In [81]:
```
from pybatfish.client.session import Session
from pybatfish.datamodel import *
```

In [82]:
```
bf = Session()
```

In [83]:
```
bf.set_network()
```

Out[83]:
```
'pcp283fe8ab-6985-4bfc-81a0-b46d84966148'
```

In [84]:
```
bf.init_snapshot('networks/8859')
```

Out[84]:
```
'ss_5f1a8319-b3dc-43f1-8cf9-eee06b48d910'
```

In [85]:
```
bf.q.fileParseStatus().answer().frame()
```

Out[85]:

	File_Name	Status	File_Format	Nodes
0	configs/dev1.txt	PASSED	CISCO_IOS	['dev1']
1	configs/dev2.txt	PASSED	CISCO_IOS	['dev2']
2	configs/dev3.txt	PASSED	CISCO_IOS	['dev3']
3	configs/dev4.txt	PASSED	CISCO_IOS	['dev4']
4	configs/dev5.txt	PASSED	CISCO_IOS	['dev5']

In [86]:
```
tr = bf.q.traceroute(startLocation='dev1', headers=HeaderConstraints(dstIps='10.0.45.5
tr
```

Out[86]:

	Flow	Traces	TraceCount
0	start=dev1 [10.0.12.1:49152->10.0.45.5:33434 UDP]	[((ORIGINATED(default), FORWARDED(Forwarded ou...	1

In [87]:
```
tr.Traces[0]
```

Out[87]: ACCEPTED

1. node: dev1
ORIGINATED(default)
FORWARDED(Forwarded out interface: GigabitEthernet0/1 with resolved next-hop IP: 10.0.12.2, Routes: [static (Network: 0.0.0.0/0, Next Hop: ip 10.0.12.2)])
TRANSMITTED(GigabitEthernet0/1)

2. node: dev2
RECEIVED(GigabitEthernet0/0)
FORWARDED(Forwarded out interface: GigabitEthernet0/1 with resolved next-hop IP: 10.0.23.3, Routes: [static (Network: 10.0.45.0/24, Next Hop: ip 10.0.23.3)])
TRANSMITTED(GigabitEthernet0/1)

3. node: dev3
RECEIVED(GigabitEthernet0/0)
FORWARDED(Forwarded out interface: GigabitEthernet0/1 with resolved next-hop IP: 10.0.34.4, Routes: [static (Network: 10.0.45.0/24, Next Hop: ip 10.0.34.4)])
TRANSFORMED(SOURCE_NAT srcIp: 10.0.12.1 -> 20.0.12.1)
SETUP_SESSION(Incoming Interfaces: [GigabitEthernet0/1], Action: PostNatFibLookup, Match Criteria: [ipProtocol=UDP, srcIp=10.0.45.5, dstIp=20.0.12.1, srcPort=33434, dstPort=49152], Transformation: [dstIp: 20.0.12.1 -> 10.0.12.1])
TRANSMITTED(GigabitEthernet0/1)

4. node: dev4
RECEIVED(GigabitEthernet0/0)
FORWARDED(Forwarded out interface: GigabitEthernet0/1, Routes: [connected (Network: 10.0.45.0/24, Next Hop: interface GigabitEthernet0/1)])
TRANSMITTED(GigabitEthernet0/1)

5. node: dev5
RECEIVED(GigabitEthernet0/0)
ACCEPTED(GigabitEthernet0/0)

