



中华人民共和国国家标准

GB/T 25931—2010/IEC 61588:2009

网络测量和控制系统的精确 时钟同步协议

Precision clock synchronization protocol for networked
measurement and control systems

(IEC 61588:2009, IDT)

2011-01-14 发布

2011-05-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

目 次

前言	XI
引言	XIII
1 范围	1
2 规范性引用文件	1
3 术语和定义、缩略语	1
3.1 术语和定义	1
3.2 缩略语	6
4 约定	7
4.1 描述性词汇形式的语法	7
4.2 词用法	7
4.3 行为规范表示法	8
5 PTP 系统的数据类型和传输格式	9
5.1 概述	9
5.2 基本数据类型规范	9
5.3 派生数据类型规范	10
5.4 传输格式	12
6 时钟同步模型	13
6.1 概述	13
6.2 关于网络的基本假设和实现推荐	13
6.3 PTP 系统	14
6.4 PTP 报文类	14
6.5 PTP 设备类型	15
6.6 同步综述	24
6.7 PTP 通信综述	31
7 PTP 实体的特性	34
7.1 域	34
7.2 PTP 时标	35
7.3 PTP 通信	35
7.4 PTP 通信媒体	38
7.5 PTP 端口	39
7.6 PTP 设备特性	44
7.7 PTP 时间特性	50
8 PTP 数据集	51
8.1 数据集成员通用规范	51
8.2 普通时钟和边界时钟的数据集	53
8.3 透明时钟的数据集	60
9 PTP 普通时钟和边界时钟	61
9.1 PTP 普通时钟和边界时钟的通用协议要求	61

9.2	状态协议	61
9.3	最佳主时钟算法	67
9.4	最高级时钟	74
9.5	报文处理语义	75
9.6	本地时钟的改变	84
10	透明时钟的 PTP	84
10.1	端到端和点到点透明时钟的通用要求	84
10.2	端到端透明时钟要求	85
10.3	点到点透明时钟要求	85
11	时钟偏移,路径延时,驻留时间和不对称校正	85
11.1	通用规范	85
11.2	普通时钟和边界时钟的时钟偏移的计算	85
11.3	延时请求-响应机制	86
11.4	对等延时机制	87
11.5	对于 PTP 版本 2 事件的透明时钟驻留时间校正	91
11.6	对于 PTP 版本 2 事件报文的不对称校正	93
12	时钟的同步与谐振	95
12.1	谐振	95
12.2	同步	96
13	PTP 报文格式	96
13.1	概述	96
13.2	通用报文格式要求	96
13.3	报文头	96
13.4	报文尾	100
13.5	Announce 报文	100
13.6	Sync 和 Delay_Req 报文	101
13.7	Follow_Up 报文	101
13.8	Delay_Resp 报文	101
13.9	Pdelay_Req 报文	101
13.10	Pdelay_Resp 报文	102
13.11	Pdelay_Resp_Follow_Up 报文	102
13.12	信号报文	103
13.13	管理报文	103
14	TLV 实体规范	103
14.1	通用要求	103
14.2	实验性 TLV	104
14.3	制造商和标准组织扩展 TLV	105
15	管理	105
15.1	概述	105
15.2	PTP 管理机制	106
15.3	管理报文的处理	106
15.4	管理报文格式	106
15.5	管理 TLV	108

16 通用可选特征	126
16.1 单播报文协商(可选)	126
16.2 路径追踪(可选)	129
16.3 备选时标(可选)	131
17 状态配置选项	134
17.1 概述	134
17.2 选项的数据类型	135
17.3 最高级时钟簇(可选)	135
17.4 备选主时钟(可选)	137
17.5 单播发现(可选)	138
17.6 可接受主时钟表(可选)	139
18 兼容性要求	141
18.1 版本 2 与未来版本的兼容性	141
18.2 版本 1 与版本 2 之间的兼容性	141
18.3 报文格式和数据类型	141
18.4 命名变化	146
18.5 对混合版本 1 和版本 2 系统的限制	146
19 一致性	147
19.1 一致性目标	147
19.2 PTP 一致性要求	147
19.3 PTP 行规	148
附录 A (资料性附录) 使用 PTP	150
A.1 概述	150
A.2 物理设计	150
A.3 逻辑设计	151
A.4 组件问题	151
A.5 本地实现问题	152
A.5.1 概述	152
A.5.2 时序问题	152
A.5.3 精度问题	152
A.5.4 时间戳精度	153
A.5.5 稳定性问题	153
A.6 系统实现问题	153
A.7 性能	154
A.8 有助于一致性测试的推荐	154
A.9 在单播网络或具有非 PTP 网桥和路由器网络中的实现推荐	155
A.9.1 概述	155
A.9.2 单播模型中的边界时钟和透明时钟	155
A.9.3 单播选项	155
A.9.4 单播一致性	155
附录 B (资料性附录) PTP 中的时间戳和历元	157
B.1 总则	157
B.2 UTC、TAI 和 PTP 历元	157

B.3 标准时间源	158
附录 C (资料性附录) 驻留时间和不对称校正实例	159
C.1 概述	159
C.2 用延时请求-响应机制计算	159
C.2.1 给出驻留时间校正的主时钟、端到端透明时钟和所有单步从时钟	159
C.2.2 给出驻留时间和不对称计算的主时钟、点到点透明时钟和所有单步从时钟	160
C.2.3 给出驻留时间和不对称计算的双步主时钟、端到端透明时钟和单步从时钟	162
C.2.4 给出驻留时间和不对称计算的主时钟、端到端透明双步时钟和单步从时钟	163
C.2.5 给出驻留时间和不对称计算的单步主时钟、端到端透明双步时钟和单步从时钟	165
C.3 使用对等延时机制的计算	166
C.3.1 给出驻留时间和不对称计算的单步对等请求方时钟、端到端透明时钟和对等响应方时钟	166
C.3.2 给出驻留时间和不对称计算的单步对等请求方时钟、双步端到端透明时钟和单步对等响应方时钟	167
C.3.3 给出驻留时间和不对称计算的单步对等请求方时钟、双步端到端透明时钟、双步对等响应方时钟——选项 1	169
C.3.4 给出驻留时间和不对称计算的单步对等请求方时钟、双步端到端透明时钟和双步对等响应方时钟——选项 2	170
C.3.5 显示驻留时间和不对称计算的单步对等请求方时钟、单步端到端透明时钟和双步对等响应方时钟——选项 2	171
C.3.6 给出从主时钟到从时钟传输时间的单步对等主时钟、双步点到点透明时钟和单步对等从时钟	172
附录 D (规范性附录) PTP 在 IPV4 用户数据包协议上的传输	174
D.1 概述	174
D.2 UDP 端口号	174
D.3 IPv4 多播地址	174
D.4 transportSpecific 字段值	174
D.5 可选值	175
D.6 IPv4 选项	175
D.7 协议地址	175
附录 E (规范性附录) PTP 在 IPv6 用户数据协议(UDP)上的传输	176
E.1 概述	176
E.2 UDP 端口号	176
E.3 IPv6 多播地址	176
E.4 transportSpecific 字段值	177
E.5 可选值	177
E.6 协议地址	177
附录 F (规范性附录) PTP 在 IEEE 802.3/Ethernet 上的传输	178
F.1 概述	178
F.2 Ethertype	178
F.3 多播 MAC 地址	178
F.4 transportSpecific 字段值	178
F.5 可选值	179
F.6 协议地址	179

附录 G (规范性附录) PTP 在 DeviceNET 上的传输	180
G.1 协议	180
G.2 事件报文时间戳点	180
G.3 clockIdentity	180
G.4 PTP 报文格式	181
G.5 PTP 的 DeviceNet 寻址	181
G.6 transportSpecific 字段值	181
附录 H (规范性附录) PTP 在 ControlNET 上的传输	182
H.1 协议	182
H.2 clockIdentity	182
H.3 PTP 报文格式	182
H.4 PTP 的 ControlNet 寻址	182
H.5 transportSpecific 字段值	182
附录 I (规范性附录) PTP 在 IEC 61158 类型 10 上的传输	183
I.1 背景	183
I.2 报文规范	184
I.3 IEC 61158 类型 10 的 DLPDU	184
I.3.1 DLPDU 的抽象句法	184
I.3.2 DLPDU 字段目标地址的编码	184
I.3.3 字段 LT 的编码	185
I.3.4 字段 FrameID 的编码	185
I.3.5 编码规范	186
附录 J (规范性附录) 缺省 PTP 行规	189
J.1 概述	189
J.2 通用要求	189
J.3 延时请求-响应缺省 PTP 行规	189
J.3.1 标识	189
J.3.2 PTP 属性值	189
J.3.3 PTP 选项	189
J.3.4 时钟物理要求	189
J.4 点到点缺省 PTP 行规	190
J.4.1 标识	190
J.4.2 PTP 属性值	190
J.4.3 PTP 选项	190
J.4.4 时钟物理要求	190
附录 K (资料性附录) 安全协议(实验性)	191
K.1 概述	191
K.2 协议综述	191
K.3 通用要求	192
K.4 挑战-响应交换	192
K.5 安全联盟更新交换	193
K.6 完整性校验值(ICV)测试	193
K.7 安全联盟查找	194

K. 8	keyId 检验	194
K. 9	重放保护机制	194
K. 10	挑战-响应检验	194
K. 11	共享密钥分发	195
K. 12	密钥产生	195
K. 13	安全数据集	195
K. 13.1	概述	195
K. 13.2	密钥列表	195
K. 13.3	安全联盟	196
K. 13.4	defaultDS 数据集安全变量	198
K. 14	协议操作	198
K. 14.1	概述	198
K. 14.2	接收报文处理	198
K. 14.3	挑战处理	200
K. 14.4	安全传输处理	201
K. 14.5	安全事件处理	202
K. 14.6	安全透明时钟处理规则	204
K. 15	AUTHENTICATION TLV	205
K. 15.1	概述	205
K. 15.2	tlvType	206
K. 15.3	lengthField	206
K. 15.4	lifetimeId(UInteger 16)	206
K. 15.5	replayCounter(UInteger 32)	206
K. 15.6	keyId(UInteger 16)	206
K. 15.7	algorithmId(UInteger 8)	206
K. 15.8	填充(Octet[M])	206
K. 15.9	ICV(Octet[N])	206
K. 16	AUTHENTICATION_CHALLENGE TLV	207
K. 16.1	概述	207
K. 16.2	tlvType	207
K. 16.3	lengthField	207
K. 16.4	challengeType(UInteger 8)	207
K. 16.5	requestNonce(UInteger 32)	207
K. 16.6	responseNonce(UInteger 32)	207
K. 17	SECURITY_ASSOCIATION_UPDATE TLV	207
K. 17.1	概述	207
K. 17.2	tlvType	208
K. 17.3	lengthField	208
K. 17.4	addressType(UInteger 8)	208
K. 17.5	nextKeyId(UInteger 16)	208
K. 17.6	nextLifetimeId(UInteger 16)	208
附录 L (资料性附录)	累积频标因子偏移的传输(实验性)	209
L. 1	概述	209

L.2 使用累积频标因子的频率补偿方案的描述	209
L.3 累积频标因子偏移的通用规范	210
L.4 CUM_FREQ_SCALE_FACTOR_OFFSET TLV 规范	211
L.4.1 tlvType	211
L.4.2 cumulativeFrequencyScaleFactorOffset	211
参考文献	212

图1 Mealy 型状态转换图	8
图2 普通时钟模型	16
图3 边界时钟模型	17
图4 端到端透明时钟模型	18
图5 端到端驻留时间校正模型	19
图6 组合的普通时钟和端到端透明时钟	21
图7 点到点透明时钟模型	22
图8 点到点驻留时间和链路延时校正模型	23
图9 组合的普通时钟和点到点透明时钟	25
图10 简单主从时钟层次结构	26
图11 裁剪的网状拓扑	27
图12 基本同步报文交换	28
图13 链路延时测量	29
图14 时间戳产生模型	30
图15 层次拓扑	31
图16 线型拓扑	32
图17 多连接拓扑	33
图18 完全不同技术的桥接	34
图19 内部延时常数定义	37
图20 传输不对称	39
图21 端口模型	40
图22 方差对数标定值滞后	49
图23 完整实现的状态机	63
图24 slave-only 实现的状态机	64
图25 STATE_DECISION_EVENT 逻辑	65
图26 状态判定算法	70
图27 数据集比较算法,第1部分	71
图28 数据集比较算法,第2部分	72
图29 Announce 报文接收逻辑	77
图30 Sync 报文接收逻辑	78
图31 Follow_Up 报文接收逻辑	79
图32 Delay_Req 报文接收逻辑	80
图33 Delay_Resp 报文接收逻辑	81
图34 延时请求-响应路径长度的测量	86
图35 对等延时链路测量	88
图36 允许的混合系统配置	147

图 37 行规印刷形式	149
图 C.1 不带不对称校正的主时钟、端到端时钟和单步从时钟	160
图 C.2 带不对称校正的主时钟、端到端时钟和单步从时钟	161
图 C.3 带不对称校正的双步主站、端到端透明时钟和单步从时钟	162
图 C.4 带不对称校正的主时钟、端到端透明双步时钟和单步从时钟	164
图 C.5 带不对称校正的单步主时钟、双步端到端透明时钟和单步从时钟	165
图 C.6 带不对称校正的单步对等响应方时钟、端到端透明时钟和对等请求方时钟	167
图 C.7 带不对称校正的单步对等响应方时钟、双步端到端透明时钟和单步对等请求方时钟	168
图 C.8 带不对称校正的双步对等响应方时钟、双步端到端透明时钟和单步对等请求方时钟： 选项 1	169
图 C.9 带不对称校正的双步对等响应方时钟、双步端到端透明时钟和单步对等请求方时钟： 选项 2	170
图 C.10 带不对称校正的双步对等响应方时钟、单步端到端透明时钟和单步对等请求方时钟： 选项 2	171
图 C.11 带时间计算的单步对等主时钟、双步点到点透明时钟和单步对等从时钟	173
图 G.1 事件报文时间戳点	180
图 I.1 PROFINET 区域与其他域结合	183
图 K.1 PTP 安全报文处理	199
图 K.2 挑战处理	201
图 K.3 安全传输处理	202
图 K.4 安全事件处理	204
表 1 基本 PTP 数据类型	9
表 2 domainNumber 取值	35
表 3 networkProtocol 枚举	38
表 4 Non-EUI-64addressTechnology 枚举	42
表 5 clockClass 规范	45
表 6 clockAccuracy 枚举	46
表 7 timeSource 枚举	47
表 8 PTP 状态枚举	58
表 9 延时机制枚举	59
表 10 PTP portState 定义	62
表 11 边界时钟事件的可用性	67
表 12 数据集比较算法的信息源	70
表 13 状态判定代码 M1 和 M2 的更新	72
表 14 状态判定代码 M3 的更新	73
表 15 状态判定代码 P1 和 P2 的更新	73
表 16 状态判定代码 S1 的更新	73
表 17 源标识符比较	76
表 18 公共报文头	96
表 19 messageType 字段的值	97
表 20 flagField 的值	98

表 21	correctionField 语义	98
表 22	sequenceId 值的例外情况的参考	99
表 23	controlField 枚举	99
表 24	字段 logMessageInterval 的值	99
表 25	Announce 报文字段	100
表 26	Sync 和 Delay_Req 报文的字段	101
表 27	Follow_Up 报文字段	101
表 28	Delay_Resp 报文字段	101
表 29	Pdelay_Req 报文字段	102
表 30	Pdelay_Resp 报文字段	102
表 31	Pdelay_Resp_Follow_Up 报文字段	102
表 32	信号报文的接受	103
表 33	信号报文字段	103
表 34	tlvType 值	104
表 35	组织特定 TLV 字段	105
表 36	管理报文的接受	106
表 37	管理报文字段	107
表 38	actionField 的值	107
表 39	管理 TLV 字段	109
表 40	managementId 值	109
表 41	CLOCK_DESCRIPTION 管理 TLV 数据字段	111
表 42	ClockType 规范	112
表 43	USER_DESCRIPTION 管理 TLV 数据字段	113
表 44	INITIALIZE 管理 TLV 数据字段	114
表 45	initializationKey 枚举	114
表 46	故障日志 severityCode 枚举	115
表 47	FAULT_LOG 管理 TLV 数据字段	115
表 48	TIME 管理 TLV 数据字段	116
表 49	CLOCK_ACCURACY 管理 TLV 数据字段	116
表 50	DEFAULT_DATA_SET 管理 TLV 数据字段	117
表 51	Priority 1 管理 TLV 数据字段	117
表 52	Priority 2 管理 TLV 数据字段	118
表 53	DOMAIN 管理 TLV 数据字段	118
表 54	SLAVE_ONLY 管理 TLV 数据字段	118
表 55	CURRENT_DATA_SET 管理 TLV 数据字段	118
表 56	PARENT_DATA_SET 管理 TLV 数据字段	119
表 57	TIME_PROPERTIES_DATA_SET 管理 TLV 数据字段	120
表 58	UTC_PROPERTIES 管理 TLV 数据字段	120
表 59	TRACEABILITY_PROPERTIES 管理 TLV 数据字段	121
表 60	TIMESCALE_PROPERTIES 管理 TLV 数据字段	121
表 61	PORT_DATA_SET 管理 TLV 数据字段	121
表 62	LOG_ANNOUNCE_INTERVAL 管理 TLV 数据字段	122
表 63	ANNOUNCE_RECEIPT_TIMEOUT 管理 TLV 数据字段	122

表 64	LOG_SYNC_INTERVAL 管理 TLV 数据字段	122
表 65	DELAY_MECHANISM 管理 TLV 数据字段	123
表 66	LOG_MIN_PDELAY_REQ_INTERVAL 管理 TLV 数据字段	123
表 67	VERSION_NUMBER 管理 TLV 数据字段	123
表 68	TRANSPARENT_CLOCK_DEFAULT_DATA_SET 管理 TLV 数据字段	123
表 69	PRIMARY_DOMAIN 管理 TLV 数据字段	124
表 70	TRANSPARENT_CLOCK_PORT_DATA_SET 管理 TLV 数据字段	124
表 71	MANAGEMENT_ERROR_STATUS 的 TLV 格式	125
表 72	managementErrorId 枚举	125
表 73	REQUEST_UNICAST_TRANSMISSION TLV 格式	127
表 74	GRANT_UNICAST_TRANSMISSION TLV 格式	127
表 75	CANCEL_UNICAST_TRANSMISSION TLV 格式	128
表 76	ACKNOWLEDGE_CANCEL_UNICAST_TRANSMISSION TLV 格式	129
表 77	UNICAST_NEGOTIATION_ENABLE 管理 TLV 数据字段	129
表 78	PATH_TRACE TLV 格式	130
表 79	PATH_TRACE_LIST 管理 TLV 数据字段	131
表 80	PATH_TRACE_ENABLE 管理 TLV 数据字段	131
表 81	ALTERNATE_TIME_OFFSET_INDICATOR TLV 格式	132
表 82	ALTERNATE_TIME_OFFSET_ENABLE 管理 TLV 数据字段	133
表 83	ALTERNATE_TIME_OFFSET_NAME 管理 TLV 数据字段	133
表 84	ALTERNATE_TIME_OFFSET_MAX_KEY 管理 TLV 数据字段	133
表 85	ALTERNATE_TIME_OFFSET_PROPERTIES 管理 TLV 数据字段	134
表 86	GRANDMASTER_CLUSTER_TABLE 管理 TLV 数据字段	136
表 87	备选主时钟属性	137
表 88	ALTERNATE_MASTER 管理 TLV 数据字段	138
表 89	UNICAST_MASTER_TABLE 管理 TLV 数据字段	139
表 90	UNICAST_MASTER_MAX_TABLE_SIZE 管理 TLV 数据字段	139
表 91	可接受主时钟表选项的操作	140
表 92	ACCEPTABLE_MASTER_TABLE 管理 TLV 数据字段	140
表 93	ACCEPTABLE_MASTER_MAX_TABLE_SIZE 管理 TLV 数据字段	141
表 94	ACCEPTABLE_MASTER_TABLE_ENABLED 管理 TLV 数据字段	141
表 95	版本 1 的 Stratum 到版本 2 的 clockClass	142
表 96	版本 2 的 clockClass 到版本 1 的 Stratum	142
表 97	版本 1 的 grandmasterIsPreferred 字段到版本 2 的 Priority 1 字段的转换	142
表 98	版本 2 的 Priority 1 字段到版本 1 的转换	142
表 99	版本 1 的 clockIdentifier 到版本 2 的 clockAccuracy 的映射	143
表 100	版本 2 的 clockAccuracy 到版本 1 的 clockIdentifier 的映射	143
表 101	版本 1 的 grandmasterIsBoundaryClock 字段到版本 2 的 Priority 2 字段的转换	143
表 102	版本 2 的 Priority 2 字段到版本 1 的 grandmasterIsBoundaryClock 字段的转换	143
表 103	版本 1 的 control 和版本 2 的 messageType 字段的映射	143
表 104	flagField 从版本 1 到版本 2 的转换	144
表 105	flagField 从版本 2 到版本 1 的转换	145
表 106	与版本 1 无对应部分的版本 2 的字段	146

表 107	与版本 2 无对应部分的版本 1 的字段	146
表 108	名称对应	146
表 109	混合系统限制	147
表 B.1	时标间的关系	158
表 C.1	图 C.1 关键值的解释	160
表 C.2	图 C.2 关键值的解释	161
表 C.3	图 C.3 关键值的解释	163
表 C.4	图 C.4 关键值的解释	164
表 C.5	图 C.5 关键值的解释	166
表 C.6	图 C.6 关键值的解释	167
表 C.7	图 C.7 关键值的解释	168
表 C.8	图 C.8 关键值的解释	169
表 C.9	图 C.9 关键值的解释	170
表 C.10	图 C.10 关键值的解释	172
表 C.11	图 C.11 关键值的解释	173
表 D.1	IPv4 多播地址	174
表 D.2	transportSpecific 字段值	175
表 E.1	IPv6 多播地址	176
表 F.1	多播 MAC 地址	178
表 F.2	以太网传输特定字段	179
表 G.1	DeviceNet 的 clockIdentity 八位位组 0~7	180
表 G.2	用于所有 PTP 报文数据包的 DeviceNet 报文头	181
表 H.1	ControlNet 的 clockIdentity 八位位组 2~7	182
表 I.1	报文映射	184
表 I.2	IEEE 802.3DLPDU 语法	184
表 I.3	多播 MAC 地址	185
表 I.4	LT(长度/类型)	185
表 I.5	FrameID	186
表 I.6	参数和属性名称的映射	186
表 I.7	从 PTP 版本 2 到 PROFINET 的 flagField 转换	188
表 K.1	flagField.SECURE 标志	192
表 K.2	AUTHENTICATION TLV	205
表 K.3	algorithmId 值	206
表 K.4	ICV 和填充长度	206
表 K.5	AUTHENTICATION_CHALLENGE TLV	207
表 K.6	challengeType 值	207
表 K.7	SECURITY_ASSOCIATION_UPDATE TLV	208
表 K.8	addressType 值	208
表 L.1	CUM_FREQ_SCALE_FACTOR_OFFSET TLV 格式	211

前 言

本标准等同采用 IEC 61588:2009《网络测量和控制系统的精确时钟同步协议》(英文版)。本标准与 IEC 61588:2009 在技术内容上没有差异,仅对文本按 GB/T 1.1—2000 的要求进行编辑性修改。

所做的编辑性修改主要有:

- 删除 IEC 61588:2009 的前言,按 GB/T 1.1—2000 重新编写了本标准的前言;
- 删除 IEC 61588:2009 的引言,根据 IEC 61588:2009 的引言、第 1 章概述以及其他技术内容形成本标准的引言;
- 删除 IEC 61588:2009 的第 1 章概述,重新编写本标准的第 1 章范围;
- 根据标准制定要求,对部分脚注进行修改或删除;
- 将原文中的“IEEE Std 1588:2008”用“本标准”代替,将原文中的 IEEE 1588 用 IEC 61588 代替;
- 将原文中的“本标准版本 1”和“本标准版本 2”分别用“PTP 版本 1”和“PTP 版本 2”代替。PTP 版本 1 对应于 IEC 61588:2004,PTP 版本 2 对应于本标准;
- 根据 GB/T 1.1 进行编辑性修改。

本标准的附录 D、附录 E、附录 F、附录 G、附录 H、附录 I、附录 J 为规范性附录,附录 A、附录 B、附录 C、附录 K、附录 L、附录 M 为资料性附录。

本标准由中国机械工业联合会提出。

本标准由全国工业过程测量和控制标准化技术委员会归口。

本标准起草单位:机械工业仪器仪表综合技术经济研究所。

本标准主要起草人:刘丹、王麟琨、谢素芬、丁露、潘长清。

引 言

本标准规定了在使用网络通信、本地计算和分布式对象等技术实现的测量和控制系统中,实现时钟精确同步的精确时间协议(Precision Time Protocol, PTP),以及支持该精确时间协议所需的节点、系统和通信特性。

本标准主要阐述了以下几部分内容:

——时间同步模型

提供了 PTP 协议的基本模型。给出了 PTP 系统、报文类、设备类型、同步机制和 PTP 协议通信等主要方面的概述。阐明了 PTP 设备类型模型和内部相关的校正过程、请求-响应延时机制和对等延时机制下的报文交换以及延时测量、时间戳产生、PTP 通信拓扑等。

——PTP 数据类型、数据集和报文格式

定义了 PTP 系统中使用的基本数据类型和派生数据类型。给出了数据集的通用规范,以及普通时钟、边界时钟和透明时钟的数据集规范。规定了通用报文和事件报文的格式。

——PTP 协议要求和校正

分别针对普通时钟和边界时钟,以及透明时钟规定了通用协议要求。给出了实现普通时钟和边界时钟的协议状态机和最佳主时钟算法。并针对时钟同步过程中的时钟偏移、路径延时、驻留时间和不对称规定了校正机制。

——管理要求

规定了管理机制和管理报文处理过程,并针对不同实现规定了通用可选特性,包括:单播报文协商、路径追踪和备选时标等。此外为协同使用最佳主时钟算法,还规定了可选状态配置以提高性能或对端口状态选择执行更多控制。

——兼容性和一致性

为保证实现不同 PTP 协议版本的设备兼容性,对报文格式、数据类型和命名等规定了兼容性要求。也规定了一致性要求,以保证不同设备的可互操作性。

——PTP 在不同通信网络上的传输及缺省 PTP 行规

规定了 PTP 在 UDP/IPV4、UDP/IPV6、IEEE 802.3/Ethernet、DeviceNET、ControlNET 和 IEC 61158 类型 10 上的传输。规定了本标准使用的缺省 PTP 行规。

网络测量和控制系统的精确 时钟同步协议

1 范围

本标准规定了在使用网络通信、本地计算和分布式对象等技术实现的测量和控制系统中,实现时钟精确同步的精确时间协议,以及支持该精确时间协议所需的节点、系统和通信特性。

本标准适用于工业自动化、电信、制造、军事以及电力等领域的测量和控制系统。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本标准,然而,鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本标准。

IEC 61158-3-2:2007 工业通信网络 现场总线规范 第 3-2 部分:数据链路层服务定义 类型 2 元素

IEC 61158-4-2:2007 工业通信网络 现场总线规范 第 4-2 部分:数据链路层协议规范 类型 2 元素

IEC 61158-5-2:2007 工业通信网络 现场总线规范 第 5-2 部分:应用层服务定义 类型 2 元素

IEC 61158-5-10:2007 工业通信网络 现场总线规范 第 5-10 部分:应用层服务定义 类型 10 元素

IEC 61158-6-2:2007 工业通信网络 现场总线规范 第 6-2 部分:应用层协议规范 类型 2 元素

IEC 61158-6-10:2007 工业通信网络 现场总线规范 第 6-10 部分:应用层协议规范 类型 10 元素

IEC 61784-1:2007 工业通信网络 行规 第 1 部分:现场总线行规

IEC 61784-2:2007 工业通信网络 行规 第 2 部分:基于 ISO/IEC 8802.3 实时网络的附加现场总线行规

IEC 61588:2004/IEEE Std 1588:2002 网络测量和控制系统的精确时钟同步协议

IEC 62026-3:2007 低压开关设备和控制设备 控制器-设备接口(CDI) 第 3 部分:DeviceNet

IEEE Std 802 用于局域网和城域网的 IEEE 标准:概述和体系结构

IEEE Std 802.1AB 用于局域网和城域网的 IEEE 标准 第 1AB 部分:站和媒体访问控制连接发现

IEEE Std 802.1Q:2005 用于局域网和城域网的 IEEE 标准 第 1Q 部分:虚拟桥接局域网

IEEE Std 802.3:2005 用于信息技术的 IEEE 标准 远程通信和系统间信息交换 局域网和城域网 规范要求 第 3 部分:载波侦听多路访问/冲突检测(CSMA/CD)访问方法和物理层规范

ISO/IEC 10646:2003 信息技术 通用多八位编码字符集(UCS) 第 1 部分:体系结构与基本多文种平面

3 术语和定义、缩略语

3.1 术语和定义

下列术语和定义适用于本标准。

3.1.1

精度 accuracy

在一个测量集上,待测时钟与理想参考时钟的时间或频率误差的平均值。稳定性是度量平均值如何随着时间、温度等变量的变化而改变。精度是度量对平均值误差的偏离。

3.1.2

原子性过程 atomic process

具有如下特点的过程是原子性过程:直到过程的所有结果被实例化后,过程的所有输入值才允许改变,并且直到每个输出的处理完成后,过程的输出对其他过程才可见。

3.1.3

边界时钟 boundary clock

边界时钟是在一个域中具有多个精确时间协议(PTP)端口,并维护该域中所用时标的时钟。它可作为时间源,即为主时钟;也可与另一个时钟同步,即为从时钟。

3.1.4

时钟 clock

参与精确时钟协议(PTP)的一个节点,该节点能提供从一个已定义历元开始的时间消逝的测量。

注:对于完全同步的 PTP 普通时钟和边界时钟,该历元是所用时标上的时间点。对于 PTP 透明时钟,该历元是本
地定义的,无需与时标关联¹⁾。

3.1.5

缺省 default

当用于属性值和选项时,缺省表示精确时钟协议(PTP)设备的配置与制造商提供的一致。

3.1.6

直接通信 direct communication

精确时钟协议(PTP)信息在两个普通时钟或边界时钟之间的通信,该通信中间不通过边界时钟。

3.1.7

域 domain

时钟的一个逻辑组,组内的时钟可使用协议实现彼此同步,但无需与另一域内的时钟同步。

3.1.8

端到端透明时钟 end-to-end transparent clock

在从时钟和主时钟之间,支持使用端到端(end-to-end)延时测量机制的透明时钟。

3.1.9

历元 epoch

时标的原点。

3.1.10

事件 event

用于产生和表示信号或条件的机制的抽象。

3.1.11

外部主(时钟) foreign master

向其他时钟发送 Announce 报文的一个普通时钟或边界时钟,该普通时钟或边界时钟不是其他时钟所认可的当前主(时钟)。

3.1.12

小数频率偏移 fractional frequency offset

在被测频率和参考频率之间,定义小数频率偏移(FFO)如下:

1) 在本标准中,给出的表和图只用于提供信息,不包含标准实现需要的要求。

$$FFO = \frac{(FO - FR)}{FR}$$

其中:FO 是被测频率,FR 是参考频率。

3.1.13

最高级时钟 grandmaster clock

在一个域中,最高级时钟是使用协议进行时钟同步的最终时间源。

3.1.14

保持 holdover

一个时钟之前已与另一个时钟(通常为主参考或主时钟)同步/谐振,但目前基于自身内部振荡器自激振荡。当该时钟同步/谐振到其他时钟时,使用所获得的数据调整其内部振荡器的频率。也就是说,只要在精度要求范围内,时钟就处于保持或保持模式。

3.1.15

链路 link

在两个支持本标准中对等延时机制的精确时间协议(PTP)端口之间的一个网络段。对等延时机制被设计用于测量在该链路上的传输时间。

3.1.16

管理节点 management node

对时钟进行配置和监视的设备。

3.1.17

主时钟 master clock

在单个精确时间协议(PTP)通信路径情况下,作为时间源的时钟。在该路径上的所有其他时钟都同步到该时钟。

3.1.18

报文时间戳点 message timestamp point

在精确时间协议(PTP)事件报文内,作为该报文参考点的点。时间戳定义为报文时间戳点通过时钟参考面的时刻。

3.1.19

多播通信 multicast communication

一种通信模型,在该模型中从任何 PTP 端口发出的每个精确时间协议(PTP)报文,能被在同一个 PTP 通信路径上的所有 PTP 端口接收和处理。

3.1.20

节点 node

在网络上能发布或接收精确时间协议(PTP)通信的设备。

3.1.21

单步时钟 one-step clock

使用单个事件报文来提供时间信息的时钟。

3.1.22

普通时钟 ordinary clock

在一个域中具有单个精确时间协议(PTP)端口,并维护该域中所用时钟的时钟。它可作为时间源,即为主时钟;或与另一个时钟同步,即为从时钟。

3.1.23

父时钟 parent clock

一个时钟与之同步的主时钟。

3.1.24

点到点透明时钟 peer-to-peer transparent clock

该透明时钟除了提供精确时间协议(PTP)事件传输时间信息外,还提供对连接到接收 PTP 事件报文端口的链路的传输延时校正。当存在点到点透明时钟时,在从时钟和主时钟之间的延时测量使用点到点延时测量机制完成。

3.1.25

相位变化率 phase change rate

观察到的被测时间相对于参考时间的变化率。相位变化率等于在被测频率和参考频率之间的小数频率偏移。见 3.1.12。

3.1.26

端口号 portNumber

在 PTP 节点上标识一个特定精确时间协议(PTP)端口的索引。

3.1.27

精度 precision

见 3.1.1。

3.1.28

精确时间协议 Precision Time Protocol (PTP)

本标准定义的协议。当作为形容词时,表示它所修饰的名词是在本标准的上下文环境中进行规定或解释的。

3.1.29

主参考 primary reference

可溯源至国际标准的时间或频率源。见 3.1.44 可溯源性。

3.1.30

行规 profile

适用于设备的允许的精确时间协议(PTP)特征集。

3.1.31

精确时间协议(PTP)通信 Precision Time Protocol (PTP) communication

在协议运行时所使用的信息,该信息在 PTP 通信路径上以 PTP 报文进行传输。

3.1.32

精确时间协议(PTP)通信路径 Precision Time Protocol (PTP) communication path

一个特定网络的信号传输路径部分,该路径可实现在普通时钟和边界时钟之间的直接通信。

3.1.33

精确时间协议(PTP)报文 Precision Time Protocol (PTP) message

本标准中定义的报文类型之一。

3.1.34

精确时间协议(PTP)节点 Precision Time Protocol (PTP) node

PTP 普通时钟、边界时钟、透明时钟、产生或解析 PTP 报文的设备。

3.1.35

精确时间协议(PTP)端口 Precision Time Protocol (PTP) port

用于与通信网络进行 PTP 通信的时钟的逻辑访问点。

3.1.36

被认可的标准时间源 recognized standard time source

被认可的标准时间源是精确时间协议(PTP)的外部源,提供可溯源到国际标准实验室的合适的

时间和/或频率。这些国际标准实验室维护的时钟作为国际原子时(TAI)和协调世界时(UTC)时标的基础。这样的例子有:全球定位系统(GPS)、NTP 和美国国家标准与技术研究所(NIST)时间服务器。

3.1.37

请求方 requestor

实现点到点延时机制并通过发送 Pdelay_Req 报文启动该机制的端口。

3.1.38

响应方 responder

对 Pdelay_Req 报文的接收做出响应的端口,Pdelay_Req 报文的接收是点到点延时机制运行的一部分。

3.1.39

稳定性 stability

见 3.1.1。

3.1.40

同步的时钟 synchronized clocks

如果两个时钟有相同的历元,并且在任意时刻对单一事件的时间测量差异不超过规定的不确定度,则两个时钟是在该规定不确定度下的同步。

3.1.41

谐振的时钟 syntonized clocks

如果两个时钟的短时持续时间相同,即被每次时钟测量的时间按相同的速率增加,则两个时钟是谐振的。这两个时钟可共用或不共用同一个历元。

3.1.42

超时 timeout

终止被请求活动的机制,该活动至少从请求方观点看没有在规定时间内完成。

3.1.43

时标 timescale

从一个历元开始的时间的线性测量。

3.1.44

可溯源性 traceability

测量结果或标准值的一种属性,该属性可通过不间断的、都具有规定不确定度的比较链,将测量结果或标准值与规定的参考(通常为国家标准)相关联(源自国际通用计量学基本术语[B15]²⁾)。

3.1.45

转换设备 translation device

在实现不同传输和报文协议的区域之间,或在本标准的不同版本之间,或在不同精确时间协议(PTP)行规之间,进行协议报文转换的边界时钟或在某些情况下的透明时钟。

3.1.46

透明时钟 transparent clock

测量精确时间协议(PTP)事件报文通过该设备的时间,并向接收该 PTP 事件报文的时钟提供该信息的设备。见 3.1.8、3.1.24。

3.1.47

双步时钟 two-step clock

使用事件报文和后续通用报文的组合来提供时间信息的时钟。见 3.1.21。

²⁾ 方括号中的数字对应于参考文献的那些数字。

3.2 缩略语

ARB	arbitrary	任意
BMC	best master clock	最佳主时钟
CAN	Controller Area Network	控制器局域网
CP	Communication Profile [根据 IEC 61784-1:2007 ³⁾	通信行规
CPF	Communication Profile Family [根据 IEC 61784-1:2007]	通信行规族
DS	differentiated service	区分服务
E2E	end-to-end	端到端
GPS	Global Positioning System	全球定位系统
IANA	Internet Assigned Numbers Authority	国际因特网地址分配委员会
ICV	integrity check value	完整性校验值
ID	identification	标识符
IM/ST	IEEE/TC 9 Measurement Sensor Technology Committee	IEEE/TC 9 测量传感器技术委员会
IPv4、IPv6	Internet Protocol version 4/6	因特网协议版本 4/6
JD	Julian Date	儒略日
JDN	Julian Day Number	儒略日数
MAC	media access control [根据 IEEE Std 802.3:2005]	媒体访问控制
MJD	Modified Julian Day	修改的儒略日
NIST	National Institute of Standards and Technology(见 www.nist.gov)	美国国家标准与技术研究所
NTP	Network Time Protocol(见 IETF RFC 1305 [B7])	网络时间协议
OUI	organizational unique identifier(IEEE 分配) 注:典型地,在设备的规范或实现中使用 OUI 用于标识目的。OUI 可标识拥有 OUI 相关子标识符的组织,但不要求该组织定义规范或提供硬件。可访问 http://standards.ieee.org/regauth/oui/index.shtml 获得 IEEE OUI 列表。	组织唯一标识符
P2P	peer-to-peer	点到点
PHY	physical layer [根据 IEEE Std 802.3:2005]	物理层
POSIX	Portable Operating System Interface (见 ISO/IEC 9945:2003 [B16])	可移植操作系统接口
PPS	pulse per second	脉冲/s
PTP	Precision Time Protocol	精确时间协议
SA	security associations	安全联盟
SNTP	Simple Network Time Protocol	简单网络时间协议
SOF	Start of Frame	帧起始
TAI	International Atomic Time	国际原子时
TC	traffic class	业务类别

3) 参考信息见第 2 章。

TLV	type, length, value [根据 IEEE Std 802.1AB]	类型、长度、值
ToS	type of service	服务类型
UCMM	UnConnect Message Manager	无连接报文管理器
UDP/IP	User Datagram Protocol(见 IETF RFC 768 [B5])/ Internet Protocol(见 IETF RFC 791 [B6])	用户数据报协议/因特网协议
UTC	Coordinated Universal Time	协调世界时

4 约定

4.1 描述性词汇形式的语法

4.1.1 词汇形式的语法

词汇形式指：

- 名称；
- 数据类型。

在本标准中使用的关于词汇形式的约定如下：

- 类型名称：例如：ClockQuality(无词间隔，每个词的首字母大写)；
- 枚举成员和全局常数：例如：ATOMIC_CLOCK(词间用下划线分隔，所有字母大写)；
- 报文内的字段、结构实例和变量：例如：secondsField, clockQuality, clockIdentity(字段名称至少为 2 个词，无词间隔，第一个词的首字母不需大写，后续词的首字母大写)；
- 结构成员：例如：clockQuality. clockClass(结构名称后为点“.”，然后为成员名称)；
- 数据集名称：例如：defaultDS、parentDS、portDS、currentDS、timePropertiesDS(无词间隔，字母 DS 前的第一个词的首字母不大写)；
- 数据集成员：例如：defaultDS. clockQuality. clockClass(数据集名称后为点“.”，然后为类型名称，其后为点“.”，最后为变量名称)；
- 报文名称：例如：Sync、Delay_Req(词间用下划线分隔，每个词的首字母大写)；
- <localNameForSomething>：角括号<>中的文本用于标准需要参考某些内容的场合，该内容的语法和词汇格式取决于本地实现和语言。

当词汇格式在文本中，而不是在类型或格式定义中出现时，将该格式解释为符合文本上下文的单数、复数或所有格。

4.2 词用法

4.2.1 应(Shall)

“应”的含义等效于“要求”，用于指示强制性要求，为符合本标准，强制性要求应严格遵循且不允许任何偏离。

4.2.2 推荐(Recommended)

“推荐”用于指示选择的灵活性，但该选择带有强的倾向性。

4.2.3 必须(Must)

“必须”指示不可避免的状况。

4.2.4 宜(Should)

“宜”等效于“推荐”，用于指示：

- 在几个可能性中，推荐其中一个特别适合的，而不提及或不排斥其他可能性；
- 某一组动作是首选的，但不是必需的；
- 某一组动作(以否定形式)不被赞成，但不禁止。

4.2.5 可以(May)

“可以”等效于“允许”，用于指示在本标准范围内允许的一组动作。

4.2.6 能(Can)

“能”用于指示物质上或物理上的可能性和能力。

4.2.7 可选的(Optional)

标记为“可选的”条和文本不要求实现。如果实现可选项，则该条或文本的所有规范应根据本标准分别实现。

注：该定义是递归的，这意味着可选项内的可选项服从相同的规则。

4.2.8 保留(Reserved)

“保留”指示：

- 如用在为枚举或属性赋值，则表示值被保留用于 PTP 的后续版本，且不用于任何其他目的；
- 如用在报文中的字段，则表示该字段被保留用于 PTP 的后续版本。该字段应以规定长度出现在报文中。PTP 的当前版本对保留字段没有做出解释，保留字段不应用于任何其他目的。

4.3 行为规范表示法

状态转换图用于规定如图 1 所示的行为特性。每个状态转换图由下列部分构成：

- 名称框，表示状态；
- 有向箭头，指示从一个状态到另一个状态的转换。

每个转换使用下列标记：

- 转换的使能事件标记或谓词标记；
- 转换的转换动作标记。

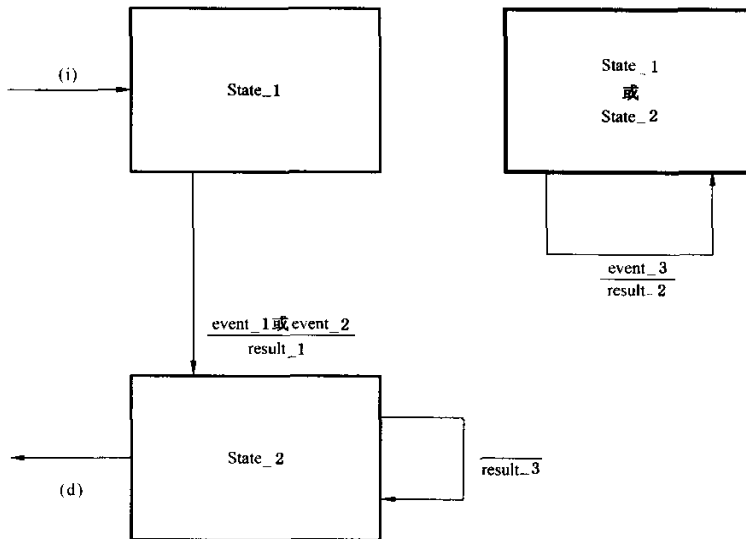


图 1 Mealy 型状态转换图

使用的表示法以抽象方式描述了状态转换图，在该转换图中动作与从一个状态到另一个状态的转换相关联。

事件(Event)，例如“event_1”、“event_2”和“event_3”标识状态机的输入。事件可以是操作请求和操作响应，或内部发生事件如定时器超时。

谓词，例如“event_1 或 event_2”标识转换的使能条件。从左向右计算，找到第一个为 TRUE 的谓词，并选择要执行的转换，然后进入下一状态。

转换动作,例如“result_1“,是转换到下一状态前执行的动作。

下一状态标识在完成选择的转换动作后状态机的状态。当下一状态的转换发生时,当前状态值改变。

粗线状态框指示该状态框代表多个状态。由该状态框开始并终止的任何转换指示在状态上无变化。

没有指示使能条件的转换,例如产生“result_3“的转换,通过未规定机制发生。除非在 PTP 中另有说明,引起这些机制的事件是实现特定的,不在本标准范围内。

进入状态机的转换,例如“(i)”,通过没有源状态的转换箭头表示。离开状态机的转换,例如“(d)”,通过没有目的状态的转换箭头表示。

注:例如:当 event_1 或 event_2 的一个结果为 TRUE 时,state_1 被下一状态值取代。在本例中,下一状态是 state_2,该状态被规定为转换箭头的目标状态框的名称。在转换前,result_1 发生。event_3 可在 state_1 或 state_2 下发生。状态不改变,但动作 result_2 作为 event_3 的结果发生。

5 PTP 系统的数据类型和传输格式

5.1 概述

为不同 PTP 变量和报文字段规定的数据类型定义了逻辑属性,这些逻辑属性是正确操作该协议或解释 PTP 报文内容所必需的。

实现可自由使用 PTP 数据类型的任意内部表示。然而,内部表示不应改变通过 PTP 通信的可见的任何量的语义,或协议的任何规定操作的语义。

5.2 基本数据类型规范

所有非基本 PTP 数据类型源自表 1 所列的基本类型。这些类型不受任何特定编程语言的约束。每个类型的主要特性如下:

- 整型(Integer):所有整型具有确定长度,与每个类型相关联的数字指示该长度。例如:UInteger 48,通过带或不带首字母“U”来指示无符号或有符号。具有这些数据类型的数遵从长度表示范围内的算术法则。数据类型范围的模数用作算术运算;例如:2 个 UInteger48 值的和用模数 2^{+48} 计算。有符号整数用二进制补码形式表示。
- 枚举型(Enumeration):所有枚举类型具有确定字段长度,与每个类型相关联的数字指示该长度。例如:Enumeration4。除非在本标准中另有说明,在枚举字段比特模式的唯一解释即为比特模式和枚举指定含义间的关联。
- 布尔型(Boolean):唯一的解释是在布尔代数中的逻辑值。
- 四位位组(Nibble)和八位位组(Octet):分别为 4 bit 和 8 bit 字段。唯一解释在本标准中明确定义。

表 1 基本 PTP 数据类型

数据类型	定义
Boolean	TRUE 或 FALSE
Enumeration4	4 bit 枚举值
Enumeration 8	8 bit 枚举值
Enumeration16	16 bit 枚举值
UInteger4	4 bit 无符号整数
Integer8	8 bit 有符号整数
UInteger 8	8 bit 无符号整数
Integer16	16 bit 有符号整数

表 1 (续)

数 据 类 型	定 义
UInteger 16	16 bit 无符号整数
Integer 32	32 bit 有符号整数
UInteger 32	32 bit 无符号整数
UInteger 48	48 bit 无符号整数
Integer64	64 bit 有符号整数
Nibble	4 bit 字段,不解释为数字
Octet	8 bit 字段,不解释为数字

5.3 派生数据类型规范

5.3.1 概述

任何基本数据类型数组的表示格式为<data type>[lengthField]<label>,其中<lengthField>指示数组中数据类型实例的个数,<label>是定义的数组数据类型的词汇名称。

结构由成员的有序列表组成,用下列语法表示:

```
struct<StructureName>
{
    <DataType1><memberName1>;
    <DataType2><memberName2>;
    ...
};
```

其中<StructureName>是定义的数据类型的词汇名称,<DataType1>是第一个成员的数据类型,<memberName1>是第一个成员的词汇名称,依此类推。

语法 typedef<DataType><TypeName>解释为:定义了派生数据类型,它与<DataType>定义的数据类型具有相同的属性,该派生数据类型的新名称由<TypeName>给出。

语法 typedef<DataType><lengthField><TypeName>解释为:定义了由类型<DataType>的元素数组组成的派生数据类型,该派生数据类型的新名称由<TypeName>给出。

5.3.2 TimeInterval

TimeInterval 类型表示时间间隔。

```
struct TimeInterval
{
    Integer64 scaledNanoseconds;
};
```

scaledNanoseconds 成员是以纳秒(ns)为单位,并乘以 2^{+16} 的时间间隔。

超出本数据类型最大范围的正或负时间间隔,应分别编码为该数据类型的最大正值或负值。

例如:2.5 ns 被表示为 0000 0000 0002 8000₁₆。

5.3.3 Timestamp

Timestamp 类型表示关于历元的正时间。

```
struct Timestamp
{
    UInteger 48 secondsField;
```

```

    UInteger 32 nanosecondsField;
};

```

secondsField 成员是以 s 为单位的 timestamp 的整数部分。

nanosecondsField 成员是以 ns 为单位的 timestamp 的小数部分。

nanosecondsField 成员总是小于 10^9 。

例如: + 2. 000 000 001 s 被表示为 secondsField = 0000 0000 0002₁₆, nanosecondsField = 0000 0001₁₆。

5.3.4 ClockIdentity

类型 ClockIdentity 标识一个时钟。

```

typedef Octet[8] ClockIdentity;

```

5.3.5 PortIdentity

类型 PortIdentity 标识一个 PTP 端口。

```

struct PortIdentity
{
    ClockIdentity clockIdentity;
    UInteger 16 portNumber;
};

```

5.3.6 PortAddress

类型 PortAddress 表示 PTP 端口的协议地址。

```

struct PortAddress
{
    Enumeration16 networkProtocol;
    UInteger 16 addressLength;
    Octet[addressLength] addressField;
};

```

networkProtocol 成员的值应从 networkProtocol 枚举中获得, 见 7.4.1。addressLength 是按八位位组个数来表示的地址长度, 其范围应为 1~16 个八位位组。

addressField 成员以协议的映射附录定义的格式来保存端口的协议地址, 协议由 networkProtocol 成员标识。addressField 的最高有效八位位组被映射到 addressField 成员中索引为 0 的八位位组。

5.3.7 ClockQuality

ClockQuality 表示时钟质量。

```

struct ClockQuality
{
    UInteger 8 clockClass;
    Enumeration 8 clockAccuracy;
    UInteger 16 offsetScaledLogVariance;
};

```

5.3.8 TLV

TLV 类型表示 TLV 扩展字段。

```

struct TLV
{

```

```

    Enumeration16 tlvType;
    UInteger 16 lengthField;
    Octet[lengthField] valueField;
};

```

所有 TLV 的长度应为偶数个八位位组。

5.3.9 PTPText

PTPText 数据类型用于表示 PTP 报文中的文本资料。

```

struct PTPText
{
    UInteger 8 lengthField;
    Octet[lengthField] textField;
};

```

textField 成员应编码为 UTF-8 符号(见 ISO/IEC 10646:2003)。起始文本符号的最高有效八位位组应为数组中索引为 0 的元素。

注：一个单一 UTF-8 符号的长度可以是 1~4 个八位位组，因此 lengthField 值可大于符号个数。

5.3.10 FaultRecord

FaultRecord 类型用于构建故障日志。

```

struct FaultRecord
{
    UInteger 16 faultRecordLength;
    Timestamp faultTime;
    Enumeration 8 severityCode;
    PTPText faultName;
    PTPText faultValue;
    PTPText faultDescription;
};

```

faultRecordLength 成员应指示 FaultRecord 中八位位组的个数，不包括 faultRecordLength 成员的 2 个八位位组。

5.4 传输格式

5.4.1 概述

PTP 协议数据单元基于 5.2、5.3 和 17.2 中定义的数据类型，由在第 13 章~第 17 章定义的 PTP 报文组成。PTP 协议数据单元字段的内部顺序在 5.4.2~5.4.4 中规定。

5.4.2 基本数据类型

5.2 中定义的数值型基本数据类型格式化为：最高有效八位位组最靠近协议数据单元的起始，其后的八位位组按有效性递减顺序排列。

Boolean 数据类型 TRUE 应格式化为等于 1 的单个比特，FALSE 格式化为等于 0 的单个比特。

任何长度的枚举类型应格式化为同样长度的无符号整型的值，例如：Enumeration16 应格式化为数据类型 UInteger16 的值。

5.4.3 基本类型数组

所有数组应格式化为：最低数字索引的成员最靠近协议数据单元的起始，其后的成员按逐次递增数字索引顺序排列。对于八位位组数组，带有最低数字索引的八位位组被称为最高有效八位位组。

当包含多个八位位组的字段用于表示一个数值时,最高有效八位位组应最靠近协议数据单元的起始,其后的八位位组按有效性递减排列。

当单个八位位组包含基本数据类型的多个字段时,在报文字段规范中定义的每个基本类型八位位组的比特位应保持不变。例如:PTP 报文头的第一个八位位组由两个字段构成的,一个是 Nibble 类型的比特 4~7,一个是 Enumeration4 类型的比特 0~3(见 13.3.1)。

5.4.4 派生数据类型

定义为结构体的派生数据类型格式化为:结构体的第一个成员最靠近协议数据单元的起始,其后依次跟随后续成员。每个成员应根据其数据类型被格式化。

用 typedef 定义的派生数据类型应根据其参考数据类型被格式化。

5.4.5 PTP 协议数据单元映射为传输格式

除非另有说明,PTP 协议数据单元和传输格式的映射应基于底层物理层传输规则。任何异常在本标准的某个传输特定附录或在适用的 PTP 行规中说明。

注:PTP 协议机制工作在协议栈的上层(即 PTP 是使用网络层或链路层服务的“应用”)。物理层传输指示传输格式。

6 时钟同步模型

6.1 概述

本条提供一个模型以理解精确时间协议的运行。这些交互作用的准确规定见后续条。

PTP 标准规定了时钟同步协议。该协议适用于由 1 个或多个节点构成的、通过网络通信的分布式系统。节点被模型化为包含一个实时时钟,该实时时钟在该节点内可被用于各种目的的应用,如为数据产生时间戳,或对由节点管理的事件进行排序。该协议提供一种机制,将参与节点的时钟同步到高精度。本标准规定了:

- 精确时间协议;
- 支持 PTP 所需的节点、系统和通信特性。

6.2 关于网络的基本假设和实现推荐

为确保协议的正确运行,应遵循下述基本假设和推荐,在后面条款中将其展开详细讨论:

- a) PTP 假设网络在每个通信路径内消除了 PTP 报文的循环转发(例如:通过使用生成树协议)。PTP 消除了通信路径间的 PTP 报文的循环转发。
- b) PTP 能容忍偶尔发生的报文丢失、重复的报文或报文到达错序。但 PTP 假设这些错误很少发生。
- c) 设计的 PTP 假定多播通信模型。只要保持协议行为,PTP 也支持单播通信模型。PTP 假设 Announce 报文被一个端口周期性地发送,且在通信路径内被传递到普通时钟或边界时钟的所有其他端口。如果通信路径由两个以上端口组成,则假设 Announce 报文以多播方式发送,或使用单播报文将 Announce 信息复制到通信路径内所有端口。PTP 端口通过接收多播 Announce 报文发现通信路径内的其他端口。当多播通信不可用时,要求另一形式的发现(例如:通过组态),见 17.5 示例。向所有端口发送的 PTP 管理报文,也要求通过多播发送报文或将管理报文复制到通信路径内的所有端口。
- d) 像所有基于报文的时间传输协议一样,事件报文选择路径的不对称会降低 PTP 时间精度,见 7.4.2。特别地,时间偏移误差是不对称值的 1/2。不对称对 PTP 是不可检测的,然而,如可知该不对称,则 PTP 可对其进行校正。物理层可以引入不对称,例如,传输媒体不对称:通过网桥和路由器,以及在大型系统中由于事件报文采用不同路由通过网络的前向和反向路径。

系统组态及部件选择应根据要求的时间精度使这些影响最小化。在几米距离范围内的单一子网系统,对于超过几十纳秒的时间精度,不对称的影响很小。

- e) 如果使用双步时钟,则网络必须被设计成通用报文采用与事件报文相同的路径通过透明时钟。如果无法做到这点,则将导致透明时钟不能正确计算路径延时;这种情况是不可检测的,且可能引入额外抖动和漂移,但不会中断协议。
- f) PTP 假设从最高级时钟到任何从时钟之间形成主从同步层次的边界时钟的数量少于 255,见 9.3.2.5。
- g) 网络部件(例如网桥)引入时间抖动和漂移,如果不对它们进行校正,则会降低时间传输精度。鉴于抖动和漂移经常是与流量相关的,所以网络流量模式应设计为使流量最小化以及使流量负荷变化最小化。推荐用比其他数据报文更高的优先级发送 PTP 事件报文,见 A.5.3.3。只要可能,应使用 PTP 边界时钟或透明时钟替换这类设备。
- h) 将网络协议结构化,以定义报文时间戳点。

6.3 PTP 系统

PTP 系统是由 PTP 设备和非 PTP 设备的组合组成的分布式网络化系统。PTP 设备包括普通时钟、边界时钟、端到端透明时钟、点到点透明时钟和管理节点。非 PTP 设备包括网桥、路由器、其他基础设备和可能使用的设备,如计算机、打印机和其他应用设备。

本协议是分布式协议,它规定了系统中的实时时钟如何实现彼此间同步。这些时钟被组织为主-从同步层次结构,主-从同步层次结构的最顶端为最高级时钟,最高级时钟决定了整个系统的参考时间。通过与使用时间信息的从时钟交换 PTP 时间报文来实现同步,以将从时钟的时钟调整到层次结构中主时钟的时间。

PTP 系统中的设备通过通信网络进行彼此间通信。网络可以包括实现不同网络通信协议的段之间的转换设备。

协议在被称为域的逻辑范围内执行。除非另有说明,所有 PTP 报文、数据集、状态机和所有其他 PTP 实体总是与一个特定域相关联。一个给定的物理网络和连接到该网络的单个设备可与多个域相关联。在本标准范围内,在一个域通过本协议建立的时间独立于其他域的时间。

6.4 PTP 报文类

协议定义事件报文和通用 PTP 报文。事件报文是计时的报文,其中的精确时间戳在发送和接收时产生,见 6.6.5。通用报文不要求精确时间戳。

事件报文集包括:

- Sync(见 13.6);
- Delay_Req(见 13.6);
- Pdelay_Req(见 13.9);
- Pdelay_Resp(见 13.10)。

通用报文集包括:

- Announce(见 13.5);
- Follow_Up(见 13.7);
- Delay_Resp(见 13.8);
- Pdelay_Resp_Follow_up(见 13.11);
- Management(管理,见第 15 章);
- Signaling(信号,见 13.12)。

Sync、Delay_Req、Follow_Up 和 Delay_Resp 报文用于产生并传递时间信息,时间信息用于同步普

通时钟和边界时钟,通过延时请求-响应机制实现。

Pdelay_Req、Pdelay_Resp 和 Pdelay_Resp_Follow_Up 报文用于测量实现对等延时机制的两个时钟端口间的链路延时。在由点到点透明时钟组成的系统内,链路延时用于校正 Sync 和 Follow_Up 报文的时间信息。实现对等延时机制的普通时钟和边界时钟可使用测量的链路延时和 Sync 及 Follow_Up 报文中的信息实现同步。

Announce 报文用于建立同步层次结构。

管理报文用于查询和更新由时钟维护的 PTP 数据集。这些报文也用于定制 PTP 系统、初始化和故障管理。管理报文在管理节点和时钟间使用。

信号报文用于时钟间所有其他目的的通信。例如:信号报文可用于协商主时钟与其从时钟间的单播报文速率。

所有报文可通过标准类型、长度、值(TLV)扩展机制进行扩展。例如:PATH_TRACE 报文扩展可用于检测欺诈帧,关于欺诈帧的详细内容见 16.2.1。

6.5 PTP 设备类型

6.5.1 概述

PTP 设备有 5 种基本类型,如下:

- 普通时钟;
- 边界时钟;
- 端到端透明时钟;
- 点到点透明时钟;
- 管理节点。

5 类设备实现协议的一个或多个部分。

PTP 使用 2 种机制测量 PTP 端口间的传输延时。第一个是延时请求-响应机制,该机制使用 Sync、Delay_Req、Delay_Resp 和 Follow_Up 报文(如需要),见 11.3。第二个是对等延时机制,该机制使用 Pdelay_Req、Pdelay_Resp、和 Pdelay_Resp_Follow_Up 报文(如需要),见 11.4。普通时钟和边界时钟的端口可以实现其中任意一个机制。端到端透明时钟的端口与这些机制无关。点到点透明时钟的端口使用对等延时机制。这两个机制不在同一通信路径上交互使用。此外,对等延时机制被限制用在某些拓扑结构,在这些拓扑结构中每个点到点端口与至多一个其他同类端口进行 PTP 报文通信,见 11.4.4。

对不同时钟类型的使用限制如下:

- 具有实现了对等延时机制的端口的普通时钟和边界时钟,以及点到点透明时钟,仅能连接到某些拓扑结构,在这些拓扑结构下实现了对等延时机制的端口与也实现了对等延时机制的单个端口进行 PTP 报文通信,见 11.4.4。除了仔细设计的网络外,这将排除使用端到端透明时钟和不支持使用对等延时机制的 PTP 的网桥,例如普通网桥。
- 实现了延时请求-响应机制的普通时钟和边界时钟端口,以及端到端透明时钟,可被连接到不使用对等延时机制的端口的任何拓扑结构。这排除了在这样的系统里使用点到点透明时钟。
- 带有支持两种机制的端口的边界时钟,可用于桥接支持不同机制的区域。

6.5.2 普通时钟

普通时钟的模型见图 2。

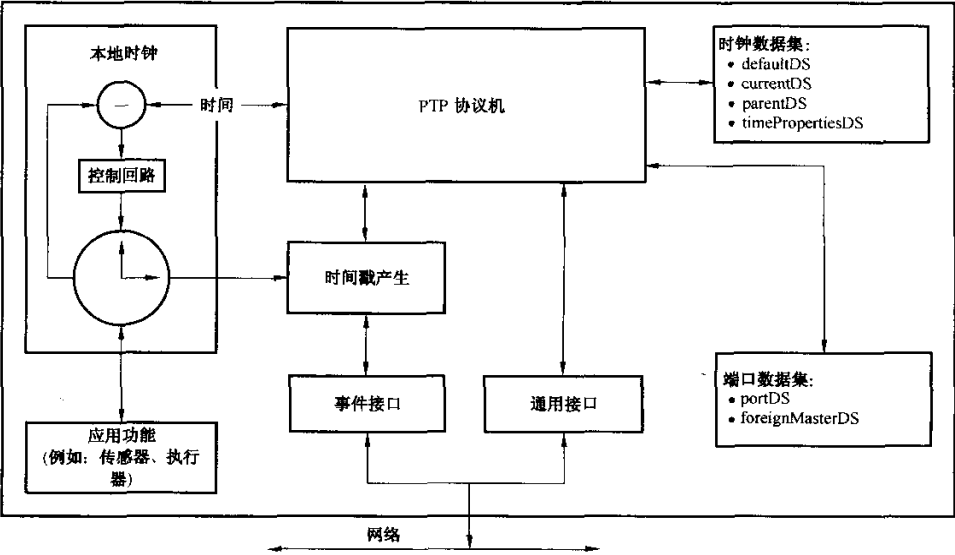


图 2 普通时钟模型

普通时钟通过基于单一物理端口的两个逻辑接口与网络通信。事件接口用于发送和接收事件报文,该事件报文由基于本地时钟值的时间戳发生块加入时间戳。通用接口用于发送和接收通用报文。在一个域内的普通时钟支持协议的单一副本且具有单一 PTP 状态。在一个系统中,普通时钟可以作为最高级时钟,或作为主从层次结构内的从时钟。

普通时钟维护两类数据集,分别是时钟数据集和端口数据集。

时钟数据集如下:

- a) defaultDS:描述普通时钟的属性;
- b) currentDS:与同步相关的属性;
- c) parentDS:描述父时钟(普通时钟与之同步的时钟)和最高级时钟(在主从层次结构中最高层的时钟)的属性;
- d) timePropertiesDS:时标属性。

端口数据集包含端口属性,端口属性包括 PTP 状态。

协议机:

- 发送和接收 PTP 报文;
- 维护数据集;
- 执行与端口关联的状态机;
- 如果端口处于从时钟状态(同步到一个主时钟),端口基于接收到的 PTP 时间报文和产生的时间戳计算主时钟时间。

如果普通时钟的端口处于从时钟状态,则本地时钟的控制回路将其时钟调整到与主时钟的时间一致。如果该端口处于主时钟状态,则本地时钟自由运行或可能同步到外部时间源,如全球定位系统(GPS)。如果该端口处于主时钟状态且普通时钟是该域的最高级时钟,则本地时钟典型地同步到可溯源到 TAI 和 UTC 的外部时间源,如 GPS 系统。

在一些应用中,如工业自动化,普通时钟也可与应用设备相关联,如传感器或执行器。在电信应用中,普通时钟可以与时间分界设备相关联。本地时钟对这些设备提供时间支持,如图 2 中所示。

6.5.3 边界时钟

边界时钟的模型如图 3 所示。

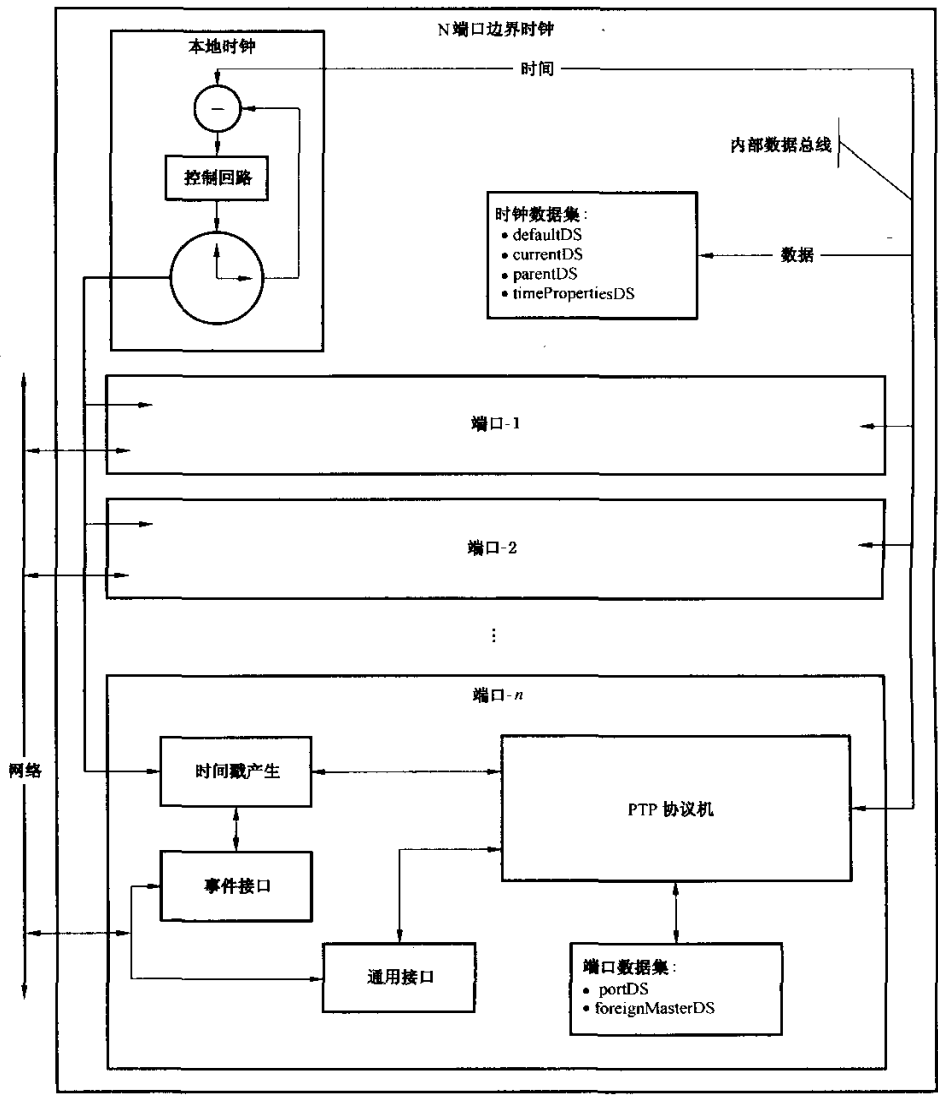


图3 边界时钟模型

边界时钟典型地具有几个物理端口，且每个物理端口通过两个逻辑接口：事件接口和通用接口，与网络通信。除了下述特性外，边界时钟的每个端口与普通时钟的端口类似：

- a) 时钟数据集对边界时钟的所有端口是共用的；
- b) 本地时钟对于所有边界时钟的所有端口是共用的；
- c) 每个协议机具有解析所有端口状态的附加功能，以确定哪个端口提供了用于同步本地时钟的时间信号。

与同步相关的报文建立了主从层次结构，并发送信号（见 6.4），这些报文在边界时钟的协议机内终止，不被转发。管理报文被边界时钟的其他端口转发，且管理报文的转发应服从系统内对这些报文传输的限制。

图 3 所示边界时钟模型仅对于 PTP 报文是适用的。对于所有非 PTP 报文,边界时钟相当于普通网络部件,例如:网桥、中继器或路由器。

边界时钟典型地仅用作网络元素,且一般不与应用设备相关联,如传感器或执行器。

6.5.4 端到端透明时钟

端到端透明时钟的模型见图 4。

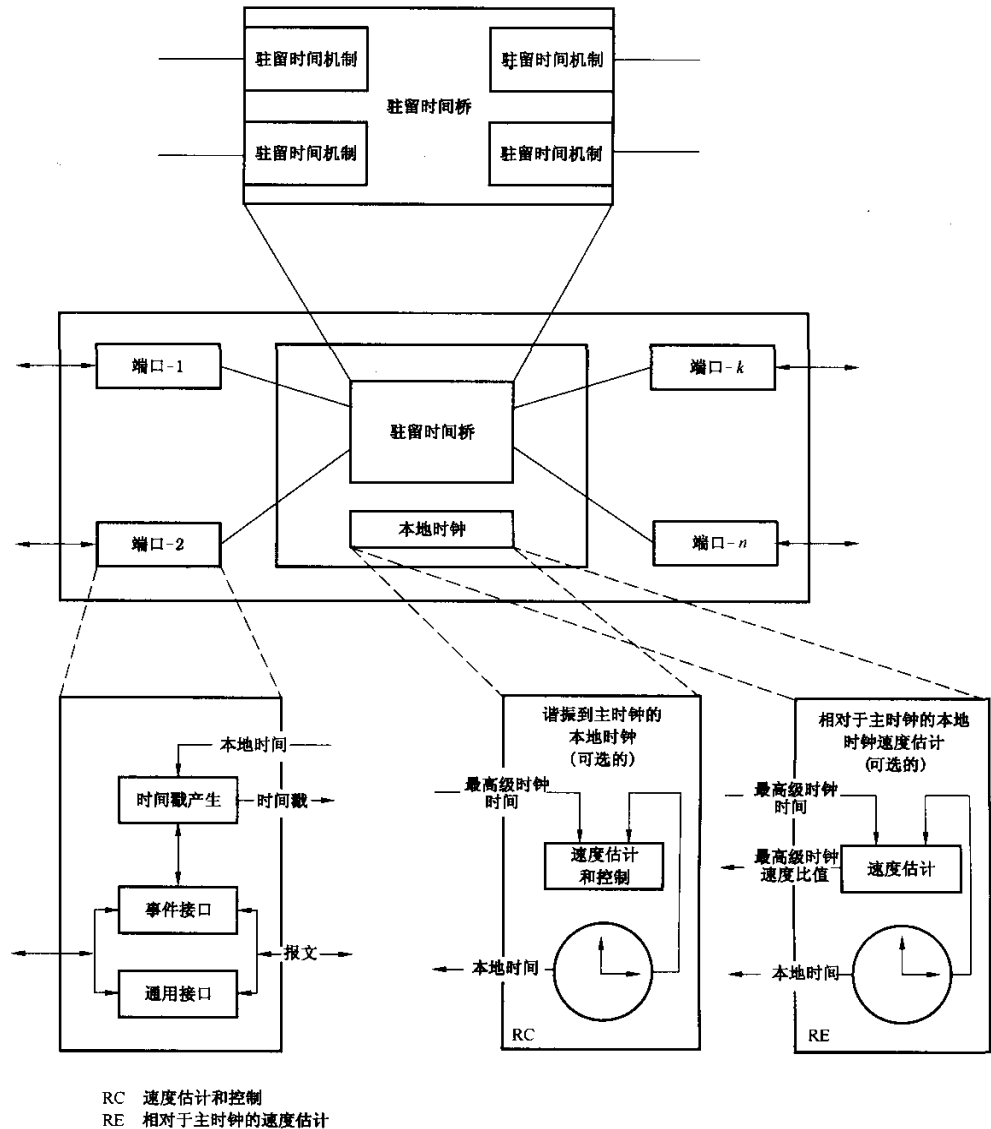


图 4 端到端透明时钟模型

端到端透明时钟像普通的网桥、路由器或中继器一样,转发所有的报文。但是对于 PTP 事件报文,如图 4 所示的驻留时间桥测量 PTP 事件报文的驻留时间(报文通过透明时钟所需的时间)。这些驻留时间在一个特定字段进行累积,该特定字段是 PTP 事件报文或相关的后续报文(Follow_Up 或 Pdelay_

Resp_Follow_Up)的 correctionField 字段。校正基于事件报文在进入和离开透明时钟时产生的时间戳的差来完成的。实现了网络协议要求的对校验和的任何更新。注意:对于通过透明时钟的所有路径或在同一路径的连续报文,驻留时间不要求相同,所以对于每个输出端口和报文,校正更新值和校验和是特定的。任意一对端口的校正过程如图 5 所示。

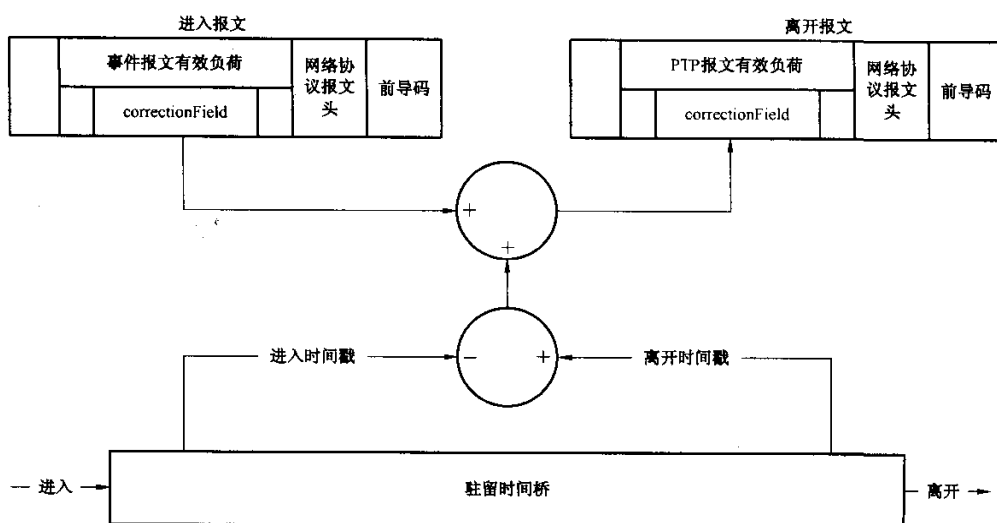


图 5 端到端驻留时间校正模型

计算驻留时间所使用的时间戳是基于本地时钟产生的时间戳。由于从时钟使用这些累积的驻留时间调整由主时钟提供的时间,所以重要的是:对于应用要求的精度,由主时钟和透明时钟的速率差异导致的任何误差应是可以忽略的。主时钟和本地时钟的速率(本质上是两个时钟对时刻的定义)可能相差 0.02%,则引入的误差能达到测量的驻留时间的 0.02%。这样,对于 1 ms 的驻留时间,最大误差是 200 ns。这个误差可能是大的不可接受,可通过几个可能的方法降低该误差。

一种方法(在 11.3 中对其展开讨论)是使本地时钟的速率与主时钟的相同,即本地时钟与主时钟同步。该同步可通过观测接收到的 Sync 报文和 Follow_Up 报文(如出现)中的时间信息来完成,此时间信息已做过上行驻留时间校正,见图 4 中的块 RC(速率控制)。已做过上行驻留时间校正的最高级时间是 RC 中速率估计和控制块的输入(见标记为“最高级时间”的箭头)。相应的本地时间也是该块的输入。速率估计和控制块使用主时钟和对应的本地时间序列,以估计主时钟和本地时钟速率的比值。然后速率估计和控制块使用此估计的速率比值来调整(即控制)本地时钟速率。注意:本地时钟速率的调整不需要物理上调整振荡器频率(即模拟实现);使用固定频率的本地振荡器来计算主时钟速率和本地速率的比值,并将本地时钟的时间戳乘以该比值(即数字实现)也是可以接受的。RC 描述方案的关键在于该方案是以闭环方式运行,即速率估计和驻留时间测量时使用的本地时间是相对于速率调整后的本地振荡器的。这意味着在一个节点上对振荡器的速率调整会影响到下行节点的调整,因为在下行节点中使用驻留时间来调整主时钟的时间。

另一种在某些应用中可能有用的降低驻留时间误差影响的方法是:使用自由运行的本地时钟,见图 4 中块 RE(速率估计)。在这种情况下,不调整本地振荡器的频率,允许它自由运行。每个时钟基于自由运行时钟测量 Sync 报文的本地驻留时间。测量的驻留时间被加入到 Sync 或 Follow_Up 报文的

correctionField 字段, 见 11.3。每个本地时钟也计算本地自由运行时钟速率与最高级时钟速率的比值。该比值是基于接收到的 Sync 和 Follow_Up 报文中的时间戳计算得到的, 这里的时间戳已经通过加入 correctionField 进行了校正。尽管 correctionField 没有反映准确的驻留时间, 得到的比值相比简单使用原时间戳, 仍然极大地降低了抖动, 而且仍然具有正确的平均值。此外, 如果驻留时间相对于速率比值的测量间隔足够大, 在最坏情况下连续通过透明时钟的相位误差累积将小于使用第一种方法的误差累积, 但对于许多非最坏情况下会导致更大相位累积。速率比值的本地值必须在每个时钟内维护, 用于校正 Delay_Req 报文的驻留时间, 如下所述。此外, 每个本地时钟计算用该比值校正过的驻留时间和未校正的驻留时间的差值, 这些差值也必须被累积并传递给从时钟。如使用该方法, 它连同用于累积的行规特定 TLV 应在行规(见 19.3)中进行规定。使用该方法, 在计算从时钟与最高级时钟的偏移量之前, 从时钟必须将累积的差值加入到接收到的 Sync 报文或 Follow_Up 报文的 correctionField。端到端的透明时钟也校正 Delay_Req 报文的驻留时间。但是, 由于该方法中的振荡器是自由运行的, Delay_Req 报文的驻留时间测量值必须使用本地时钟速率和最高级时钟速率的比值进行校正, 该比值的计算和保存如前所述。对于 Delay_Req 报文, 这是时钟操作方面的唯一变化。使用该技术, 对 Pdelay_Resp 报文的处理与 Sync 报文完全一样, 必须有规定的 TLV 用于此目的。对 Pdelay_Req 报文的处理与 Delay_Req 报文一样。

注 1: 在决定使用哪个方法之前, 重要的是评估相比应用误差预算可能的误差。例如: 如果主时钟和从时钟之间的所有时钟的总的驻留时间是 10 ms, 则在每个中间时钟的未校正的本地时钟速率的最大误差小于该值的 200×10^{-6} , 或 2 μ s。在许多设计中, 总的驻留时间应相当低, 且任一种机制都可给出满意的结果。

注 2: PTP 节点的实现需考虑支持多个域所要求的资源。由于资源所限造成的无法实时处理协议可能会导致同步性能的退化、抖动或协议失效。当选择节点和设计系统时, 用户需清楚这种限制。

端到端透明时钟可能被用作网络元件, 或可以与应用设备关联, 如传感器或执行器。在后一种情况下, 将普通时钟与透明时钟结合, 以提供对应用设备的实时支持。这种组合设备的模型见图 6。

在图 6 中, 当普通时钟是从时钟时, 时间信息包含进入时间戳。当普通时钟是主时钟时, 时间信息包含离开时间戳。驻留时间桥将进入的 PTP 时间报文、Announce 报文、由进入 Sync 报文产生的进入时间戳以及任何内部时间校正传递给普通时钟协议机。协议机计算基于此信息的准确时间, 并将其作为输入发送给本地时钟。如果普通时钟是主时钟, 它应发起带有报文时间戳的 Sync 报文和 Follow_Up 报文, 该报文时间戳参考普通时钟的本地时钟, 并且基于内部时间校正和离开时间戳。实际上端到端透明时钟和普通时钟功能可以共享一个公共物理时钟。

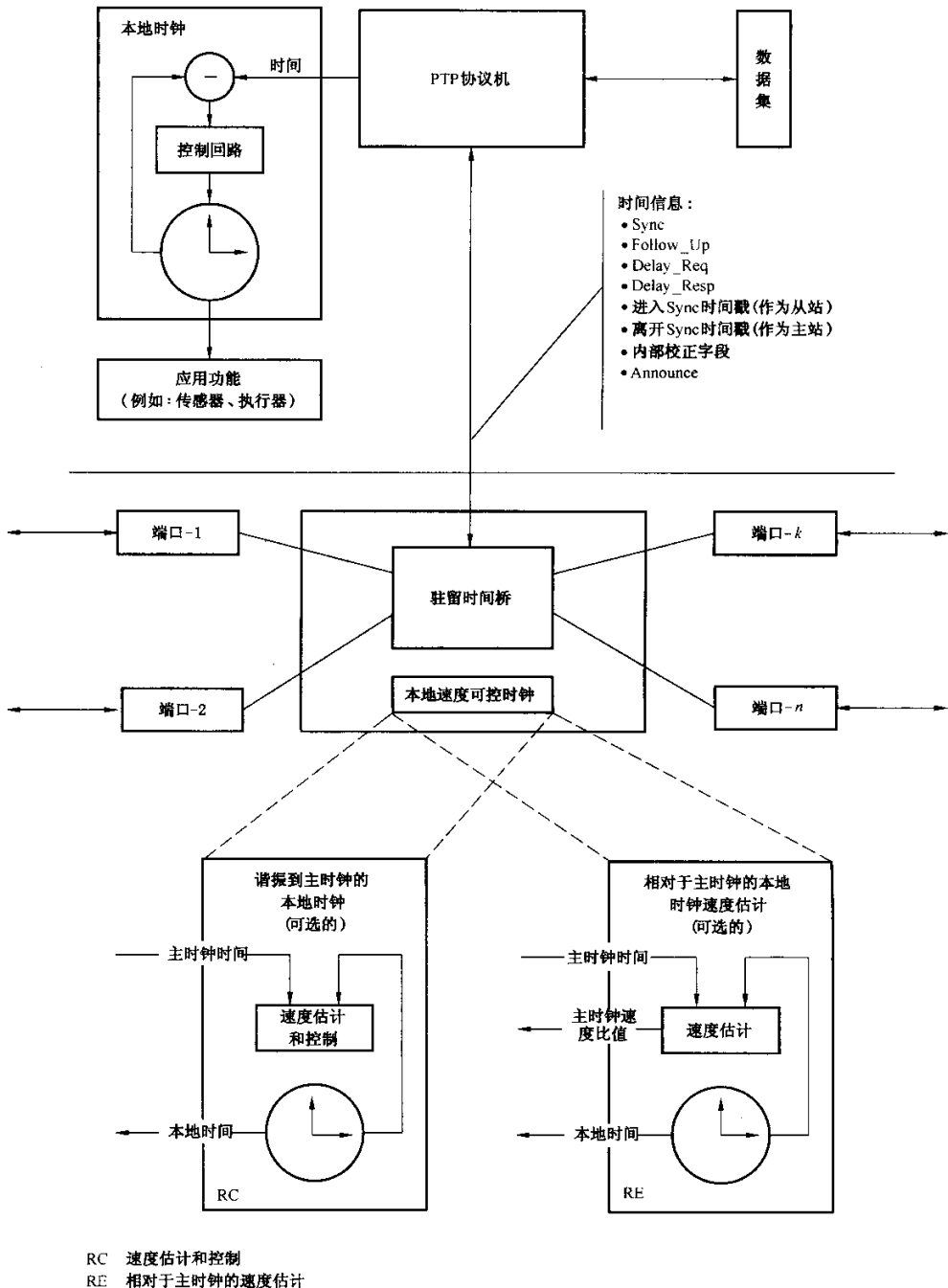


图6 组合的普通时钟和端到端透明时钟

6.5.5 点到点透明时钟

点到点透明时钟的模型如图7所示。

点到点透明时钟与端到端透明时钟在校正和处理 PTP 时间报文的方式上不同,在其他所有方面,两者完全一致。

点到点透明时钟的每个端口具有一个额外的块,如图 7 所示。该块用于计算每个端口同另一个节点上相似配置的端口间的链路延时,这两个端口共享该链路,即链路对等点。鉴于不希望在点到点透明时钟间出现非点到点设备,链路对等点存在于另一个支持对等延时机制的时钟。链路延时的计算是基于与链路对等点交换 Pdelay_Req 报文、Pdelay_Resp 报文和可能的 Pdelay_Resp_Follow_Up 报文。作为这些交换的结果,点到点透明时钟的每个端口都知道链路延时。与校正和转发所有 PTP 时间报文的端到端时钟不同,点到点透明时钟仅校正和转发 Sync 和 Follow_Up 报文。在这些报文中相应的 correctionField 被更新,包括点到点透明时钟内 Sync 报文的驻留时间以及接收 Sync 报文端口的链路延时的更新。

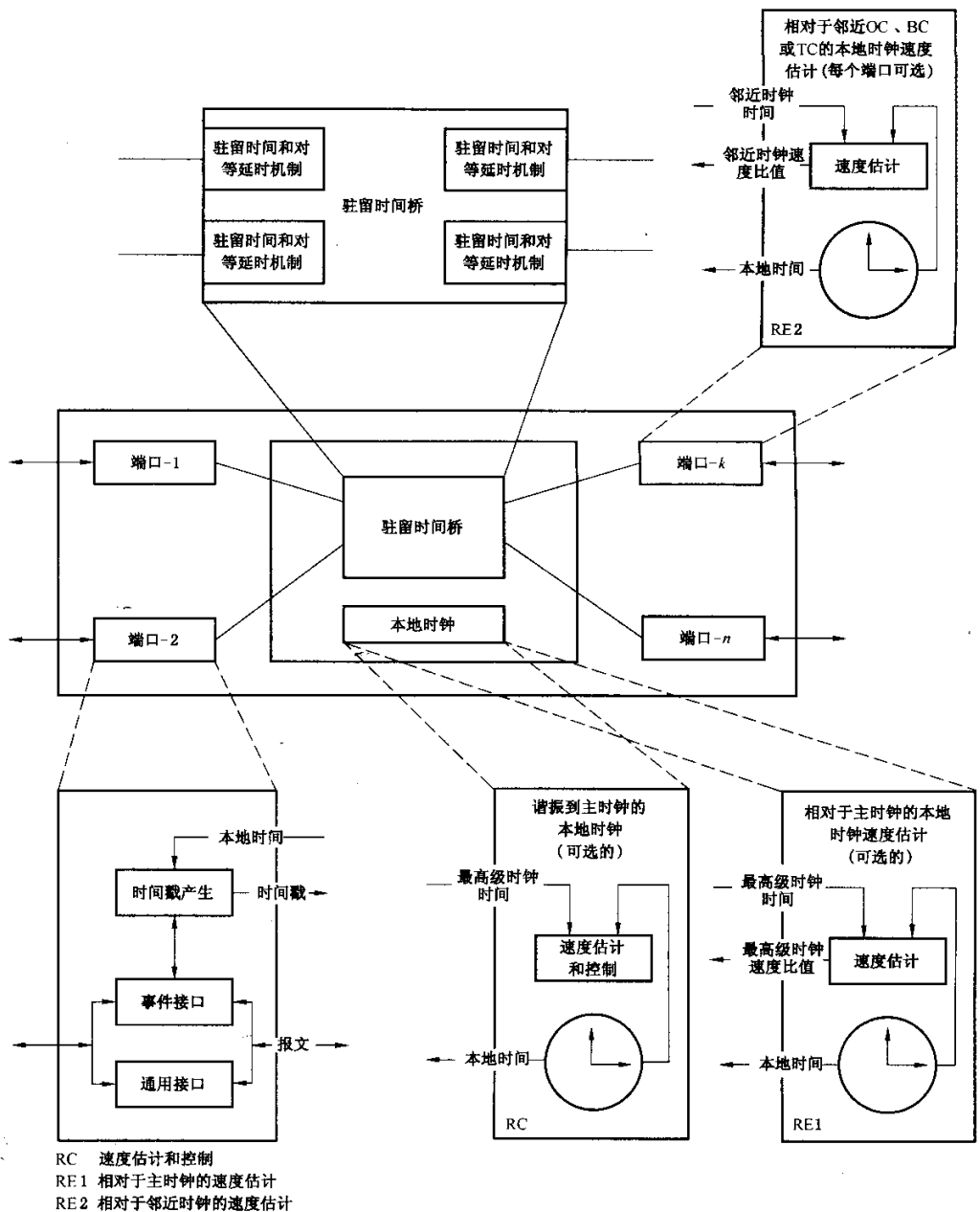


图 7 点到点透明时钟模型

校正过程如图 8 所示。离开 PTP 时间报文包含 Sync 数据包驻留时间和其通过实际进入链路的链路延时的校正。通过点到点时钟系统的 Sync 包路由的改变可能导致 Sync 包从不同的链路进入点到点透明时钟。鉴于点到点透明时钟基于每个 Sync 包选择的实际链路和内部路径来校正驻留时间和路径时间,在测量精度范围内的离开时间信息总是准确的。因此,为从时钟提供的时间信息总是反映通过点到点透明时钟网络的实际路径。对比端到端校正情况,从时钟等待基于 Sync 报文和 Delay_Resp 报文的组合的新路径延时值,鉴于在校正值出现在从时钟之前这些报文需要通过整个新路径,将等待更长的时间。

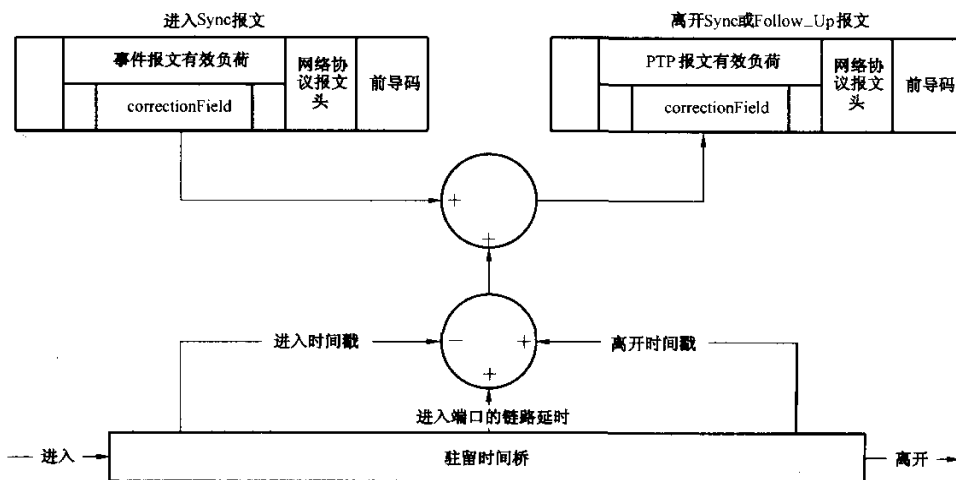


图 8 点到点驻留时间和链路延时校正模型

使用对等延时机制的路径延时测量不与基于延时请求-响应机制的路径延时测量相互作用。结果是,点到点透明时钟仅能与支持对等延时机制的时钟端口工作。使用点到点透明时钟,主时钟仅需要发布 Sync 报文和 Follow_Up 报文,及响应 Pdelay_Req 报文。它不接收 Delay_Req 报文,所以处理工作负荷独立于一个给定端口服务的点到点透明时钟数目。同样地,端到端透明时钟仅支持延时请求-响应机制,不支持对等延时机制。如果网络在一个区域内包含点到点透明时钟,在另一个区域内包含端到端透明时钟,则这些区域仅能通过边界时钟连接。

计算驻留时间和链路延时时间使用的时间戳,是基于本地时钟产生的时间戳。因为从时钟使用这些累积的驻留时间和链路延时时间校正主时钟提供的时间,重要的是:任何由于主时钟和透明时钟速率差异所导致的误差对于应用要求的精度应是可忽略的。因为主时钟和本地时钟(本质上是两个时钟对时刻的定义)的速率可能相差 0.02%,则引入的误差为测量的驻留时间的 0.02%。这样对于 1 ms 的驻留时间,最大误差是 200 ns。该误差可能大的无法接受,有几个可能降低误差的方法。

一种降低最大误差的方法(在 11.4 中对其展开讨论)是使本地时钟的速率与主时钟相同,即本地时钟与主时钟同步。该同步可通过观测的时间信息来完成,时间信息在接收到的 Sync 报文和 Follow_Up 报文(如存在)中,且已校正过上行驻留时间和链路延时,见图 7 中的块 RC(速率控制)。已校正过任何上行驻留时间的最高级时间是 RC 中速率估计和控制块的输入(见标记为“最高级时钟时间”的箭头)。相应的本地时间也是该块的输入。速率估计和控制块使用主时钟和对应的本地时间序列,以估计主时钟和本地时钟速率的比值。速率估计和控制块使用此估计的速率比值来调整(即控制)本地时钟速率。注意:本地时钟速率的调整不必物理改变(即模拟实现)振荡器频率;使用固定频率的本地振荡器来计算主时钟速率和本地速率的比值,并将本地时钟的时间戳乘以该比值(即数字实现)也是可以接受的。RC 描述的方案的关键方面在于该方案是以闭环方式运行,即速率估计和驻留时间测量时使用的本地时间是相对于速率调整过的本地振荡器的。这意味着在一个节点上对振荡器的速率调整影响到下行节点的

调整,因为在下行节点中驻留时间被用于调整主时钟时间。

另一种在某些应用中可能有用的降低驻留时间误差影响的方法是使用自由运行的本地时钟,见图 7 中块 RE1(主时钟速率比值估计)。在这种情况下,不调整本地振荡器的频率,允许它自由运行。每个时钟测量基于自由运行时钟的 Sync 报文的本地驻留时间。测量的驻留时间被加到 Sync 或 Follow_Up 报文的 correctionField 字段,见 11.4。每个本地时钟也计算本地自由运行时钟速率与最高级时钟速率的比值。该比值是基于接收到的 Sync 和 Follow_Up 报文中的时间戳计算得到的,这里的时间戳已经通过加入 correctionField 进行了校正。尽管 correctionField 没有反映准确的驻留时间,得到的比值相比简单使用原时间戳,仍然极大地降低了抖动,而且仍然具有正确的平均值。此外,如果驻留时间相对于速率比值的测量间隔足够大,在最坏情况下在连续通过透明时钟的阶段误差累积将小于使用第一种方法的误差累积,但对于许多非最坏情况下会导致更大阶段累积。此外,每个本地时钟计算用该比值校正过的驻留时间和未校正的驻留时间的差值,这些差值也必须被累积并传递给从时钟。如使用该方法,它连同用于累积的行规特定 TLV 应在行规(见 19.3)中进行规定。使用该方法,在计算从时钟与最高级时钟的偏移量之前,从时钟必须将累积的差值加到接收到的 Sync 报文或 Follow_Up 报文的 correctionField。

使用这两种方法,Sync 或 Follow_Up 报文的 correctionField 也必须进行链路延时校正,见 11.4。使用对等延时机制测量链路延时(见 6.6.4)。如果点到点透明时钟谐振到如 RC 块所示的主时钟,则对等延时测量也可使用谐振时钟完成。但是如果测量透明时钟相对于最高级时钟的速率,并如 RE1 块使用该速率,则使用自由运行的本地时钟完成对等延时测量。在这种情况下,链路延时误差等于接收 Pdelay_Req 报文和发送 Pdelay_Resp 报文之间所经过的时间,即 $t_3 - t_2$ (见 6.6.4 和图 13)乘以透明时钟和相邻时钟之间的小数频率偏移。为消除链路延时的这部分误差,使用包含在连续 Pdelay_Resp 报文,和可能的 Pdelay_Resp_Follow_Up 报文中的时间戳信息来测量透明时钟和相邻时钟之间的频率偏移,见图 7 中 RE2(相邻速率比值估计)块。注意:为完成该测量,分开发送 Pdelay_Req 接收时间(t_2)和 Pdelay_Resp 发送时间(t_3)是必要的(见 11.4.3)。

点到点透明时钟可用作网络元件,或可以与应用设备关联,如传感器或执行器。在后一种情况下,将普通时钟与透明时钟结合,以提供对应用设备的实时支持。这种组合设备的模型见图 9。

在图 9 中,当普通时钟是从时钟时,时间信息包含进入时间戳。当普通时钟是主时钟时,时间信息包含离开时间戳。当时钟是从时钟时,驻留时间桥将进入的 PTP 时间报文、Announce 报文、由进入 Sync 报文产生的进入时间戳,以及任何内部时间校正传递给普通时钟协议机。协议机计算基于这些报文的准确时间,并作为输入发送给本地时钟。如果普通时钟是主时钟,它应发送带有发送时间戳的 Sync 报文和 Follow_Up 报文,该时间戳参考普通时钟的本地时钟,并且基于内部时间校正和离开时间戳。实际上点到点透明时钟和普通时钟功能可以共享一个公共物理时钟。

6.5.6 管理节点

管理节点是具有下述特点的 PTP 设备:

- a) 具有一个或多个到网络的物理连接;
- b) 作为 PTP 管理报文的人机接口或编程接口;
- c) 可以与任何时钟类型结合。

6.6 同步综述

6.6.1 概述

在协议的正常执行中有两个阶段:

- 建立主从层次结构;
- 同步时钟。

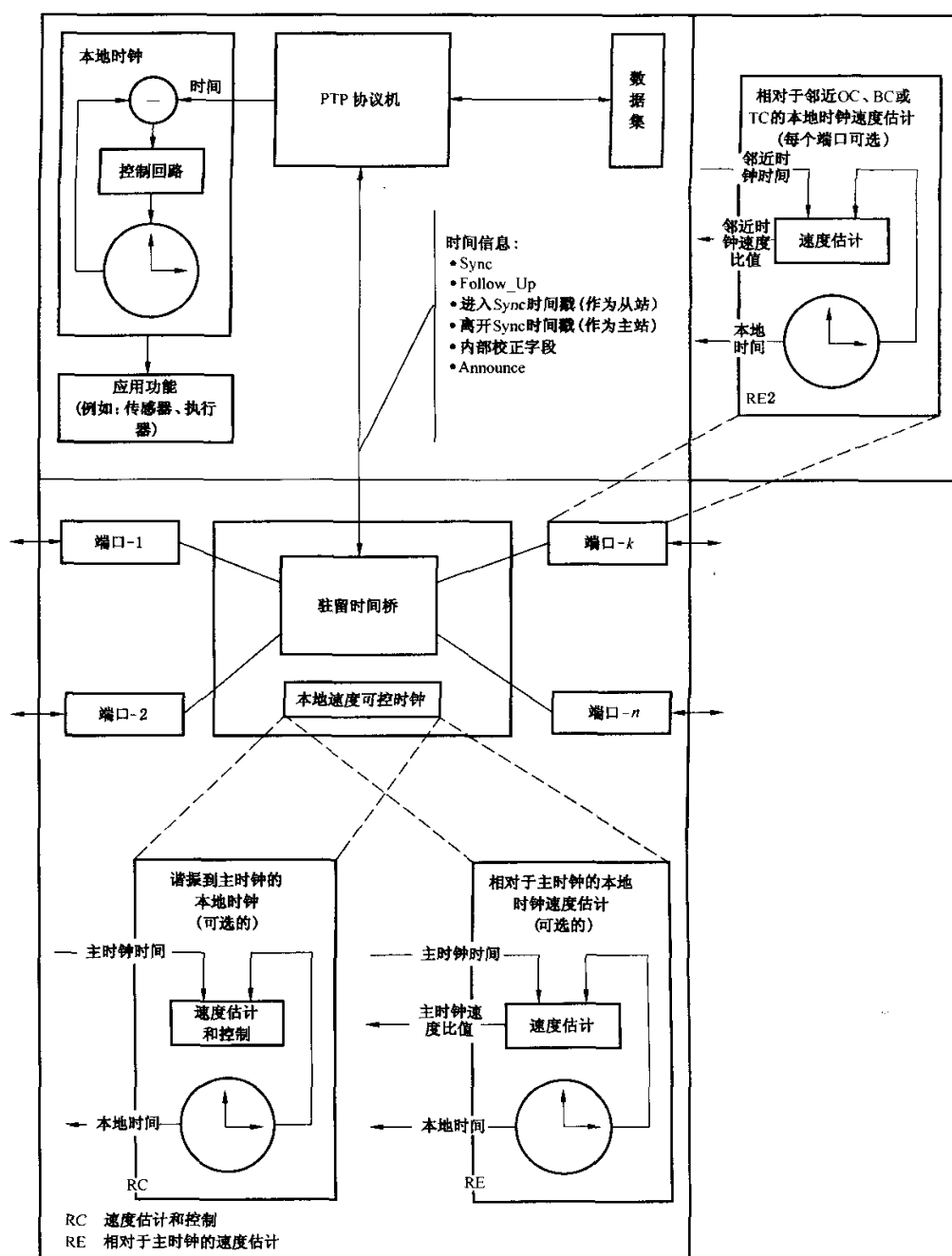


图9 组合的普通时钟和点到点透明时钟

6.6.2 建立主从层次结构

6.6.2.1 概述

在域内,普通时钟和边界时钟的每个端口执行协议状态机的单独副本。对于“状态决定事件”,每个端口检查该端口接收到的所有 Announce 报文的内容。使用最佳主时钟算法,对 Announce 报文内容与普通时钟或边界时钟相关联的数据集进行分析,以决定时钟每个端口的状态。

6.6.2.2 PTP 状态机

普通时钟和边界时钟的每个端口维护 PTP 状态机的单独副本。该状态机定义端口的允许状态和状态间的转换规则。决定主从层次结构的主要“状态决定事件”是接收 Announce 报文和 announceInterval(Announce 报文的间隔)结束。决定主从层次结构的端口状态如下所述：

- a) MASTER:端口是在其服务路径上的时间源；
- b) SLAVE:端口同步到路径上具有处于 MASTER 状态的端口的设备；
- c) PASSIVE:端口不是路径上的主时钟,也不同步到主时钟。

6.6.2.3 最佳主时钟算法

最佳主时钟算法比较描述两个时钟的数据,以确定哪些数据描述更好时钟。该算法用于确定由本地时钟端口接收到的几个 Announce 报文描述的时钟中,哪一个最佳时钟。它也可用于确定一个新发现的时钟——外部主时钟是否好于本地时钟。描述外部主时钟的数据包含在 Announce 报文中的 grandmaster 字段。描述本地时钟的数据包含在时钟的 defaultDS 数据集中。

最佳主时钟算法由两个单独的算法组成：

- 数据集比较算法；
- 状态决定算法。

数据集比较算法是基于按下述先后顺序对属性的成对比较：

- a) Priority 1:用户可配置标记,表示一个时钟属于一有序时钟集,从该时钟集中选出主时钟。
- b) clockClass:定义时钟的 TAI 可溯源性的属性。
- c) clockAccuracy:定义时钟精度的属性。
- d) offsetScaledLogVariance:定义时钟稳定性的属性。
- e) Priority 2:用户可配置标记,为其他相当时钟提供良好排序。
- f) clockIdentity:基于唯一标识符的决定性判据。

除了此先后顺序,当两个 Announce 报文反映相同的外部主时钟时,“距离”被使用,“距离”由本地时钟和外部主时钟之间的边界时钟个数来测量。“距离”在 Announce 报文的 stepsRemoved 字段指示。在带有循环路径的 PTP 系统中该条件会发生,这些循环路径不能由 PTP 以外的协议移除。数据集比较算法无歧义地选择两个时钟中的一个作为“更好的”或“拓扑更好的”。

状态决定算法在数据集比较算法结果的基础上确定端口的下一个状态——推荐的状态——是 MASTER、SLAVE,或 PASSIVE 中的哪一个,以及确定本地时钟类是否小于 128,见 7.6.2.4。端口协议机根据协议状态机的当前状态估计该推荐的状态,以确定端口的下一个实际状态。

6.6.2.4 简单主从层次结构

在域内的普通时钟和边界时钟中建立主从层次结构的过程,如图 10 所示。

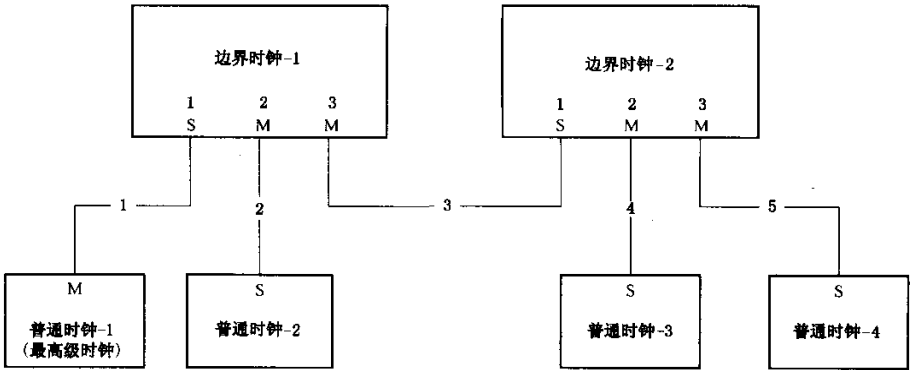


图 10 简单主从时钟层次结构

在这个示例中,普通时钟-1 处于层次结构中的最高级位置,被称为最高级时钟。边界时钟-1 的端口-1 对于最高级时钟是从时钟(被标记为 S)。在边界时钟-1 上的所有其他端口对于连接到这些端口的时钟是主时钟。这样,边界时钟-2 的端口-1 对于边界时钟-1 是从时钟,依此类推。仅普通时钟和边界时钟维护这种形式的状态,且仅边界时钟建立主从层次结构中的分支点(即路径 1、2、3、4、5 可能包含透明时钟,但这些时钟不加入到主从层次结构,也不维护这种形式的状态)。

6.6.2.5 裁剪网状拓扑

图 11 所示为协议将一个网状网络简化为树形主从层次结构的案例。当底层的桥接或路由协议不能消除循环路径时,上述处理发生。在图 11 中,认为普通时钟-1 已被最佳主时钟算法选定为最高级时钟。在边界时钟中,最佳主时钟算法已选择了端口状态以构建树,如图所示。裁剪的路径用虚线标记。对于每个边界时钟,最佳主时钟算法选择一个端口作为从时钟端口,其他端口被设置为 MASTER 状态或 PASSIVE 状态。最佳主时钟算法确保在每个段上只选择一个单主时钟端口。

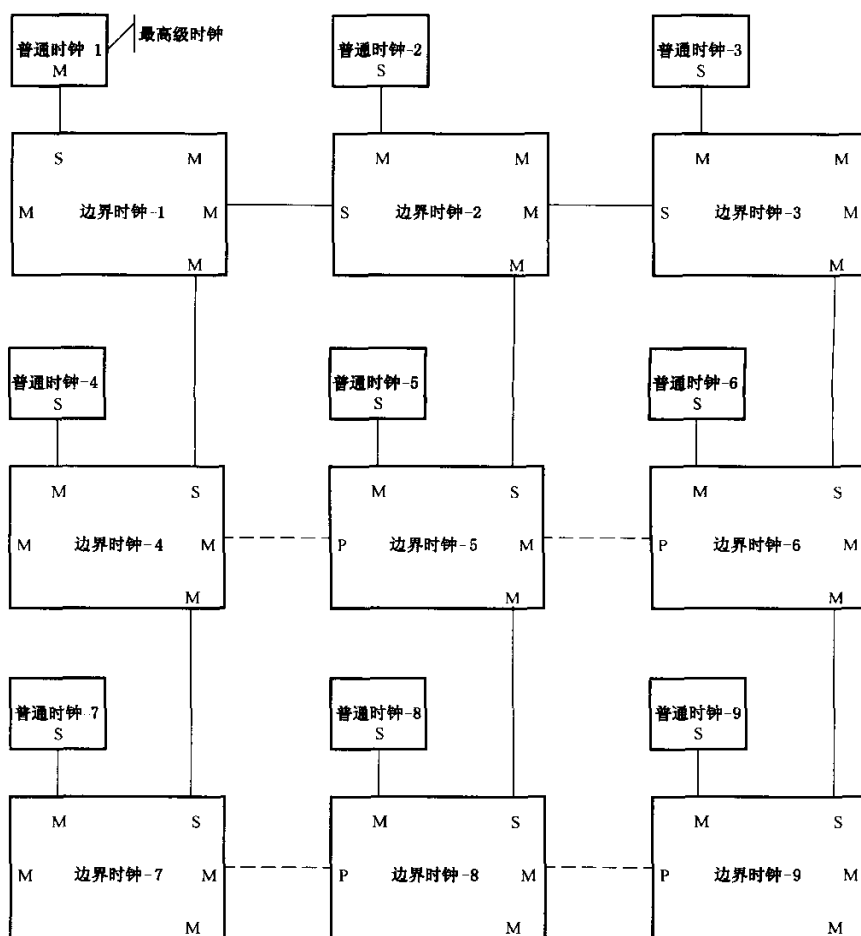


图 11 裁剪的网状拓扑

6.6.3 同步普通时钟和边界时钟

在 PTP 系统中,通过在链接两个时钟的通信路径上交换 PTP 时间报文,实现普通时钟或边界时钟的同步。例如,在图 10 中,边界时钟-1 通过在通信路径-1 上交换报文实现与普通时钟-1 的同步。同步报文交换的基本模式如图 12 所示。

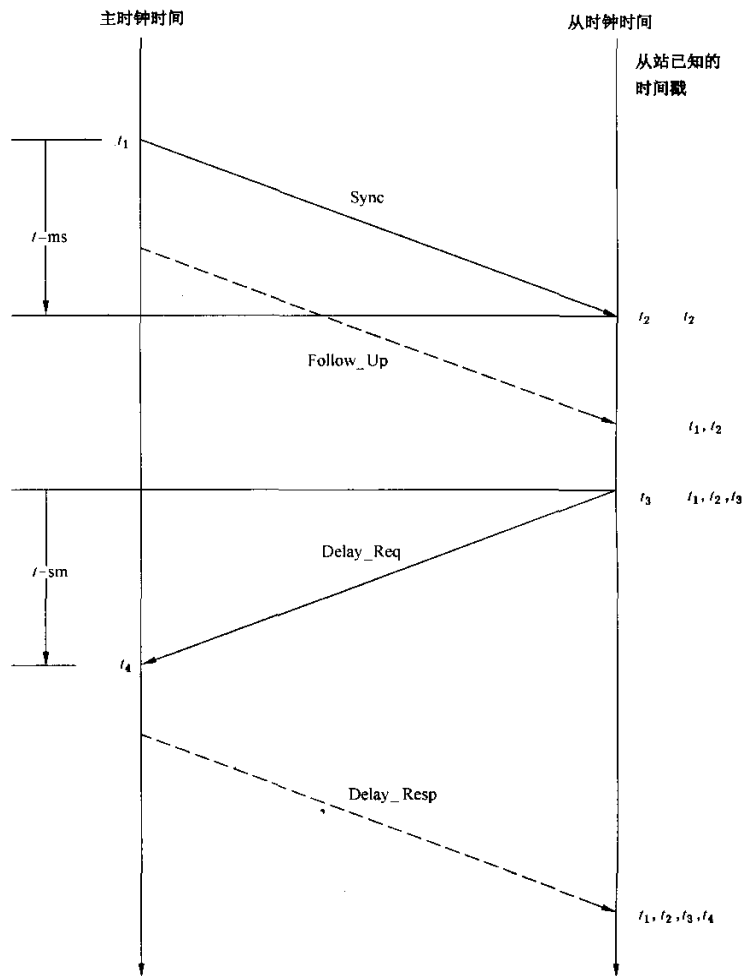


图 12 基本同步报文交换

报文交换模式如下所述：

- 主时钟向从时钟发送 Sync 报文,并注明发送报文的时间 t_1 。
- 从时钟接收 Sync 报文,并注明接收报文的时间 t_2 。
- 通过如下方式,主时钟向从时钟传输时间戳 t_1 :
 - 在 Sync 报文中嵌入时间戳 t_1 。这要求某些类型的硬件处理,以得到最高的精度。
 - 在 Follow_Up 报文中嵌入时间戳 t_1 。
- 从时钟向主时钟发送 Delay_Req 报文,并注明发送报文时的时间 t_3 。
- 主时钟接收 Delay_Req 报文,并注明接收报文时的时间 t_4 。
- 通过在 Delay_Resp 报文中嵌入时间戳 t_4 ,主时钟将时间戳传输给从时钟。

在交换报文结束时,从时钟拥有所有 4 个时间戳。这些时间戳可被用于计算从时钟与主时钟的偏移量,以及计算报文在两个时钟间的平均传输时间(在图 12 中即为 $t-ms$ 和 $t-sm$ 的平均值)。

偏移量和传输时间的计算是假设主时钟到从时钟的传输时间与从时钟到主时钟的传输时间相等。任何在传输时间上的不对称将导致时钟偏移量计算值的误差。不对称会造成计算的平均传输时间与实际的传输时间不同。

6.6.4 支持点到点路径校正的时钟的链路传输延时测量

在实现对称延时机制(见 6.5.5)的两个端口间测量链路延时的机制如图 13 所示。该测量被所有实现此机制的端口执行。共享一个链路的两个端口独立实现测量,因此两个端口都知道链路延时。这允许在 6.5.5 中描述的校正不考虑 Sync 报文的方向。重要的是:即使端口被用于消除循环拓扑的非 PTP 算法所阻塞(blocked),仍可在其上进行该测量,以便如果 Sync 报文通过的路径发生改变时,最新的链路延时测量仍可在所有链路上进行(见 6.5.5)。

链路延时测量开始时,端口-1 发出 Pdelay_Req 报文并为 Pdelay_Req 报文产生时间戳 t_1 。端口-2 接收 Pdelay_Req 报文并为该报文产生时间戳 t_2 。端口-2 返回 Pdelay_Resp 报文并为该报文产生时间戳 t_3 。为最小化由于两个端口之间的频率偏移产生的误差,端口-2 在接收到 Pdelay_Req 报文后应尽快返回 Pdelay_Resp 报文。

端口-2 执行下面任意一个动作:

- a) 在 Pdelay_Resp 报文中返回时间戳 t_2 和 t_3 的差值;
- b) 在 Pdelay_Resp_Follow_Up 报文中返回时间戳 t_2 和 t_3 的差值;
- c) 分别在 Pdelay_Resp 和 Pdelay_Resp_Follow_Up 报文中返回时间戳 t_2 和 t_3 。

当端口-1 接收到 Pdelay_Resp 报文时产生时间戳 t_4 。端口-1 使用这 4 个时间戳计算平均链路延时。

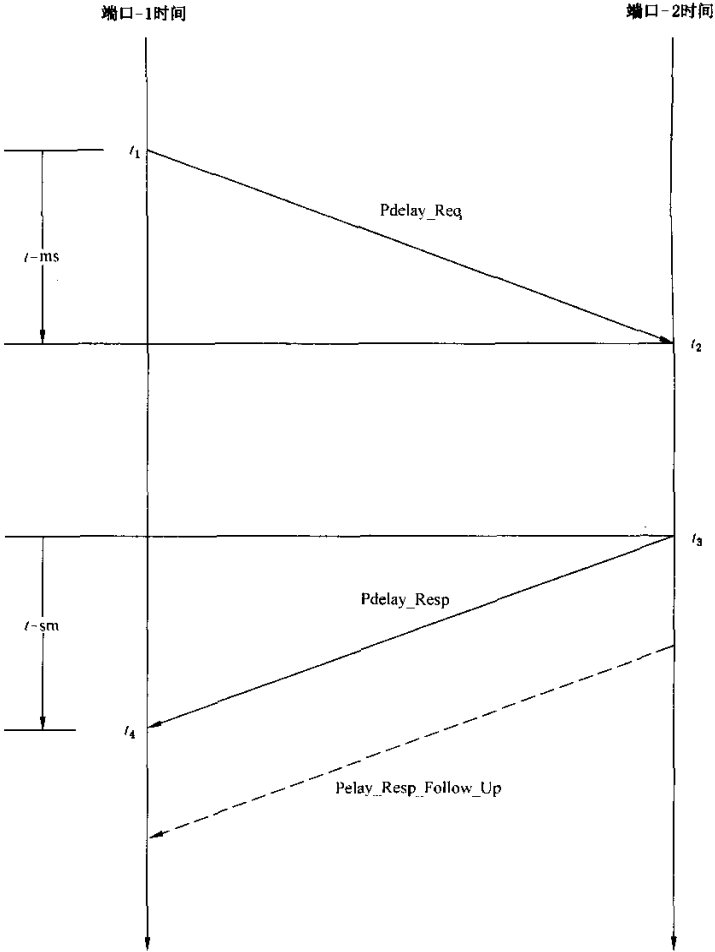


图 13 链路延时测量

传输时间 t_{ms} 和 t_{sm} 的任何不对称会将误差引入链路延时的计算值中。如果产生时间戳 t_1 、 t_2 、 t_3 和 t_4 的机制不与发出 Sync 报文的最高级时钟使用相同的时刻定义,则小的但可能很严重的误差被引入链路延时测量。该误差的源是在最高级时钟和点到点时钟中振荡器频率的微小差异。此外,如果在两个点到点时钟的振荡器频率之间有差异时,则周转时间 $t_3 - t_2$ 会造成在点到点时钟中存在小的但可能很严重的误差。如果该误差很严重,则在点到点时钟中的振荡器能频率同步(即谐振)到最高级时钟的振荡器,或在计算的两个振荡器间的频率差异基础上,计算和校正该误差。

6.6.5 报文时间戳产生

在发送和接收任何事件报文时,产生时间戳事件。当报文的时间戳点通过节点和网络间的边界时,时间戳事件发生。

时间戳(见图 12 和图 13)产生的模型见图 14。

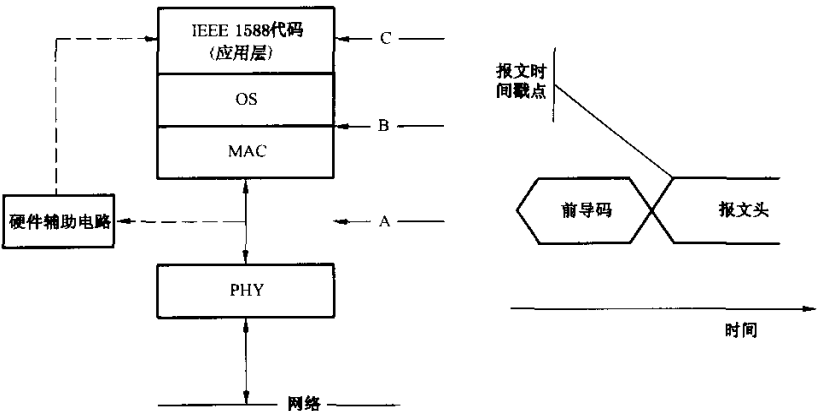


图 14 时间戳产生模型

PTP 时间报文从一个时钟的 PTP 应用代码中发出,由另一个时钟的 PTP 应用代码接收和处理。这些报文典型地具有在网络中使用的通信协议物理层规定的前导码。前导码之后是一个或多个协议特定报文头,然后是用户数据如 PTP 有效负荷。对于每个这样的传输机制,本标准在 PTP 时间报文中规定了一个特定点,通常为帧信号的开始,作为可辨识点被称为报文时间戳点。当 PTP 报文通过节点的协议栈,且报文时间戳点通过协议栈内的定义点时,产生时间戳。该定义点可以在应用层,在图 14 中标记为“C”;在内核或中断服务程序中,标记为“B”;或在协议栈的物理层中,标记为“A”。通常,该点离实际网络连接越近,则由通过较低层所用时间的波动引入的时间误差越小。时间戳在物理层产生的情况下,经常会使用某些类型的硬件辅助电路(用虚线标记)。在这种情况下,通过正常路径外的路径将时间戳传递给 PTP 代码,时间戳后跟着 PTP 时间报文本身。为确保时间戳与正确报文相关联,硬件辅助电路经常从 PTP 时间报文(与时间戳一起被传递给 PTP 代码)中捕捉附加信息。

当设计数据包识别器、时间戳发生器和在物理层完成的任何形式的 PTP 硬件辅助电路时,在物理层产生 PTP 时间戳的实现必须考虑传输格式的映射(见 5.4)。

设计在媒体访问控制(MAC)层和物理层(PHY)之间,或甚至在物理层内产生时间戳的设备是可能的。在这样的设备内,所有或部分 PTP 代码很可能在低级半导体内嵌入并执行,而不使用操作系统。

6.7 PTP 通信综述

6.7.1 PTP 通信拓扑

6.7.1.1 概述

在仅由边界时钟和普通时钟组成的系统中,PTP 操作为 PTP 报文产生非循环图结构,而不考虑实际底层网络连接,后续技术由 Perlman[B22]描述。

在所有其他情况下,PTP 假设底层桥接或路由协议确保 PTP 报文转发避免环路。特别地,协议假设在通信路径内多播 PTP 报文不会形成不确定的环路。协议不假设:由 PTP 端口发送的多播报文仅在接收边界时钟的一个端口接收。协议不假设在一个 PTP 端口发送的多播报文的仅一个副本被另一个 PTP 端口接收;然而,对多个副本的接收影响传输时间的精度,因此应对网络进行设计以避免这种行为。

6.7.1.2 层次拓扑

不同的应用倾向不同的拓扑。对于许多系统,优先使用图 15 所示的层次拓扑。

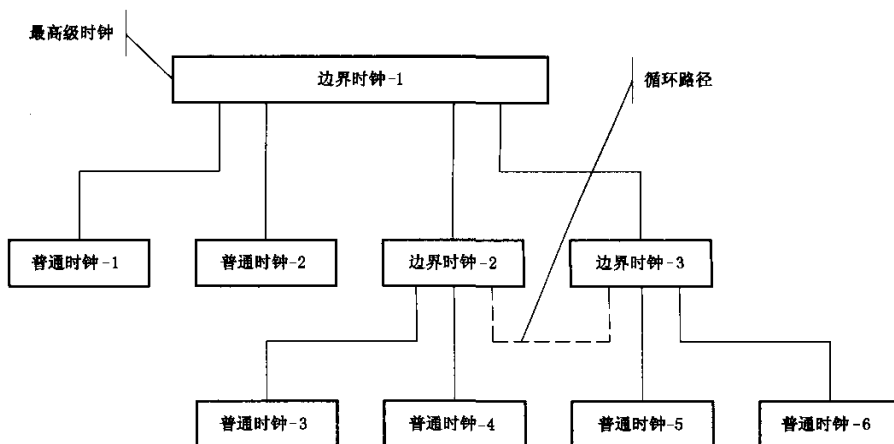


图 15 层次拓扑

除了用虚线标记的“循环”路径,图 15 中所示的其他设备形成树型层次结构,边界时钟在树中形成了分支点。因为边界时钟可有许多端口,这种拓扑允许大量的设备与在任何从时钟和最高级时钟之间的少数边界时钟同步。在此例中,假定虚线路径不存在。如果边界时钟-1 被选为最高级时钟,如图所示,则中间边界时钟的最大数量是 1。最坏的情况是其中一个普通时钟,如时钟-3,被选为最高级时钟。则到最远从时钟,如时钟-6,边界时钟的最大数量是 3。

如 6.2 所述,协议期望底层拓扑避免转发环路。在图 15 中,如果有虚线路径,则存在涉及到 3 个边界时钟的环路。在 6.6.2 中描述最高级时钟和状态决定算法的操作作为 PTP 报文打开该环路。

6.7.1.3 线型拓扑

一些应用要求如图 16 所示的长线型拓扑,而不是图 15 所示的星型或层次拓扑。图 16 中所示为端到端透明时钟的两个长线型链,其中边界时钟-1 作为最高级时钟。

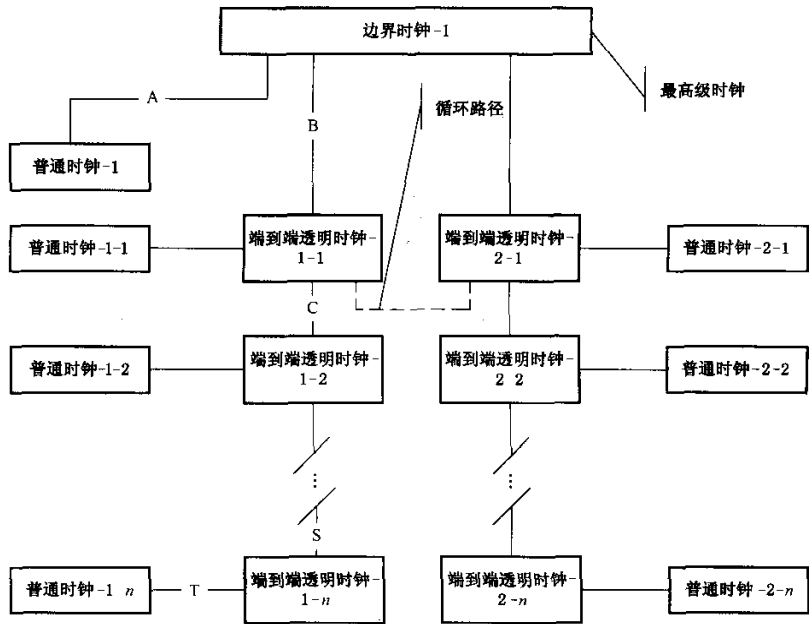


图 16 线型拓扑

普通时钟和边界时钟间的同步涉及两个时钟间 PTP 时间报文的交换,例如在路径-A 上。这些 PTP 时间报文对于系统中的其他时钟是不可见的。基于 PTP 时间报文,这种主-从时钟对中的从时钟执行某些类型的伺服机制以降低时钟偏移误差。

按普通网桥或路由器方式工作的透明时钟通过该时钟转发 PTP 时间报文。此外透明时钟测量在其内部 PTP 时间报文所用的时间。这些“驻留”时间在 PTP 时间报文的 correctionField 字段中累积,这允许从时钟校正时间戳,从而有效移除可能由网桥引入的时间波动。缺点是:主时钟,本例中的边界时钟-1,不得不处理来自线型链中所有从时钟的 PTP 时间报文(即 Delay_Req),而不是来自一个临近的边界时钟。

注:这种缺点对主时钟的影响能通过使用点到点透明时钟或边界时钟避免。

6.7.1.4 在多连接拓扑中的快速重新配置

在许多应用中,设备被放置在多个相连的拓扑中,例如:图 17 所示的网状或环路。在这些网状或环路中,备选路径被非 PTP 协议逻辑地移除以形成非循环拓扑。路径失效的情况下,这些外部协议重新配置网络以恢复连接。由于 PTP 运行在底层快速重新配置的网络的顶部,在重新配置后,PTP 可能不得不重新调整在主时钟和从时钟之间路径长度的校正。

点到点透明时钟被设计用于在本环境下使用。

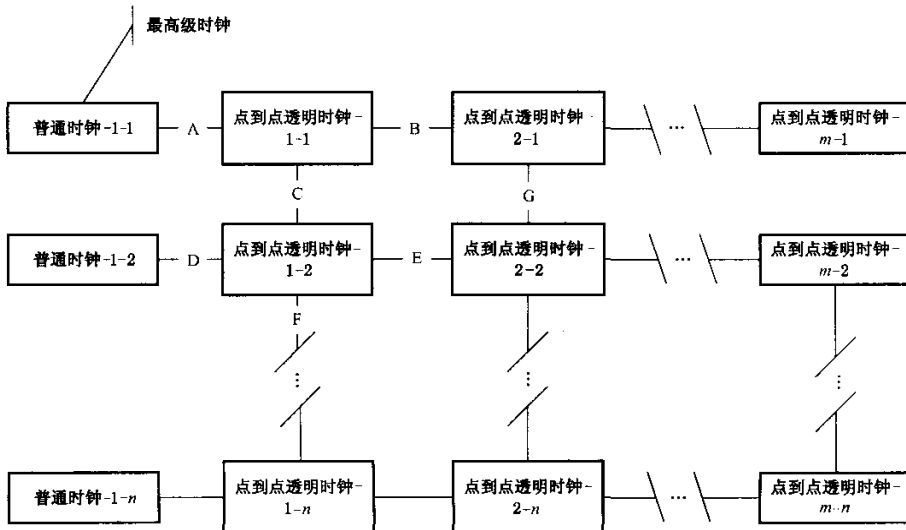


图 17 多连接拓扑

图 17 所示是连接到网状拓扑的几个点到点透明时钟。点到点透明时钟的运行依赖一些能消除网络中循环路径的外部协议。与端到端透明时钟的情况一样，点到点透明时钟常与普通时钟相关联用于传感器或其他设备。两类时钟的差异在于路径长度校正的方法。

假设在普通时钟-1-1(最高级时钟)和普通时钟-1-2(从时钟)间的初始路径是非 PTP 协议确定的 A、B、G、E、D。来自普通时钟-1-1 的 Sync 报文，由点到点时钟-1-1 针对其在点到点时钟内的驻留时间和链路延时 A 进行校正。同样，点到点时钟-2-1 进一步对其驻留时间和链路延时 B 进行校正，依此类推。所以普通时钟-1-2 接收到的 Sync 报文，其在点到点时钟-1-1、2-1、2-2 和 1-2 中的驻留时间和链路延时 A、B、G 和 E 已校正。也支持对等链路延时机制的普通时钟-1-2 校正最后一个链路延时 D。

假设对网络进行重新配置，使这两个时钟之间的新路径是 A、C、D。点到点时钟-1-1 完成与之前相同的校正。但是在此情况下，点到点时钟-1-2 直接从点到点时钟-1-1 接收 Sync 报文，所以对驻留时间和之前已经测量过的链路延时 C 进行校正。正是由于对所有链路的链路延时的预先测量(无论是否起作用)，才允许这种快速重新配置。

6.7.1.5 在不同网络协议间的桥接

不要求所有 PTP 通信路径使用相同的底层通信媒体或技术。边界时钟用于在不同网络传输技术间桥接，如图 18 所示。

在图 18 中，边界时钟-1 和普通时钟-1、普通时钟-3 和普通时钟-5 被假定通过标记为 A 的实现了一种技术(例如：UDP/IP)的网络路径进行通信。边界时钟-2 和普通时钟-2、普通时钟-4 和普通时钟-6 通过标记为 B 的实现了第二种技术(例如：DeviceNet)的网络路径进行通信。其中一个边界时钟，图 18 中的边界时钟-2，是一个在端口-2 支持技术 A 以及在其他端口支持技术 B 的网桥。

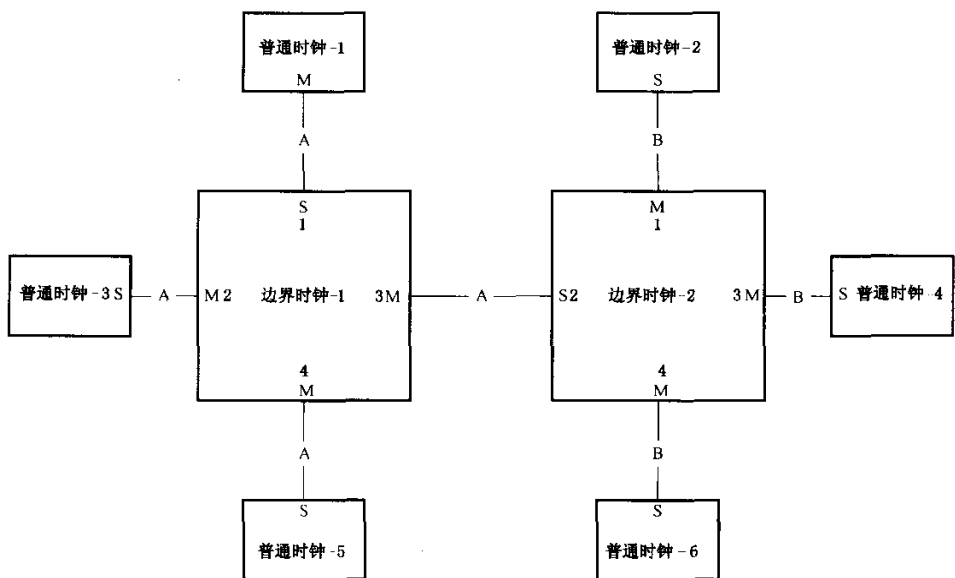


图 18 完全不同技术的桥接

在大多数情况下,这种网桥不仅涉及网络传输协议的改变,如在上面例子中 UDP/IP 和 DeviceNet 之间,也可能涉及其他 PTP 特性,如更新速率。在这种情况下,边界时钟是维护足够的 PTP 状态以完成桥接功能的唯一设备。在一些情况下,要求的唯一桥接功能是转换数据包格式或其他物理层发出的(信号)。在这些情况下,因为不要求 PTP 状态信息,所以设计透明时钟完成桥接是可能的。

6.7.2 系统启动

当时钟上线时,为提供更多有序行为,普通时钟或边界时钟在可配置的时间间隔内侦听来自主时钟的 Announce 报文。如果在这个时间段内没有接收到 Announce 报文,该时钟假定在“更好的”时钟出现前它就是主时钟。

