

DOI: 10.19783/j.cnki.pspc.190041

基于 Wireshark 的智能变电站 SV 报文分析工具设计及实现

栾庆武, 岳蔚, 周永华

(国电南瑞科技股份有限公司深圳分公司, 广东 深圳 518055)

摘要: Wireshark 是一款开源免费、功能强大、可扩展性强的网络通信报文捕捉分析软件, 但是其在解析 SV 报文方面有很多不足, 这限制了 Wireshark 在智能变电站中的使用。通过分析 Wireshark 的设计思想、模块架构、主要的数据结构和处理流程, 以及 SV 报文的帧结构和编码, 给出了基于 Wireshark 开发 SV 报文分析工具的方案和实现过程。并展示了基于该方案开发的 SV 报文分析工具的运行结果, 达到了预期目的。基于开源软件 Wireshark 进行的二次开发, 拓展了 Wireshark 的应用范围, 同时也能吸收开源软件中的精华部分。为开源软件在电力系统中的应用提供一种思路, 对于理解智能变电站通信协议也有重要意义。

关键词: 智能变电站; Wireshark; SV 报文; 开源软件; 协议解析器; SV 波形; SV 异常识别

Design and development of a smart substation SV packet analysis tool based on Wireshark

LUAN Qingwu, YUE Wei, ZHOU Yonghua

(NARI Technology Development Co., Ltd. Shenzhen Branch, Shenzhen 518055, China)

Abstract: Wireshark is an open source, free, powerful and extensible network communication packet capture analysis software, but it has many shortcomings in analyzing SV packets, which limits the use of Wireshark in smart substation. By analyzing the design idea, module architecture, main data structure and processing flow, and the frame structure and coding of SV packet, the design idea and implementation process of Wireshark-based SV packet analysis tool are presented. The operation result of the SV packet analysis tool designed and developed based on this scheme has achieved the intended purpose. The further development based on the open source software Wireshark expands the application scope of Wireshark, and can absorb the essence of open source software. It provides an idea for the application of open source software in power systems and has great significance for understanding the smart substation communication protocol.

This work is supported by Science and Technology Project of State Grid Corporation of China “Research and Application of Substation Automation Equipment Wide-area Operation and Maintenance Techniques” (No. 5210EF170027).

Key words: smart substation; Wireshark; SV packet; open source software; protocol dissector; SV wave; SV abnormal judgment

0 引言

近年来, 国内智能变电站发展迅速^[1], 其中过程层网络是智能变电站的核心, SV 报文又是过程层网络的重要组成部分, 具有流量大、可靠性高、语法语义清晰明确等特点。SV 报文的质量优劣直接关系到继电保护装置的动作行为, 甚至影响到智能变电站的稳定运行^[2-7]。因此, 对 SV 报文的精确分析十分重要。Wireshark 是一款开源、免费、功能强大的网络

通信报文捕捉、嗅探和分析软件, 在智能变电站调试检修中广泛使用^[8]。其内置大量的通信协议解析器, 包括智能变电站中使用到的 MMS 协议、SV 协议、GOOSE 协议、104 协议、PTP 协议、SNTP 协议等。通过 Wireshark 可以对这些协议报文进行捕捉、分析、统计, 快捷方便, 但在分析 SV 协议报文方面, 却有一些缺陷和不足。例如不能解析 SV 报文的通道值, 不能对 SV 报文中语义异常进行判断, 不能直观展示 SV 波形信息, 不能关联 SCD 模型文件, 可视化较差等^[9-11]。这些不足限制了 Wireshark 在智能变电站中的使用。通过分析 Wireshark 的源代码, 总结其设计思想和模块架构, 给出了在 Wireshark 软件基础上完

基金项目: 国家电网公司科技项目资助 (5210EF170027) “变电站自动化设备广域运维技术研究及应用”

善 SV 报文解析的方案,并进行了程序的设计和实现,最终达到预期的效果^[12-15]。

1 Wireshark 简介及源码分析

Wireshark 是开源软件项目,所有的源代码在 GPL 框架下都可以免费使用。因此,全世界志愿者和开发人员都可以很容易在 Wireshark 上添加新的协议解析器,或者将其作为插件整合到自己的程序里。基于以上原因,Wireshark 包含了极其丰富的协议解析器,应用范围广泛。目前的 Wireshark 版本,报文解析器已经有 1 700 多个,几乎包含了所有主流的网络通信协议。

1.1 设计思想

Wireshark 设计初衷是希望各通信专业领域的开发人员都能参与进来,编写其熟悉领域的通信协议解析器代码,丰富项目的使用范围。基于此,Wireshark 设计了一套完整框架,具有良好的可扩展性系统架构,主要体现在以下几个方面。

- 1) 根据 TCP/IP 与 OSI 模型设计的分层解析机制。
- 2) 协议解析器注册机制。
- 3) 统一的报文捕捉、文件处理、界面显示、内存管理机制。

通过上述机制及其提供的丰富的数据类型和接口,使得各通信专业领域的开发人员只需要关注本层次协议的解析,不需要了解具体的实现细节,就可以完成相应通信协议报文的捕捉、分析、展示、统计、查询等。