In [88]:
```
bireachability = bf.q.bidirectionalReachability(pathConstraints=PathConstraints(startL
```

```
---------------------------------------------------------------------------
BatfishException                          Traceback (most recent call last)
Cell In [88], line 1
----> 1 bireachability = bf.q.bidirectionalReachability(pathConstraints=PathConstrain
ts(startLocation='dev5'),headers=HeaderConstraints(srcIps='10.0.45.5', dstIps='20.0.1
2.1', srcPorts='32875', dstPorts='22')).answer().frame()

File /usr/local/lib/python3.8/dist-packages/pybatfish/question/question.py:192, in Qu
estionBase.answer(self, snapshot, reference_snapshot, include_one_table_keys, backgro
und, extra_args)
    190 if include_one_table_keys is not None:
    191     self._set_include_one_table_keys(include_one_table_keys)
--> 192 return _bf_answer_obj(
    193     session=session,
    194     question_str=self.json(),
    195     question_name=self.get_name(),
    196     background=background,
    197     snapshot=real_snapshot,
    198     reference_snapshot=reference_snapshot,
    199     extra_args=extra_args,
    200 )

File /usr/local/lib/python3.8/dist-packages/pybatfish/client/internal.py:60, in _bf_a
nswer_obj(session, question_str, question_name, background, snapshot, reference_snaps
hot, extra_args)
     56 # Answer the question
     57 work_item = workhelper.get_workitem_answer(
     58     session, question_name, snapshot, reference_snapshot
     59 )
---> 60 workhelper.execute(work_item, session, background, extra_args)
     62 if background:
     63     return work_item.id

File /usr/local/lib/python3.8/dist-packages/pybatfish/client/workhelper.py:140, in ex
ecute(work_item, session, background, extra_args)
    138             log_file_handle.write(str(log))
    139         log_file_msg = "Full log written to {}\n".format(log_file)
--> 140     raise BatfishException(
    141         "Work terminated abnormally\nwork_item: {item}\n\n{msg}log: {prefix}
{log}".format(
    142             item=work_item.to_json(),
    143             msg=log_file_msg,
    144             log=log[-MAX_LOG_LENGTH:],
    145             prefix="..." if log_file_msg else "",
    146         )
    147     )
    149 return {"status": status}

BatfishException: Work terminated abnormally
work_item: {"containerName": "pcp283fe8ab-6985-4bfc-81a0-b46d84966148", "id": "593117
35-0b44-44c4-a5fc-f5b8e68f1009", "requestParams": {"answer": "", "questionname": "__b
idirectionalReachability_fa940fa7-15c5-4102-8ed7-e90b47531484", "testrig": "ss_5f1a83
19-b3dc-43f1-8cf9-eee06b48d910"}, "testrigName": "ss_5f1a8319-b3dc-43f1-8cf9-eee06b48
d910"}

log: Loading configurations for NetworkSnapshot{network=49e8afcc-5cf0-4f98-8b37-b2333
fe4d8bf, snapshot=a219496e-28c0-4542-aa67-2f2f48546aa0}
```

```
Loading configurations for NetworkSnapshot{network=49e8afcc-5cf0-4f98-8b37-b2333fe4d8
bf, snapshot=a219496e-28c0-4542-aa67-2f2f48546aa0}
Loading configurations for NetworkSnapshot{network=49e8afcc-5cf0-4f98-8b37-b2333fe4d8
bf, snapshot=a219496e-28c0-4542-aa67-2f2f48546aa0}
Loading configurations for NetworkSnapshot{network=49e8afcc-5cf0-4f98-8b37-b2333fe4d8
bf, snapshot=a219496e-28c0-4542-aa67-2f2f48546aa0}
Exception in container:49e8afcc-5cf0-4f98-8b37-b2333fe4d8bf, testrig:a219496e-28c0-45
42-aa67-2f2f48546aa0; exception:org.batfish.common.QuestionException: Exception answe
ring question
        at org.batfish.datamodel.answers.Answer.append(Answer.java:46)
        at org.batfish.main.Batfish.run(Batfish.java:2106)
        at org.batfish.main.Driver.lambda$runBatfish$0(Driver.java:247)
        at java.base/java.lang.Thread.run(Thread.java:829)
Caused by: org.batfish.common.BatfishException: Failed to answer question
        at org.batfish.main.Batfish.answer(Batfish.java:545)
        ... 3 more
Caused by: java.lang.UnsupportedOperationException: Reachability does not yet support
PreNatFibLookup
        at org.batfish.bddreachability.SessionInstrumentation$1.visitPreNatFibLookup
(SessionInstrumentation.java:240)
        at org.batfish.bddreachability.SessionInstrumentation$1.visitPreNatFibLookup
(SessionInstrumentation.java:223)
        at org.batfish.datamodel.flow.PreNatFibLookup.accept(PreNatFibLookup.java:22)
        at org.batfish.bddreachability.SessionInstrumentation.computeNewSuccessEdges
(SessionInstrumentation.java:222)
        at org.batfish.bddreachability.SessionInstrumentation.computeNewEdges(Session
Instrumentation.java:212)
        at java.base/java.util.stream.ReferencePipeline$7$1.accept(ReferencePipeline.
java:271)
        at java.base/java.util.Collections$2.tryAdvance(Collections.java:4747)
        at java.base/java.util.Collections$2.forEachRemaining(Collections.java:4755)
        at java.base/java.util.stream.ReferencePipeline$Head.forEach(ReferencePipelin
e.java:658)
        at java.base/java.util.stream.ReferencePipeline$7$1.accept(ReferencePipeline.
java:274)
        at java.base/java.util.Iterator.forEachRemaining(Iterator.java:133)
        at java.base/java.util.Spliterators$IteratorSpliterator.forEachRemaining(Spli
terators.java:1801)
        at java.base/java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.jav
a:484)
        at java.base/java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeli
ne.java:474)
        at java.base/java.util.stream.StreamSpliterators$WrappingSpliterator.forEachR
emaining(StreamSpliterators.java:312)
        at com.google.common.collect.CollectSpliterators$FlatMapSpliterator.lambda$fo
rEachRemaining$1(CollectSpliterators.java:378)
        at java.base/java.util.Spliterators$ArraySpliterator.forEachRemaining(Spliter
ators.java:948)
        at com.google.common.collect.CollectSpliterators$FlatMapSpliterator.forEachRe
maining(CollectSpliterators.java:374)
        at java.base/java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.jav
a:484)
        at java.base/java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeli
ne.java:474)
        at java.base/java.util.stream.StreamSpliterators$WrappingSpliterator.forEachR
emaining(StreamSpliterators.java:312)
        at com.google.common.collect.CollectSpliterators$FlatMapSpliterator.lambda$fo
```

