Solution

```
gre_local_physical('IGW-L-0','Tunnel0') = '2.1.0.9'
gre_remote_physical('IGW-C-0','Tunnel0') = '2.1.0.9'
gre_remote_physical('IGW-L-0','Tunnel0') = '1.1.0.3'
gre_local_physical('IGW-C-0','Tunnel0') = '1.1.0.3'
ip_address('IGW-L-0',eth0) = '2.1.0.9'
ip_address('IGW-C-0',eth0) = '1.1.0.3'
ip_address('IGW-L-0','Tunnel0') = '192.168.21.2'
ip_address('IGW-C-0','Tunnel0') = '192.168.21.1'
mask('IGW-L-0','Tunnel0') = 30
mask('IGW-C-0','Tunnel0') = 30
ipsec_ea('IGW-L-0',eth0,'IGW-C-0',eth0) = 'esp-des'
ipsec_ea('IGW-C-0',eth0,'IGW-L-0',eth0) = 'esp-des'
ipsec_ha('IGW-L-0',eth0,'IGW-C-0',eth0) = 'esp-md5-hmac'
ipsec_ha('IGW-C-0',eth0,'IGW-L-0',eth0) = 'esp-md5-hmac'
ipsec_key('IGW-L-0',eth0,'IGW-C-0',eth0) = 1234567890
ipsec_key('IGW-C-0',eth0,'IGW-L-0',eth0) = 1234567890
ipsec_remote('IGW-L-0',eth0,'IGW-C-0',eth0) = '1.1.0.3'
ipsec_remote('IGW-C-0',eth0,'IGW-L-0',eth0) = '2.1.0.9'
ipsec_acl_id('IGW-L-0',eth0,'IGW-C-0',eth0) = 'gre_ipsec_IGW-
L-0_eth0_IGW-C-0_eth0'
ipsec_acl_id('IGW-C-0',eth0,'IGW-L-0',eth0) = 'gre_ipsec_IGW-
C-0_eth0_IGW-L-0_eth0'
acl_action('0000','IGW-L-0','gre_ipsec_IGW-L-0_eth0_IGW-C-
0_eth0','2.1.0.9',32,0,'1.1.0.3',32,0,47) = permit
acl_action('0000','IGW-C-0','gre_ipsec_IGW-C-0_eth0_IGW-L-
0_eth0','1.1.0.3',32,0,'2.1.0.9',32,0,47) = permit
gre_local_physical('IGW-L-0','Tunnel1') = '2.1.0.9'
gre_remote_physical('IGW-S-0','Tunnel0') = '2.1.0.9'
gre_remote_physical('IGW-L-0','Tunnel1') = '4.1.0.3'
gre_local_physical('IGW-S-0','Tunnel0') = '4.1.0.3'
ip_address('IGW-S-0',eth0) = '4.1.0.3'
ip_address('IGW-L-0','Tunnel1') = '192.168.21.6'
ip_address('IGW-S-0','Tunnel0') = '192.168.21.5'
mask('IGW-L-0','Tunnel1') = 30
mask('IGW-S-0','Tunnel0') = 30
ipsec_ea('IGW-L-0',eth0,'IGW-S-0',eth0) = 'esp-des'
```

