Wireshark.org の web サイトにある「Wireshark Developer's Guide」より抜粋した lua ファイルのサンプル

local p_multi = Proto("multi", "MultiProto");	独自プロトコル p_multi の定義 "multi":独自プロトコルの名前、"MultiProto":独自プロトコルの概要

local vs_protos = {
[2] = "mtp2",
[3] = "mtp3",
[4] = "alcap",
[5] = "h248",
[6] = "ranap",
[7] = "rnsap",
[8] = "nbap"
}
	フィールドのパラメーター

local f_proto = ProtoField.uint8("multi.protocol", "Protocol", base.DEC, vs_protos)	独自プロトコル p_multi のフィールドの定義 "multi.protocol":フィールド名略称(フィルターに使用される文字列) "Protocol":フィールド名 base.DEC:表示オプション(10 進数、16 進数など) vs_protos:値に対応する文字列(の配列)
local f_dir = ProtoField.uint8("multi.direction", "Direction", base.DEC, { [1] = "incoming", [0] = "outgoing"}) local f_text = ProtoField.string("multi.text", "Text")	
p_multi.fields = { f_proto, f_dir, f_text }	独自プロトコル p_multi に上記で定義したフィールドを設定
local data_dis = Dissector.get("data")	既存の Dissector を呼び出す
local protos = { [2] = Dissector.get("mtp2"), [3] = Dissector.get("mtp3"), [4] = Dissector.get("alcap"), [5] = Dissector.get("h248"), [6] = Dissector.get("ranap"), [7] = Dissector.get("rnsap"), [8] = Dissector.get("nbap"),	既存の Dissector を呼び出す

```
        [9] = Dissector.get("rrc"),
        [10] = DissectorTable.get("sctp.ppi"):get_dissector(3), -- m3ua
        [11] = DissectorTable.get("ip.proto"):get_dissector(132), -- sctp
}

function p_multi.dissector(buf, pkt, tree)




        local subtree = tree:add(p_multi, buf(0,2))
        subtree:add(f_proto, buf(0,1))
        subtree:add(f_dir, buf(1,1))

        local proto_id = buf(0,1):uint()

        local dissector = protos[proto_id]

        if dissector ~= nil then
                -- Dissector was found, invoke subdissector with a new Tvb,
                -- created from the current buffer (skipping first two bytes).
                dissector:call(buf(2):tvb(), pkt, tree)
        elseif proto_id < 2 then
                subtree:add(f_text, buf(2))
                -- pkt.cols.info:set(buf(2, buf:len() - 3):string())
        else
                -- fallback dissector that just shows the raw data.
                data_dis:call(buf(2):tvb(), pkt, tree)
        end


end

local wtap_encap_table = DissectorTable.get("wtap_encap")
local udp_encap_table = DissectorTable.get("udp.port")

wtap_encap_table:add(wtap.USER15, p_multi)
wtap_encap_table:add(wtap.USER12, p_multi)
udp_encap_table:add(7555, p_multi)
```

既存の DissectorTable を呼び出す

独自プロトコルの Dissector 関数の定義
buf：独自プロトコル以降のデータを格納したもの
pkt：パケット情報
tree：パケット詳細部

tree にサブツリー情報を設定
サブツリーにフィールド値を追加
サブツリーにフィールド値を追加

コメント：パケット情報を GUI の Info 列に表示する設定

既存の DissectorTable を呼び出す

独自プロトコルと既存の Dissector を紐付ける

独自プロトコルとポート 7555 を紐付ける