1.2 模块架构

Wireshark 采用模块化的架构设计,如图 1 所示,模块划分清晰,极大地方便了开发人员灵活的添加扩展^[16]。当前版本主要包含以下几个模块。

GUI(QT/GTK): 图形界面模块,用于人机接口、分析结果展示、参数输入等。Wireshark 提供了基于 QT 和 GTK 两种图形窗口的开发框架。

Core: 系统整合模块,用于连接、调度、串联架构中其他模块,构建完整系统。

Epan: 报文解析核心模块,内存管理、协议解析器注册和调度、协议树生成等全部在该模块中完成,开发人员新添加的协议解析器也放置在该模块中。

Wiretap: 报文文件生成读取模块,支持多种报文格式,例如 pcap、pcapng 等。

Capture: 报文捕捉接口模块。

Dumpcap: 报文实时捕捉模块,通过调用底层函数库实现捕捉。

WinPcap and libpcap: 抓包函数库,完成从网

卡上读取数据,转交上层模块,分别应用于 Windows 系统和 Linux 系统。

Utilities: 公用模块,包括一些公用的字符处理函数、时间处理函数等。

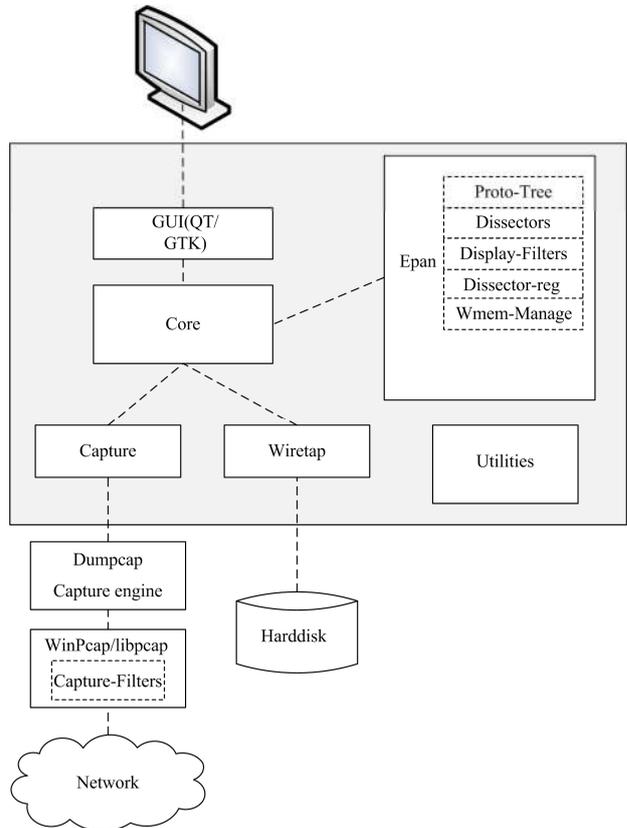


图 1 Wireshark 架构

Fig. 1 Wireshark architecture

1.3 主要数据结构

Wireshark 定义大量的数据结构,其中重要的数据结构有两个,分别是 tvbuff_t 和 proto_tree。

tvbuff_t 结构用来存放当前帧报文每个层次的原始信息,包括该层报文原始数据位置,待解析的报文长度等。所有层次链接在一起形成链表结构。

tvbuff_t 结构主要的数据成员如下。

```

struct tvbuff_t {
...
const guint8 *real_data; //指向报文原始数据
区中待解析位置
guint length; //待解析的报文长度
tvbuff_t *next; //指向链表下一个结点
...
};

```

以图 2 一帧 SV 原始报文为例,在解析过程中,调用 frame_tvbuff_new()函数创建了 tvbuff_t 链表的

头结点, 其中数据域变量 `real_data` 指向该帧报文原始数据区, 即位置 1。length 记录了该帧报文长度, 即 269。其后, 每解析一个层次就会根据情况创建一个链表结点。SV 报文的应用层使用了 BER 编码, 每个 TLV 结构也会创建一个结点, 最终构成的链表结构如图 3 所示。

字节计数	原始报文数据
0000	01 ^{hex} 0C CD 04 00 16 00 0C CD 04 03 01 81 00 80 ^{hex} 00
0010	88 BA 40 ^{hex} 16 00 FB 00 00 00 00 60 81 F0 80 ^{hex} 01 01 ^{hex}
0020	A2 81 EA 30 ^{hex} 81 E7 80 1D 4D ^{hex} 4D 32 32 30 31 41 4D
0030	55 53 56 30 31 2F 4C 4C 4E 30 24 53 56 24 4D 53
0040	56 43 42 30 31 82 02 05 ^{hex} 52 83 04 00 ^{hex} 00 00 01 85
0050	01 01 ^{hex} 87 81 B8 00 ^{hex} 01 04 10 00 00 00 00 00 00 00
0060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070	00 00 00 00 00 FF FF EE DF 00 00 00 00 00 00 00
0080	00 00 00 00 00 FF FE B1 F2 00 00 00 00 00 00 00
0090	00 00 00 00 01 00 00 00 00 00 00 00 01 00 00 00
00A0	00 00 00 00 00 FF FF 21 4C 00 00 00 00 FF FE A9
00B0	61 00 00 00 00 FF FE C7 5C 00 00 00 00 FF FF 7B
00C0	3C 00 00 00 00 FF FF 8C 5E 00 00 00 00 00 00 00
00D0	00 00 00 00 01 00 00 00 00 00 00 00 01 00 00 00
00E0	00 00 00 00 01 00 00 00 00 00 00 00 01 00 00 00
00F0	00 00 00 00 01 00 00 00 00 00 00 00 01 00 00 00
0100	00 00 00 00 01 00 00 00 00 00 00 00 01 00 00 00