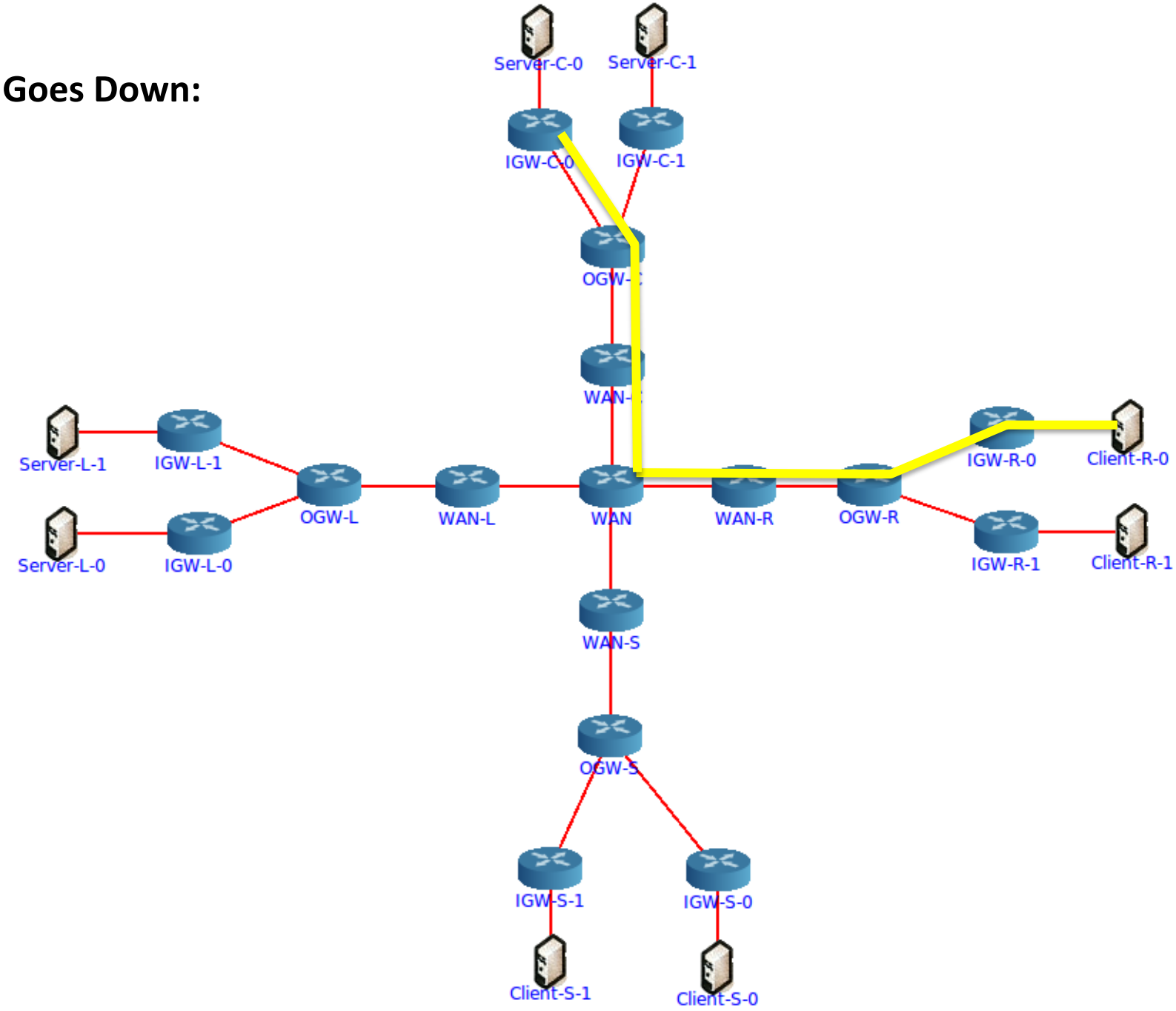
Vendor-Specific Configuration

```
===== IGW-C-0.sh =====
ip tunnel add Tunnel0 mode gre local 1.1.0.3 remote 2.1.0.9
ttl 225
ifconfig Tunnel0 multicast
ifconfig Tunnel0 192.168.21.1 netmask 255.255.255.252 up
ip tunnel add Tunnel1 mode gre local 1.1.0.3 remote 3.1.0.17
ttl 225
ifconfig Tunnel1 multicast
ifconfig Tunnel1 192.168.21.10 netmask 255.255.255.252 up
ifconfig eth0 1.1.0.3 netmask 255.255.255.0 up
ifconfig eth1 1.1.100.2 netmask 255.255.255.0 up
ifconfig eth2 172.16.16.12 netmask 255.255.255.0 up

cat >> /usr/local/etc/quagga/Quagga.conf <<HEAR_HEAR
hostname IGW-C-0
password adcadc
enable password adcadc

router ospf
  network 192.168.21.1/30 area 0
  network 192.168.21.10/30 area 0
  network 1.1.100.2/24 area 0
HEAR_HEAR

cat >> /usr/local/etc/quagga/daemons <<HEAR_HEAR
zebra=yes
ospfd=yes
HEAR_HEAR

service quagga restart
iptables -A INPUT -j ACCEPT -s 1.1.0.3/32 -d 2.1.0.9/32
iptables -A INPUT -j ACCEPT -s 1.1.0.3/32 -d 3.1.0.17/32
PCONF=`mktemp --tmpdir psk.XXXXXX`
```