rEachRemaining$1(CollectSpliterators.java:378)
	at java.base/java.util.Spliterators$ArraySpliterator.forEachRemaining(Spliterators.java:948)
	at com.google.common.collect.CollectSpliterators$FlatMapSpliterator.forEachRemaining(CollectSpliterators.java:374)
	at java.base/java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:484)
	at java.base/java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:474)
	at java.base/java.util.stream.ReduceOps$ReduceOp.evaluateSequential(ReduceOps.java:913)
	at java.base/java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)
	at java.base/java.util.stream.ReferencePipeline.collect(ReferencePipeline.java:578)
	at org.batfish.bddreachability.BDDReachabilityUtils.computeForwardEdgeTable(BDDReachabilityUtils.java:52)
	at org.batfish.bddreachability.BDDReachabilityAnalysis.<init>(BDDReachabilityAnalysis.java:70)
	at org.batfish.bddreachability.BDDReachabilityAnalysisFactory.bddReachabilityAnalysis(BDDReachabilityAnalysisFactory.java:1566)
	at org.batfish.bddreachability.BidirectionalReachabilityAnalysis.computeReturnPassAnalysis(BidirectionalReachabilityAnalysis.java:255)
	at com.google.common.base.Suppliers$NonSerializableMemoizingSupplier.get(Suppliers.java:183)
	at org.batfish.bddreachability.BidirectionalReachabilityAnalysis.computeReturnPassForwardReachableBdds(BidirectionalReachabilityAnalysis.java:173)
	at com.google.common.base.Suppliers$NonSerializableMemoizingSupplier.get(Suppliers.java:183)
	at org.batfish.bddreachability.BidirectionalReachabilityAnalysis.computeForwardPassStartLocationToReturnPassSuccessBdds(BidirectionalReachabilityAnalysis.java:289)
	at com.google.common.base.Suppliers$NonSerializableMemoizingSupplier.get(Suppliers.java:183)
	at org.batfish.bddreachability.BidirectionalReachabilityAnalysis.getResult(BidirectionalReachabilityAnalysis.java:343)
	at org.batfish.main.Batfish.bidirectionalReachability(Batfish.java:2885)
	at org.batfish.question.bidirectionalreachability.BidirectionalReachabilityAnswerer.answer(BidirectionalReachabilityAnswerer.java:78)
	at org.batfish.main.Batfish.answer(Batfish.java:542)
	... 3 more
{"answerElements":[{"class":"org.batfish.common.BatfishException$BatfishStackTrace","answer":["org.batfish.common.BatfishException: Failed to answer question","\tat org.batfish.main.Batfish.answer(Batfish.java:545)","\tat org.batfish.main.Batfish.run(Batfish.java:2106)","\tat org.batfish.main.Driver.lambda$runBatfish$0(Driver.java:247)","\tat java.base/java.lang.Thread.run(Thread.java:829)","Caused by: java.lang.UnsupportedOperationException: Reachability does not yet support PreNatFibLookup","\tat org.batfish.bddreachability.SessionInstrumentation$1.visitPreNatFibLookup(SessionInstrumentation.java:240)","\tat org.batfish.bddreachability.SessionInstrumentation$1.visitPreNatFibLookup(SessionInstrumentation.java:223)","\tat org.batfish.datamodel.flow.PreNatFibLookup.accept(PreNatFibLookup.java:22)","\tat org.batfish.bddreachability.SessionInstrumentation.computeNewSuccessEdges(SessionInstrumentation.java:222)","\tat org.batfish.bddreachability.SessionInstrumentation.computeNewEdges(SessionInstrumentation.java:212)","\tat java.base/java.util.stream.ReferencePipeline$7$1.accept(ReferencePipeline.java:271)","\tat java.base/java.util.Collections$2.tryAdvance(Collections.java:4747)","\tat java.base/java.util.Collections$2.forEachRemaining(Collections.java:4755)","\tat java.base/java.util.stream.ReferencePipeline$Head.forEac