图 2 SV 原始报文示例

Fig. 2 SV packet data example

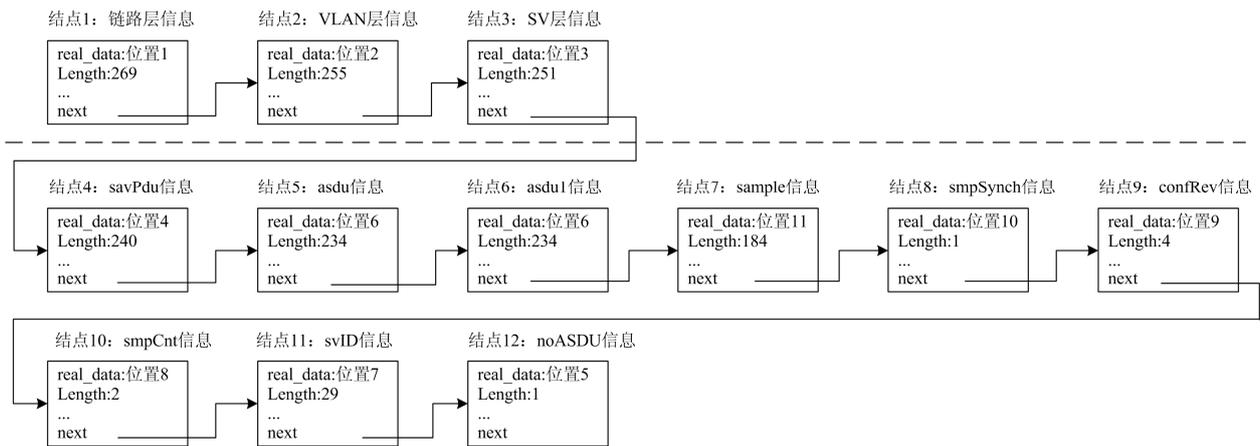


图 3 tvbuff_t 链路结构示例

Fig. 3 Link structure of tvbuff_t example

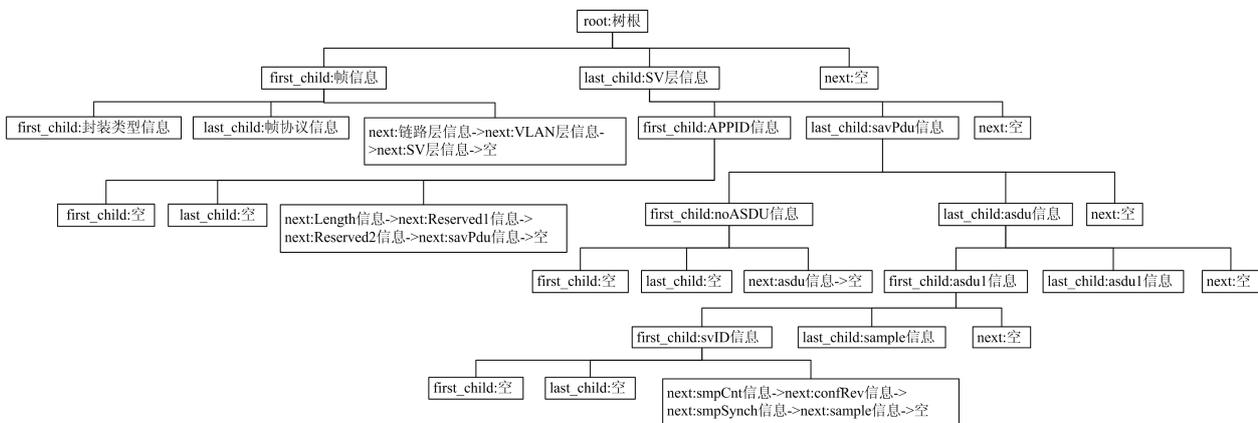


图 4 proto_tree 树结构示例

Fig. 4 Tree structure of proto_tree example

`proto_tree` 结构用来存放当前帧报文每个层次或每个字段解析后的信息, 包括解析结果、字段描述、显示方式、对应的原始报文位置等。每个层次或字段按树形结构组织起来, 供人机界面进行后续展示。`proto_tree` 结构主要的数据成员如下。

```
typedef struct _proto_node {
```

```
...
    struct _proto_node *first_child; //指向第一个子结点
    struct _proto_node *last_child; //指向最后一个子结点
    struct _proto_node *next; //指向邻居结点
...
} proto_node;
```

同样, 以图 2 一帧 SV 原始报文为例, 在解析过程中, 调用 `proto_tree_create_root()` 函数创建了树根, 在逐层解析报文时, 根据报文结构创建子结点和邻居结点, 最终构成的树结构如图 4 所示。

1.4 主要处理流程

Wireshark 解析过程主要包含四个环节：“文件读取或在线捕捉”、“逐帧读取报文”、“逐层解析报文”、“解析展示结果展示”。以解析一份 SV 报文文件为例，主要的解析流程如图 5 所示。

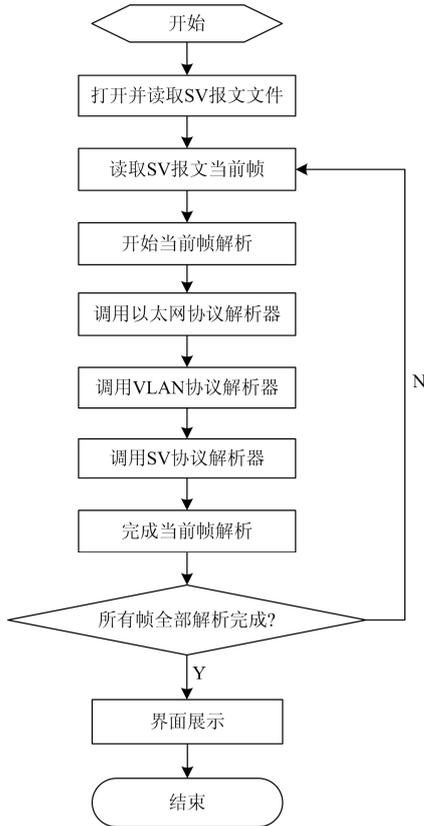


图 5 SV 解析流程

Fig. 5 Process of SV parsing

2 SV 报文简介及帧格式分析

SV 报文，即采样值报文，是在智能变电站中用于传输电流、电压等模拟量的过程层报文，属于 DL/T860.5-2006 标准中规定的“类型 4-原始数据报文”。SV 报文信息交换基于发布/订阅机制。为了访问相关的采样值控制块，标准中规定了两种应用协议子集和传输协议子集的使用组合。两个不同的协议组合分别适用于：(1) 基于 MMS 的 Client/Server 服务；(2) 基于数据链路层的采样值服务。本文所描述的 SV 报文指传输采样值报文，即组合 2，同时遵循 DL/T860.92 标准规定的报文。SV 报文传输使用的协议集映射关系如图 6 所示。可以看出，为了保证 SV 报文的传输效率和稳定，在通信映射层次上，跳过了网络层、传输层、会话层三个层次^[17-20]。

SV 报文链路层帧格式符合 8802-3 规定，其中以以太网类型字段固定为“0X88BA”，如图 7 所示^[17]。

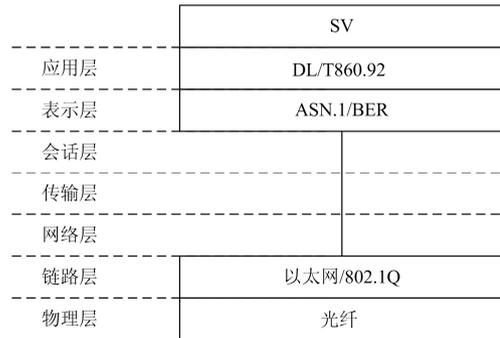


图 6 SV 通信协议栈

Fig. 6 SV protocol stack

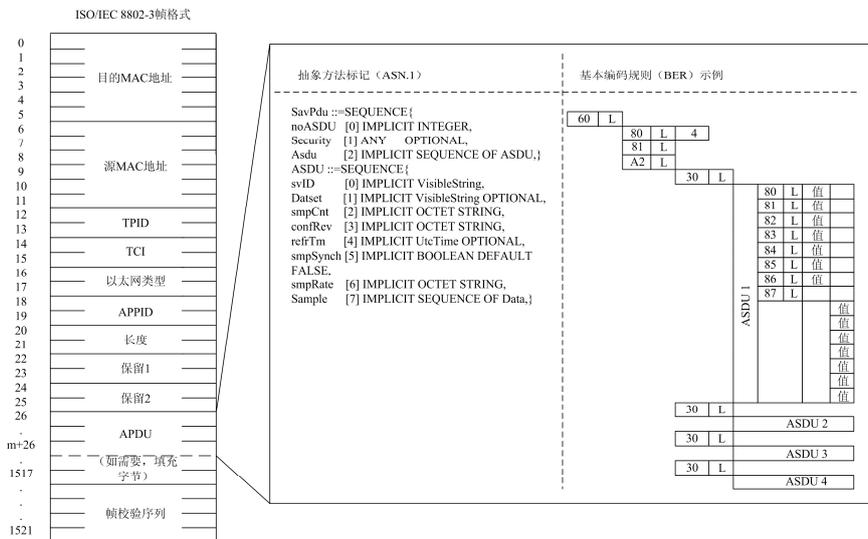


图 7 SV 帧结构及应用层数据编码示例

Fig. 7 SV frame structure and its encode example

SV 报文应用层部分, 即应用协议数据单元 (APDU), 采用 ASN.1 语法描述, 并采用基本编码规则 (BER) 进行编码, 如图 7 所示。BER 编码的基本原则是“类型-长度-值”的三段式结构, 简称 TLV (Type-Length-Value) 结构。任何编码的第一个字段是类型 (T) 字段, 表明了关于标签和编码格式的信息; 长度 (L) 字段定义数值的长度 (按 8 位位组计); 值 (V) 字段表示实际的数值。每一个字段是按 8 位位组对齐的, 并且遵循靠左对齐的原则。编码的值部分是递归的, 即它本身也可以是 TLV 结构, 如图 8 所示^[17]。

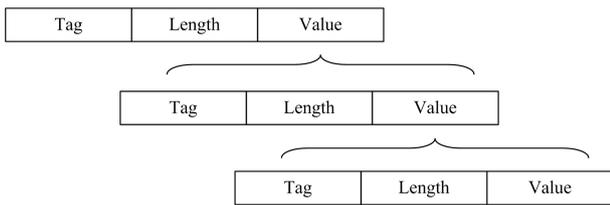


图 8 TLV 结构

Fig. 8 Structure of TLV

3 方案设计及实现

Wireshark 在捕捉解析 SV 报文方面, 主要的缺陷表现在以下四个方面:

- 1) 不能在树形结构中展示 SV 通道数据;
- 2) 不能对 SV 语义上的异常进行识别;
- 3) 不能以波形方式展示 SV 通道数据;
- 4) 不能给每帧报文打上精准的时标。

要弥补上述缺陷, 需要在 Wireshark 框架基础上进行二次开发, 基本思路如下。

- 1) 增加 SCD 配置解析和 SV 通道数据解析功能。
- 2) 根据解析结果进行异常识别和波形展示。
- 3) 裁剪掉 Wireshark 自带的报文捕捉模块, 分析基于网络报文分析记录装置捕捉到的报文。
- 4) 通过比对分析, 选择使用 Qt 作为界面开发框架^[21-22]。

综合上述考虑, 需要增加配置解析、通道数据解析、波形展示和异常识别 4 个模块, 删除 Capture 模块。根据 Wireshark 既定的框架, 新增加的模块分布如图 9 的虚线部分。

综合上述考虑, 需要增加配置解析、通道数据解析、波形展示和异常识别 4 个模块, 删除 Capture 模块。根据 Wireshark 既定的框架, 新增加的模块分布如图 9 的虚线部分。

3.1 配置解析模块

智能变电站 SCD 配置文件包含了全站二次设备的信息, 需要从中解析出 SV 控制块的相关信息, 同时对于 SCD 文件中缺失的 SV 通道相别、变比等信息进行手动配置, 用于波形展示和异常识别, 具体信息见表 1。

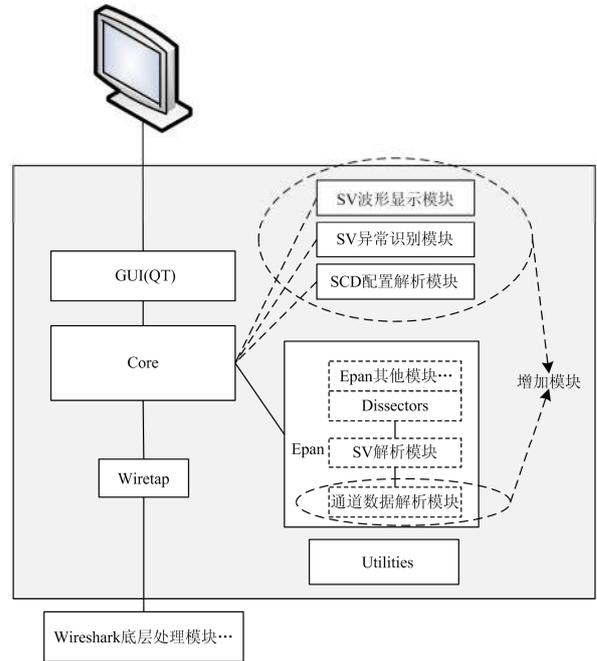


图 9 调整后的架构

Fig. 9 Modified Wireshark architecture

表 1 SV 控制块信息

Table 1 SVCB information

序号	名称	备注	
1	MAC	从 SCD 配置文件中解析	
2	APPID	从 SCD 配置文件中解析	
3	svID	从 SCD 配置文件中解析	
4	DataSet	从 SCD 配置文件中解析	
5	confRev	从 SCD 配置文件中解析	
6	数据集元素个数	从 SCD 配置文件中解析	
7	通道 1 信息	通道描述	从 SCD 配置文件中解析
		通道类型	从 SCD 配置文件中解析
		通道相别	需要手动配置
		一次变比	需要手动配置
		二次变比	需要手动配置
...	通道 n 信息	...	

SCD 配置文件使用 XML 语言编写, 解析 XML 文件相关的库较多, 最终使用 Qt 提供的 QtXml 模块完成对 SCD 配置文件的解析。

3.2 通道数据解析模块

获取 SV 报文中通道数据值是进行 SV 波形展示的基础。通过分析 SV 报文帧格式可知, 通道数据是直接编码在报文中, 非 ANS.1 的 BER 编码。在 Wireshark 中, 只解析至 BER 编码部分, 没有对通道数据进一步解析, 结果如图 10 所示。因此, 需要在 Wireshark 的 SV 协议解析器代码中, 增加通道解析处理模块。

```

# IEC61850 Sampled Values
  APPID: 0x4001
  Length: 262
  Reserved 1: 0x0000 (0)
  Reserved 2: 0x0000 (0)
# savPdu
  noASDU: 1
  seqASDU: 1 item
  # ASDU
    svID: MT1111MUSV01/LLN0.smvcb0
    smpCnt: 2189
    confRef: 1
    smpSynch: local (1)
    seqData: 0000000400000000000022ed88000000000022ed8800000000...

```

图 10 Wireshark 解析 SV 通道数据结果
Fig. 10 Result of Wireshark parsing SV data

SV 的每个通道数据占用了 8 个字节，前 4 个字节为值，后 4 个字节为品质，采用大端字节序码。根据每个通道数据的长度和所有通道数据的总长度，即可以计算出通道数目。根据通道数目，定义数据缓冲区，存储解析出的通道数据。调用 Wireshark 的协议树相关接口，在图 4 的树形结构中增加 SV 通道数据的结点，如图 11 虚线框所示。同时提供数据接口给波形模块，用来完成 SV 通道数据在波形中展示。

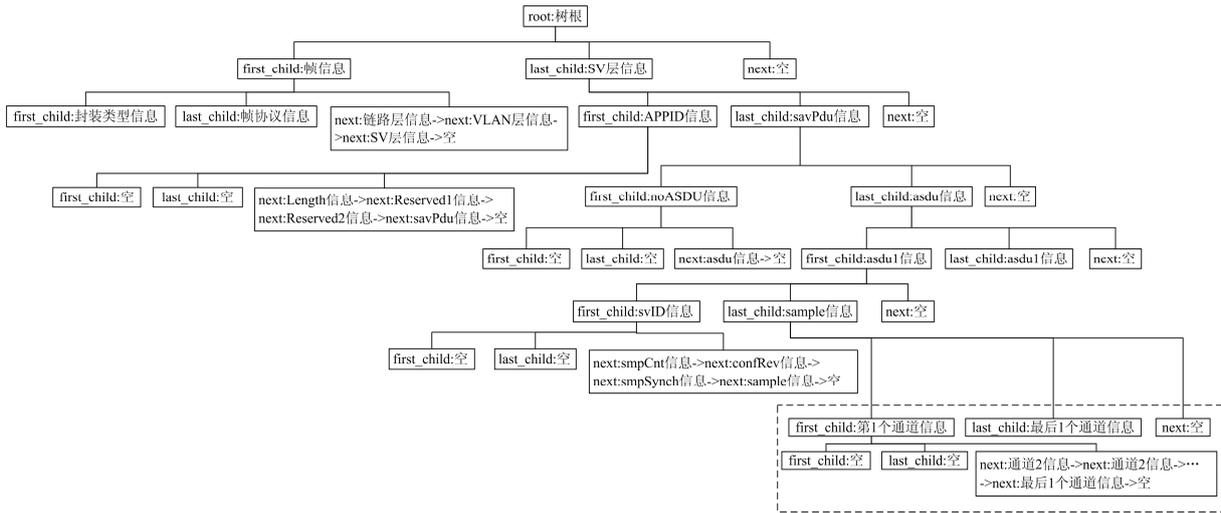


图 11 增加通道数据解析后的协议树
Fig. 11 Proto tree of added SV data parsing

3.3 异常识别模块

Wireshark 只对 SV 报文格式中一些语法错误进行了异常判断，例如不符合 BER 编码规则、长度异常等，没有对 SV 报文语义上的一些异常进行识别。常见的语义上的异常见表 2。

