CocoaPacketAnalyzer Dissector PlugIn

Since Analyzer PlugIns are Cocoa bundles you will need to create a new Cocoa bundle project in Xcode. Add the CPAPlugInManager.framework to it. Dont embed - just link against it. Objective-C interfaces and used protocols are included.

On launch CPA will try to load plugIns found in the various plugIns directories. An instance of the PlugInPrincipal class will be created and initialized using the following init message:

```
public required init!(plugInManager: CPAPlugInManagement!)
```

Further more the PlugInPrincipal class should implement the following class methods:

```
public class func plugInVersion() -> String!
public class func plugInAuthor() -> String?
public class func plugInTitle() -> String?
public class func plugInDescription() -> String!
```

Next the PlugInPrincipal instance should register needed parameters using plugInManager. CPAPlugInManagement will respond to the following messages:

```
func register(plugIn: CPAPlugInStructure, forBundle: Bundle) -> Bool
func register(networkProtocol: CPAPlugInNetworkProtocol, forBundle: Bundle) -> Bool
func unregister(networkProtocol: CPAPlugInNetworkProtocol, forBundle: Bundle) -> Bool
```

Instances of the analyzer class will be used for protocol analysis. The analyzer class has to implement the following instance methods:

```
required public init!(bytes:UnsafeRawPointer!, length:UInt16, packet:Any!)
public func protocolSize() -> UInt16
public func protocolNodes() -> [Any]! (should be CPANode)
public func nextProtocol() -> String?
public func protocolName() -> String!
public func protocolInfo() -> String!
public func hasPayload() -> Bool
```

The preference instance should respond to the following messages:

```
public func usesColor() -> Bool
public func preferencesSheet() -> NSWindow
@objc public func protocolColor() -> NSColor
```

Please have a look at the "TestPlugIn" project. If you need additional infos feel free to contact me!