

How to send a broadcast request for discovery of SNMP entities?

To send a broadcast SNMP message to a target, simply

1. use a broadcast address as target
2. use SNMP v1, v2c, or v3 with noAuthNoPriv security level (broadcast with authentication does not work obviously)
3. use the asynchronous response processing model of SNMP4J

For example:

Setup a `Smp` instance and let it listen to responses

```
TransportMapping transport = new DefaultUdpTransportMapping();
Session snmp = new Smp(transport);
USM usm = new USM(SecurityProtocols.getInstance(),
    new OctetString(MPv3.createLocalEngineID()), 0);
SecurityModels.getInstance().addSecurityModel(usm);
snmp.listen();
```

Create a broadcast target

```
Address targetAddress = GenericAddress.parse("udp:255.255.255.255/161");
```

- SNMP v1

```
CommunityTarget target = new CommunityTarget();
target.setCommunity(new OctetString("public"));
target.setAddress(targetAddress);
target.setRetries(1);
target.setTimeout(5000);
target.setVersion(SmpConstants.version1);
```

- SNMP v2c

```
CommunityTarget target = new CommunityTarget();
target.setCommunity(new OctetString("public"));
target.setAddress(targetAddress);
target.setRetries(1);
target.setTimeout(5000);
target.setVersion(SmpConstants.version2c);
```

- SNMPv3

```
UserTarget target = new UserTarget();
target.setAddress(targetAddress);
target.setRetries(1);
target.setTimeout(5000);
target.setVersion(SmpConstants.version3);
target.setSecurityLevel(SecurityLevel.NOAUTH_NOPRIV);
target.setSecurityName(new OctetString("unsecUser"));
```

Create the PDU

```
PDU pdu = new PDU();
pdu.add(new VariableBinding(new OID(new int[] {1,3,6,1,2,1,1,1})));
pdu.add(new VariableBinding(new OID(new int[] {1,3,6,1,2,1,1,2})));
pdu.setType(PDU.GETNEXT);
```

Send the message asynchronously

```
class MyResponseListener implements ResponseListener {
    boolean finished = false;
    public void onResponse(ResponseEvent event) {
        System.out.println("Received response PDU is: "+event.getResponse());
        if (event.getResponse() == null) {
            finished = true;
            listener.notify();
        }
    }

    public boolean isFinished() {
        return finished;
    }
};

MyResponseListener listener = new MyResponseListener() {
snmp.sendPDU(pdu, target, null, listener);
try {
    while (!listener.isFinished()) {
        listener.wait(target.getTimeout()*2);
    }
} catch (InterruptedException iex) {
    System.out.println("Request cancelled: "+iex.getMessage());
}
```