Propane

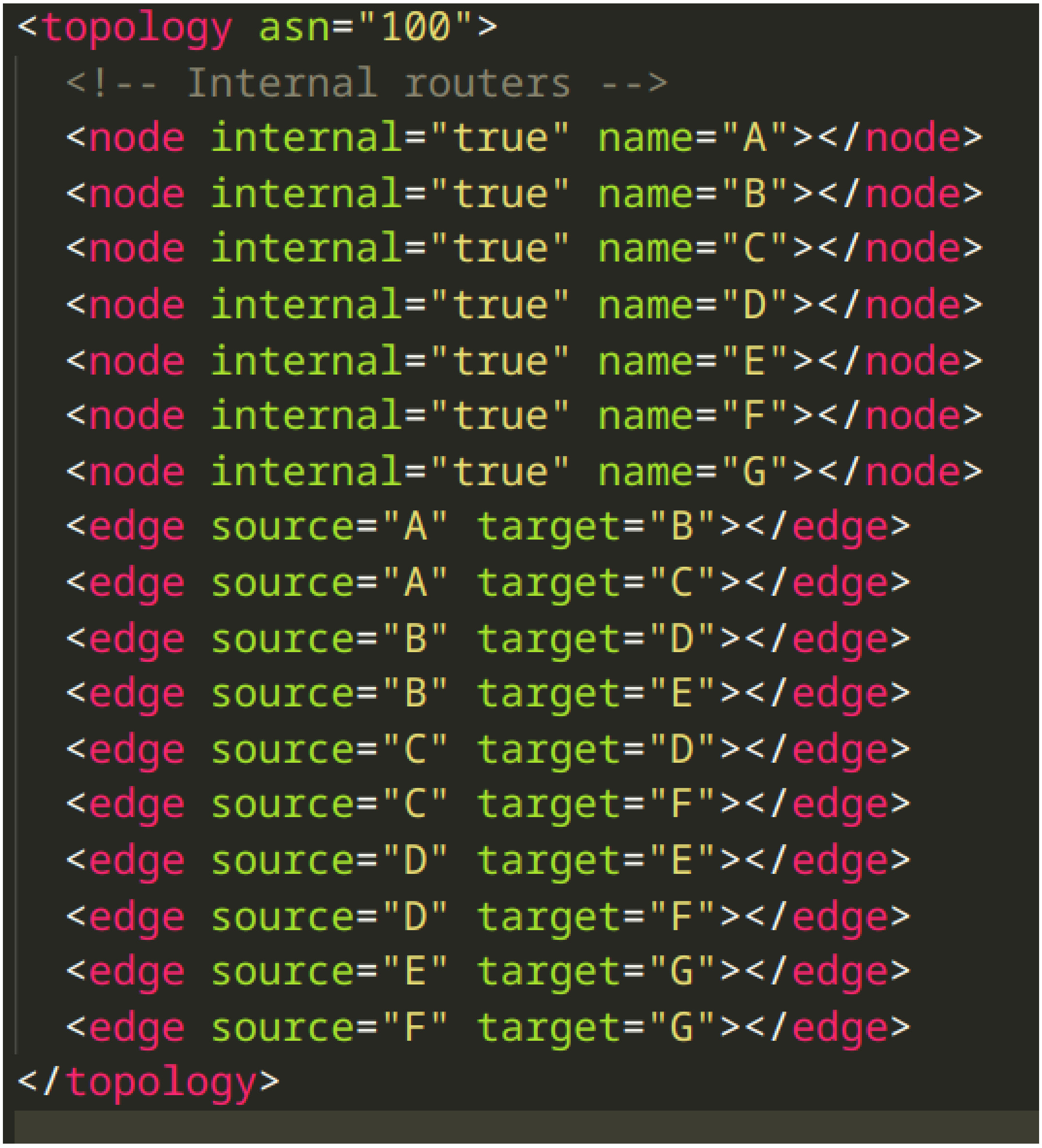
tags: `class`

compile: `propane --policy ex1.pro --topo ex1.xml`

emulator: `core-gui output/core.imn`

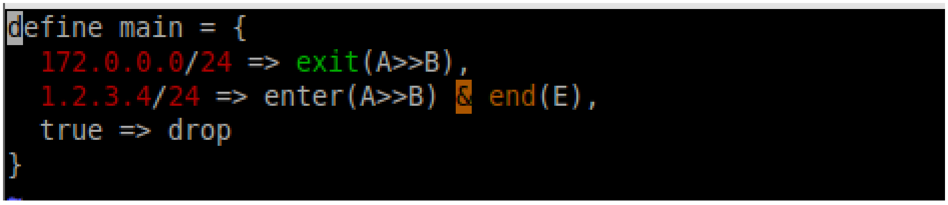
Part 1: Cold Potato Routing

Topology:



```
<topology asn="100">
  <!-- Internal routers -->
  <node internal="true" name="A"></node>
  <node internal="true" name="B"></node>
  <node internal="true" name="C"></node>
  <node internal="true" name="D"></node>
  <node internal="true" name="E"></node>
  <node internal="true" name="F"></node>
  <node internal="true" name="G"></node>
  <edge source="A" target="B"></edge>
  <edge source="A" target="C"></edge>
  <edge source="B" target="D"></edge>
  <edge source="B" target="E"></edge>
  <edge source="C" target="D"></edge>
  <edge source="C" target="F"></edge>
  <edge source="D" target="E"></edge>
  <edge source="D" target="F"></edge>
  <edge source="E" target="G"></edge>
  <edge source="F" target="G"></edge>
</topology>
```