h(ReferencePipeline.java:658)","   at java.base/java.util.stream.ReferencePipeline$7
$1.accept(ReferencePipeline.java:274)","   at java.base/java.util.Iterator.forEachRem
aining(Iterator.java:133)","   at java.base/java.util.Spliterators$IteratorSpliterato
r.forEachRemaining(Spliterators.java:1801)","   at java.base/java.util.stream.Abstrac
tPipeline.copyInto(AbstractPipeline.java:484)","   at java.base/java.util.stream.Abst
ractPipeline.wrapAndCopyInto(AbstractPipeline.java:474)","   at java.base/java.util.s
tream.StreamSpliterators$WrappingSpliterator.forEachRemaining(StreamSpliterators.jav
a:312)","   at com.google.common.collect.CollectSpliterators$FlatMapSpliterator.lambd
a$forEachRemaining$1(CollectSpliterators.java:378)","   at java.base/java.util.Splite
rators$ArraySpliterator.forEachRemaining(Spliterators.java:948)","   at com.google.co
mmon.collect.CollectSpliterators$FlatMapSpliterator.forEachRemaining(CollectSpliterat
ors.java:374)","   at java.base/java.util.stream.AbstractPipeline.copyInto(AbstractPi
peline.java:484)","   at java.base/java.util.stream.AbstractPipeline.wrapAndCopyInto
(AbstractPipeline.java:474)","   at java.base/java.util.stream.StreamSpliterators$Wra
ppingSpliterator.forEachRemaining(StreamSpliterators.java:312)","   at com.google.com
mon.collect.CollectSpliterators$FlatMapSpliterator.lambda$forEachRemaining$1(CollectS
pliterators.java:378)","   at java.base/java.util.Spliterators$ArraySpliterator.forEa
chRemaining(Spliterators.java:948)","   at com.google.common.collect.CollectSpliterat
ors$FlatMapSpliterator.forEachRemaining(CollectSpliterators.java:374)","   at java.ba
se/java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:484)","   at jav
a.base/java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:47
4)","   at java.base/java.util.stream.ReduceOps$ReduceOp.evaluateSequential(ReduceOp
s.java:913)","   at java.base/java.util.stream.AbstractPipeline.evaluate(AbstractPipe
line.java:234)","   at java.base/java.util.stream.ReferencePipeline.collect(Reference
Pipeline.java:578)","   at org.batfish.bddreachability.BDDReachabilityUtils.computeFo
rwardEdgeTable(BDDReachabilityUtils.java:52)","   at org.batfish.bddreachability.BDDR
eachabilityAnalysis.<init>(BDDReachabilityAnalysis.java:70)","   at org.batfish.bddre
achability.BDDReachabilityAnalysisFactory.bddReachabilityAnalysis(BDDReachabilityAnal
ysisFactory.java:1566)","   at org.batfish.bddreachability.BidirectionalReachabilityA
nalysis.computeReturnPassAnalysis(BidirectionalReachabilityAnalysis.java:255)","   at
com.google.common.base.Suppliers$NonSerializableMemoizingSupplier.get(Suppliers.java:
183)","   at org.batfish.bddreachability.BidirectionalReachabilityAnalysis.computeRet
urnPassForwardReachableBdds(BidirectionalReachabilityAnalysis.java:173)","   at com.g
oogle.common.base.Suppliers$NonSerializableMemoizingSupplier.get(Suppliers.java:18
3)","   at org.batfish.bddreachability.BidirectionalReachabilityAnalysis.computeForwa
rdPassStartLocationToReturnPassSuccessBdds(BidirectionalReachabilityAnalysis.java:28
9)","   at com.google.common.base.Suppliers$NonSerializableMemoizingSupplier.get(Supp
liers.java:183)","   at org.batfish.bddreachability.BidirectionalReachabilityAnalysi
s.getResult(BidirectionalReachabilityAnalysis.java:343)","   at org.batfish.main.Batf
ish.bidirectionalReachability(Batfish.java:2885)","   at org.batfish.question.bidirec
tionalreachability.BidirectionalReachabilityAnswerer.answer(BidirectionalReachability
Answerer.java:78)","   at org.batfish.main.Batfish.answer(Batfish.java:542)","   ...
3 more",""]}],"question":{"class":"org.batfish.question.bidirectionalreachability.Bid
irectionalReachabilityQuestion","headers":{"dstIps":"20.0.12.1","dstPorts":"22","srcI
ps":"10.0.45.5","srcPorts":"32875"},"pathConstraints":{"startLocation":"dev5"},"retur
nFlowType":"SUCCESS","differential":false,"includeOneTableKeys":true,"instance":{"des
cription":"Searches for successfully delivered flows that can successfully receive a
response.","instanceName":"__bidirectionalReachability_fa940fa7-15c5-4102-8ed7-e90b47
531484","longDescription":"Performs two reachability analyses, first originating from
specified sources, then returning back to those sources. After the first (forward) pa
ss, sets up sessions in the network and creates returning flows for each successfully
delivered forward flow. The second pass searches for return flows that can be success
fully delivered in the presence of the setup sessions.","orderedVariableNames":["path
Constraints","headers","returnFlowType"],"tags":["dataplane","reachability"],"variabl
es":{"headers":{"description":"Packet header constraints.","displayName":"Headers","f
ields":{"applications":{"optional":true,"type":"applicationSpec"},"dscps":{"optiona
l":true},"dstIps":{"optional":true,"type":"ipSpaceSpec"},"dstPorts":{"optional":tru