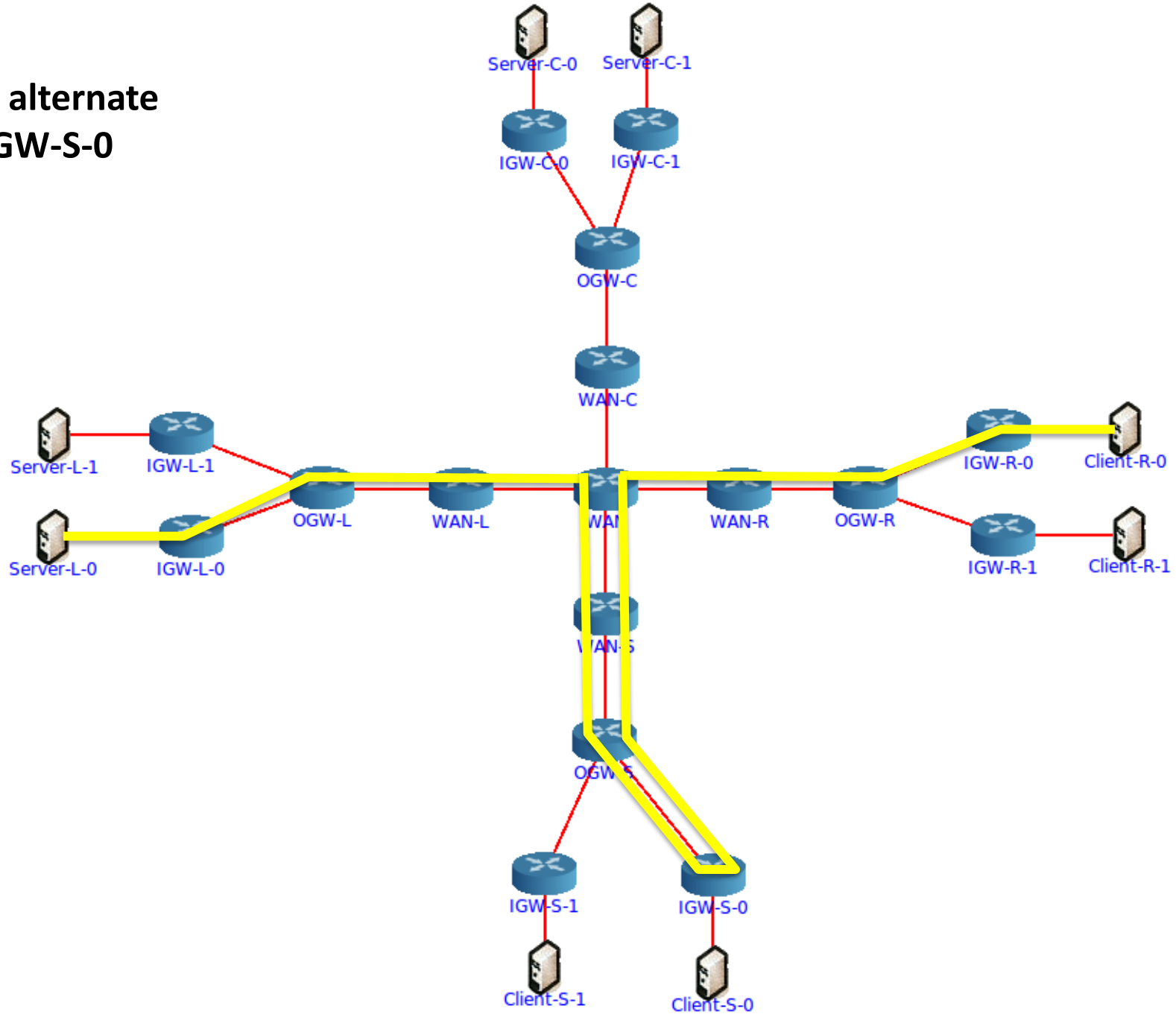
Successful Ping:
Client-R-0 to Server-L-0