表 2 SV 异常

Table 2 Abnormal SV

序号	名称	判据
1	SV 失步	smpSynch 为 false
2	SV 检修	通道的品质位中的测试位(Test)为 true
3	SV 数据无效	通道的品质位中的数据有效位(Validity)置无效
4	SV 配置不一致	svID、ConfRev、条目个数、组播地址与 SCD 配置不一致
5	SV 丢帧	smpCnt 不连续
6	SV 抖动	报文间隔时间与额定采样率相差较大

为实现上述异常判别，单独定义一个异常识别模块。该模块从 Wireshark 解析模块中读取需要的值，从配置解析模块中读取需要的配置，综合判断即可完成异常识别，架构如图 12 所示。

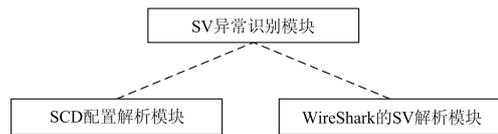


图 12 SV 异常识别模块架构

Fig. 12 SV abnormal judgment architecture

3.4 波形展示模块

通过通道数据解析模块提供的接口，可以获取每个 SV 通道的原始数据。通过配置解析模块可以获取每个 SV 通道的描述信息、类型信息、相别信息、变比信息等。综合两者信息，可以获取采样间隔、采样点数，结合 Wireshark 读取的每个采样点的瞬时值，完成波形的绘制和展示，模块架构如图 13 所示。

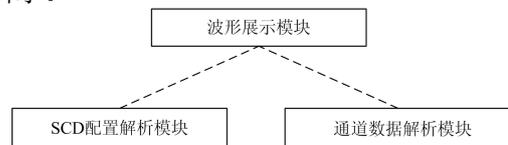


图 13 波形展示模块架构

Fig. 13 Wave display architecture

使用 Qt 提供的 QPainter 模块完成波形绘制。

4 运行结果展示

4.1 SV 报文通道数据解析

SV 报文通道数据解析结果如图 14 所示, 可以解析并展示每帧 SV 报文中通道的名称、原始采样值、品质等信息。

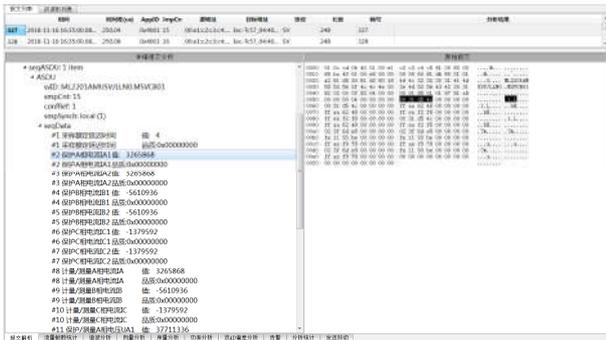


图 14 SV 通道数据解析展示

Fig. 14 Display of SV data parsing

4.2 SV 报文异常识别

SV 报文异常识别结果如图 15—图 20 所示, 可以识别 SV 失步、检修、数据无效、配置不一致、丢帧、抖动等异常。

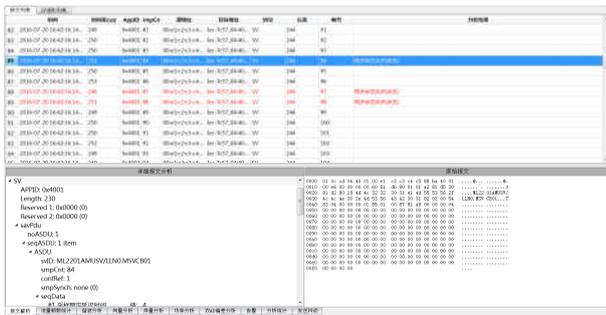


图 15 SV 失步

Fig. 15 SV not synchronised

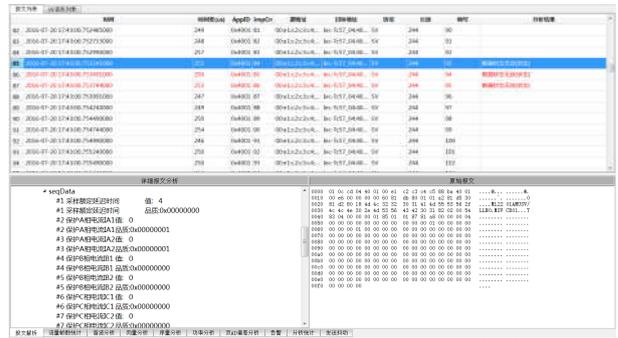


图 17 SV 数据无效

Fig. 17 SV data invalid

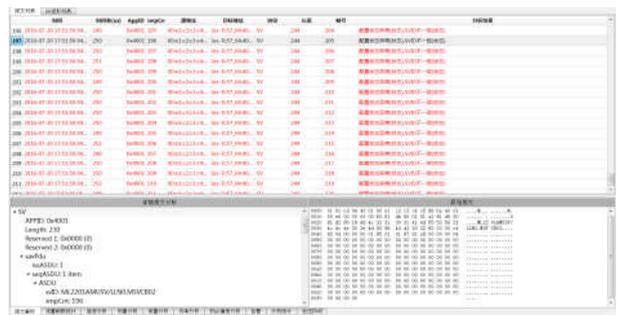


图 18 SV 配置不一致

Fig. 18 SV config inconsistent

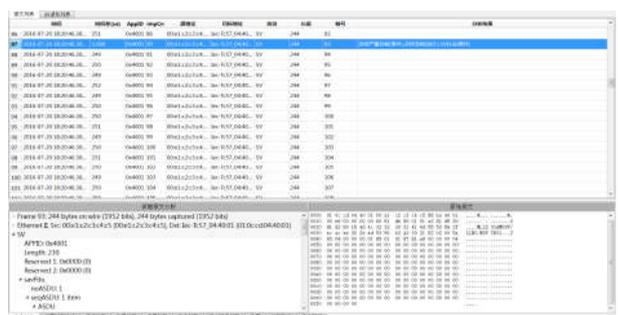


图 19 SV 丢帧

Fig. 19 SV frame lost

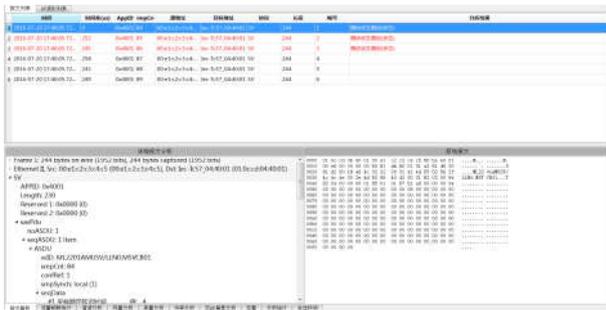


图 16 SV 检修

Fig. 16 SV test

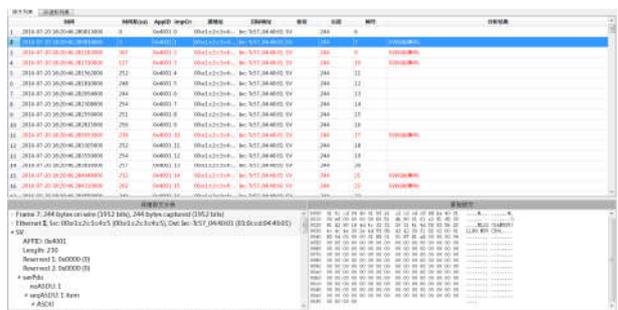


图 20 SV 抖动

Fig. 20 SV frame rate instability

4.3 SV 波形显示

SV 波形展示结果如图 21 所示, 可以直观展示每个数据通道的名称、相别、有效值、采样值、波形等信息, 极大地方便相关专业人员进行报文查看及问题处理。

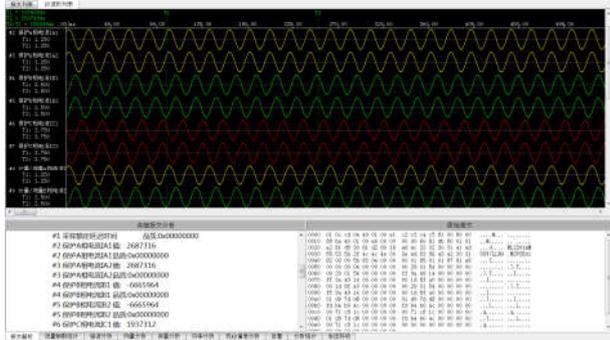


图 21 SV 波形

Fig. 21 SV wave

5 结论

基于 Wireshark 开发的智能变电站 SV 报文分析工具, 能够快速友好地解析 SV 报文, 并进行异常判断、波形显示等。该工具已进行大量的现场使用, 得到了工程调试人员的良好反馈。另一方面, 基于开源软件 Wireshark 进行的二次开发, 拓展了 Wireshark 的应用范围, 同时也能吸收开源软件中的精华部分, 为开源软件在电力系统中的应用提供一种思路。未来, 该工具还可以扩展 MMS/GOOSE 等智能变电站使用的通信协议, 提高工具的使用范围, 为智能变电站的建设发展提供技术支持。

参考文献

[1] 高志远, 黄海峰, 徐昊亮, 等. IEC 61850 应用剖析及其发展探讨[J]. 电力系统保护与控制, 2018, 46(1): 162-169.
GAO Zhiyuan, HUANG Haifeng, XU Haoliang, et al. Discussion on applications of IEC 61850 and its development[J]. Power System Protection and Control, 2018, 46(1): 162-169.

[2] 国家电力监管标准化委员会. 智能变电站技术导则: GB/T 30155—2013[S]. 北京: 中华人民共和国质量监督检验检疫总局, 2013.
State Electricity Regulatory Commission of People's Republic of China. Technical guide for smart substation: GB/T 30155—2013[S]. Beijing: General Administration of Quality Supervision, Inspection and Quarantine of the People's Republic of China, 2013.

[3] 王海柱, 张延旭, 蔡泽祥, 等. 智能变电站过程层网络

信息流潮流模型与计算方法[J]. 电网技术, 2013, 37(9): 2602-2607.
WANG Haizhu, ZHANG Yanxu, CAI Zexiang, et al. Information flow calculation model and method for process bus network in smart substation[J]. Power System Technology, 2013, 37(9): 2602-2607.

[4] 余越, 周春霞, 詹荣荣, 等. 智能变电站继电保护装置网络压力产生原因及测试方法研究[J]. 电力系统保护与控制, 2017, 45(18): 156-162.
YU Yue, ZHOU Chunxia, ZHAN Rongrong, et al. Research on network stress causes and test method of relay protection in smart substation[J]. Power System Protection and Control, 2017, 45(18): 156-162.

[5] 徐春霞, 张艳霞, 葛跃田. 智能变电站过程层网络数据流的分析与研究[J]. 中国电力, 2013, 46(8): 153-159.
XU Chunxia, ZHANG Yanxia, GE Yuetian. Analysis and study on intelligent substation process level network data stream[J]. Electric Power, 2013, 46(8): 153-159.

[6] 陶文伟, 高红亮, 杨贵, 等. 智能变电站过程层冗余组网模式及网络延时累加技术研究[J]. 电力系统保护与控制, 2018, 46(8): 124-129.
TAO Wenwei, GAO Hongliang, YANG Gui, et al. Research on redundant networking mode and network delay accumulation technology of smart substation process layer[J]. Power System Protection and Control, 2018, 46(8): 124-129.

[7] 李旭, 倪传坤, 李宝伟, 等. 基于自适应采样算法及数据处理的过程层设备时延优化设计[J]. 电力系统保护与控制, 2018, 46(18): 164-170.
LI Xu, NI Chuankun, LI Baowei, et al. Optimal design of process level device delay based on self-adaption sampling algorithm and data processing[J]. Power System Protection and Control, 2018, 46(18): 164-170.

[8] 程冀川, 吴占贵. 抓包工具在变电站缺陷处理中的应用[J]. 电力系统通信, 2013, 34(2): 97-100.
CHENG Jichuan, WU Zhanguai. Application of Wireshark in substation defect treatment[J]. Telecommunications for Electric Power System, 2013, 34(2): 97-100.

[9] 胡煜, 张惠刚. 基于 GOOSE 报文解析的变电站配置方法及图形化技术[J]. 南京工程学院学报(自然科学版), 2016, 14(4): 85-90.
HU Yu, ZHANG Huigang. Substation configuration and graphic technology based on GOOSE message[J]. Journal of Nanjing Institute of Technology (Natural Science Edition), 2016, 14(4): 85-90.

[10] 黄志高, 李妍, 李腾, 等. 智能变电站 SCD 文件虚回

- 路自动生成技术的设计和实现[J]. 电力系统保护与控制, 2017, 45(17): 106-111.
- HUANG Zhigao, LI Yan, LI Teng, et al. Design and implementation of automatic generation technology of SCD file virtual circuit in smart substation[J]. Power System Protection and Control, 2017, 45(17): 106-111.
- [11] 孙一民, 刘宏君, 姜健宁, 等. 智能变电站 SCD 文件管控策略完备性分析[J]. 电力系统自动化, 2014, 38(16): 105-109.
- SUN Yimin, LIU Hongjun, JIANG Jianning, et al. Analysis on completeness of substation configuration description file control strategy for smart substation[J]. Automation of Electric Power Systems, 2014, 38(16): 105-109.
- [12] 王丽萍, 孙蕾. 基于 Ethereal 开源代码构建协议解析器的方法研究[J]. 计算机技术与发展, 2007, 17(10): 27-30.
- WANG Liping, SUN Lei. Study on establishment of protocol parser using Ethereal public code[J]. Computer Technology and Development, 2007, 17(10): 27-30.
- [13] 罗青林, 徐克付, 臧文羽, 等. Wireshark 环境下的网络协议解析与验证方法[J]. 计算机工程与设计, 2011, 32(3): 770-773.
- LUO Qinglin, XU Kefu, ZANG Wenyu, et al. Network protocol parser and verification method based on Wireshark[J]. Computer Engineering and Design, 2011, 32(3): 770-773.
- [14] 曹利, 王丹丹. 基于 Wireshark 的协议解析器开发研究和实现[J]. 计算机时代, 2015(6): 13-16.
- CAO Li, WANG Dandan. Research and implementation of protocol parser based on Wireshark[J]. Computer Era, 2015(6): 13-16.
- [15] 范建忠, 马千里. 基于 WINPCAP 的 GOOSE 报文捕获分析工具开发[J]. 电力系统自动化, 2007, 31(23): 52-56.
- FAN Jianzhong, MA Qianli. Development of a GOOSE packet capture and analysis tool based on WINPCAP[J]. Automation of Electric Power Systems, 2007, 31(23): 52-56.
- [16] Wireshark developer's guide for wireshark 2.5[EB/OL]. [2017]. <https://www.wireshark.org/docs/>.
- [17] IEC 61850 communication networks and systems for power utility automation: Ed 2.0—2010[S].
- [18] 冯亚东, 李彦, 王松, 等. IEC 61850-9-2 点对点采样值传输在继电保护中的实现与应用[J]. 电力系统自动化, 2012, 36(2): 82-85.
- FENG Yadong, LI Yan, WANG Song, et al. Realization and application of peer-to-peer IEC 61850-9-2 sampled value transmission in digital substation protections[J]. Automation of Electric Power Systems, 2012, 36(2): 82-85.
- [19] 童晓阳, 廖晨淞, 周立龙, 等. 基于 IEC 61850-9-2 的变电站通信网络仿真[J]. 电力系统自动化, 2010, 34(2): 69-74.
- TONG Xiaoyang, LIAO Chensong, ZHOU Lilong, et al. The simulation of substation communication network based on IEC 61850-9-2[J]. Automation of Electric Power Systems, 2010, 34(2): 69-74.
- [20] 范建忠, 马千里. GOOSE 通信与应用[J]. 电力系统自动化, 2007, 31(19): 85-90.
- FAN Jianzhong, MA Qianli. GOOSE and its application[J]. Automation of Electric Power Systems, 2007, 31(19): 85-90.
- [21] 李文帆, 刘志刚, 伍文城, 等. 基于 Qt 的电力系统地理接线图绘制软件设计[J]. 电力系统自动化, 2013, 37(7): 72-76.
- LI Wenfan, LIU Zhigang, WU Wencheng, et al. Design of power system geographical wiring diagram drawing software based on Qt[J]. Automation of Electric Power Systems, 2013, 37(7): 72-76.
- [22] 汤伟, 李强. Qt/E 的嵌入式 Linux GUI 研究与实现[J]. 计算机应用与软件, 2011, 28(10): 260-263.
- TANG Wei, LI Qiang. Research and realization of Qt/E embedded Linux GUI[J]. Computer Applications and Software, 2011, 28(10): 260-263.

收稿日期: 2019-01-10; 修回日期: 2019-07-04

作者简介:

栾庆武(1982—), 男, 学士, 工程师, 研究方向为电力系统继电保护和智能变电站; E-mail: 109001739@qq.com

岳蔚(1977—), 女, 硕士, 高级工程师, 研究方向为电力系统继电保护和智能变电站;

周永华(1979—), 男, 硕士, 高级工程师, 研究方向为电力系统继电保护和智能变电站。

(编辑 许威)