e},"ecns":{"optional":true},"flowStates":{"optional":true},"fragmentOffsets":{"optional":true},"icmpCodes":{"optional":true},"icmpTypes":{"optional":true},"ipProtocols":{"optional":true,"type":"ipProtocolSpec"},"packetLengths":{"optional":true},"srcIps":{"optional":true,"type":"ipSpaceSpec"},"srcPorts":{"optional":true},"tcpFlags":{"optional":true}},"optional":false,"type":"headerConstraint","value":{"applications":null,"dscps":null,"dstIps":"20.0.12.1","dstPorts":"22","ecns":null,"fragmentOffsets":null,"icmpCodes":null,"icmpTypes":null,"ipProtocols":null,"packetLengths":null,"srcIps":"10.0.45.5","srcPorts":"32875","tcpFlags":null}},"pathConstraints":{"description":"Constraint the path a flow can take (start/end/transit locations).","displayName":"Path Constraints","fields":{"endLocation":{"optional":true,"type":"nodeSpec"},"forbiddenLocations":{"optional":true,"type":"nodeSpec"},"startLocation":{"optional":true,"type":"locationSpec"},"transitLocations":{"optional":true,"type":"nodeSpec"}},"optional":true,"type":"pathConstraint","value":{"endLocation":null,"forbiddenLocations":null,"startLocation":"dev5","transitLocations":null}},"returnFlowType":{"allowedValues":["FAILURE","MULTIPATH_INCONSISTENT","SUCCESS"],"description":"Specifies the type of return flows to search.","displayName":"Return Flow Type","optional":true,"type":"string","value":"SUCCESS","values":[{"description":"Flows that are successful","name":"SUCCESS"},{"description":"Flows that fail","name":"FAILURE"},{"description":"Flows that succeed or fail depending on the path","name":"MULTIPATH_INCONSISTENT"}]}}}},"status":"FAILURE","summary":{"numFailed":0,"numPassed":0,"numResults":0}}

In [ ]: