CCNx Testrig

Christopher A. Wood

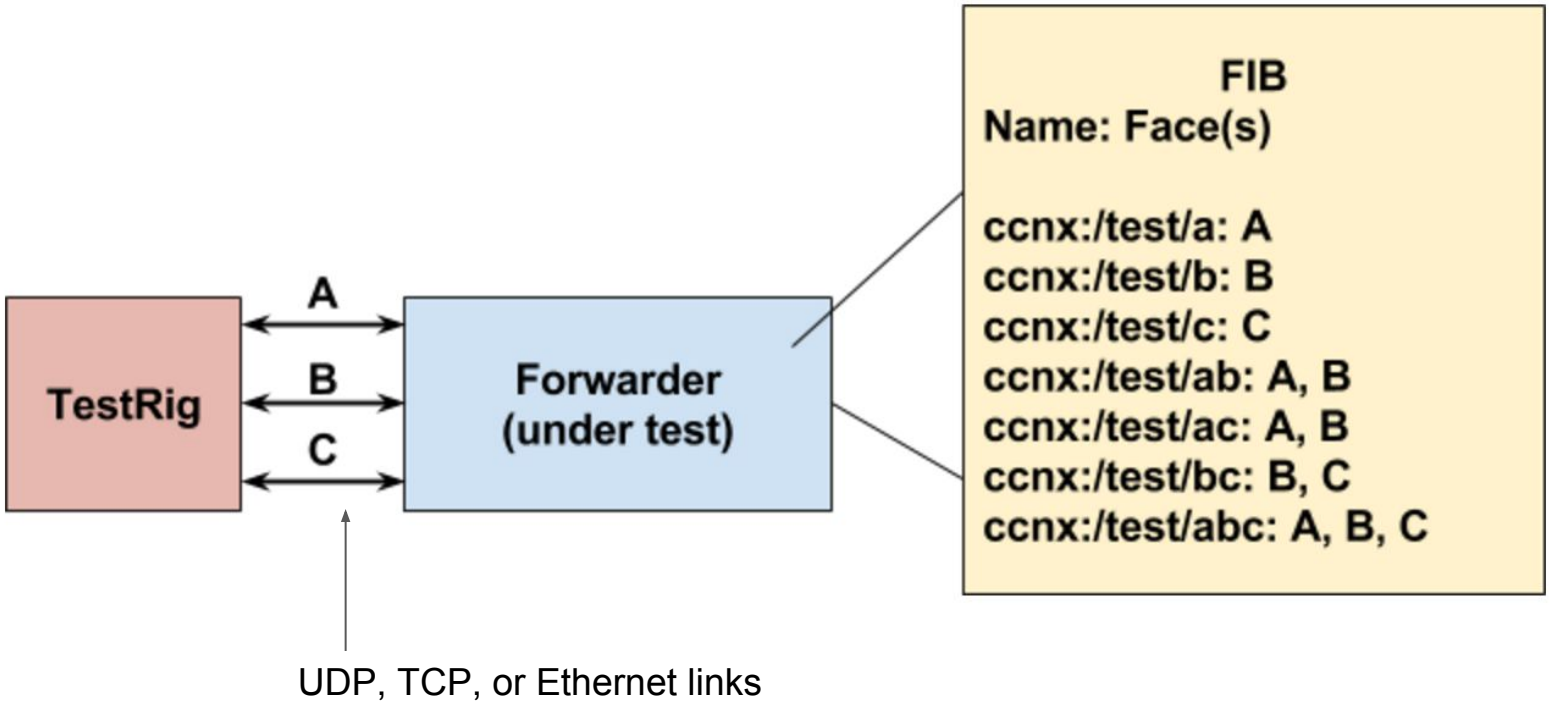
ICNRG Interim Meeting - IETF 96 - Berlin

July 17, 2016

Goals

- Build a system to enable easy forwarder verification testing
  - Did we build this forwarder right?
  - Not a replacement for unit testing, functional testing, etc.
- Cleanly separate strict rules from policy-specific behaviors
- Aim for “scriptable tests”

Testrig and FUT Setup



Example Behavior Test

Test script:

1. Send Interest for “ccnx:/test/**b**/0x12341” to link A
2. Receive Interest on link **B**
3. Send Content with name “ccnx:/test/b/0x12341” to link B
4. Receive Content with name “ccnx:/test/b/0x12341” on link A

Sample Code

```
// Create the packets to send
CCNxName *testName = _createRandomName("ccnx:/test/b");
CCNxInterest *interest = ccnxInterest_Create(testName, 1000, NULL, NULL);
PARCBuffer *testPayload = parcBuffer_Allocate(1024);
CCNxContentObject *content = ccnxContentObject_CreateWithNameAndPayload(testName, testPayload);

// Build the script
CCNxTestrigScript *script = ccnxTestrigScript_Create(testCaseName);
CCNxTestrigScriptStep *step1 = ccnxTestrigScript_AddSendStep(script, interest, CCNxTestrigLinkID_LinkA);
CCNxTestrigScriptStep *step2 = ccnxTestrigScript_AddReceiveOneStep(script, step1,
       ccnxTestrig_GetLinkVector(rig, CCNxTestrigLinkID_LinkB));
CCNxTestrigScriptStep *step3 = ccnxTestrigScript_AddRespondStep(script, step2, content);
CCNxTestrigScriptStep *step4 = ccnxTestrigScript_AddReceiveOneStep(script, step3,
       ccnxTestrig_GetLinkVector(rig, CCNxTestrigLinkID_LinkA));

// Execute it...
CCNxTestrigSuiteTestResult *testCaseResult = ccnxTestrigScript_Execute(script, rig);
```

See https://github.com/PARC/ccnxTestrig

Send and Receive Checks

For every send and receive step:

- Check that message bodies (names, payloads, etc.) are intact
- Check that the hop limits are decremented where appropriate
- Check per-hop headers for equality
- ...

Policy-Specific Check

Test script:

1. Send Interest for “ccnx:/test/**bc**/0x12341” to link A
2. Depending on the policy, receive the Interest on link **B,** link **C**, or both.
3. Reply accordingly...

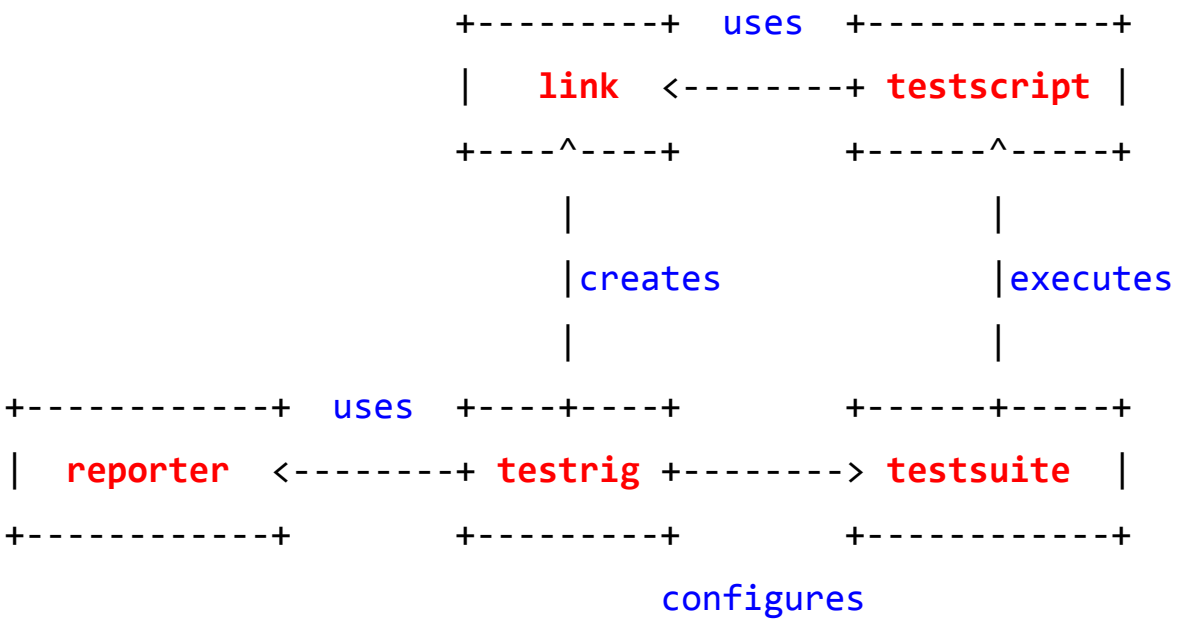
Sample Code

```
// Create the test packets
CCNxName *testName = _createRandomName("ccnx:/test/bc");
CCNxInterest *interest = ccnxInterest_Create(testName, 1000, NULL, NULL);
PARCBuffer *testPayload = parcBuffer_Allocate(1024);
CCNxContentObject *content = ccnxContentObject_CreateWithNameAndPayload(testName, testPayload);

// Build the script
CCNxTestrigScript *script = ccnxTestrigScript_Create(testCaseName);
CCNxTestrigScriptStep *step1 = ccnxTestrigScript_AddSendStep(script, interest, CCNxTestrigLinkID_LinkA);
CCNxTestrigScriptStep *step2 = ccnxTestrigScript_AddReceiveOneStep(script, step1,
       ccnxTestrig_GetLinkVector(rig, CCNxTestrigLinkID_LinkB, CCNxTestrigLinkID_LinkC));
CCNxTestrigScriptStep *step3 = ccnxTestrigScript_AddRespondStep(script, step2, content);
CCNxTestrigScriptStep *step4 = ccnxTestrigScript_AddReceiveOneStep(script, step3,
       ccnxTestrig_GetLinkVector(rig, CCNxTestrigLinkID_LinkA));

// Execute it
CCNxTestrigSuiteTestResult *testCaseResult = ccnxTestrigScript_Execute(script, rig);
```

Code Structure



Future Work

- Extend the suite of tests available
  - Go to https://goo.gl/86NvrV to add, revise, or discuss tests
  - … Or air issues out on the mailing list
- Test with more forwarders
- Build Python bindings to the CCNxTestrig API to make testing easier