Policy:



```
define main = {
  172.0.0.0/24 => exit(A>>B),
  1.2.3.4/24 => enter(A>>B) & end(E),
  true => drop
}
```

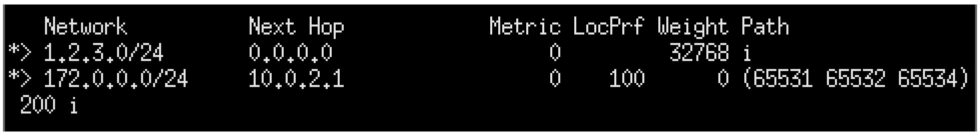
AS number:

A: 65534

B: 65533

C: 65532
D: 65531
E: 65530

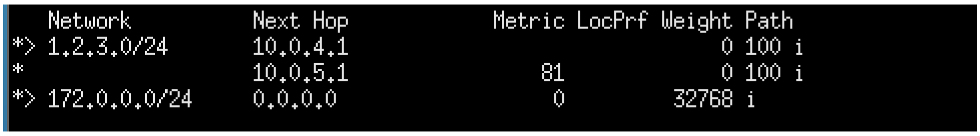
1



```
   Network          Next Hop            Metric LocPrf Weight Path
*> 1.2.3.0/24       0.0.0.0                  0         32768 i
*> 172.0.0.0/24     10.0.2.1                 0    100      0 (65531 65532 65534)
 200 i
```

On the second line, we can see the route E->D->C->A has a local preference of 100 and 3 stops in the path columns(65531, 65532, 65534), giving us an AS path length of 3.

2

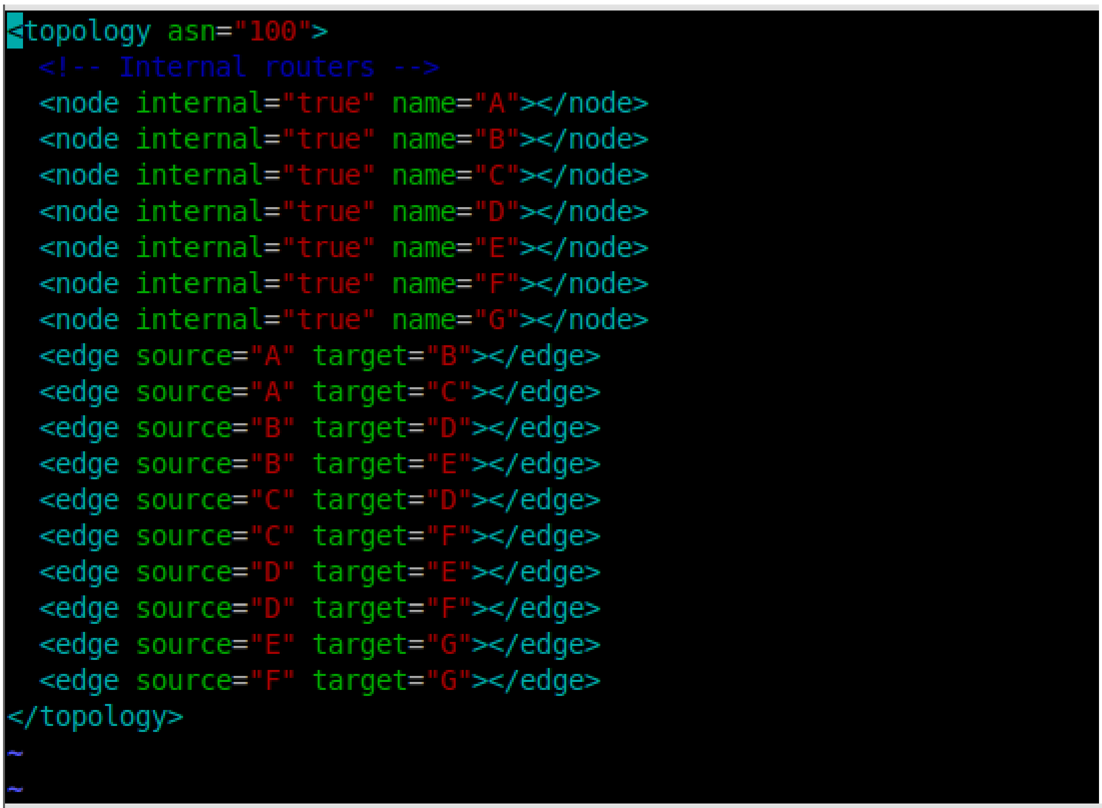


```
   Network          Next Hop            Metric LocPrf Weight Path
*> 1.2.3.0/24       10.0.4.1                               0 100 i
*                   10.0.5.1                81             0 100 i
*> 172.0.0.0/24     0.0.0.0                  0         32768 i
```

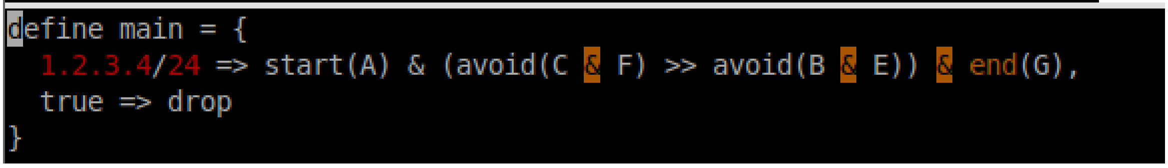
For this part, I enabled the --med for the second constraint to be compiled, specifying A as an exit point is more preferable than B. The Metric column is different for 10.0.4.1, which connects to A, and 10.0.5.1, which connects to B.

Part 2: BGP Expressiveness

1

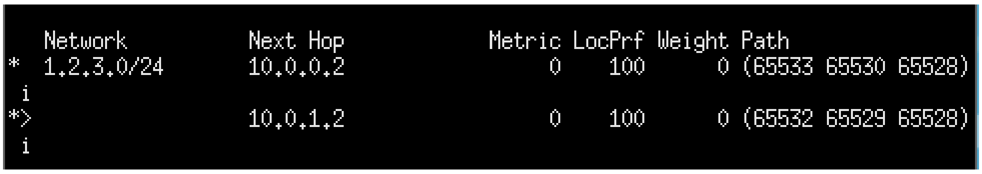


```
<topology asn="100">
  <!-- Internal routers -->
  <node internal="true" name="A"></node>
  <node internal="true" name="B"></node>
  <node internal="true" name="C"></node>
  <node internal="true" name="D"></node>
  <node internal="true" name="E"></node>
  <node internal="true" name="F"></node>
  <node internal="true" name="G"></node>
  <edge source="A" target="B"></edge>
  <edge source="A" target="C"></edge>
  <edge source="B" target="D"></edge>
  <edge source="B" target="E"></edge>
  <edge source="C" target="D"></edge>
  <edge source="C" target="F"></edge>
  <edge source="D" target="E"></edge>
  <edge source="D" target="F"></edge>
  <edge source="E" target="G"></edge>
  <edge source="F" target="G"></edge>
</topology>
~
~
```



```
define main = {
  1.2.3.4/24 => start(A) & (avoid(C & F) >> avoid(B & E)) & end(G),
  true => drop
}
```