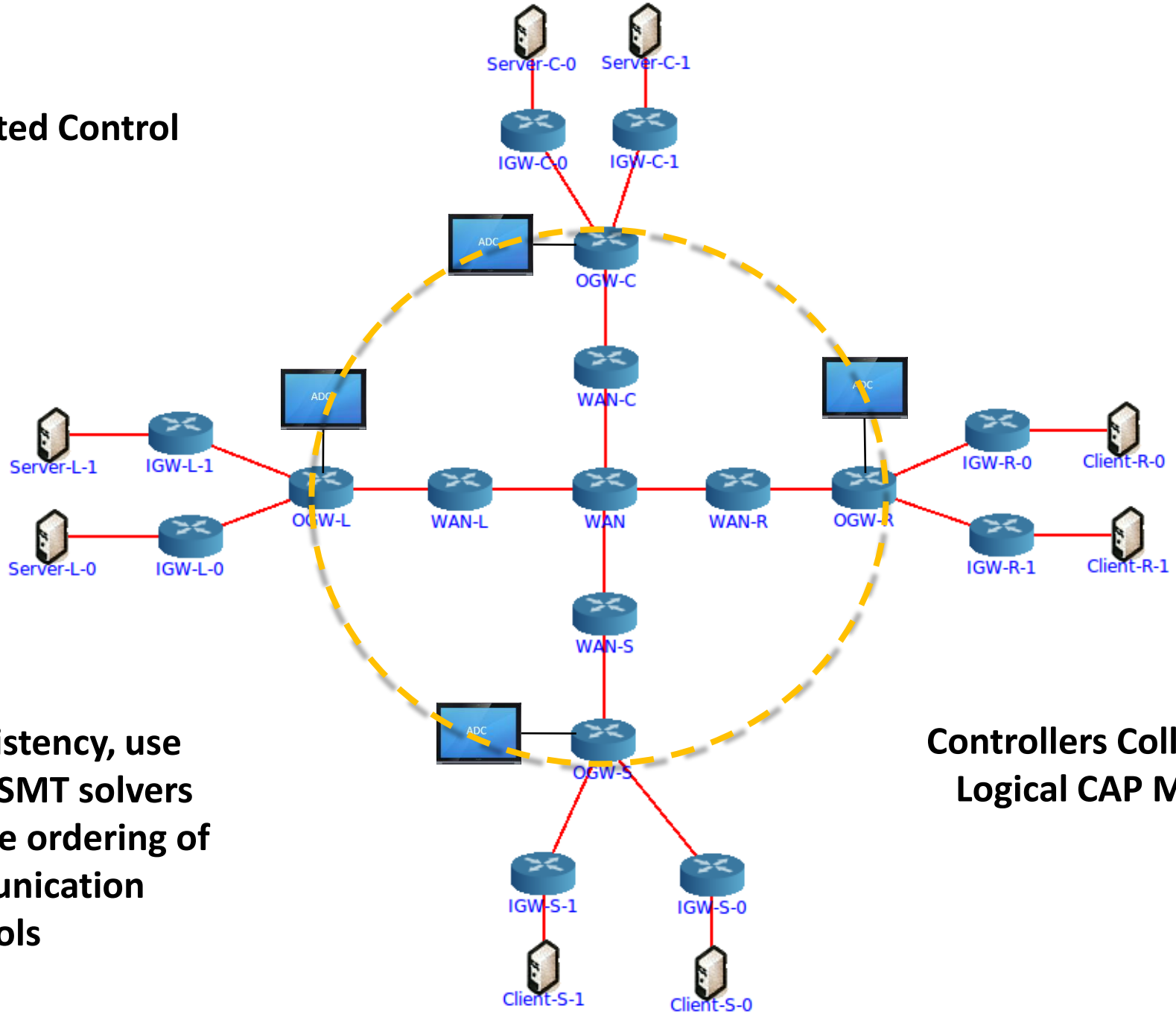
Router-IGW-C-0 Goes Down:



OSPF establishes alternate routes through IGW-S-0



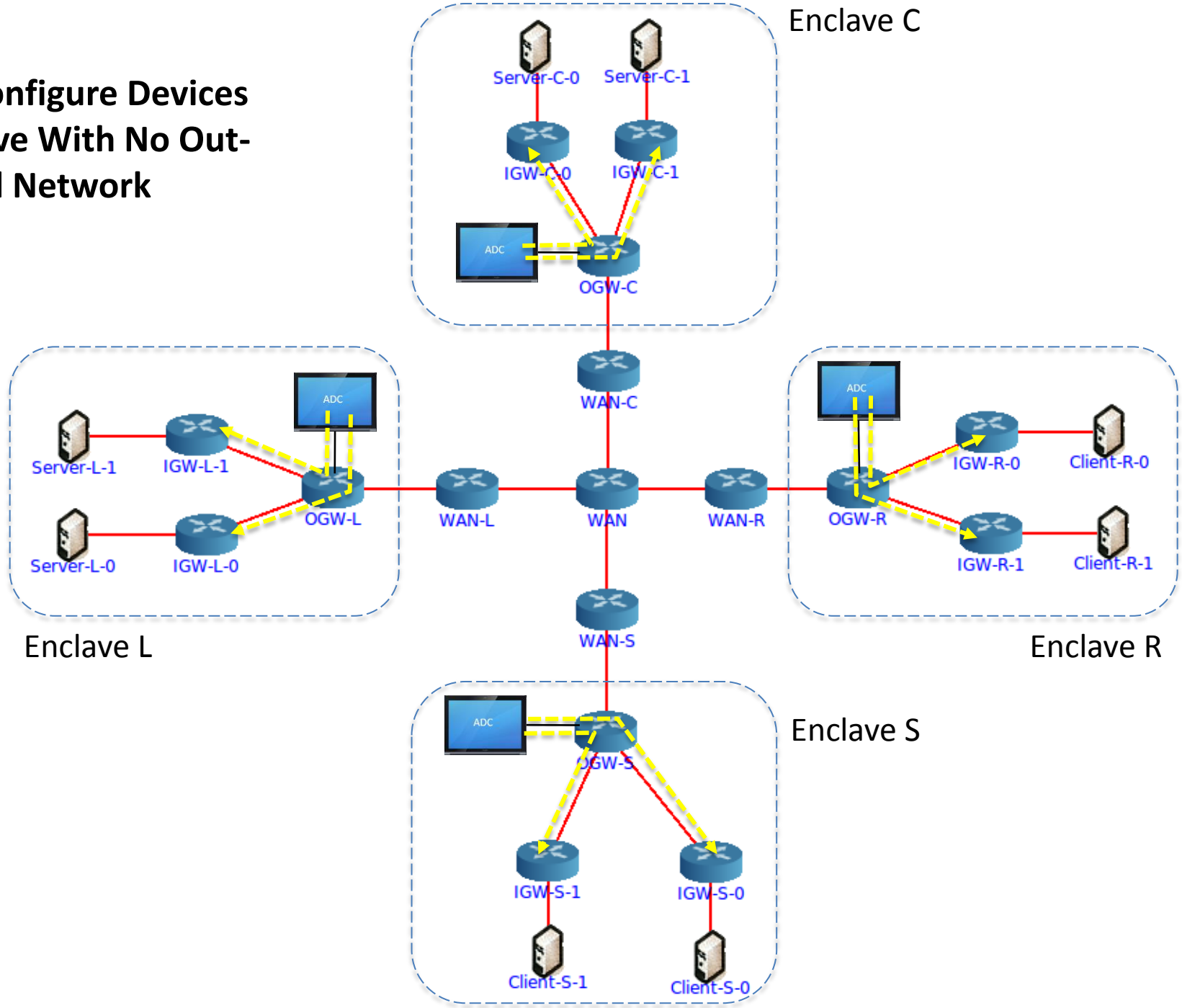
Distributed Control



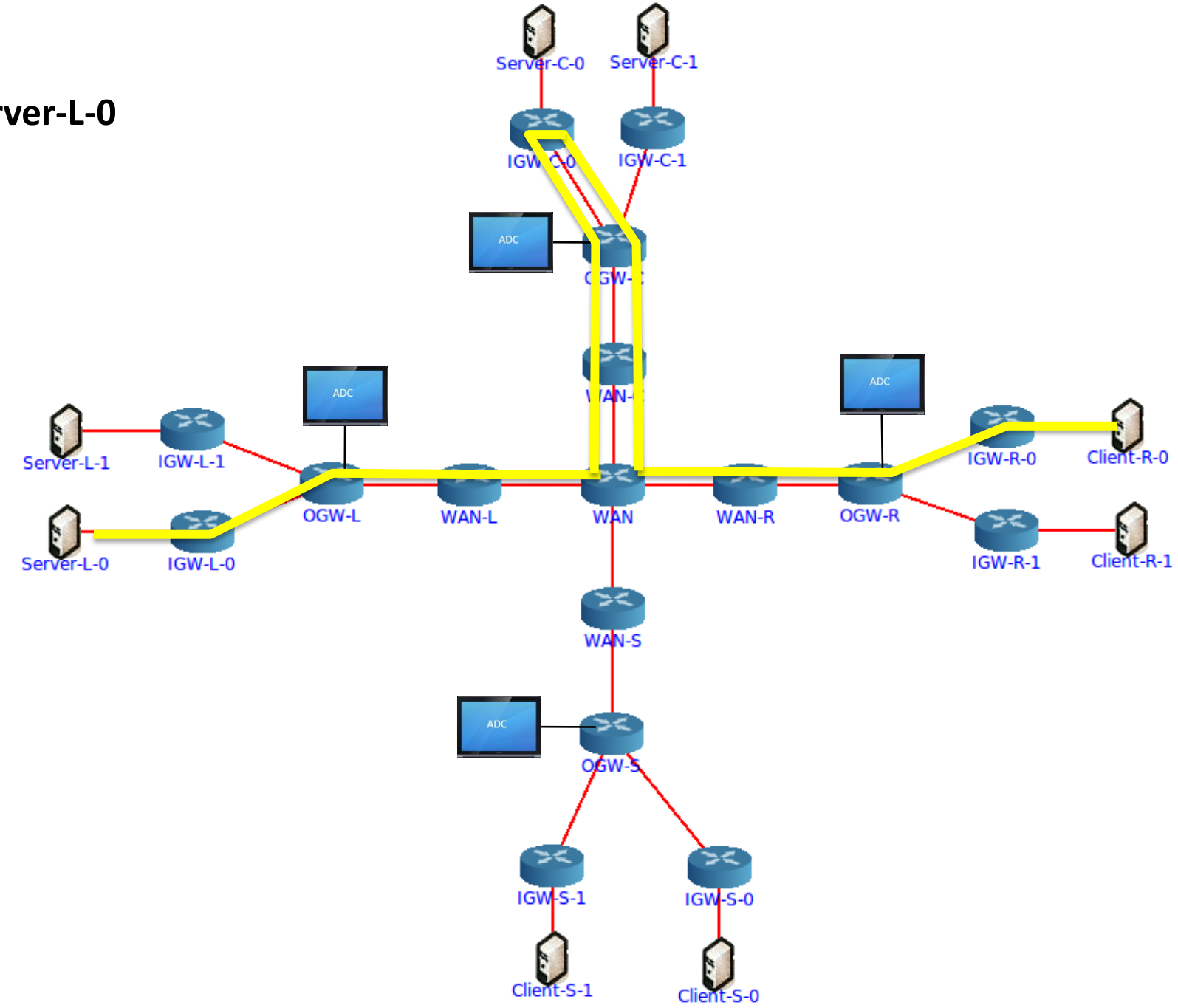
For global consistency, use determinism of SMT solvers and total message ordering of group communication protocols

Controllers Collaborate Over Logical CAP Message Bus

Controllers Configure Devices In Their Enclave With No Out-of-Band Network



Successful Ping:
Client-R-0 to Server-L-0



In-Band Update for Network Routing Policy Migration
To Appear in ICNP-2014

- Devise an **ILP-based symbolic encoding** of the network and use it to model how packets are forwarded by routers before and after they are configured
- **Encode constraints** that will ensure a safe reconfiguration order