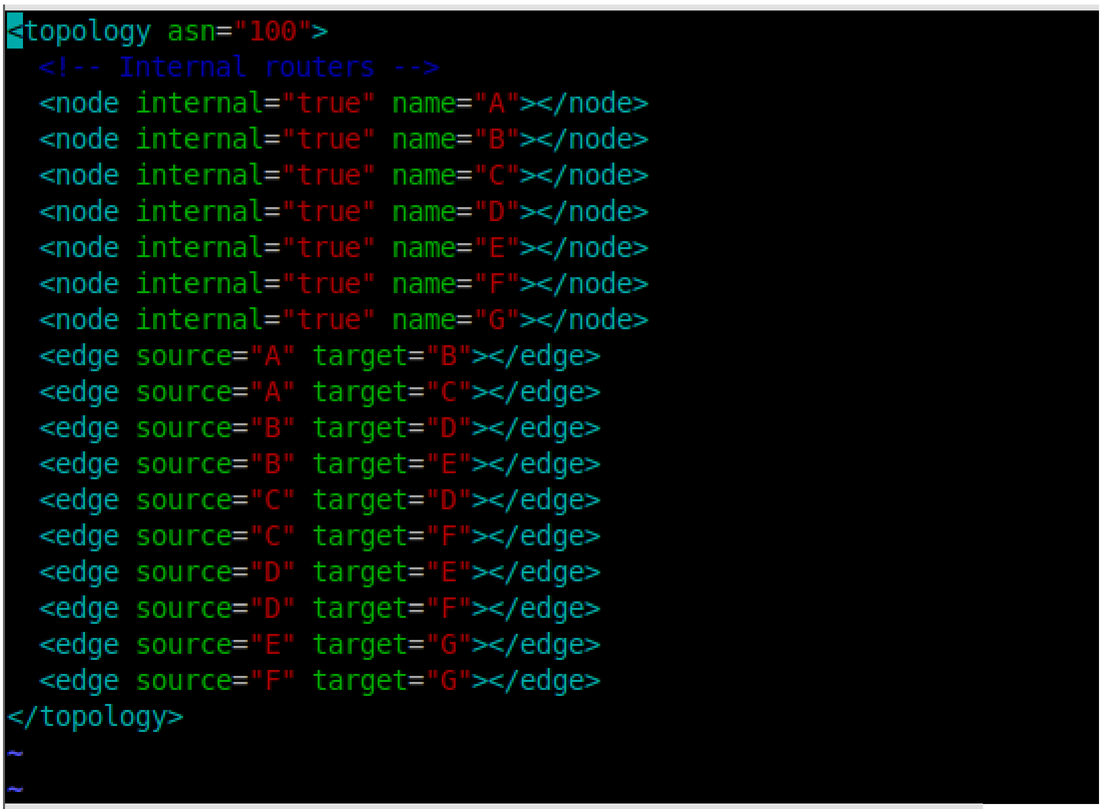
BGP on A:



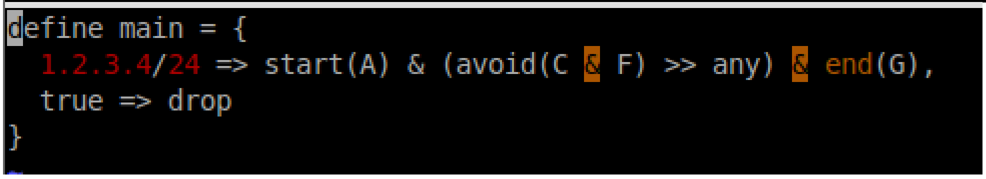
```
   Network          Next Hop            Metric LocPrf Weight Path
*  1.2.3.0/24       10.0.0.2                 0    100      0 (65533 65530 65528)
 i
*>                  10.0.1.2                 0    100      0 (65532 65529 65528)
 i
```

Yes, although I got a warning saying `Unused preference 2 policy for predicate` propane was able to express this policy.

2

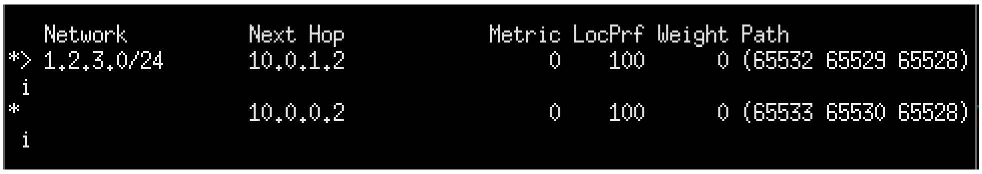


```
<topology asn="100">
  <!-- Internal routers -->
  <node internal="true" name="A"></node>
  <node internal="true" name="B"></node>
  <node internal="true" name="C"></node>
  <node internal="true" name="D"></node>
  <node internal="true" name="E"></node>
  <node internal="true" name="F"></node>
  <node internal="true" name="G"></node>
  <edge source="A" target="B"></edge>
  <edge source="A" target="C"></edge>
  <edge source="B" target="D"></edge>
  <edge source="B" target="E"></edge>
  <edge source="C" target="D"></edge>
  <edge source="C" target="F"></edge>
  <edge source="D" target="E"></edge>
  <edge source="D" target="F"></edge>
  <edge source="E" target="G"></edge>
  <edge source="F" target="G"></edge>
</topology>
~
~
```



```
define main = {
  1.2.3.4/24 => start(A) & (avoid(C & F) >> any) & end(G),
  true => drop
}
```

BGP on B:



```
   Network          Next Hop            Metric LocPrf Weight Path
*> 1.2.3.0/24       10.0.1.2                 0    100      0 (65532 65529 65528)
 i
*                   10.0.0.2                 0    100      0 (65533 65530 65528)
 i
```

Propane was able to express this policy as well. Got the same warning: `Unused preference 2 policy for predicate`

3

Both of the policies compiled successfully