当时钟被加入或删除、时钟特性改变或连接拓扑改变,在 PRE_MASTER 状态下实现了一个支持系统更有序重新配置的附加机制。在这种状态下,除了不将某些报文类放置在端口通信路径上,时钟端口的行为应与它处于 MASTER 状态时的行为一致。时钟端口应保持在 PRE_MASTER 状态足够长的时间,以允许在系统内点间传输的改变,这些点在本地时钟和该端口可见的可能主时钟之间。

7 PTP 实体的特性

7.1 域

域由一个或多个 PTP 设备组成,这些设备按照协议的定义相互通信。域应定义 PTP 报文通信范围、状态、操作、数据集和时标。PTP 设备可以加入多个域中,但除非在本标准中另有说明,不同域的协议运行和时标是独立的。

注 1: 限制 PTP 在一个域内运行和通信的机制可能涉及到,例如:通信特定技术,如路由器表配置,限制物理连接以及应用层处理 PTP 报文的 domainNumber 字段的可靠性。

注 2: PTP 节点的实现者需要考虑支持多个域所要求的资源。这对于普通时钟和边界时钟尤为正确,因为普通时钟和边界时钟相比透明时钟需维护更多的状态。由于资源的限制造成不能以实时模式处理协议,可能导致协议同步性能降低、波动,或协议失效。当选择节点和设计系统时,用户需要意识到该限制。

由取值范围为 0~255 的整数(见表 2),domainNumber,对域进行标识。

表 2 domainNumber 取值

值(十进制)	定 义
0	缺省域
1	备选域 1
2	备选域 2
3	备选域 3
4~127	用户定义域
128~255	保留

domainNumber 为 0 的域作为缺省域。domainNumber 的值应是可配置的,且遵从 PTP 行规规定限制。

7.2 PTP 时标

7.2.1 概述

域的时标由最高级时钟建立。

PTP 支持下述两类时标:

- 时标 PTP:在正常运行时,历元是 PTP 历元且时标是连续的,见 7.2.4,时间测量的单位是 SI_s (在旋转的大地水准面上实现);
- 时标 ARB(任意):在正常运行时,历元由管理程序设置,且在正常运行期间允许对其复位,在管理程序调用之间,时标是连续的。额外的调用管理程序可能导致整个时标的不连续。

7.2.2 历元

历元是一个域的时标的原点。

PTP 历元是 1970 年 1 月 1 日 00:00:00 TAI,也是 1969 年 12 月 31 日 23:59:51.999918 UTC。

注 1: PTP 历元与通用可移植操作系统接口 (POSIX)算法的历元相同,该算法将从历元开始经过的时间转换到日时的 ISO 8601:2004 印刷表示法,见 ISO/IEC 9945:2003 [B16]和 ISO 8601:2004 [B17]。

注 2: 在通用时标间转换的信息见附录 B。

7.2.3 UTC 偏移量

当历元是 PTP 历元时,使用 timePropertiesDS.currentUtcOffset 的值计算 UTC 时间是可行的。

timePropertiesDS.currentUtcOffset 的值应为:

$\text{timePropertiesDS.currentUtcOffset} = \text{TAI} - \text{UTC}$ 。

注: 在 2006 年 1 月 1 日 0 时 UTC,UTC 比 TAI 延迟 33 s。在该时刻 currentUtcOffset 的 PTP 定义的价值为 +33 s (Service de la Rotation Terrestre, Observatoire de Paris [B23])。

7.2.4 域内的时间测量

在域内,时间是从历元开始测量的经过时间。

7.3 PTP 通信

7.3.1 报文传输模型

尽管本标准是基于多播模型制定的,如果协议的行为被保持也允许基于单播模型的实现。

7.3.2 报文属性

所有 PTP 相关通信通过 PTP 报文实现。PTP 报文具有下述属性:

- 报文类;
- 报文 sourcePortIdentity;
- 报文类型;
- 报文的 sequenceId;
- 定义可选项的标记。

7.3.3 报文类

7.3.3.1 事件报文

事件报文类由下述报文类型组成:

- a) Sync:主时钟将 Sync 报文发送给它的从时钟。Sync 报文包含其发送时间或跟随的 Follow_Up 报文包含该时间。接收节点可以使用 Sync 报文测量从主时钟到从时钟的数据包传输延时。Sync 报文在 PTP 端口参考面的出现是一个事件,本地时钟应基于本地时钟值为该事件分配时间戳(syncEventIngressTimestamp)或(syncEventEgressTimestamp)。
- b) Delay_Req:Delay_Req 报文请求接收节点使用 Delay_Resp 返回 Delay_Req 报文被接收到的时间。Delay_Req 报文在 PTP 端口参考面的出现是一个事件,本地时钟应基于本地时钟值为该事件分配时间戳(delayReqEventIngressTimestamp)或(delayReqEventEgressTimestamp)。
- c) Pdelay_Req:PTP 端口将 Pdelay_Req 报文发送给作为对等延时机制(见 11.4)一部分的另一个端口,以确定端口间链路的延时。Pdelay_Req 在 PTP 端口的参考面的出现是一个事件,本地时钟应基于本地时钟值为该事件分配时间戳(pdelayReqEventIngressTimestamp)或(pdelayReqEventEgressTimestamp)。
- d) Pdelay_Resp:PTP 端口发送 Pdelay_Resp 报文作为对接收 Pdelay_Req 报文的响应。在 Pdelay_Resp 报文中传输时间戳信息有如下几个选项:
 - 1) Pdelay_Resp 报文发送时间和相应 Pdelay_Req 报文的接收时间的差值在 Pdelay_Resp 报文中传输;
 - 2) Pdelay_Resp 报文发送时间和相应 Pdelay_Req 报文的接收时间的差值在跟随 Pdelay_Resp 报文的 Pdelay_Resp_Follow_Up 报文中传输;
 - 3) 相应 Pdelay_Req 报文的接收时间在 Pdelay_Resp 报文中传递,Pdelay_Resp 报文的发送时间在跟随 Pdelay_Resp 报文的 Pdelay_Resp_Follow_Up 报文中传递。
 Pdelay_Resp 报文在 PTP 端口的参考面的出现是一个事件,本地时钟应基于本地时钟值为该事件分配时间戳(pdelayRespEventIngressTimestamp)或(pdelayRespEventEgressTimestamp)。

应按 7.3.4 中的规定为事件报文分配上述时间戳。

7.3.3.2 通用报文

通用报文类由下述报文类型组成:

- a) Announce:Announce 报文提供发送节点和它的最高级时钟的状态和特性的信息。当执行最佳主时钟算法时接收节点使用该信息;
- b) Follow_Up:在双步普通时钟或边界时钟(见 3.1.4.7)中,Follow_Up 报文为特殊 Sync 报文传送(syncEventEgressTimestamp)值;
- c) Delay_Resp:Delay_Resp 报文向发送 Delay_Req 报文的从时钟端口传送(delayReqEventIngressTimestamp)值;
- d) Pdelay_Resp_Follow_Up:在支持对等延时机制的双步时钟中,Pdelay_Resp_Follow_Up 报文传递发送时间戳(pdelayRespEventEgressTimestamp),该时间戳由 PTP 端口在发送 Pdelay_Resp 报文时产生;
- e) Management:管理报文传输用来管理时钟的信息和命令;
- f) Signaling:信号报文在时钟间传递信息、请求和命令。

不要求为通用报文加上时间戳。

7.3.4 事件报文时间戳的产生

7.3.4.1 事件报文时间戳点

除非在本标准的传输特定附录中另有说明,事件报文的报文时间戳点应在帧起始定界符(SOF)之后第一个符号的起始处。

7.3.4.2 事件时间戳产生

所有 PTP 事件报文在进入和离开时被加上时间戳。时间戳应是事件报文时间戳点通过参考面的时间,该参考面标记了 PTP 节点和网络的边界。

- 注 1: 如果实现使用一个点而不是报文时间戳点产生事件报文时间戳,则产生的时间戳应由检测的实际时间和报文时间戳点通过参考面时间的时间间隔进行适当校正。校正失败会导致从时钟和主时钟之间的时间偏移量。
- 注 2: 通常,时间戳可以从参考面移开的点上产生。此外,与参考面的时间偏移量可能对于进入和离开事件报文是不同的。为满足本条的要求,应对产生的时间戳进行偏移校正。图 19 示出这些偏移。基于该模型,合适的校正如下:

$$\langle \text{egressTimestamp} \rangle = \langle \text{egressMeasuredTimestamp} \rangle + \text{egressLatency}$$

$$\langle \text{ingressTimestamp} \rangle = \langle \text{ingressMeasuredTimestamp} \rangle - \text{ingressLatency}$$

其中在参考面上测量的实际时间戳 $\langle \text{egressTimestamp} \rangle$ 和 $\langle \text{ingressTimestamp} \rangle$ 根据它们相应的等待时间从检测(即测量)的时间戳中计算得到。无法完成校正会导致从时钟和主时钟之间的时间偏移。

7.3.5 报文的 sourcePortIdentity

每个 PTP 报文包含标识离开端口的 sourcePortIdentity 字段,见 13.3.2.8。

7.3.6 报文类型

每个 PTP 报文包含命名 PTP 报文的 messageType 字段,见 13.3.2.2。

注: PTP 报文头也包含一个附加字段“controlField”(见 13.3.2.10),提供该字段以保持基于 PTP 版本 1 的硬件设计的向后兼容性。

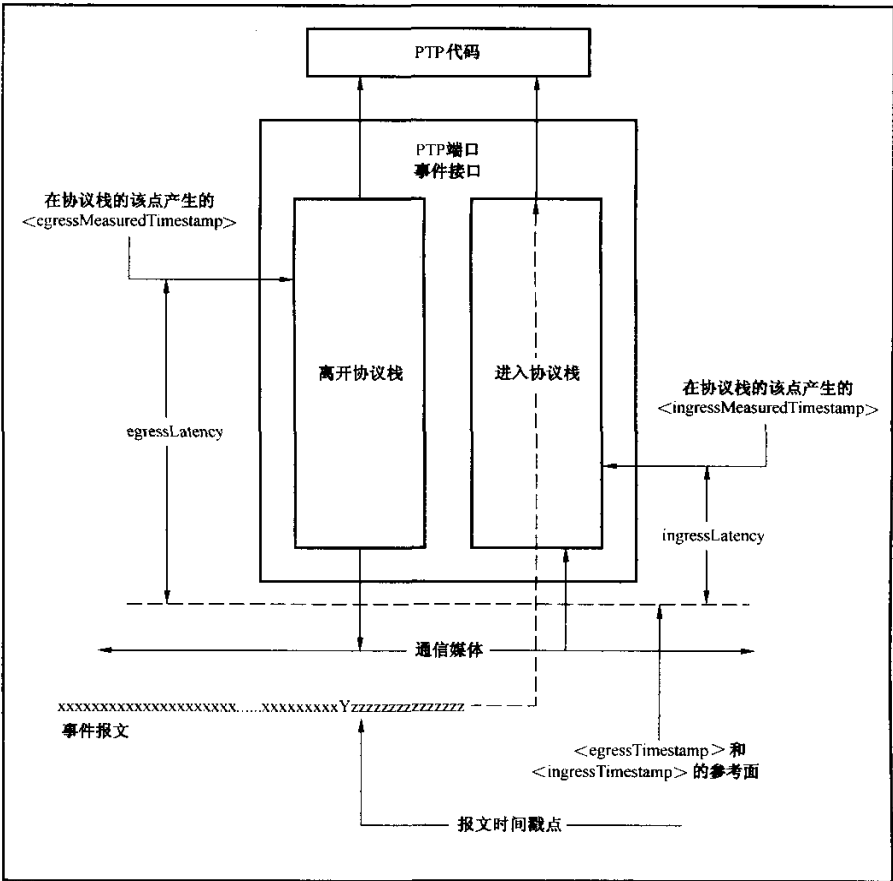


图 19 内部延时常数定义

7.3.7 报文的 sequenceId

除了后面提到的例外情况,在 PTP 普通时钟、边界时钟或透明时钟的每个端口应为发送到目的地址的每个报文类型维护一个单独的 sequenceId 池。多播地址被认为是一组目的地址,每个单播地址被认为是一个目的地址。

报文的 sequenceId 应比发送端口发送给同一报文目的地址的相同报文类型的前一个报文的 sequenceId 大一,这要服从 sequenceId 字段使用的 UInteger 16 数据类型的翻转约束。

如果两个 PTP 报文的 sequenceId 字段的值相同,且符合上面定义的 sequenceId 的翻转特性,具有相同的 messageType 和 domainNumber 字段值,并且是从 PTP 节点的另一发送协议地址发送的,则两个 PTP 报文是完全相同的。

sequenceId 的单独池的维护不用于下列报文类型:

- Pdelay_Resp
- Follow_Up
- Delay_Resp
- Pdelay_Resp_Follow_Up
- 响应另一个管理报文的管理报文

对于这些例外,sequenceId 值在 13.3.2.9 和 15.4.1.2 中规定。

7.3.8 基于标志的指示器

7.3.8.1 unicastFlag

标志 unicastFlag(见 13.3.2.6)为 TRUE 指示将报文作为单播报文传送。

7.3.8.2 alternateMasterFlag

标志 alternateMasterFlag(见 13.3.2.6)为 TRUE 指示报文从一个非 MASTER 状态的端口发送。

7.3.8.3 twoStepFlag

twoStepFlag(见 13.3.2.6)为 TRUE 指示报文从双步时钟发送。

7.4 PTP 通信媒体

7.4.1 网络传输协议

PTP 通信在使用传输协议的路径上发生,该传输协议由本标准的映射附录定义。传输路径上使用的传输协议的标识在表 3 中通过 networkProtocol 枚举指示。

表 3 networkProtocol 枚举

名 称	值(十六进制)	应用附录
保留	0000	—
UDP/IPv4	0001	附录 D
UDP/IPv6	0002	附录 E
IEEE 802.3	0003	附录 F
DeviceNet	0004	附录 G
ControlNet	0005	附录 H
PROFINET	0006	附录 I
保留,由 IM/ST 委员会精确网络化时钟工作组分配	0007~FFFF	—
保留,由 PTP 行规分配	F000~FFFD	—
未知协议	FFFE	—
保留	FFFF	—

注:其他传输机制的枚举分配可以通过向 IM/ST 委员会精确网络化时钟工作组申请获得。

7.4.2 通信路径不对称

在任何网络路径上包含双步时钟的系统中,主时钟到从时钟的报文和从时钟到主时钟的报文应通过同一网络路径。在任何网络路径上包含双步时钟的系统中,从请求方到响应方的报文和从响应方到请求方的报文应通过同一网络路径。在所有的系统中报文应通过同一路径,以使不对称最小化。

精确时间协议要求在主时钟和从时钟以及从时钟和它的主时钟之间测量某些报文的传输时间。对于对等延时机制,要求测量响应方和请求方,以及该请求方和该响应方之间的传输时间。典型地,这些时间并不相同,PTP 使用以下属性描述传输时间:

- $\langle \text{meanPathDelay} \rangle$;
- delayAsymmetry 。

这些属性的基础在图 20 中描述。

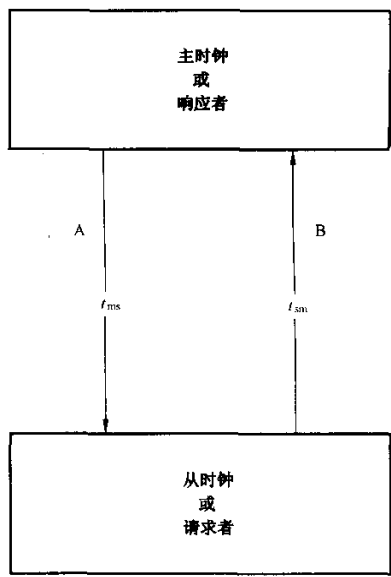


图 20 传输不对称

$\langle \text{meanPathDelay} \rangle$ 是 t_{ms} 和 t_{sm} 的均值,即 $\langle \text{meanPathDelay} \rangle = (t_{ms} + t_{sm})/2$ 。 delayAsymmetry 的值对于计算主时钟到从时钟或响应方到请求方方向的实际延时(即在 11 章中使用的 t_{ms})是必需的。在许多情况下, delayAsymmetry 的值低于同步应用的误差要求。

属性 delayAsymmetry 的定义如下:

$$t_{ms} = \langle \text{meanPathDelay} \rangle + \text{delayAsymmetry}$$

$$t_{sm} = \langle \text{meanPathDelay} \rangle - \text{delayAsymmetry}$$

换句话说,当主时钟到从时钟或响应方到请求方的传输时间大于从时钟到主时钟或请求方到响应方的传输时间时,定义 delayAsymmetry 为正。

对 delayAsymmetry 的测量不在本标准范围内。但是如可知,则应按照本条规定,模型化传输不对称用于校正时间计算,见 11.6。

7.5 PTP 端口

7.5.1 概述

PTP 系统中的节点通过被称为端口的实体与网络进行交互,见图 21。

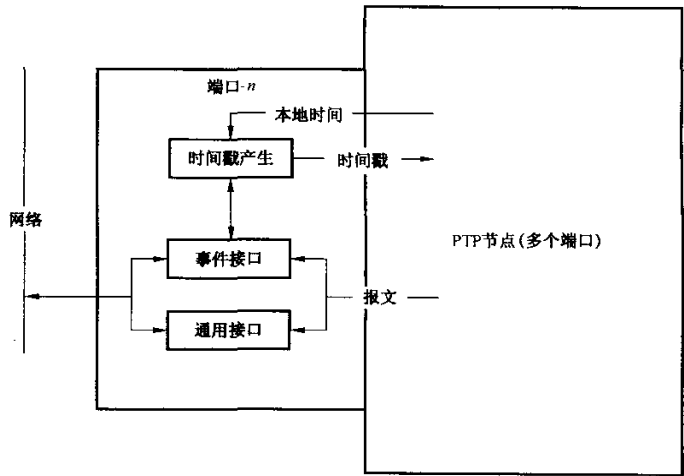


图 21 端口模型

PTP 普通时钟、边界时钟或透明时钟的每个端口被模型化为支持 2 个接口：事件接口和通用接口，如图 21 所示。该模型说明为事件报文（见 7.3.4.2）产生时间戳，但不要求为普通报文产生时间戳。

注：图 21 所示为端口模型，除非在本标准中另有说明，不排除其他实现，例如：有单一接口、为所有报文打上时间戳并之后选出事件报文的实现。

每个 PTP 端口实现该协议的单一版本，且使用单一的传输协议。多个 PTP 端口能通过单一物理端口连接到网络。

PTP 端口的属性在 7.5.2~7.5.5 中描述。

7.5.2 PortIdentity

7.5.2.1 概述

PTP 端口用类型 PortIdentity 的 portIdentity 属性标识，见 5.3.5。portIdentity 的值在 portDS 数据集的成员 portIdentity 中维护，见 8.2.5.2.1。portIdentity 由下述两个属性组成：

- portIdentity.clockIdentity；
- portIdentity.portNumber。

7.5.2.2 clockIdentity

7.5.2.2.1 概述

clockIdentity 是 8 个八位位组的数组。

clockIdentity 值应从 7.5.2.2.2 规定的 IEEE EUI-64 单个分配数⁴⁾中获得，或从 7.5.2.2.3 规定的数值集中获得。

7.5.2.2.2 IEEE EUI-64 clockIdentity 值

clockIdentity 的最高有效 3 个八位位组应为 OUI。OUI 的最高有效八位位组的两个最低有效位都应为 0。OUI 的最高有效八位位组的最低有效位用于区分本条规定的 clockIdentity 值和 7.5.2.2.3 中规定的 clockIdentity 值。

注 1：OUI 的最高有效八位位组的最低有效位和次低有效位的值分别指示该地址是组地址或单个地址，以及该地址由 IEEE 统一管理或本地管理。

ClockIdentity 应为 EUI-64 或从 EUI-48 构造的 EUI-64。

4) IEEE EUI-64、IEEE EUI-48、IEEE MAC-48 分配数可从 IEEE 注册管理委员会获得，见 <http://standards.ieee.org/regauth/>。这些分配数的指南也可从该网址获得。

对于将 IEEE EUI-64 用于 clockIdentity 值的设备：

- OUI 应由依据本条规定创建 clockIdentity 实例的组织拥有。
- 拥有 OUI 的组织应确保 EUI-64 的剩余 5 个八位位组在该组织分配的 clockIdentity 值范围内是唯一的。
- 将 IEEE EUI-64 的 8 个八位位组按序分配给 8 个八位位组数组 clockIdentity，其中将 IEEE EUI-64 的最高有效八位位组分配给八位位组数组 clockIdentity 中索引为 0 的成员。

示例：

X 公司的 OUI 是 ACDE48₁₆。如果 X 公司希望产生 EUI-64 clockIdentity，则合法值应为 ACDE48234567ABCD₁₆，X 公司确保 5 个八位位组数组 234567ABCD₁₆ 在 X 公司用作 clockIdentity 值的所有 EUI-64 分配数中是唯一的。来自 IEEE 注册管理委员会的 ClockIdentity 的字节表示和比特表示如下所示⁴⁾：

OUI			扩展标识符					字段
addr+0	addr+1	addr+2	addr+3	addr+4	addr+5	addr+6	addr+7	顺序
AC	DE	48	23	45	67	AB	CD	十六进制
10101100	11011110	01001000	00100011	01000101	01100111	10101011	11001101	比特
		组地址比特						
	最高有效字节					最低有效字节		
最高有效位						最低有效位		

实现者可以选择使用 EUI-48 产生 EUI-64 clockIdentity，在此情况下则：

- EUI-48 应为以太网 MAC 地址，以太网 MAC 地址由依据本条规定创建 clockIdentity 实例的组织拥有。拥有 MAC 地址的该组织应确保 MAC 地址用于产生仅仅一个 clockIdentity 实例，例如：通过要求 MAC 地址是嵌入到由 clockIdentity 标识的设备的 MAC 地址。
- 从 EUI-48 中构造 EUI-64 的映射规则应由 IEEE 注册管理委员会规定。
- 应将生成的 IEEE EUI-64 的 8 个八位位组按序分配给 8 个八位位组数组 clockIdentity，其中将 IEEE EUI-64 的最高有效八位位组分配给八位位组数组 clockIdentity 中索引为 0 的成员。

注 2：当使用 EUI-48 时，将 IEEE EUI-48 的前 3 个八位位组，即 OUI 部分，按序分配给 clockIdentity 的前 3 个八位位组，其中将 IEEE EUI-64 的最高有效八位位组，即 OUI 部分的最高有效八位位组，分配给八位位组数组 clockIdentity 中索引为 0 的成员，索引为 3 和 4 的八位位组的值分别为 FF 和 FE(十六进制)。IEEE EUI-48 的余下 3 个八位位组按序分配给 clockIdentity 的最后 3 个八位位组。

注 3：IEEE 注册管理委员会不赞成将 MAC-48 用于任何新设计中⁴⁾。

示例：

如果 X 公司希望在自己的设备 ACDE48234567₁₆ 内使用基于 MAC 地址的 EUI-48 作为 clockIdentity 的一部分，则产生的 clockIdentity 应为：ACDE48FFFE234567₁₆。来自 IEEE 注册管理委员会的 ClockIdentity 的字节表示和比特表示如下所示⁴⁾：

OUI			扩展标识符					字段
addr+0	addr+1	addr+2	addr+3	addr+4	addr+5	addr+6	addr+7	顺序
AC	DE	48	FF	FE	23	45	67	十六进制
10101100	11011110	01001000	11111111	11111110	00100011	01000101	01100111	比特
		组地址比特						
	最高有效字节					最低有效字节		
最高有效位						最低有效位		

除了在本标准定义的说明和功能行为外，不允许其他任何依赖 IEC 61588 的 clockIdentity 值的说

明和功能行为。

注 4：PTP 使用 clockIdentity 值作为唯一标识符，而不是作为网络地址。尽管对于使用于此目的的 IEEE 分配数的技术，网络地址可能从 clockIdentity 推断出来，但这种说明不在本标准范围内。

7.5.2.2.3 non-IEEE EUI-64 clockIdentity 值

对于不使用 IEEE EUI-64 或基于 EUI-48 的 clockIdentity 值的设备：

- 最高有效八位位组应被设置为 FF₁₆。该八位位组的最高有效位用于区分本条规定的 clockIdentity 值和在 7.5.2.2.2 中规定的值。
 - 紧接的 8 个 bit，即索引为 1 的八位位组，应分配来自表 4 的 non-EUI-64 addressTechnology 枚举。
 - 应对索引为 2~7 的八位位组分配值：使得到的 6 个八位位组数组的值在索引为 0 和 1 的八位位组定义的协议技术内是唯一的。使用此选项且唯一技术特定标识符小于 6 个八位位组的技术，应在该字段内左对齐设备的该唯一特定技术标识符，且将未使用的八位位组填 0。换句话说，索引为 2 的 clockIdentity 八位位组对于所有使用此选项的技术应有效。
 - 协议技术特定的唯一标识符的分配应由管理应用技术规范的标准化团体或工业团体负责。
- Non-IEEE 管理号的 16 比特值，及 non-EUI-64 addressTechnology 的 16 比特值的定义见表 4。

表 4 Non-EUI-64addressTechnology 枚举

八位位组[0]（十六进制）	八位位组[1]（十六进制）	通信协议
FF	00	保留
	01	DeviceNet
	02	ControlNet
	03	PROFINET
	04~FD	保留，由 IM/ST 委员会精确网络化时钟工作组分配
	FE	版本 1 设备
	FF	不在本标准范围内的封闭系统

注 1：PTP 使用 clockIdentity 作为唯一标识符，而不是网络地址。对于使用于此目的的 IEEE 分配数的技术，网络地址可能从 clockIdentity 可以推断出来，但这种说明不在本标准范围内。

注 2：在表 4 中列出的协议使用不准备映射到 EUI-64 框架的地址。IEC 61588 版本 1 的设备使用以太网 MAC 地址用于版本 1 的 UUID 字段。在边界时钟内版本 1 和版本 2 之间的转换可通过将版本 1 设备的特定规定加入到本表来简化，见 18.3.7。

示例：

XYZ 协会是标准化组织，负责管理当前或将来作为本标准附录的 XYZ 网络协议。为 XYZ 协会分配来自表 4 的值 FF05（十六进制）。假设其中一个成员，Y 公司从 XYZ 协会获得 6 个八位位组的唯一 clockIdentity：23456789ABCD₁₆。产生的 8 个八位位组 clockIdentity 应为：FF0523456789ABCD₁₆。字节和比特的表示如下所示⁴⁾：

OUI			扩展标识符					字段
addr+0	addr+1	addr+2	addr+3	addr+4	addr+5	addr+6	addr+7	顺序
FF	05	23	45	67	89	AB	CD	十六进制
11111111	00000101	00100011	01000101	01100111	10001001	10101011	11001101	比特
组地址比特								
最高有效字节								
最高有效位								
						最低有效字节		
						最低有效位		

7.5.2.2.4 保留的 clockIdentity 值

全为 1 的 clockIdentity 值应保留用于指定域内的所有时钟。

注：这符合 IEEE 注册管理委员会制定的规则⁴⁾。

7.5.2.3 portNumber

支持单一 PTP 端口的 PTP 节点的端口的 portNumber 值应为 1。支持 N 个 PTP 端口的 PTP 节点的 N 个端口的端口号应分别为 1, 2, ..., N。全 0 和全 1 的 portNumber 值被保留。全 1 的 portNumber, 即 FFFF_{16} , 用作管理报文(见 15.3.1)和信号报文(见 13.12.1)中的“所有端口”指示器。全 0 的 portNumber 用在 portIdentity 和 clockIdentity(见 7.5.2.4 和表 12)之间的数据集比较, 表示时钟的内部 portNumber。全 0 的 portNumber 也可用于表示 portNumber 值为 NULL, 例如它可表示未初始化或无效的 portNumber 值。

7.5.2.4 clockIdentity 值和 portIdentity 值的排序

两个 clockIdentity 值 X 和 Y 进行如下比较：

- 如果 X 内的每个八位位组等于 Y 内的对应八位位组, 则 $X=Y$;
- 否则, 考虑八位位组不同的最高有效位置, 并将此位置所在的八位位组作为无符号整数。如果属于 X 的八位位组小于属于 Y 的八位位组, 则 $X<Y$; 否则 $X>Y$ 。

具有属性 clockIdentity 和 portNumber 的类型 PortIdentity 的两个 portIdentity A 和 B 进行如下比较：

- 如果 A. clockIdentity 小于 B. clockIdentity, 则 $A<B$;
- 否则, 如果 A. clockIdentity 大于 B. clockIdentity, 则 $A>B$;
- 否则, 如果 A. portNumber 值小于 B. portNumber 值, 则 $A<B$;
- 否则, 如果 A. portNumber 值大于 B. portNumber 值, 则 $A>B$;
- 否则, $A=B$ 。

具有属性 clockIdentity 和 portNumber 的类型 PortIdentity 的 portIdentity A 和类型 ClockIdentity 的 clockIdentity B 进行如下比较：

- 如果 A. clockIdentity 小于 B. clockIdentity, 则 $A<B$;
- 否则, 如果 A. clockIdentity 大于 B. clockIdentity, 则 $A>B$;
- 否则, $B<A$ 。

7.5.3 状态

有 2 种 PTP 端口: 有状态和无状态。PTP 有状态的端口支持 9.2 中的状态机。有状态的 PTP 端口由与该端口关联的 PTP 状态机的当前状态描述。无状态的 PTP 端口不支持 PTP 状态机和不具有该状态属性。

注：有状态 PTP 端口和无状态 PTP 端口的定义不排除 9.2 以外的可能适用 PTP 端口的状态机。

7.5.4 路径延时测量机制

用于测量事件报文传输时间的两个机制如下：

- 延时请求-响应机制(见 11.3), 测量两个有状态 PTP 端口间的传输时间;
- 对等延时机制(见 11.4), 测量两个支持对等延时机制的端口间的传输时间。

注：对等延时机制可以被有状态以及无状态的 PTP 端口支持。

7.5.5 versionNumber

属性 versionNumber 指示端口实现 PTP 的版本。对于 PTP 当前版本, versionNumber 的值为 2。

注：VersionNumber 的改变指示 PTP 报文的格式或语义或协议的运行已经改变, 这使得早期版本将无法正确解析 PTP 报文。定义新的 TLV 或之前保留的 PTP 报文字段将不一定会导致 versionNumber 的改变, 仅会导致 PTP 当前版本日期的改变。符合 PTP 早期版本的节点不能利用这些 TLV 或以前保留字段的定义所产生的新功能。尽管不在当前版本的范围之内, 但将来版本可规定自主协商或其他技术, 能够兼容实现 PTP 早期版本的节点。在本版本中, 第 18 章定义了版本 1 和版本 2 间的转换。

7.6 PTP 设备特性

7.6.1 PTP 设备类型

有如下 5 种类型的 PTP 设备：

- 普通时钟；
- 边界时钟；
- 端到端透明时钟；
- 点到点透明时钟；
- 管理节点。

所有 PTP 设备由 clockIdentity 属性标识。

此外，普通时钟和边界时钟的特点由下述属性描述：

- Priority 1；
- Priority 2；
- clockClass；
- clockAccuracy；
- timeSource；
- offsetScaledLogVariance；
- numberPorts。

普通时钟和边界时钟可以使用下述属性保留父时钟性能的统计数据：

- observedParentOffsetScaledLogVariance；
- observedParentClockPhaseChangeRate。

7.6.2 PTP 设备属性

7.6.2.1 clockIdentity

时钟或管理节点的 clockIdentity 值应在 7.5.2.2 中规定。

7.6.2.2 Priority 1

属性 Priority 1 用于执行最佳主时钟算法，见 9.3.2，较低值优先。Priority 1 的初始化值在 PTP 行规中规定。Priority 1 的值应可配置为 0~255 范围内的任何值，除非受适用的 PTP 行规规定限值的限定。

注：最佳主时钟算法的执行从具有较低 Priority 1 值的时钟集中选择时钟，而不是从具有较高 Priority 1 值的时钟集中选择时钟。

7.6.2.3 Priority 2

属性 Priority 2 用于执行最佳主时钟算法，见 9.3.2。较低值优先。Priority 2 的初始化值在 PTP 行规中规定。Priority 2 的值应可配置为 0~255 范围内的任何值，除非受适用的 PTP 行规规定限值的限定。

注：如果最佳主时钟算法执行基于 Priority 1、clockClass、clockAccuracy 和 scaledOffsetLogVariance 的值选择时钟失败，Priority 2 属性允许在决定性判据前产生最高 256 个优先级用于计算。决定性判据是基于 clockIdentity。

7.6.2.4 clockClass

普通时钟或边界时钟的 clockClass 属性指示最高级时钟分发的时间或频率的可溯源性。clockClass 的解释和允许值应基于表 5 中的定义。

注 1：clockClass 数范围 68~122 和 133~170 被保留用于备选 PTP 行规的定义。例如：这些范围预期由定义了只分发频率的应用的 PTP 行规使用。值 128 允许这些 PTP 行规选择没有被最佳主时钟算法选为主时钟的时钟的端口处于 PASSIVE 还是 SLAVE 状态。

注 2：被设计为将其时钟与当前主时钟同步的节点，只要它没有被选作主时钟，该节点不应将其 clockClass 数设置为小于 128。例如：由用户使用管理报文 TIME 设置时间（见 15.5.3.2.1），并随后同步到当前主时钟的时钟，应设置 clockClass 数为 187，而不是 6 或 7。“HAND_SET”的描述见表 7。

注 3：clockClass 数范围 216~232 预期由 PTP 行规使用，PTP 行规要求基于某些应用特定优先级为时钟分配优先级。例如：在工业应用中，控制器可能优先于传感器。

除非在 PTP 行规中另有说明,应选择降级可选 B。

除非时钟被特别设计为在导致执行 POWERUP 事件的环境下维护时钟精度,该事件的执行不应为 clockClass 分配 6、7、13 或 14。

在一个给定域中,clockClass 值小于 128 的时钟至少应固有地与任何 clockClass 值大于其自身的时钟一样稳定(低方差)。

表 5 clockClass 规范

clockClass(十进制)	规 范
0	保留,以兼容将来版本
1~5	保留
6	应指定同步到主参考时间源的时钟,分发的时标为 PTP。clockClass 6 时钟不应是此域内另一个时钟的从时钟
7	应指定一个时钟,该时钟之前已指定为 clockClass 6 但已失去同步到主参考时间源的能力,且处于保持状态并符合保持规范。分发的时标为 PTP。ClockClass 7 时钟不应是此域内另一个时钟的从时钟
8	保留
9~10	保留,以兼容将来版本
11~12	保留
13	应指定同步到应用特定时间源的时钟,分发的时标为 ARB。clockClass 13 时钟不应是此域内另一个时钟的从时钟
14	应指定一个时钟,该时钟之前已指定为 clockClass 13 但已失去同步到应用特定时间源的能力,且处于保持状态并符合保持规范。分发的时标为 ARB。ClockClass 14 时钟不应是此域内另一个时钟的从时钟
15~51	保留
52	不符合保持规范的 clockClass 7 时钟的降级可选 A,ClockClass 52 时钟不应是此域内另一个时钟的从时钟
53~57	保留
58	不符合保持规范的 clockClass 14 时钟的降级可选 A,ClockClass 58 时钟不应是此域内另一个时钟的从时钟
59~67	保留
68~122	由备选 PTP 行规使用
123~127	保留
128~132	保留
133~170	由备选 PTP 行规使用
171~186	保留
187	不符合保持规范的 clockClass 7 时钟的降级可选 B,ClockClass 187 时钟可以是此域内另一个时钟的从时钟
188~192	保留
193	不符合保持规范的 clockClass 14 时钟的降级可选 B,ClockClass 193 时钟可以是此域内另一个时钟的从时钟
194~215	保留

表 5 (续)

clockClass(十进制)	规 范
216~232	由备选 PTP 行规使用
233~247	保留
248	缺省值,如果没有其他的定义适用,则使用该 clockClass
249~250	保留
251	保留,以兼容版本 1,见 18 章
252~254	保留
255	应仅为作为从时钟的时钟的 clockClass,见 9.2.2

如果时钟的固有特性改变,使得 clockClass 或 clockAccuracy 的指示不再适用,时钟应:

- 升级或降级 clockClass 或 clockAccuracy,以正确规定当前时钟特性;
- 或者被置于 FAULTY 状态。

7.6.2.5 clockAccuracy

clockAccuracy 为最佳主时钟(BMC)算法描述时钟特性。clockAccuracy 值应从表 6 中的枚举获得。该时钟以符合选定枚举值的精度来估计该属性的值。例如:23₁₆表示精度为±0.5 μs。该估计应基于 timeSource 属性(7.6.2.6)、从上一次同步到该时间源开始的经过时间和时钟的保持规范。如果确定该估计的信息不可用,则应使用“未知”枚举规范。

clockAccuracy 指示当时钟是最高级时钟或成为最高级时钟时期望的时钟精度。

在执行最佳主时钟算法时,对时钟精度的排序规定如下。当比较时钟精度时,如果时钟 A 的 clockAccuracy 值比时钟 B 的低,则认为时钟 A 好于时钟 B。

注:备选 PTP 行规保留 80~FD₁₆ 范围。期望此范围由定义了应用的 PTP 行规使用,该应用仅分发频率,以定义与频率分发相适用的精度规范。

表 6 clockAccuracy 枚举

值(十六进制)	规 范
00~1F	保留
20	时间精度为 25 ns
21	时间精度为 100 ns
22	时间精度为 250 ns
23	时间精度为 1 μs
24	时间精度为 2.5 μs
25	时间精度为 10 μs
26	时间精度为 25 μs
27	时间精度为 100 μs
28	时间精度为 250 μs
29	时间精度为 1 ms
2A	时间精度为 2.5 ms
2B	时间精度为 10 ms
2C	时间精度为 25 ms
2D	时间精度为 100 ms

表 6 (续)

值(十六进制)	规 范
2E	时间精度为 250 ms
2F	时间精度为 1 s
30	时间精度为 10 s
31	时间精度大于 10 s
32~7F	保留
80~FD	由备选 PTP 行规使用
FE	未知
FF	保留

7.6.2.6 timeSource

该纯信息属性指示最高级时钟使用的时间源。该值不用于选择最高级时钟,见表 7。这些值标识分类。例如, GPS 登录项应不仅包括美国国防部的 GPS 系统,也包括欧洲 Galileo 系统和其他目前及将来的基于人造卫星的时间系统。

表 7 timeSource 枚举

值(十六进制) ^a	timeSource	描 述
10	ATOMIC_CLOCK	基于原子共振产生频率,并已按国际标准对频率和时间(如果使用 PTP 时标)进行校准的任何设备,或直接连接到此类设备的设备。
20	GPS	同步到人造卫星系统的任何设备,人造卫星系统按照国际标准分发时间和频率。
30	TERRESTRIAL_RADIO	通过任何电波配时系统同步的任何设备,电波配时系统按照国际标准分发时间和频率。
40	PTP	同步到位于域外部的基于 PTP 的时间源的任何设备。
50	NTP	通过 NTP 或简单网络时间协议(SNTP)与服务器同步的任何设备,服务器按照国际标准分发时间和频率。
60	HAND_SET	基于观测的国际标准时间源,通过人机接口将其时间设定到声明时钟精度范围内的设备。
90	OTHER	没有被其他值覆盖的其他时间和/或频率源
A0	INTERNAL_OSCILLATOR	其频率不是基于原子共振也不按国际标准进行校准,且其时间是基于自由运行振荡器(其历元在任意或未知模式下确定)的任意设备。
F0~FE	由备选 PTP 行规使用	—
FF	保留	—

^a 保留所有在表 7 中未使用的值。

- 注 1: clockClass、clockAccuracy 和 timeSource 的值应是一致的。例如:被直接同步到 GPS 系统的类 6 原子时钟可能声明的精度为 25 ns,但对于没有直接同步到 GPS 系统,而是用户通过网络观察美国标准技术研究院(NIST)服务器,使用用户接口设置的同一原子时钟,可能声明为类 6 但精度为 10 s;
- 注 2: F0₁₆~FE₁₆ 范围被保留用于备选 PTP 行规。该范围预期由定义了仅分发频率的应用的 PTP 行规使用,以定义与频率分发相适应的源的特性。
- 注 3: 这些指定在电源故障重启后可能存在也可能不存在,但无论如何应反应节点的当前状态。例如:一个简单石英振荡器在启动后状态应为 INTERNAL_OSCILLATOR。如果稍后手动设置历元,则为 HAND_SET;而如果稍后同步到 GPS,则为 GPS;如果有备份电池的实时时钟,尽管在电源故障重启下 clockAccuracy 和 clockClass 可能降低,但该状态仍存在。

7.6.2.7 numberPorts

属性 numberPorts 应指示 PTP 设备的 PTP 端口数量。

7.6.3 PTP 方差

7.6.3.1 概述

两个方差估计(见 7.6.3.2)表示了 PTP 系统中普通时钟和边界时钟的特性:

- 每个此类时钟应维护一个对其固有精度的估计,offsetScaledLogVariance(见 7.6.3.5)。当时钟没有同步到使用协议的另一个时钟时,该估计是包含在时钟发布报文中的时间戳精度;
- 如果此类时钟 clock_A 被同步到另一个使用 PTP 协议的时钟。clock_A 可以维护其观察到的与之同步的时钟的精度估计,observedParentOffsetScaledLogVariance(见 7.6.4.3)。

7.6.3.2 方差算法

offsetScaledLogVariance 和 observedParentOffsetScaledLogVariance 源自 PTP 方差,PTP 方差基于下述 Allan 偏差理论:

Allan 偏差 $\sigma_y(\tau)$ 按下式估计(见 ITU-T Recommendation G.810 [B19]):

$$\sigma_y(\tau) = \left[\frac{1}{2(N-2)\tau^2} \times \sum_{k=1}^{N-2} (x_{k+2} - 2x_{k+1} + x_k)^2 \right]^{\frac{1}{2}}$$

其中, x_k, x_{k+1} 和 x_{k+2} 是在时间 $t_k, t_k + \tau$ 和 $t_k + 2\tau$ 时刻完成的时间差测量, τ 是测量之间的采样周期, N 是数据采样个数。“差”意味着不变的系统影响已从数据中移除。

按照规定,Allan 偏差是振荡器频率变化的二阶统计量,振荡器作为时间基的基础。

PTP 方差定义如下:

$$\sigma_{\text{PTP}}^2 = \tau^2 \times \frac{1}{3} \sigma_y^2$$

PTP 方差的无偏估计应按下式计算:

$$\sigma_{\text{PTP}}^2 = \frac{1}{3} \left[\frac{1}{2(N-2)} \times \sum_{k=1}^{N-2} (x_{k+2} - 2x_{k+1} + x_k)^2 \right]$$

其中, x_k, x_{k+1} 和 x_{k+2} 是在时间 $t_k, t_k + \tau$ 和 $t_k + 2\tau$ 时刻完成的在测量时钟和本地参考时钟之间的时间差测量, τ 是采样周期, N 是数据采样个数。对于 PTP 方差,采样周期 τ 应在适用的 PTP 行规中定义的值。8.2.1.3.1.3 和 8.2.3.4 规定了使用本条规定计算的方差。如果在执行期间计算这些方差,只有是 syncInterval 倍数时数据才是可用的,见 7.7.2.3。在此情况下, τ 应该是 syncInterval 的倍数。

实现可以计算 PTP 方差的保守估计值,而不是计算此处给定的准确值。注意:在有限计算资源或存储资源的实现中这可能是必要的。

注:Allan 偏差对采样周期的依赖提供了关于基本噪声过程类型的信息。Allan 偏差对于时间或频率的常数偏移不敏感,即使这些偏移可能在本标准的一些应用中是重要的。此外,例如当噪声频谱包含“亮线”(电力线引入的 50 Hz 变化),Allan 偏差不提供有用的诊断。最后,Allan 偏差作为观察集的平均值被计算,且当数据是统计不变时 Allan 偏差最有效。偏差不提供对偶然干扰的频率或幅值的良好测量,即使这些事件可能在本标准的一些应用中是重要的。

7.6.3.3 方差表示

PTP 方差应表示如下:

- a) 在 7.6.3.2 中规定的 σ_{PTP}^2 的估计以秒平方(s^2)为单位来计算。
- b) 计算此估计以 2 为底的对数。该对数的计算不必比该方差估计的精度更精确。
- c) 该对数乘以 2^8 产生一个标定值。
- d) 根据本条的滞后规范修改该标定值以产生报告值。
- e) 该报告值被表示为二进制补码形式的 Integer 16。将值 8000_{16} 加到以此形式表示的报告值,并忽略任何溢出。其结果(即该偏移标定的报告值)为 UInteger 16。

f) 该偏移标定的报告值(表示为 UInteger 16)应是方差(见 7.6.3.1)的对数。

注 1: 例如,假设 PTP 方差值是 $1.414 \times 2^{-73} = 1.497 \times 10^{-22} \text{ s}^2$, 则 $\log_2(1.414 \times 2^{-73}) = -73 + 0.5 = -72.5$ 。如果将它表示为 Integer16, 则它应截短为 -72 。为保持一定的精度, 通过该值乘以 2^8 来标定以产生 scaledLogVariance 为 -18560 (即 $B780_{16}$), 它保持更高的 8 比特精度。对此值加上 $8\,000_{16}$ 以产生偏移标定报告值 3780_{16} 。

注 2: 可以表示的最小方差是 2^{-128} 或约等于 $3 \times 10^{-39} \text{ s}^2$, 它产生 offsetScaledLogVariance 为 0000_{16} 。可以表示的最大方差是约等于 $2^{+127.996\,09}$, 它产生 offsetScaledLogVariance 为 $FFFF_{16}$ 。

注 3: 本表示确保 7.6.3.4 中的方差算法的计算顺序在所有实现中产生同样结果。使用浮点表示不能保证这样的结果。

offsetScaledLogVariance 属性的最大可能正数($FFFF_{16}$)指示方差太大以致不能表示或还没有计算方差。

方差值用于选择最佳主时钟(见 9.3.2), 因此在运行期间计算方差值的实现应包含方差估计的滞后, 以排除选择主时钟过程中的波动。报告估计值以 2 为底对数的 2^8 标定值使用的滞后的量值为 PTP_SCALED_LOG_VARIANCE_HYSTERESIS, 如图 22 所示。该滞后用于估计值对数的标定值, 该估计值用于产生实际报告的估计值对数的偏移标定值, 并用于最佳主时钟算法的计算(见 9.3)。需要维护足够的本地状态, 以允许正确实现实际方差估计中增加或减少趋势的滞后特性, PTP_SCALED_LOG_VARIANCE_HYSTERESIS 的值是 2^7 。

注: 2^7 对应 \log_2 (实际估计值)为 $1/2$ 的改变。因为滞后要求, \log_2 (方差估计)低于 $1/2$ 的波动不被报告。

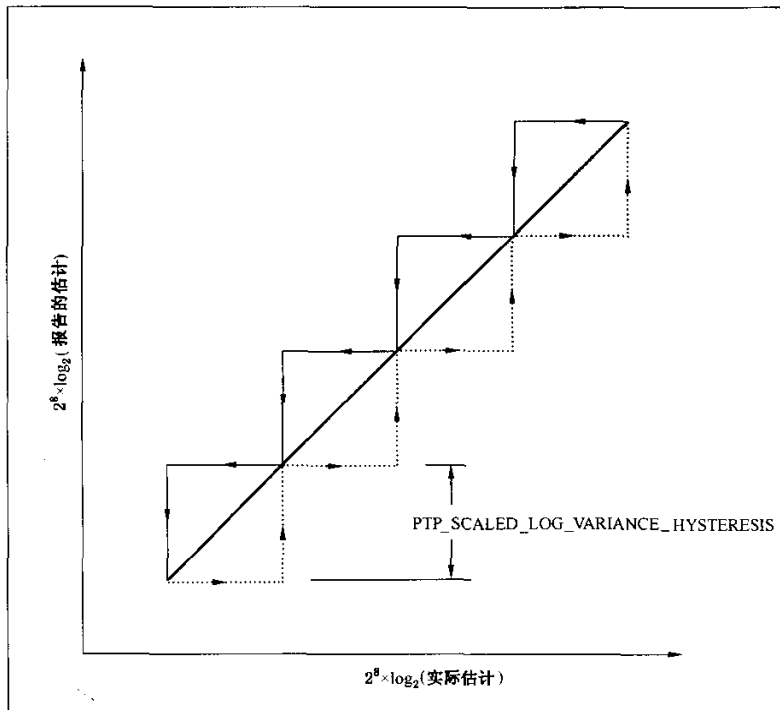


图 22 方差对数标定值滞后

7.6.3.4 方差计算顺序

方差应按对数表示、标定值、偏移量的顺序计算。

7.6.3.5 计算 defaultDS, offsetScaledLogVariance

defaultDS, offsetScaledLogVariance 值的计算应基于由理想时钟测量的本地时钟的特性。defaultDS, offsetScaledLogVariance 值如下:

- 由制造商确定的静态常数；
- 基于本地时钟部件测量行为或模型化行为和环境计算得到。

defaultDS.offsetScaledLogVariance 值的计算和表示见 7.6.3.2 和 7.6.3.3。

当本地时钟没有与使用协议的另一个时钟同步时,其 defaultDS.offsetScaledLogVariance 值应为相对于线性时标的本地时钟变化的估计。当没有与另一个时钟同步时,其参考时钟可能是原子时钟、GPS 接收器、稳定本地振荡器、通过 NTP 被同步的时钟组,等等。这些源可能影响方差估计。

defaultDS.offsetScaledLogVariance 值应为适用于一个测量集的方差,测量包括来自下述方面的误差:

- 同步到它的参考时钟;
- 时钟相位变化率的变化和本地振荡器的噪声特性;
- 采样量化误差、进入延时和离开延时的波动和本地时钟的其他波动。

7.6.4 父时钟统计数据

7.6.4.1 概述

在从时钟状态的时钟可维护观测到的其父时钟性能的统计数据。这两个可选统计数据如下:

- parentDS.observedParentOffsetScaledLogVariance
- parentDS.observedParentClockPhaseChangeRate

7.6.4.2 parentDS, parentStats

parentDS.parentStats 的值应指示 parentDS.observedParentOffsetScaledLogVariance 和 parentDS.observedParentClockPhaseChangeRate 的值是否已测量和有效。值为 TRUE 表示有效数据。

7.6.4.3 parentDS.observedParentOffsetScaledLogVariance

本地时钟的 parentDS.observedParentOffsetScaledLogVariance 应为本地时钟测量的父时钟相位的方差。该测量是可选的。parentDS.observedParentOffsetScaledLogVariance 值的计算和表示见 7.6.3.2 和 7.6.3.3。

7.6.4.4 parentDS.observedParentClockPhaseChangeRate

parentDS.observedParentClockPhaseChangeRate 应是从时钟测量的父时钟相位变化率的估计,见 3.1.25。报告值应为小数频率偏移乘以 2^{+40} 。如果估计值超过了数据类型的范围,应将该值设为适当的最大或最小允许值。正符号表示父时钟比从时钟快。该测量是可选的。

注 1: 能表示的最小相位变化率是 2^{-40} 或近似 1 ps/s。

注 2: 该值取决于使用的测量时间间隔。当该值对应用域十分重要时,应该在适用的 PTP 行规中规定时间间隔。

7.6.4.5 defaultDS.numberPorts

属性 defaultDS.numberPorts 应为支持 PTP 的设备中端口的数量。

7.7 PTP 时间特性

7.7.1 概述

7.7 规定了协议的时间和超时属性。

7.7.2 报文传输间隔

7.7.2.1 通用间隔规范

对于每个报文类型 Announce、Sync、Delay_Req 和 Pdelay_Req,连续报文之间的平均时间间隔应表示为以 2 为底的时间间隔的对数,该时间间隔以秒为单位在发送报文设备的本地时钟上测量。这些以对数表示的属性值应选择 -128~127 范围内的整数,并且服从适用的 PTP 行规规定的更多限制。这些间隔通过 PTP 报文的 logMessageInterval 字段传送。logMessageInterval 的解释取决于报文类型,见 13.3.2.11。除了 Delay_Req 报文(见 9.5.11.2),节点应以一定的间隔发送报文,该间隔在属性设定值的 $\pm 30\%$ 范围内,置信度为 90%。

7.7.2.2 Announce 报文发送间隔

portDS.logAnnounceInterval 应规定连续 Announce 报文间的平均时间间隔,即 announceInterval。

portDS.logAnnounceInterval 应该在整个域内是统一的,与之不符的域的行为不在本标准范围内。

注 1: portDS.logAnnounceInterval 值是对域内确定主从层次结构的网络内的变化作出期望的响应速度,同发送报文造成的通信和计算工作量之间的折衷值。

注 2: 在使用不同通信技术的区域,例如:有线和无线技术,采用不同 portDS.logAnnounceInterval 是理想的。portDS.logAnnounceInterval 随区域改变的系統仍应正常工作。但是相比系統使用统一间隔值的情况,当等待较慢区域使用最佳主时钟算法选择主时钟时,短间隔的区域可能进行更多的重新配置。

7.7.2.3 Sync(多播)报文发送间隔

portDS.logSyncInterval 应规定作为多播报文发送的连续 Sync 报文间的平均时间间隔,即 syncInterval。

注 1: 在使用不同通信技术的区域,例如:有线和无线技术,采用不同 portDS.logSyncInterval 是理想的。

注 2: portDS.logSyncInterval 值是在本地时钟稳定性和精度,同发送这些报文造成的通信和计算工作量之间的折衷值。

7.7.2.4 Delay_Req 报文发送间隔

portDS.logMinDelayReqInterval 应规定在连续 Delay_Req 报文间所允许的最小平均时间间隔,即 minDelayRequestInterval。Delay_Req 报文由从时钟发送给主时钟的一个特定端口,见 9.5.11.2。

主时钟根据其处理 Delay_Req 报文流量的能力确定并通报该值。该值应为一整数,最小值为 portDS.logSyncInterval,即与 Sync 报文的速率相同,最大值为 logSyncInterval+5,即每 32 个 Sync 报文有 1 个 Delay_Req 报文。

注: portDS.logDelayReqInterval 值是对路径延时改变的响应速度,同发送报文造成的通信和计算工作量之间的折衷值。

7.7.2.5 Pdelay_Req 报文发送间隔

portDS.logMinPdelayReqInterval 应规定在链路上发送的连续 Pdelay_Req 报文间允许的最小平均时间间隔,即 minPdelayReqInterval。

注: portDS.logMinPdelayReqInterval 值是在链路延时和启动时间波动,同发送报文造成的通信和计算工作量之间的折衷值。

7.7.3 PTP 超时

7.7.3.1 portDS.announceReceiptTimeout

portDS.announceReceiptTimeout 值应规定在 ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES (见 9.2.6.11)事件发生前无 Announce 报文接收情况下必须经过的 announceInterval 个数。值范围应为 2~255,且要符合 PTP 行规中的其他限定。最小值宜为 3。

$(\text{portDS.announceReceiptTimeout}) \times 2^{\text{portDS.logAnnounceInterval}}$ 的值应该在整个域内统一,与之不符的域的行为不在本标准范围内。

注: portDS.announceReceiptTimeout 值是对失效设备或失效网络的快速响应,同由于偶然性的 Announce 报文丢失所造成的用于确定主-从层次结构的最佳主时钟算法过度波动之间的折衷值。

8 PTP 数据集

8.1 数据集成员通用规范

8.1.1 数据集概述

8.1.1.1 数据集规范概述

每个数据集成员规范包括:

- 成员的正式名称;
- 成员关联语义的参考或定义;
- 特性的初始化及组态,见 8.1.3。

8.1.1.2 普通时钟和边界时钟

对于每个普通时钟和边界时钟,应本地维护以下“时钟数据集”,作为协议判定和为报文字段提供值

的基础:

- defaultDS(见 8.2.1);
- currentDS(见 8.2.2);
- parentDS(见 8.2.3);
- timePropertiesDS(见 8.2.4);
- portDS(普通时钟或边界时钟每个端口的一个数据集,见 8.2.5)。

8.1.1.3 透明时钟

对于每个透明时钟,应本地维护以下“透明时钟数据集”,作为协议判定和为报文字段提供值的基础:

- transparentClockDefaultDS 数据集(见 8.3.2);
- transparentClockPortDS 数据集(见 8.3.3)。

8.1.2 初始化分类

8.1.2.1 概述

一个数据集的每个成员都分类为静态的、动态的或可组态的。

8.1.2.1.1 静态成员

静态成员的值是时钟或协议固有的物理或操作特性。

8.1.2.1.2 动态成员

动态成员的值不可直接被用户修改,但可进行改变:

- 作为协议运行的结果。例如, portDS. portState 可由于协议事件而改变。一个管理报文可引发一个事件而导致 portDS. portState 的改变,但只能通过状态机的动作间接修改。在一些设计中, portDS. portState 能通过管理报文进行设置。
- 由于时钟内部特性的变化而改变。例如, defaultDS. offsetScaledLogVariance 可由于温度对本地振荡器的影响而改变。
- 由于同 PTP 外部时间系统的交互而改变。例如,一个能够与 GPS 系统同步的时钟,当第一次锁定 GPS 时,可改变其 defaultDS. clockQuality、时间或 timePropertiesDS 数据集成员的值。

8.1.2.1.3 可组态成员

可组态成员的值仅能通过管理报文或实现特定的组态方式来修改。

除非在本标准中另有说明,否则当可组态成员值更新时,其更新值应立即起作用。

除非在本标准中另有说明,可组态成员值的更新应限制在第 7 章或在适用 PTP 行规中规定的最大限制范围值的范围内。

注:例如, defaultDS. domainNumber 的值可通过管理报文修改,但不受协议或时钟内部变化的影响。

8.1.3 时钟数据集初始化特性

8.1.3.1 通用初始化规范

数据集成员的初始化特性应根据其分类(静态、动态或可组态)来决定。

对于普通时钟或边界时钟,数据集成员的初始化应在离开 INITIALIZATION 状态之前进行;对于透明时钟,数据集成员的初始化应在正常操作开始之前进行,见 8.1.3.2~8.1.3.4。

8.1.3.2 静态数据集成员的初始化

静态成员应被初始化为实现特定的值以满足成员规范。

8.1.3.3 动态数据集成员的初始化

动态成员应被初始化为以下值中首先适用的值:

- a) 在规范中对数据集成员要求的值;
- b) 表示时钟或协议在初始化时的特性值;

注:例如, defaultDS. clockQuality 的值可取决于初始化时刻时钟是否与 GPS 同步。

c) 如实现,在非易失读写存储器中的值;

注:例如,假定网络不太可能被频繁重新配置,currentDS.meanPathDelay 的值可被保存在非易失存储器中。

d) 实现特定的值。

如果动态成员的值被保存在非易失读写内存中,制造商应使用 a)、b)或 d)的适用值预加载该内存。

a)、b)或 d)的值被称为动态数据集的初始化值。

8.1.3.4 可组态数据集成员的初始化

可组态成员应被初始化为以下值中首先适用的值:

a) 在成员规范中说明作为初始化值的值而非根据 PTP 行规规范的值;

b) 上一个由管理报文或实现特定方式组态且保存在非易失读写存储器中(如实现)的值;

c) 设备实现的 PTP 行规中规定的成员的缺省初始化值。

如果组态值被保存在非易失读写内存中,制造商应使用 a)或 c)的适用初始化值预加载该内存。

a)或 c)的值被称为可组态数据集的初始化值。

8.1.3.5 数据集成员的非易失读写存储操作

一个或多个动态或可组态数据集成员的当前值可保存在非易失读写存储器中。

注:例如,本方式允许系统停电后更快速的组态。

当接收到管理报文 RESET_NON_VOLATILE_STORAGE 或实现特定的等效报文,非易失读写内存的内容应被复位为适用的动态或可组态数据集的初始化值(见 8.1.3.3 和 8.1.3.4)。

适用的动态和可组态数据集成员的当前值应被复制到非易失读写内存中:

——当接收到管理报文 SAVE_IN_NON_VOLATILE_STORAGE 或实现特定的等效报文;

——在实现特定时间。

8.2 普通时钟和边界时钟的数据集

8.2.1 defaultDS 数据集成员规范

8.2.1.1 概述

该数据集的成员如下:

——defaultDS.twoStepFlag;

——defaultDS.clockIdentity;

——defaultDS.numberPorts;

——defaultDS.clockQuality;

——defaultDS.Priority 1;

——defaultDS.Priority 2;

——defaultDS.domainNumber;

——defaultDS.slaveOnly。

8.2.1.2 defaultDS 数据集的静态成员

8.2.1.2.1 defaultDS.twoStepFlag

如果时钟是一个双步时钟,则 defaultDS.twoStepFlag 的值应为 TRUE;否则,值应为 FALSE。

8.2.1.2.2 defaultDS.clockIdentity

defaultDS.clockIdentity 的值应为本地时钟的 clockIdentity(见 7.6.2.1)。

8.2.1.2.3 defaultDS.numberPorts

defaultDS.numberPorts 的值应为设备上 PTP 端口的个数。对于普通时钟,该值应为 1。

8.2.1.3 defaultDS 数据集的动态成员

8.2.1.3.1 defaultDS.clockQuality

8.2.1.3.1.1 defaultDS.clockQuality.clockClass

defaultDS.clockQuality.clockClass 的值应遵循 7.6.2.4 的 clockClass 规范。

defaultDS.clockQuality.clockClass 的初始化值应按如下选择：

- a) 该值取决于 defaultDS 数据集成员 defaultDS.slaveOnly 的初始化值, defaultDS.slaveOnly 应在 defaultDS.clockQuality.clockClass 成员初始化之前被初始化；
- b) 如果 defaultDS.slaveOnly 是 TRUE, 则该初始化值应为 255, 见 7.6.2.4；
- c) 如果 defaultDS.slaveOnly 是 FALSE, 并且 PTP 行规规定 clockClass 为 52、58、187、193 或在 68~122、133~170 或 216~232 范围内, 则应使用 PTP 行规规定的 clockClass 值进行初始化；
- d) 如果 defaultDS.slaveOnly 是 FALSE, 并且设备按照 clockClass 6 或 13 进行设计, 若这些数字表示时钟离开 INITIALIZING 状态的 clockClass, 则 clockClass 初始化值应分别设为 6 或 13；若 clockClass 6 或 13 不表示离开 INITIALIZING 状态的时钟, 则 clockClass 初始化值应为：
 - 1) 对于被设计为 class 6 的时钟, 根据行规规定, 应为 52、187 或 248；
 - 2) 对于被设计为 class 13 的时钟, 根据行规规定, 应为 58、193 或 248。
- e) 否则, 值应为 248。

注：设计为 clockClass 6 或 13 的时钟, 在初始化 defaultDS.clockClass 成员之前, 可包括实现特定的措施以确保其满足 7.6.2.4 对这些 clockClass 的规范。例如, 在初始化 defaultDS.clockClass 成员之前, 可将时钟与 GPS 系统同步作为在 INITIALIZING 状态的部分动作。如果这样的时钟在离开 INITIALIZING 状态之前不能满足 7.6.2.4 的规范, 则使用时钟的降级或缺省可选值之一。

8.2.1.3.1.2 defaultDS.clockQuality.ClockAccuracy

defaultDS.clockQuality.ClockAccuracy 的值是 defaultDS.clockQuality 成员的 clockAccuracy 成员, 见 5.3.7。

defaultDS.clockQuality.ClockAccuracy 的初始化按如下选择：

- a) 该值取决于 defaultDS.clockQuality.clockClass 的初始化值(见 8.2.1.3.1.1), defaultDS.clockQuality.clockClass 应在 clockAccuracy 初始化之前被初始化；
- b) clockAccuracy 初始化值应表示初始化时刻时钟的 clockAccuracy, 见 7.6.2.5。

8.2.1.3.1.3 defaultDS.clockQuality.offsetScaledLogVariance

defaultDS.clockQuality.offsetScaledLogVariance 的值是 defaultDS.clockQuality 成员的 offsetScaledLogVariance 成员, 见 5.3.7。

defaultDS.clockQuality.offsetScaledLogVariance 的初始化值应反映初始化时刻本地时钟的固有特性, 见 7.6.3.5。

8.2.1.4 defaultDS 数据集的可组态成员

8.2.1.4.1 defaultDS.Priority 1

defaultDS.Priority 1 的值是本地时钟的 Priority 1 属性(见 7.6.2.2)。

8.2.1.4.2 defaultDS.Priority 2

defaultDS.Priority 2 的值是本地时钟的 Priority 2 属性(见 7.6.2.3)。

8.2.1.4.3 defaultDS.domainNumber

defaultDS.domainNumber 是本地时钟的域的属性(见 7.1)。

8.2.1.4.4 defaultDS.slaveOnly

如果时钟是 slave-only 时钟, 则 defaultDS.slaveOnly 的值应为 TRUE, 见 9.2.2。如果时钟是 non-slave-only 时钟, 则 defaultDS.slaveOnly 的值应为 FALSE, 见 9.2.3。

8.2.2 currentDS 数据集成员规范

8.2.2.1 概述

该数据集的成员如下：

- currentDS.stepsRemoved;
- currentDS.offsetFromMaster;
- currentDS.meanPathDelay。

currentDS 数据集的所有成员都是动态的。

8.2.2.2 currentDS.stepsRemoved

currentDS.stepsRemoved 的值为本地时钟和最高级时钟之间穿过的通信路径的个数。

初始化值应为 0。

注：例如，处于与最高级时钟相同 PTP 通信路径的从时钟，其 currentDS.stepsRemoved 的值为 1，表示仅一个路径被穿过。

8.2.2.3 currentDS.offsetFromMaster

currentDS.offsetFromMaster 的值是主时钟与从时钟之间时间差当前值的实现特定表示，该值由从时钟计算；即 $\langle \text{offsetFromMaster} \rangle = \langle \text{从时钟的时间} \rangle - \langle \text{主时钟的时间} \rangle$ （见 11.2）。数据类型应为 TimeInterval。初始化值应为：

- 如实现，为非易失读写存储器中的值；
- 或实现特定。

8.2.2.4 currentDS.meanPathDelay

currentDS.meanPathDelay 的值是主时钟与从时钟之间平均传输时间当前值的实现特定表示，该值由从时钟计算，即 $\langle \text{meanPathDelay} \rangle$ ；见 11.3 和 11.4。数据类型应为 TimeInterval。初始化值应为：

- 如实现，为非易失读写存储器中的值；
- 或实现特定。

8.2.2.5 parentDS 数据集成员规范

8.2.2.6 概述

该数据集的成员如下：

- parentDS.parentPortIdentity;
- parentDS.parentStats;
- parentDS.observedParentOffsetScaledLogVariance;
- parentDS.observedParentClockPhaseChangeRate;
- parentDS.grandmasterIdentity;
- parentDS.grandmasterClockQuality;
- parentDS.grandmasterPriority 1;
- parentDS.grandmasterPriority 2。

parentDS 数据集的值应在 defaultDS 数据集的值的初始化之后被初始化。

parentDS 数据集的所有成员都是动态的。

8.2.2.7 parentDS.parentPortIdentity

parentDS.parentPortIdentity 的值是主时钟端口的 portIdentity，该主时钟发出用于同步本时钟的 Sync 报文。

初始化值应为：

- parentDS.parentPortIdentity.clockIdentity 成员是 defaultDS.clockIdentity 字段的值；
- parentDS.portNumber 成员为 0。

8.2.2.8 parentDS.parentStats

如果满足所有下列条件，则 parentDS.parentStats 的值应为 TRUE：

- 时钟有一个端口处于 SLAVE 状态；
- 时钟已统计计算了 parentDS.observedParentOffsetScaledLogVariance 和 parentDS.observe-

edParentClockPhaseChangeRate 成员的有效估计。

否则,值应为 FALSE。

初始化值应为 FALSE。

8.2.2.9 parentDS. observedParentOffsetScaledLogVariance

parentDS. observedParentOffsetScaledLogVariance 的值应为从时钟观察到的父时钟 PTP 方差的估计,计算和表示见 7.6.3.5。该值的计算是可选的。若不计算,则 parentDS. parentStats 的值应为 FALSE。

初始化值应为 FFFF_{16} ,见 7.6.3.3。

8.2.2.10 parentDS. observedParentClockPhaseChangeRate

parentDS. observedParentClockPhaseChangeRate 的值应为从时钟观察到的父时钟相位变化率的估计,见 7.6.4.4。如果估计超出该数据类型的范围(见 15.5.3.5.1.4),该值应被适当地设为 7FFFFFF_{16} 或 $80\,000\,000_{16}$ 。正号表示父时钟的相位变化率大于从时钟的。该值的计算是可选的。若不计算,则 parentDS. parentStats 的值应为 FALSE。

不论本地时钟是否实现该计算,初始化值应为 7FFFFFF_{16} 。

值为 7FFFFFF_{16} 时表示该值超出了数据类型范围,或者值未被计算。

注:该值取决于所用的测量时间间隔。当该值对于应用领域至关重要时,时间间隔应在适用的 PTP 行规中规定。

8.2.2.11 parentDS. grandmasterIdentity

parentDS. grandmasterIdentity 的值是最高级时钟的 clockIdentity 属性(见 7.6.2.1)。

初始化值应为 defaultDS. clockIdentity 成员的值。

8.2.2.12 parentDS. grandmasterClockQuality

parentDS. grandmasterClockQuality 的值是最高级时钟的 clockQuality 属性(见 7.6.2.4、7.6.2.5 和 7.6.3)。

初始化值应为 defaultDS. clockQuality 成员的值。

8.2.2.13 parentDS. grandmasterPriority 1

parentDS. grandmasterPriority 1 的值是最高级时钟的 Priority 1 属性(见 7.6.2.2)。

初始化值应为 defaultDS. Priority 1 成员的值。

8.2.2.14 parentDS. grandmasterPriority 2

parentDS. grandmasterPriority 2 的值是最高级时钟的 Priority 2 属性(见 7.6.2.3)。

初始化值应为 defaultDS. Priority 2 成员的值。

8.2.3 timePropertiesDS 数据成员规范

8.2.3.1 概述

该数据集的成员如下:

- timePropertiesDS. currentUtcOffset;
- timePropertiesDS. currentUtcOffsetValid;
- timePropertiesDS. leap59;
- timePropertiesDS. leap61;
- timePropertiesDS. timeTraceable;
- timePropertiesDS. frequencyTraceable;
- timePropertiesDS. ptpTimescale;
- timePropertiesDS. timeSource。

timePropertiesDS 数据集的所有成员都是动态的。

timePropertiesDS. ptpTimescale 成员应在该数据集其他成员的初始化之前被初始化。

8.2.3.2 timePropertiesDS. currentUtcOffset

PTP 历元的 PTP 系统中, timePropertiesDS. currentUtcOffset 的值是 TAI 和 UTC 之间的偏移; 否则该值没有意义。值的单位为 s。

初始化值应按如下选择:

- a) 如果 timePropertiesDS. ptpTimescale(见 8.2.4.8)为 TRUE,若该值在初始化时刻是已知的,则为从主参考获得的值;否则
- b) 该值应为节点被设计时闰秒(见 7.2.3)的当前个数。

注: 设计为 clockClass 6 的时钟,在初始化 timePropertiesDS. currentUtcOffset 成员之前,可包括实现特定的措施以确保其满足 7.6.2.4 中 clockClass 6 的规范,并因此可访问 UTC 偏移值。例如,在初始化 currentUtcOffset 成员之前,时钟可将与 GPS 系统同步作为在 INITIALIZING 状态的部分动作。

8.2.3.3 timePropertiesDS. currentUtcOffsetValid

如果 timePropertiesDS. currentUtcOffset 已知是正确的,则 timePropertiesDS. currentUtcOffsetValid 的值是 TRUE。

如果 timePropertiesDS. currentUtcOffset 的值已知是正确的,则 timePropertiesDS. currentUtcOffsetValid 的初始化值是 TRUE;否则,是 FALSE。

8.2.3.4 timePropertiesDS. leap 59

PTP 历元的 PTP 系统中, timePropertiesDS. leap 59 为 TRUE 时表示当前 UTC 日的最后 1 min 包含 59 s。

如果不是 PTP 历元,则值应设为 FALSE 见 7.6.3.3。

初始化值应按如下选择:

- a) 如果 timePropertiesDS. ptpTimescale(见 8.2.4.8)的值为 TRUE,若在初始化时刻是已知的,则该值为从主参考获得的值;否则
- b) 该值应为 FALSE。

8.2.3.5 timePropertiesDS. leap 61

PTP 历元的 PTP 系统中, timePropertiesDS. leap 61 为 TRUE 时表示当前 UTC 日的最后 1 min 包含 61 s。

如果不是 PTP 历元,则值应设为 FALSE。

初始化值应按如下选择:

- a) 如果 timePropertiesDS. ptpTimescale(见 8.2.4.8)的值为 TRUE,若在初始化时刻是已知的,则该值为从主参考获得的值;否则
- b) 该值应为 FALSE。

8.2.3.6 timePropertiesDS. timeTraceable

如果时标和 timePropertiesDS. currentUtcOffset 的值可溯源到主参考,则 timePropertiesDS. timeTraceable 的值为 TRUE;否则,值应为 FALSE。

初始化值应按如下选择:

- a) 如果 timePropertiesDS. ptpTimescale(见 8.2.4.8)为 TRUE,并且在初始化时刻 timePropertiesDS. currentUtcOffset 可溯源到主参考,则该值应为 TRUE;否则
- b) 该值应为 FALSE。

8.2.3.7 timePropertiesDS. frequencyTraceable

如果决定时标的频率可溯源到主参考,则 timePropertiesDS. frequencyTraceable 的值为 TRUE;否则,值应为 FALSE。

初始化值应按如下选择:

- a) 如果在初始化时刻频率可溯源到主参考,则该值应为 TRUE;否则
- b) 该值应为 FALSE。

8.2.3.8 timePropertiesDS. ptpTimescale

如果最高级时钟的时标是 PTP,则 timePropertiesDS. ptpTimescale 的值为 TRUE;否则,值为 FALSE。

初始化值应按如下选择:

- a) 如果在初始时刻可知时标(见 7.2.1)是 PTP,则该值应设为 TRUE;否则
- b) 该值应为 FALSE,表示时标为 ARB。

8.2.3.9 timePropertiesDS. timeSource

timePropertiesDS. timeSource 的值为最高级时钟使用的时间源。

初始化值应按如下选择:

- a) 如果在初始化时刻时间源是可知的,则该值应被设为那个值;否则
- b) 该值应为 INTERNAL_OSCILLATOR。

8.2.4 portDS 数据集规范

8.2.4.1 概述

对于普通时钟的单个端口和边界时钟的每个端口,应维护以下“端口数据集”,作为协议判定和为报文字段提供值的基础。这样的数据集的个数应为 defaultDS. numberPorts 的值。

该数据集的成员如下:

- portDS. portIdentity;
- portDS. portState;
- portDS. logMinDelayReqInterval;
- portDS. peerMeanPathDelay;
- portDS. logAnnounceInterval;
- portDS. announceReceiptTimeout;
- portDS. logSyncInterval;
- portDS. delayMechanism;
- portDS. logMinPdelayReqInterval;
- portDS. versionNumber。

8.2.4.2 portDS 数据集静态成员

8.2.4.2.1 portDS. portIdentity

portDS. portIdentity 的值应为本地端口的 PortIdentity 属性,见 7.5.2。

8.2.4.3 portDS 数据集动态成员

8.2.4.3.1 portDS. portState

portDS. portState 的值应为与本端口相关联协议机的当前状态值(见 9.2),并应取自表 8 中的枚举。

表 8 PTP 状态枚举

PTP 状态枚举	值(十六进制)
INITIALIZING	01
FAULTY	02
DISABLED	03
LISTENING	04
PRE_MASTER	05

表 8 (续)

PTP 状态枚举	值(十六进制)
MASTER	06
PASSIVE	07
UNCALIBRATED	08
SLAVE	09
—	其他值保留

初始化值应为 INITIALIZING。

8.2.4.3.2 portDS.logMinDelayReqInterval

portDS.logMinDelayReqInterval 的值是以 2 为底 minDelayReqInterval 的对数,见 7.7.2.4。初始化值是实现特定的,并符合 7.7.2.4。

8.2.4.3.3 portDS.peerMeanPathDelay

如果 portDS.delayMechanism 成员的值是点到点(P2P),则 portDS.peerMeanPathDelay 的值应为链路上当前单向传输延时的估计值,即附加到本端口并使用对等延时机制进行计算的〈meanPathDelay〉,见 11.4。数据类型应为 TimeInterval。如果 portDS.delayMechanism 成员的值是端到端(E2E),则值应为 0。初始化值应为 0。

8.2.4.4 portDS 数据集可组态成员

8.2.4.4.1 portDS.logAnnounceInterval

portDS.logAnnounceInterval 的值应为以 2 为底 announceInterval 平均值的对数,见 7.7.2.2。

8.2.4.4.2 portDS.announceReceiptTimeout

portDS.announceReceiptTimeout 的值应为 announceInterval 的整数倍,见 7.7.3.1。

注: announceInterval 等于 $2^{\text{portDS.logAnnounceInterval}}$ 的值。

8.2.4.4.3 portDS.logSyncInterval

portDS.logSyncInterval 的值应为以 2 为底用于多播报文的 SyncInterval 平均值的对数,见 7.7.2.3。

注:对于单播传输速率应在单个端口基础上分别协商,不受本条约束。

8.2.4.4.4 portDS.delayMechanism

portDS.delayMechanism 的值应指出端口在计算〈meanPathDelay〉时所用的传输延时测量选项,值应取自表 9 中的枚举。除了在 PTP 行规中说明外,初始化值是实现特定的。

表 9 延时机制枚举

延时机制	值(十六进制)	规 范
E2E	01	端口被配置为使用延时请求-响应机制
P2P	02	端口被配置为使用对等延时机制
DISABLED	FE	端口未实现延时机制,见注。
注:除非适用 PTP 行规规定时钟仅谐振且不使用任何路径延时机制,否则都不应设为该值。		

注:9.1 允许重新配置。允许自动配置,但超出本标准范围。

8.2.4.4.5 portDS.logMinPdelayReqInterval

portDS.logMinPdelayReqInterval 的值应是以 2 为底 minPdelayReqInterval 的对数,见 7.7.2.5。

8.2.4.4.6 portDS.versionNumber

portDS.versionNumber 的值应指出端口所使用的 PTP 版本。

8.3 透明时钟的数据集

8.3.1 概述

可选地,透明时钟应维护每个缺省数据集和端口数据集的单个副本。

注:不同于普通时钟和边界时钟,透明时钟不为每个域维护单独的数据集。除了谐振外,透明时钟是域无关的。

8.3.2 transparentClockDefaultDS 数据集成员规范

8.3.2.1 概述

该数据集成员如下:

- transparentClockDefaultDS.clockIdentity;
- transparentClockDefaultDS.numberPorts;
- transparentClockDefaultDS.delayMechanism;
- transparentClockDefaultDS.primaryDomain。

8.3.2.2 transparentClockDefaultDS 数据集静态成员

8.3.2.2.1 transparentClockDefaultDS.clockIdentity

transparentClockDefaultDS.clockIdentity 的值应是本地时钟的 clockIdentity 属性(见 7.6.2.1)。

8.3.2.2.2 transparentClockDefaultDS.numberPorts

transparentClockDefaultDS.numberPorts 的值应为设备 PTP 端口的个数。

8.3.2.3 transparentClockDefaultDS 数据集的可组态成员

8.3.2.3.1 transparentClockDefaultDS.delayMechanism

如果透明时钟是端到端透明时钟,则 transparentClockDefaultDS.delayMechanism 的值应为 E2E,见表 9。如果透明时钟是点到点透明时钟,则值应为 P2P。

8.3.2.3.2 transparentClockDefaultDS.primaryDomain

transparentClockDefaultDS.primaryDomain 的值应是主谐振域的 domainNumber,见 10.1。初始化值应为 0。

8.3.3 transparentClockPortDS 数据集成员规范

8.3.3.1 概述

该数据集成员如下:

- transparentClockPortDS.portIdentity;
- transparentClockPortDS.logMinPdelayReqInterval;
- transparentClockPortDS.faultyFlag;
- transparentClockPortDS.peerMeanPathDelay。

8.3.3.2 portDS 数据集静态成员

8.3.3.2.1 transparentClockPortDS.portIdentity

transparentClockPortDS.portIdentity 的值应是本地端口的 PortIdentity 属性,见 7.5.2。

8.3.3.3 portDS 数据集动态成员

8.3.3.3.1 transparentClockPortDS.logMinPdelayReqInterval

transparentClockPortDS.logMinPdelayReqInterval 的值应是以 2 为底 MinPdelayReqInterval 的对数,见 7.7.2.5。

8.3.3.3.2 transparentClockPortDS.faultyFlag

如果端口故障,则 transparentClockPortDS.faultyFlag 的值应为 TRUE;如果端口运行正常,则值应为 FALSE。初始化值应为 FALSE。

8.3.3.3.3 transparentClockPortDS.peerMeanPathDelay

如果 transparentClockDefaultDS.delayMechanism 成员的值是 P2P,则 transparentClockPortDS.

peerMeanPathDelay 的值应为链路上当前单向传输延时的估计值,即附加到本端口并使用对等延时机制进行计算的〈meanPathDelay〉,见 11.4。如果 transparentClockDefaultDS.delayMechanism 成员的值是 E2E,则值应为 0。数据类型应为 TimeInterval。初始化值应为 0。

9 PTP 普通时钟和边界时钟

9.1 PTP 普通时钟和边界时钟的通用协议要求

普通时钟和边界时钟:

- a) 可在多个域(见 7.1)中运行。每个域的运行应独立于其他域;
- b) 当 9.2 中的状态机要求时,应根据 12.2 同步;
- c) 应通过使用以下选项对路径延时进行校正:
 - 1) 延时请求-响应机制,见 11.3;
 - 2) 对等延时机制,见 11.4。

只要任意时刻仅有一个机制是活动的,一个端口可同时实现延时请求-响应机制和对等延时机制。选择的方法不属于本标准范围。不禁止自动配置。

时钟仅与使用最佳主时钟算法所选的时钟同步。

除满足第 17 章条件外,当接收到的 Announce、Sync、Follow_Up 或 Delay_Resp 报文的报文头标志 altermateMasterFlag 值为 TRUE 时,普通时钟或边界时钟应丢弃该报文。

普通时钟应包含一个 PTP 端口,该端口遵循 9.2 的要求。

边界时钟应包含多个 PTP 端口,每个 PTP 端口都遵循 9.2 的要求。

9.2 状态协议

9.2.1 通用状态要求

所有普通时钟和边界时钟都应实现 9.2 的状态机和状态行为。

9.2.2 slave-only(只作为从时钟)普通时钟

普通时钟可被设计为 slave-only 时钟或 non-slave-only 时钟。实现可以通过管理报文 SLAVE_ONLY 或实现相关方法,可选地提供配置到 slave-only 模式的能力。slave-only 时钟应实现图 24 所示的状态机。

注: slave-only 时钟不能进入 MASTER 状态。因此,系统应至少包含一个 non-slave-only 时钟。slave-only 时钟使用的状态机不同于 non-slave-only 时钟,并且具有不同的 clockClass 号,见 7.6.2.4。

9.2.3 non-slave-only(不仅仅作为从时钟)时钟

边界时钟不应为 slave-only 时钟。边界时钟以及不被设计为或不被配置为 slave-only 的普通时钟,应实现图 23 所示的状态机。

9.2.4 状态定义

除了 16.1 定义的单播报文条件外,与图 23 和图 24 状态机相关联的端口状态的行为应如表 10 定义。

除了 DISABLED 状态外,端口可使用接收到的 PTP 报文中的任何信息,只要该用法不违反协议要求。

9.2.5 状态机

本条的状态机应决定具有状态的 PTP 端口的允许状态转换。

4.3 规定图 23 和图 24 所用的符号。

表 10 PTP portState 定义

PTP portState	描 述
INITIALIZING	当端口处于 INITIALIZING 状态时,端口初始化其数据集、硬件和通信设备。时钟端口不应向其通信路径放置任何 PTP 报文。如果边界时钟有一个端口处于 INITIALIZING 状态,则其所有端口都应处于 INITIALIZING 状态
FAULTY	协议的故障状态。当端口处于该状态时,除了要求的对另外管理报文响应的管理报文外,端口不应向其通信路径放置任何 PTP 报文。在边界时钟,故障端口的行为不应影响设备的其他端口。如果处于该状态的端口的故障行为,不能限制在故障端口,则所有端口都应处于 FAULTY 状态
DISABLED	端口不应向通信路径放置任何 PTP 报文。在边界时钟,该端口的行为不允许影响边界时钟任何其他端口的行为。除管理报文外,处于该状态的端口应丢弃所有接收到的 PTP 报文
LISTENING	端口等待 announceReceiptTimeout 超时,或者等待接收来自主时钟的 Announce 报文。该状态的目的是允许有序地将时钟增加到一个域。当端口处于该状态时,除 Pdelay_Req、Pdelay_Resp、Pdelay_Resp_Follow_Up、信号报文或要求的对另外管理报文响应的管理报文外,端口不应向其通信路径放置任何 PTP 报文
PRE_MASTER	除了处于该状态的端口不能向其通信路径上放置除 Pdelay_Req、Pdelay_Resp、Pdelay_Resp_Follow_Up、信号报文或管理报文以外的任何报文外,端口在所有方面的行为应同处于 MASTER 状态一样
MASTER	端口作为主时钟端口进行动作
PASSIVE	除 Pdelay_Req、Pdelay_Resp、Pdelay_Resp_Follow_Up、信号报文或要求的对另外管理报文响应的管理报文外,端口不应向其通信路径放置任何报文
UNCALIBRATED	在域中检测到一个或多个主时钟端口。已选择了合适的主时钟端口,并且本地端口正准备同步到所选的主时钟端口。这是一个暂态,以便允许初始化同步动作,当选择一个新主时钟端口时的数据集更新,以及其他实现特定行为
SLAVE	端口同步到所选的主时钟端口

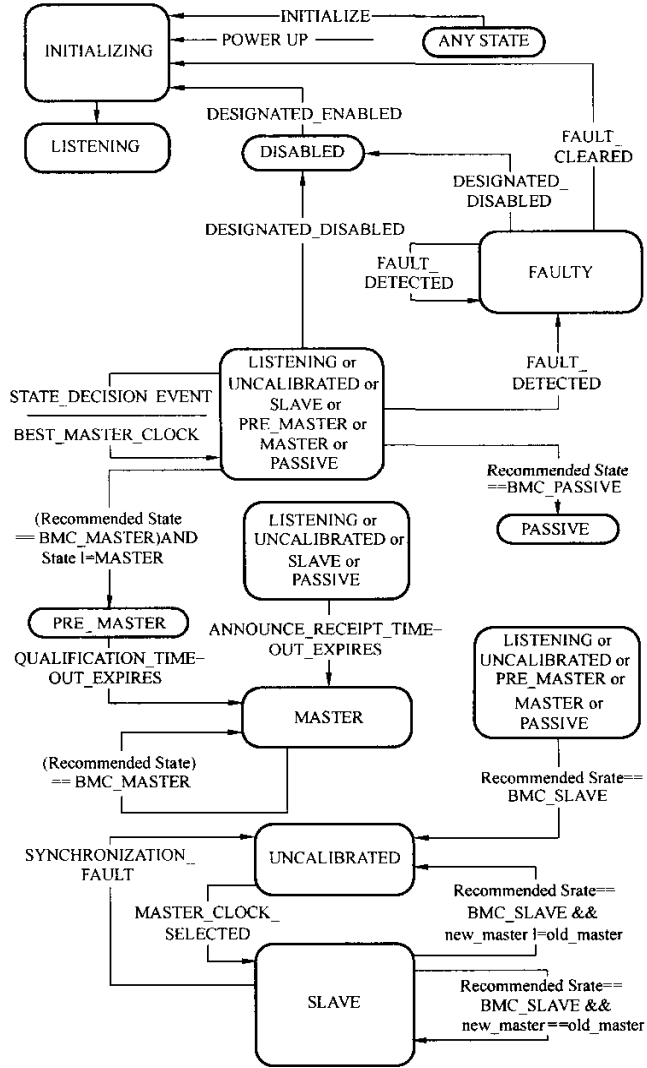


图 23 完整实现的状态机

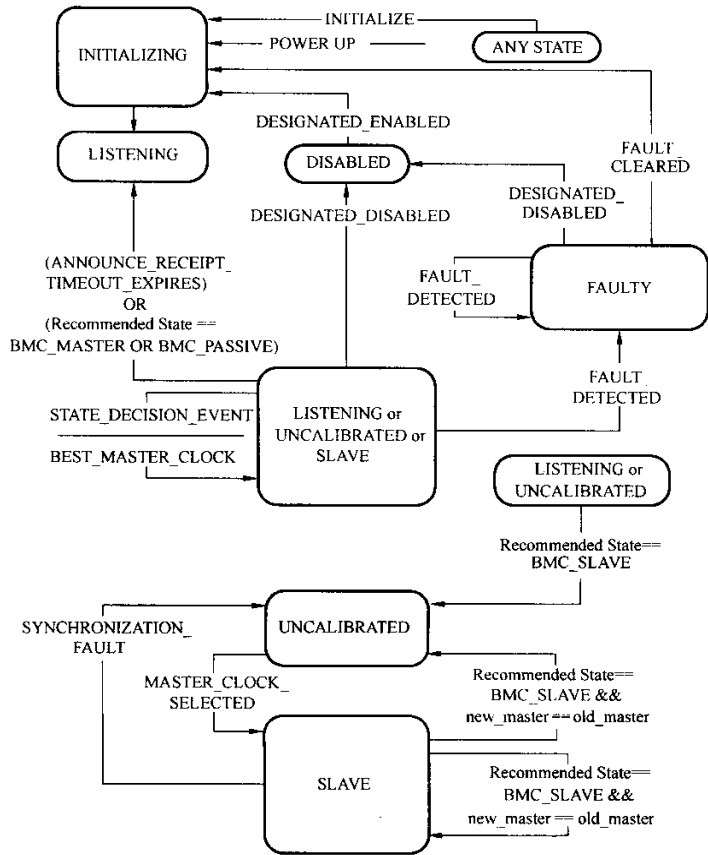


图 24 slave-only 实现的状态机

在图 23 和图 24 中, 当一个故障被清除, 或先前被禁用的端口启用了, 状态机就转换到 INITIALIZING 状态。

注: 在后续转换到 LISTENING 状态之前, 不要求实现执行 INITIALIZE 命令、上电或复位之后的所有步骤, 但要求达到执行这些步骤的结果。

9.2.6 引发 PTP 状态转换的事件

9.2.6.1 概述

引发图 23 和图 24 状态机中状态转换的每个事件的规范见 9.2.6。

9.2.6.2 POWERUP

POWERUP 事件应通过打开设备电源实现实例化。也可由实现特定的复位机制实例化, 如复位按钮。

9.2.6.3 INITIALIZE

INITIALIZE 事件应由 INITIALIZE 管理报文的接收实例化, 或如果该报文 initializationKey 字段要求, 由其等价方法实例化。

9.2.6.4 DESIGNATED_ENABLED

DESIGNATED_ENABLED 事件应由 ENABLE_PORT 管理报文的接收实例化。

9.2.6.5 DESIGNATED_DISABLED

DESIGNATED_DISABLED 事件应由 DISABLE_PORT 管理报文的接收实例化。

9.2.6.6 FAULT_CLEARED

FAULT_CLEARED 事件应由故障条件或阻止端口正确操作条件的清除实例化。

注：所有检测到的故障条件的清除可源于管理报文的动作或内部规程。

9.2.6.7 FAULT_DETECTED

FAULT_DETECTED 事件应由阻止端口正确操作的内部条件的发生实例化。

9.2.6.8 STATE_DECISION_EVENT

STATE_DECISION_EVENT 机制通过使用接收到的 Announce 报文中的数据判断哪个是最佳主时钟，以及本地时钟端口是否需改变状态。每个时钟应实现一个生成 STATE_DECISION_EVENT 的机制，并且该事件的发生应实现图 25 所示逻辑。图 25 中所示的最佳主时钟算法为本标准规定的缺省最佳主时钟算法。如果规定可选最佳主时钟算法（见 9.3.1），则图中方框标记的最佳主时钟算法的部分可不同。

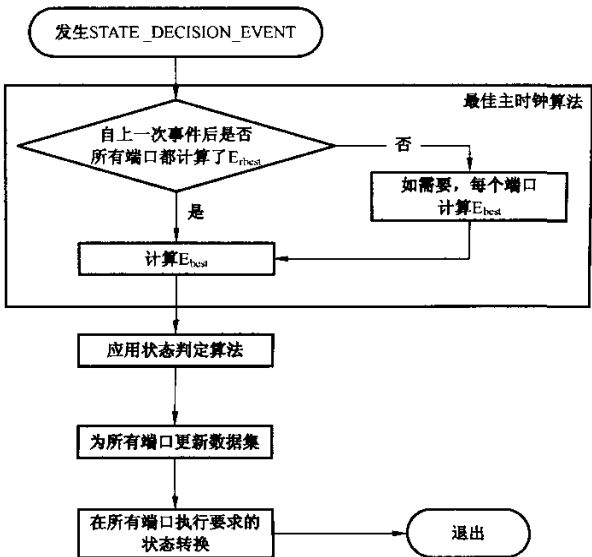


图 25 STATE_DECISION_EVENT 逻辑

STATE_DECISION_EVENT 应：

- 在一个时钟的所有端口逻辑地同时发生；
- 每 Announce 报文传输间隔内至少发生一次；
- 当任意端口处于 INITIALIZING 状态时不得发生。

对于实现缺省最佳主时钟算法（见 9.3.1）的节点，在 STATE_DECISION_EVENT 逻辑的第一个动作之前，或作为 STATE_DECISION_EVENT 逻辑的第一个动作，每个不处于 DISABLED 或 FAULTY 状态的端口 N，应计算 E_{rbest} 的更新值，以反映自上一次 STATE_DECISION_EVENT 以后对 Announce 报文的接收。所有端口都计算 E_{rbest} 集之后，时钟应计算 E_{best} 。

对所有节点，时钟应按给定顺序完成以下任务：

- a) 采用最佳主时钟算法；
- b) 更新相应的数据集；
- c) 在状态机中实例化推荐状态事件；
- d) 在所有端口执行要求的状态改变。

这些任务应原子性地执行，见 3.1.2。该输入信息应包含 E_{rbest} 值的集合。这些任务的执行见 9.3 定义。

9.2.6.9 推荐状态

推荐状态事件是由 STATE_DECISION_EVENT 引发的执行最佳主时钟算法的结果。

9.2.6.10 QUALIFICATION_TIMEOUT_EXPIRES

〈qualificationTimeoutInterval〉的超时定义 QUALIFICATION_TIMEOUT_EXPIRES 事件。该超时机制决定了时钟处于 PRE_MASTER 状态的时间。

当端口进入 PRE_MASTER 状态时,启动〈qualificationTimeoutInterval〉。应在按如下方法计算出的时间间隔后发生超时:

qualificationTimeoutInterval 应为 announceInterval(见 7.7.2.2)的 N 倍,单位为秒(s),当:

- a) 如果根据图 26 的判定点 M1 或 M2,使得推荐状态=MASTER 事件,则 N 应为 0;
- b) 如果根据图 26 的判定点 M3,使得推荐状态=MASTER 事件,则 N 应为 currentDS.stepsRemoved 字段值加 1。

9.2.6.11 ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES

每个协议机应支持定义 announceReceiptTimeoutInterval 的超时机制,其值为 portDS. announceReceiptTimeout 乘以 announceInterval 的值,见 7.7.3.1。

该超时值加上一个在(0,1)期间均匀分布的随机数乘以 announceInterval 的值,该和超时将引发 ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES 事件。

当任何以下事件发生时,应启动或重新启动端口的本超时机制:

- a) 对处于 UNCALIBRATED 或 SLAVE 状态的端口,当从父时钟接收 Announce 报文时。父时钟由 parentDS.parentPortIdentity 指示。
- b) 除非第 9 章另有规定,当前 portDS. announceReceiptTimeoutInterval 超时时刻。
- c) 当进入 LISTENING、UNCALIBRATED、SLAVE 或 PASSIVE 状态时。
- d) 对处于 PASSIVE 状态的端口,当从由于传输 Announce 报文导致端口进入 PASSIVE 状态的时钟接收 Announce 报文时。该时钟的辨别通过比较各自报文 sourcePortIdentity 字段实现。

当进入 INITIALIZING、PRE_MASTER、FAULTY、DISABLED 或 MASTER 状态时,应停止或不重新启动本超时机制。

除图 23 和图 24 的状态改变外,当 ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES 事件发生时,处于 LISTENING、PASSIVE、UNCALIBRATED 或 SLAVE 状态的端口,应在进入 MASTER 状态前更新数据集,操作如下:

- 对于普通时钟,按判据 M1 规定将端口数据集更新为 MASTER 状态配置(见 9.3.5);
- 对于没有其他端口处于 SLAVE 状态的边界时钟,按判据 M1 规定将端口数据集更新为 MASTER 状态配置(见 9.3.5);
- 对于有其他端口处于 SLAVE 状态的边界时钟,按判据 M3 规定更新端口数据集为 MASTER 状态配置(见 9.3.5)。

9.2.6.12 SYNCHRONIZATION_FAULT

SYNCHRONIZATION_FAULT 事件的实例化是实现特定的。只要时钟处于 SLAVE 状态,并且实现检测到要求重新执行发生在 UNCALIBRATED 状态下的功能的实现环境(重新执行保证正确的同步),该事件应被实例化。

9.2.6.13 MASTER_CLOCK_SELECTED

MASTER_CLOCK_SELECTED 事件的实例化是实现特定的。只要处于 UNCALIBRATED 状态的时钟,已满足保证在 SLAVE 状态同步的所有实现和协议必备要求,该事件应被实例化。

9.2.7 边界时钟端口使用 PTP 事件

对于边界时钟,引发图 23 和图 24 状态机状态转换的事件(见 9.2.6)应被用于设备端口的状态机,如表 11 所示。

表 11 边界时钟事件的可用性

事件名称	端口可用性
POWERUP	所有端口
INITIALIZE	所有端口
FAULT_DETECTED	所有受故障影响的端口
FAULT_CLEARED	所有受故障影响的端口
STATE_DECISION_EVENT	所有端口
推荐状态,见注	所有端口
ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES,或 QUALIFICATION_TIMEOUT_EXPIRES	发出超时机制到时信号的端口
DESIGNATED_ENABLED,或 DESIGNATED_DISABLED	初始化管理报文规定的端口
MASTER_CLOCK_SELECTED	发出事件信号的端口
SYNCHRONIZATION_FAULT	发出事件信号的端口
注: 推荐状态是由 STATE_DECISION_EVENT 引发的最佳主时钟算法执行结果的状态。	

9.3 最佳主时钟算法

9.3.1 最佳主时钟算法的选择

PTP 允许使用 2 种方式的最佳主时钟算法:

- 缺省地,9.3.2~9.3.4 规定的机制;
- 如果 PTP 行规中有规定,备选最佳主时钟算法。

任何备选最佳主时钟算法应满足以下要求:

- a) 算法的输出应提供执行 9.2.5、9.2.6.8 和 9.2.6.9 中 PTP 状态机和状态判定事件所要求的推荐状态。推荐状态应满足 9.2.4 的要求。可选地,备选算法可以是动态算法,或是在运行节点的端口上简单配置推荐状态值的静态算法。
- b) 算法的输出应提供用于更新数据集(见 9.3.5)的状态判定代码,以及根据这些代码进行更新所要求的任意数据。这些判定代码应如下:
 - 1) M1:端口处于 MASTER 状态,因为端口在 clockClass 1~127 节点上,并且是系统最高级时钟的端口;
 - 2) M2:端口处于 MASTER 状态,因为端口在 clockClass 128 或更高节点上,并且是系统最高级时钟的端口;
 - 3) M3:端口处于 MASTER 状态,但并不是系统最高级时钟的端口;
 - 4) S1:端口处于 SLAVE 状态;
 - 5) P1:端口处于 PASSIVE 状态,因为端口在 clockClass 1~127 节点上,并且不在系统最高级时钟上,或者处于 PASSIVE 以中断一个时间环;
 - 6) P2:端口处于 PASSIVE 状态,因为端口在 clockClass 128 或更高节点上,并且处于 PASSIVE 以中断一个时间环。

域中每个普通时钟和边界时钟的所有端口都本地运行 BMC 算法。因为连续运行,该算法不断重新适应网络或时钟的改变。

9.3.2 BMC 算法

9.3.2.1 概述和术语定义

9.3.2 规定了本地时钟用来判断在所有时钟(包括自身)中哪个是“最佳”时钟的方法,以决定其所有端口的下一状态,见表 10。在域中每个时钟独立执行算法。换言之,时钟并不协商哪个应为主时钟

哪个应为从时钟——相反,每个时钟仅计算自己端口的状态。该算法避免了两个主时钟、没有主时钟,或主时钟来回替换的配置。

最佳主时钟算法 BMC 包括两部分:

- a) 数据集比较算法,用于判定两个时钟端口中哪个是较好的,见 9.3.4;
- b) 用于计算涉及端口的推荐状态的算法,见 9.3.3。

本判据是基于给定时钟端口接收到的 Announce 报文中包含的信息,以及该给定时钟的 defaultDS 数据集的值。

9.3.2.2 通用 BMC 规范

通过 defaultDS 数据集 D_0 和其他数据集描述特性,普通时钟 C_0 或具有“N”个端口的边界时钟 C_0 上的 BMC 算法应遵循以下:

- a) 对于 C_0 的每个端口“r”,应比较合格的 Announce 报文(见 9.3.2.5)。这些报文是从连接到端口“r”所在通信路径的其他时钟的端口接收到的。
- b) 对于 C_0 的每个端口“r”,报文中的最佳报文 $E_{r\text{best}}$,应通过使用数据集比较算法判定。
- c) 对于时钟 C_0 的 N 个端口的集合,报文中的最佳报文 E_{best} ,应通过使用数据集比较算法从 N 个 $E_{r\text{best}}$ 报文中判定。
- d) 对于时钟 C_0 N 个端口的每个端口,报文 E_{best} 、 $E_{r\text{best}}$ 和 defaultDS 数据集 D_0 应同状态判定算法一起使用,以决定适用每个端口状态机(见 9.2.5)的 BMC 事件。
- e) 除了主时钟簇选项的条款外(17.3),当根据 9.3.2 执行最佳主时钟算法时,端口应丢弃所有 alternateMasterFlag 为 TRUE 的 Announce 报文。这些报文不应进入用于最佳主时钟算法的 foreignMasterDS。备选主时钟选项(见 17.4)规定了该标志为 TRUE 的 Announce 报文的其他用法。

9.3.2.3 $E_{r\text{best}}$ 的计算

每个端口可判定 $E_{r\text{best}}$ 而与其他端口活动无关。 E_{best} 的判定、状态判定算法的使用,以及这些判定结果应用所要求的任何状态改变的实例化,都是原子性的,见 3.1.2。

当计算 $E_{r\text{best}}$ 时,端口“r”应:

- a) 仅考虑端口“r”上接收到的合格的 Announce 报文(见 9.3.2.5)。
- b) 包括至少两个合格的来自外部主时钟的 Announce 报文(如果这样的报文存在)。在这种情况下,“r”应从其实现特定的 foreignMasterDS 数据集(见 9.3.2.4)中,删除所有被考虑但未被选为 $E_{r\text{best}}$ 的记录。
- c) 如果端口“r”处于 SLAVE 状态,包括先前在端口“r”上的 $E_{r\text{best}}$ 计算结果。然而,如果端口“r”从相同发送端口接收到一个更新的合格 Announce 报文,则应考虑该报文中的值。如果在端口“r”上先前 $E_{r\text{best}}$ 计算期间,为所选时钟发生了一个 ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES 事件,则不应包括先前在端口“r”上的 $E_{r\text{best}}$ 计算值。

注:应包括来自许多外部主时钟的 Announce 报文,外部主时钟的个数与端口资源允许的一样多。

9.3.2.4 foreignMasterDS 数据集规范

9.3.2.4.1 概述

每个端口应维护一个实现特定的 foreignMasterDS 数据集,以便对 Announce 报文进行合格判定。数据集中的每个登入项包含两个成员:

- foreignMasterDS, foreignMasterPortIdentity;
- foreignMasterDS, foreignMasterAnnounceMessages。

9.3.2.4.2 foreignMasterDS, foreignMasterPortIdentity

foreignMasterDS, foreignMasterPortIdentity 的值应是从外部主时钟接收到的 Announce 报文中 sourcePortIdentity 字段的值。

9.3.2.4.3 foreignMasterDS, foreignMasterAnnounceMessages

foreignMasterDS, foreignMasterAnnounceMessages 的值应是从 foreignMasterDS, foreignMaster-PortIdentity 成员标识的外部主时钟接收到的 Announce 报文的个数,这些 Announce 报文是在 FOREIGN_MASTER_TIME_WINDOW 的时间窗内接收到的。

9.3.2.4.4 FOREIGN_MASTER_TIME_WINDOW 和 FOREIGN_MASTER_THRESHOLD

这两个属性值决定了在执行最佳主时钟算法时,考虑是否从先前静默的外部主时钟接受 Announce 报文的准则。这些属性值应为:

- FOREIGN_MASTER_TIME_WINDOW: 4 倍的 announceInterval;
- FOREIGN_MASTER_THRESHOLD: 在 FOREIGN_MASTER_TIME_WINDOW 内接收到的 2 个 Announce 报文。

9.3.2.4.5 foreignMasterDS 数据集大小

实现特定的 foreignMasterDS 数据集应具有最小 5 个外部主时钟记录的容量。

9.3.2.5 Announce 报文的合格判定

端口“r”接收到的 Announce 报文 S,应按如下方法判定其是否合格,以用于 BMC 算法:

- a) 如果 S 是端口“r”发出的,或从包含端口“r”的时钟的其他任何端口发出(见 9.5.2),则 S 不应是合格的;
- b) 如果 S 不是端口“r”从给定时钟接收到的最新 Announce 报文,则 S 应是不合格的;
- c) 如果 S 的发送方是外部主时钟 F,并且在最新的 FOREIGN_MASTER_TIME_WINDOW 间隔内,从 F 接收了少于 FOREIGN_MASTER_THRESHOLD 个不完全相同的 Announce 报文,则 S 应是不合格的;

注:本时间窗口和阈值的目的是为了只对稳定的潜在主时钟进行合格判定,并防止最佳主时钟的伪转移。

- d) 如果 S 的 stepsRemoved 字段为 255 或更大,则 S 应是不合格的;

注:本条确保杜绝欺诈帧。这是对出于此目的的 PATH_TRACE 选项的必要后备,见 16.2。如果网络规模大到可能存在涉及 255 个边界时钟的回路,则该基于 stepsRemoved 的机制可能导致 PTP 失效。这在实际应用中极不可能发生。

- e) 否则,S 是合格的。

9.3.3 状态判定算法

图 26 定义了 in 最佳主时钟算法中使用的状态判定算法。当使用该算法获得一个判定后,本地时钟的数据集应更新,见 9.3.5。

D_0 表示时钟 C_0 的特性,见 C_0 数据集。

D_0 同 E_{best} 或 $E_{r_{best}}$ 的比较应使用数据集比较算法。对于图 26 中的判定块“根据拓扑 E_{best} 比 $E_{r_{best}}$ 好”, E_{best} 和 $E_{r_{best}}$ 的比较应使用数据集比较算法。

应根据 D_0 的 defaultDS, clockQuality, clockClass 成员的值,判断 D_0 是否为 clockClass 1~127。

图 23 和图 24 的协议机状态机中所用的推荐状态值,应为图 26 给定的推荐状态值。

9.3.4 数据集比较算法

最佳主时钟算法通过比较代表各自时钟的数据集,实现一个时钟同另一个时钟的比较。数据集比较算法定义如图 27 和图 28 所示。数据集在这些图中被表示为集 A 和集 B。表 12 给出数据集值的来源。如果在比较 D_0 、 E_{best} 或 $E_{r_{best}}$ 的任意两个数据集中,一个数据集是空集,则非空集被视为较好集。 D_0 从来不是空集,但 E_{best} 、 $E_{r_{best}}$ 或两个都可能是空集。

注:该比较随后的通用过程如下:

- a) 寻找哪个时钟将自己的时间溯源到最佳最高级时钟。找到这个时钟而不是较好时钟,对于算法的稳定性是很重要的。
- b) 如果特性都相同,则使用决定性判断技术。

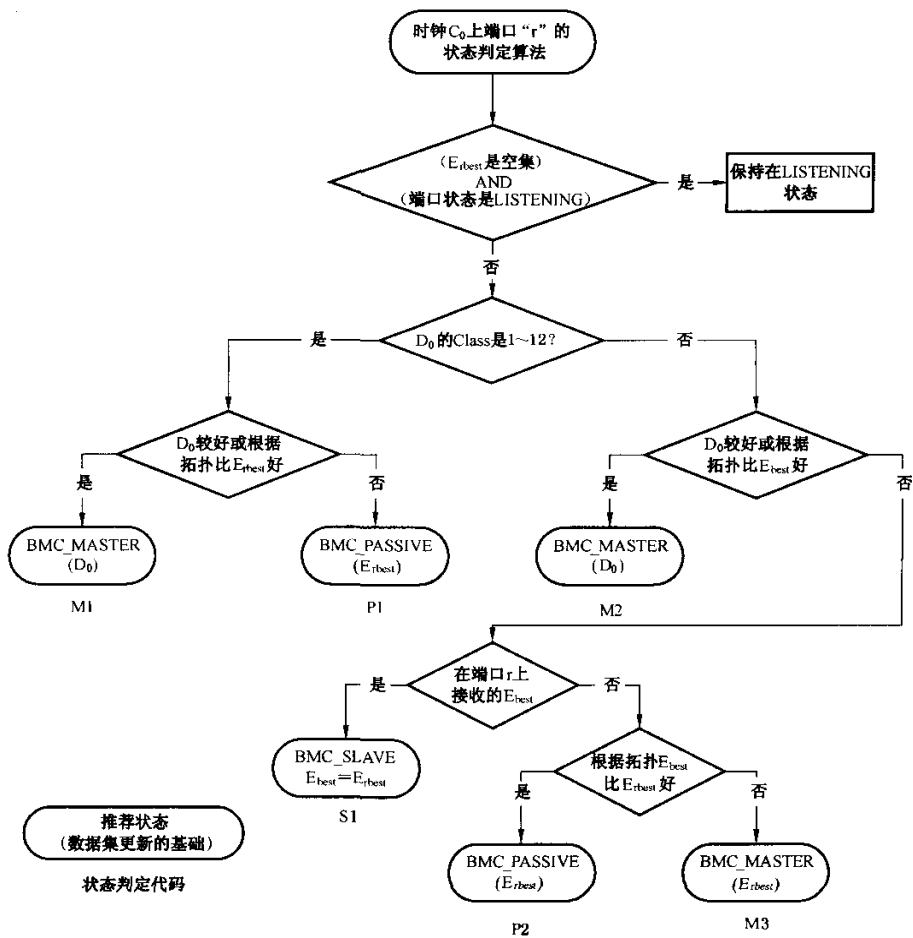


图 26 状态判定算法

表 12 数据集比较算法的信息源

当按照图 27 和图 28 考虑本特性时	如果数据集是 E _{best} 或 E _{rbest} , 使用相关联的 Announce 报文的这些字段	如果数据集是 D ₀ , 使用本地时钟 defaultDS 数据集的这些字段
GM Priority 1	grandmasterPriority 1	defaultDS. Priority 1
GM identity	grandmasterIdentity	defaultDS. clockIdentity
GM class	grandmasterClockQuality. clockClass	defaultDS. clockQuality. clockClass
GM accuracy	grandmasterClockQuality. clockAccuracy	defaultDS. clockQuality. clockAccuracy
GM offsetScaledLogVariance	grandmasterClockQuality. offsetScaledLogVariance	defaultDS. clockQuality. offsetScaledLogVariance
GM Priority 2	grandmasterPriority 2	defaultDS. Priority 2
经过几步 (Steps Removed)	stepsRemoved	值为 0
发送方的 Identity	sourcePortIdentity	defaultDS. clockIdentity

表 12 (续)

当按照图 27 和图 28 考虑本特性时	如果数据集是 E_{best} 或 $E_{r_{best}}$, 使用相关联的 Announce 报文的这些字段	如果数据集是 D_0 , 使用本地时钟 defaultDS 数据集的这些字段
接收方的 Identity	parentDS, portIdentity (接收报文端口的 portDS 数据集的)	defaultDS, clockIdentity
接收方的端口号	parentDS, portIdentity, portNumber (接收报文端口的 portDS 数据集的)	值为 0

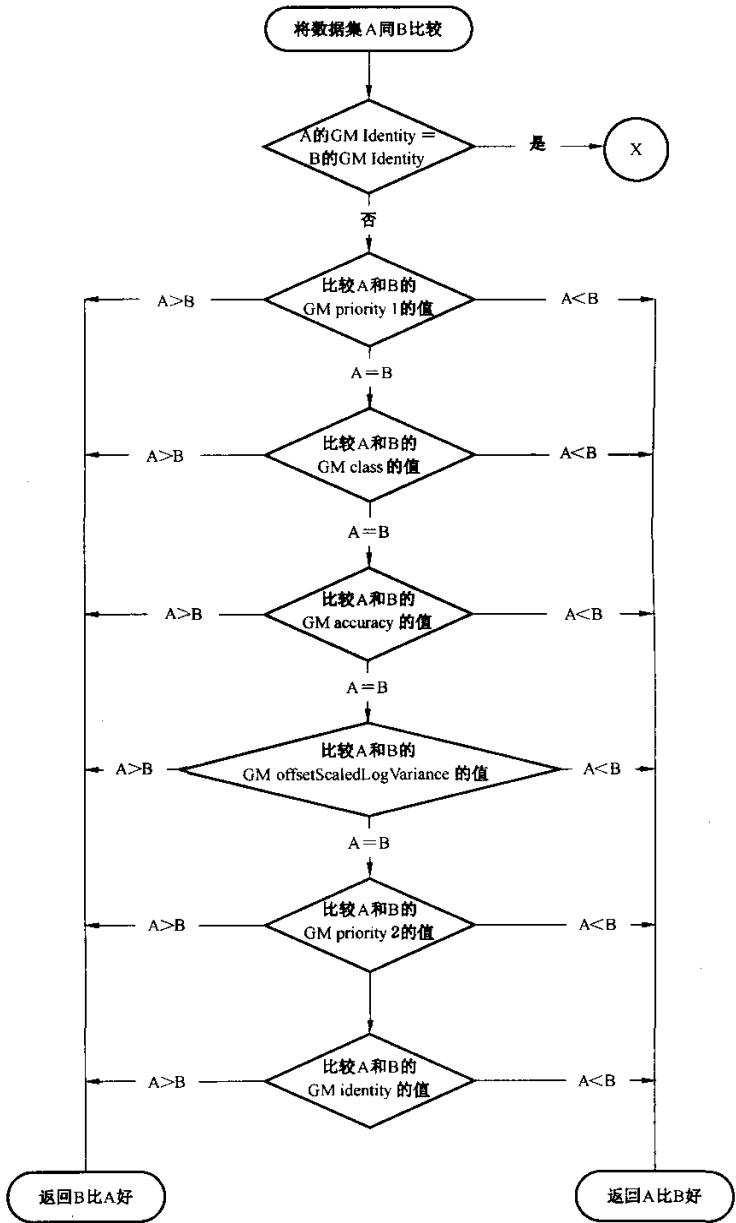


图 27 数据集比较算法, 第 1 部分

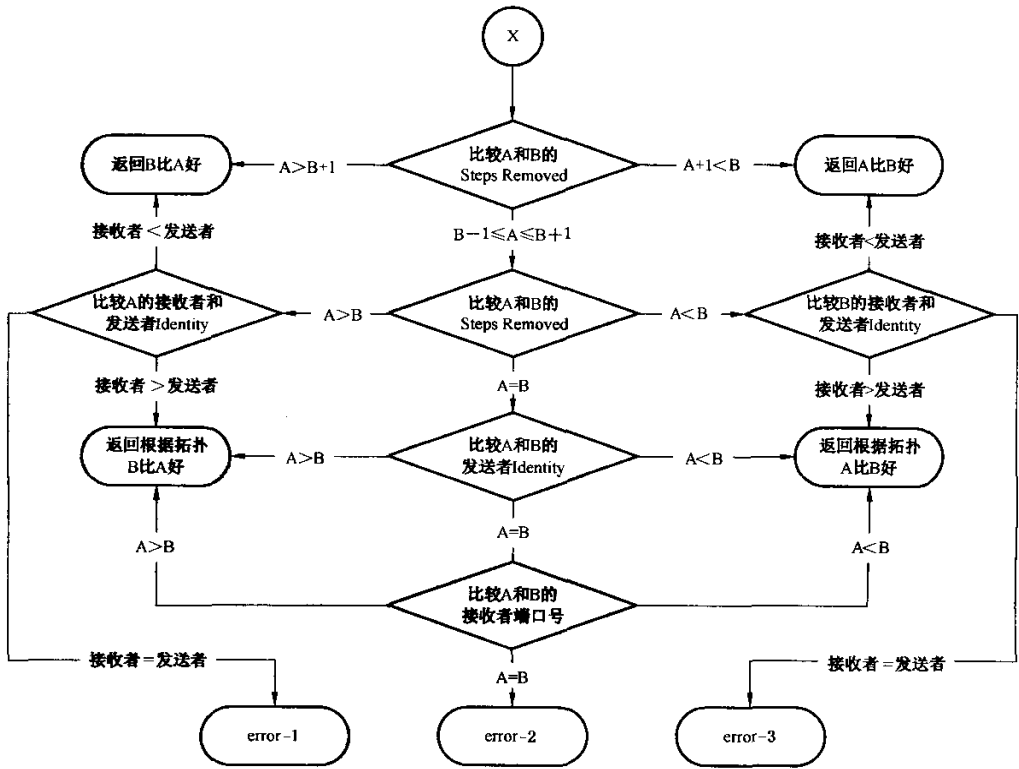


图 28 数据集比较算法，第 2 部分

注：图 28 所示返回的错误不用于图 26 的状态判定算法。这些返回值不属于 9.3.2.5，但有助于诊断或错误检测。当违反 9.3.2.5 时，错误情况就产生。Error-1 是指在相同端口发送和接收了其中一个报文。Error-2 是指有重复报文，或它们是来自相同最高级时钟的前后两个报文。

9.3.5 数据集的更新

currentDS、parentDS、portDS 和 timePropertiesDS 数据集的更新，应按表 13～表 16 定义的状态判定代码的更新。判定代码由图 26 中缺省最佳主时钟算法的“状态判定代码”，以及 9.3.1PTP 行规定的替代算法所表示。

portDS 数据集的 portDS、portState 成员应随着图 23 和图 24 所示的、与每个端口相关联的协议状态机的改变而更新，状态机的改变根据 9.2.6.8。

表 13～表 16 未包括的数据集字段不更新。

表 13 状态判定代码 M1 和 M2 的更新

需更新的字段	除非另有说明，来自时钟 defaultDS 数据集的指定字段
currentDS 数据集	
currentDS.stepsRemoved	设为 0
currentDS.offsetFromMaster	设为 0
currentDS.meanPathDelay	设为 0
parentDS 数据集	
parentDS.parentPortIdentity	parentDS.clockIdentity 成员设为 defaultDS.clockIdentity 字段的值，parentDS.parentPortIdentity.portNumber 成员设为 0

表 13 (续)

需更新的字段	除非另有说明,来自时钟 defaultDS 数据集的指定字段
parentDS. grandmasterIdentity	defaultDS. clockIdentity
parentDS. grandmasterClockQuality	defaultDS. clockQuality
parentDS. grandmasterPriority 1	defaultDS. Priority 1
parentDS. grandmasterPriority 2	defaultDS. Priority 2
timePropertiesDS 数据集	
timePropertiesDS. currentUtcOffset	遵循 9.4 的规则
timePropertiesDS. currentUtcOffsetValid	遵循 9.4 的规则
timePropertiesDS. leap59	遵循 9.4 的规则
timePropertiesDS. leap61	遵循 9.4 的规则
timePropertiesDS. timeTraceable	遵循 9.4 的规则
timePropertiesDS. frequencyTraceable	遵循 9.4 的规则
timePropertiesDS. ptpTimescale	遵循 9.4 的规则
timePropertiesDS. timeSource	遵循 9.4 的规则
portDS 数据集	
portDS. portState	状态是将图 26 中的推荐状态应用到相应的图 23 或图 24 状态机的结果

表 14 状态判定代码 M3 的更新

更新该字段	来自指定源
portDS 数据集	
portDS. portState	状态是将图 26 中的推荐状态应用到相应的图 23 或图 24 状态机的结果

表 15 状态判定代码 P1 和 P2 的更新

更新该字段	来自指定源
portDS 数据集	
portDS. portState	状态是将图 26 中的推荐状态应用到相应的图 23 或图 24 状态机的结果

表 16 状态判定代码 S1 的更新

更新该字段	来自指定源
currentDS 数据集	
currentDS. stepsRemoved	1 + E _{best} 的 stepsRemoved 值
parentDS 数据集	
parentDS. parentPortIdentity	E _{best} 的 sourcePortIdentity
parentDS. grandmasterIdentity	E _{best} 的 grandmasterIdentity
parentDS. grandmasterClockQuality	E _{best} 的 grandmasterClockQuality
parentDS. grandmasterPriority 1	E _{best} 的 grandmasterPriority 1
parentDS. grandmasterPriority 2	E _{best} 的 grandmasterPriority 2
timePropertiesDS 数据集	

表 16 (续)

更新该字段	来自指定源
timePropertiesDS. currentUtcOffset	E _{best} 的 currentUtcOffset 字段
timePropertiesDS. currentUtcOffsetValid	E _{best} 的 flagField 的 currentUtcOffsetValid 位的逻辑值
timePropertiesDS. leap59	E _{best} 的 flagField 的 leap59 位的逻辑值
timePropertiesDS. leap61	E _{best} 的 flagField 的 leap61 位的逻辑值
timePropertiesDS. timeTraceable	E _{best} 的 flagField 的 timeTraceable 位的逻辑值
timePropertiesDS. frequencyTraceable	E _{best} 的 flagField 的 frequencyTraceable 位的逻辑值
timePropertiesDS. ptpTimescale	E _{best} 的 flagField 的 ptpTimescale 位的逻辑值
timePropertiesDS. timeSource	E _{best} 的 timeSource 字段的值
portDS 数据集	
portDS. portState	状态是将图 26 中的推荐状态应用到相应的图 23 或图 24 状态机的结果
注: E _{best} 是一个 Announce 报文。	

9.4 最高级时钟

最高级时钟决定域的时标以及相关时标属性。

注: 作为图 26 中状态判定算法 M1 和 M2 的结果而处于 MASTER 状态的时钟是最高级时钟。

如果时标是 PTP, 并且时钟是 class 6、7 或 52, 则 timePropertiesDS 数据集成员应按如下设置:

- timePropertiesDS. leap59 和 timePropertiesDS. leap61: 如果已知, 设为从主参考获得的值; 否则, 设为 FALSE。
- timePropertiesDS. currentUtcOffset: 当节点被设计时, 如果已知, 设为从主参考获得的值; 否则, 设为当前闰秒数, 见 7. 2. 3。
- timePropertiesDS. currentUtcOffsetValid: 如果已知 currentUtcOffset 的值是正确的, 设为 TRUE; 否则, 设为 FALSE。
- timePropertiesDS. ptpTimescale: 设为 TRUE。
- timePropertiesDS. timeTraceable: 如果时间可溯源到主参考, 设为 TRUE; 否则, 设为 FALSE。
- timePropertiesDS. frequencyTraceable: 如果频率可溯源到主参考, 设为 TRUE; 否则, 设为 FALSE。
- timePropertiesDS. timeSource: 如果已知, 设为表 7 中的适当值; 否则, 设为 INTERNAL_OSCILLATOR。

如果时标是 ARB, 则对于除 6、7 或 52 之外的所有 clockClass, 除非另有说明, timePropertiesDS 数据集成员应被设为:

- timePropertiesDS. leap59 和 timePropertiesDS. leap61: 设为 FALSE;
- timePropertiesDS. currentUtcOffset: 当节点被设计时, 设为当前闰秒数, 见 7. 2. 3;
- timePropertiesDS. currentUtcOffsetValid: 如果已知 currentUtcOffset 是正确的, 设为 TRUE; 否则, 设为 FALSE;
- timePropertiesDS. ptpTimescale: 设为 FALSE;
- timePropertiesDS. timeTraceable: 如果时间可溯源到主参考, 设为 TRUE; 否则, 设为 FALSE;
- timePropertiesDS. frequencyTraceable: 如果频率可溯源到主参考, 设为 TRUE; 否则, 设为 FALSE;

——timePropertiesDS. timeSource:如果已知,设为表 7 中的适当值;否则,设为 INTERNAL_OSCILLATOR。

如果 timePropertiesDS. leap59 或 timePropertiesDS. leap61 是 TRUE,则在当日结束的午夜(UTC)之前的时间段里都应被设为 TRUE,以及在以下时间段之后被设为 TRUE:

- 当最高级时钟判定该值为 TRUE 时;
- 当日午夜(UTC)之前的 12 个小时。

当第一个具有 timePropertiesDS. currentUtcOffset 更新值的 Announce 报文出现时,这些变量值应被设为 FALSE。

timePropertiesDS. currentUtcOffset、timePropertiesDS. leap59 和 timePropertiesDS. leap61 的更新应在午夜(UTC)的 ± 2 个 announceInterval 时间内完成。

注 1: 闰秒数和 currentUtcOffset 的改变在午夜(UTC)按照国际标准进行。但在实际实现中,由于代码执行时间的差异,currentUtcOffset 的实际更新以及 timePropertiesDS. leap59 和 timePropertiesDS. leap61 的 FALSE 值设置,可在午夜(UTC)稍微提前或错后的时间进行。 ± 2 个 announceInterval 更新的要求可在主时钟很容易地实现,以确保即使在非常大的拓扑结构中,信息在下一个 UTC 日的中午之前被传送给所有从时钟。这避免混淆 UTC 所指跳变在哪个午夜完成。

注 2: 使用经认可的标准时间源的最高级时钟应分发 PTP 时标。

9.5 报文处理语义

9.5.1 普通时钟和边界时钟的报文传输行为

9.5 规定了在普通时钟或边界时钟的每个 PTP 端口,可能发生的以下事件的行为:

- 报文的接收;
- 报文的发送。

由这些事件产生的行为可取决于普通时钟或边界时钟 PTP 端口的当前状态,见 9.2.5。

除管理报文外,所有多播 PTP 报文应在边界时钟或普通时钟终止。

同时是网桥或路由器的边界时钟,根据网络转发规则转发所有单播 PTP 报文。任何其他行为不在本标准范围内。

除 9.5 规定外,协议的执行不应导致通信路径上的任何通信。

由于任何事件或合格事件序列的发生,而导致本地时钟的任何状态或数据的改变,或数据集的更新都是原子性的,见 3.1.2。合格事件序列被定义为在相同 announceInterval 内接收到多个 Announce 报文(见 7.7.2.2)。

状态或数据改变,或一个事件发生而导致发送一个 PTP 报文的请求,应按先进先出顺序(FIFO)由报文类处理。应为事件和通用报文维护一个逻辑独立的 FIFO 排序。一旦报文请求进入队列,无论发生什么后续事件,都应发送该报文。

只有当 PTP 报文头(见 13.3.2.5)中的 domainNumber 字段与 defaultDS. domainNumber 一致时,PTP 报文才被协议接受并进行处理。

9.5.2 时钟对任何自己发出报文的接收

9.5.2.1 概述

精确时间协议不使用时钟自己发出并接收到的报文,并且实现应防止协议处理这样的报文。

9.5.2.2 和 9.5.2.3 规定了这样的报文被接收时应如何处理。

9.5.2.2 普通时钟和边界时钟的任何单个端口

在一个端口上发出的又被同一端口接收到的报文应被忽略。由于实现特定的诊断目的可能存在特例,这超出本标准范围。

这种情况的识别可通过将接收报文的 sourcePortIdentity 字段成员同进入端口 portDS. portIdentity 的相应成员进行比较来实现。可能性和解释见表 17。

注: 由于通信路径的正常或非正常特性可发生本情况。

9.5.2.3 边界时钟的附加限制

边界时钟上的端口“N”可能直接接收相同边界时钟的其他端口“M”发出的 Announce 报文。如果端口 N 和 M 在相同的通信路径通信,则可发生该情况。这是最佳主时钟算法不能检测的非正常情况。当该情况在一组端口被检测出时,除具有最小 portNumber 端口(假定 N)外,边界时钟应将涉及的其他所有端口状态都置为 PASSIVE 状态,直到协议正常运行不再判定端口 N 处于 MASTER 状态为止。

这种情况的识别可通过将接收报文的 sourcePortIdentity 字段成员同进入端口 portDS.portIdentity 的相应成员进行比较来实现。可能性和解释见表 17。

表 17 源标识符比较

假设报文 m 到达时钟 c 的端口 n, n 的 portDS.portIdentity 字段具有 clockIdentity 为 a 和 portNumber 为 n 的成员:	
如果 m 的 sourcePortIdentity 包含:	解释为:
clockIdentity a, portNumber n	m 是由时钟 c 的端口 n 发出的
clockIdentity f≠a, portNumber n	m 是由其他时钟(不是时钟 c)的端口发出的
clockIdentity a, portNumber q≠n	m 是由时钟 c 发出的,但不是从 n 端口

9.5.3 来自其他时钟的 Announce 报文的接收

处理 Announce 报文的逻辑见图 29。图中表示的状态是指接收 Announce 报文的端口的当前状态。

如果端口在 INITIALIZING 或 DISABLED 状态接收 Announce 报文,则报文应被丢弃。如果端口在 FAULTY 状态接收 Announce 报文,则除出于实现特定目的以满足 9.2 要求外,报文应被丢弃。

如果接收到的 Announce 报文 sourcePortIdentity 字段的成员与接收时钟的数据集的 parentDS.parentPortIdentity 相应成员相同,则报文来自当前父时钟,即该时钟同步到的主时钟。

当来自端口 F 的 Announce 报文被端口 N 在与 N 相关的通信路径上接收时,如果任何以下条件为 TRUE,则端口 F 被指定为外部主时钟:

- 端口 N 不处于 SLAVE 状态;
- 端口 N 处于 SLAVE 状态,并且接收报文的 sourcePortIdentity 字段的成员不完全与接收时钟的数据集的 parentDS.parentPortIdentity 相应成员相同。

如果从外部主时钟接收 Announce 报文,则进入端口的实现特定的 foreignMasterDS 数据集(见 9.3.2.4)应按如下更新:

- 如果接收到的 Announce 报文 sourcePortIdentity 字段的成员与进入端口数据集的 foreignMasterDS.foreignMasterPortIdentity 的相应成员相同,则该记录的 foreignMasterAnnounceMessages 字段应增加。
- 如果接收到的 Announce 报文 sourcePortIdentity 字段的成员与进入端口数据集的 foreignMasterDS.foreignMasterPortIdentity 的相应成员不相同,则在数据集中应创建一个新记录,该记录 foreignMasterDS.foreignMasterPortIdentity 字段被设为接收到的 Announce 报文的 sourcePortIdentity 字段值,并且其 foreignMasterDS.foreignMasterAnnounceMessages 字段值应设为 0。对 foreignMasterDS 数据集容量的实现特定的局限性,限制了这样记录的个数。如果 foreignMasterDS 满,则应丢弃 Announce 报文。

如果从当前父时钟接收一个 Announce 报文,则进入端口的数据集应根据表 16 进行更新,除非每个字段的源都应接收到的 Announce 报文而不是 E_{best}。

注:在 foreignMasterDS 数据集(见 9.3.2.4.5)中可保存的最大记录个数不影响协议运行的正确性,但是可能影响选择主时钟和从时钟的收敛速度。更高容量允许节点在每个 announceInterval 内处理并排除考虑更多外部主时钟,而不必等到数据集中存储来自任何剩余外部主时钟的后续报文的空间可用。

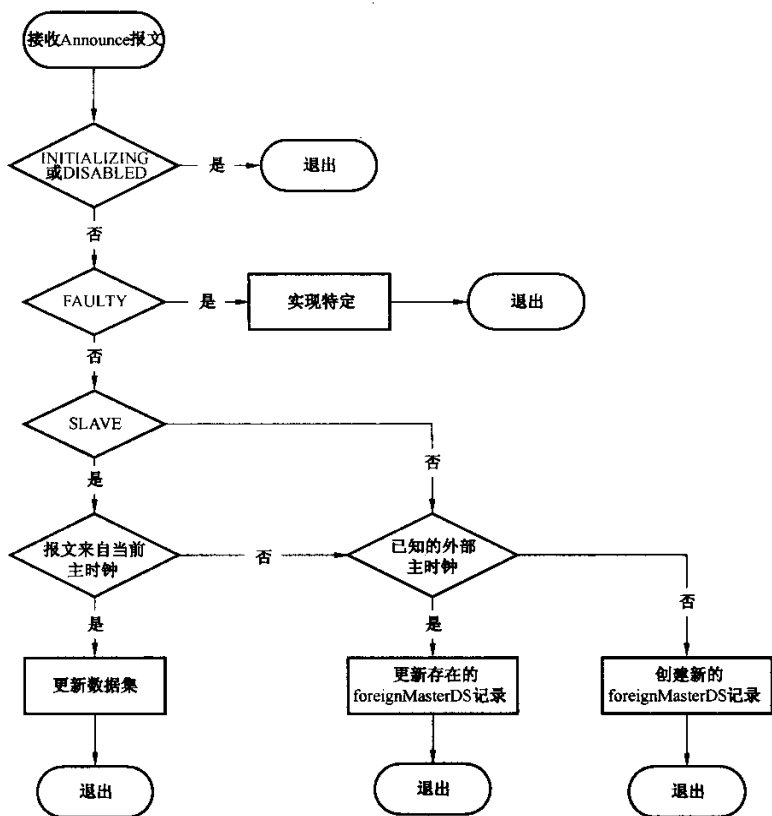


图 29 Announce 报文接收逻辑

9.5.4 来自其他时钟的 Sync 报文的接收

处理 Sync 报文的逻辑见图 30。图中表示的状态是指接收 Sync 报文的端口的当前状态。

如果端口在 INITIALIZING 或 DISABLED 状态接收 Sync 报文，则报文应被丢弃。如果端口在 FAULTY 状态接收 Sync 报文，则除出于实现特定目的以满足 9.2 要求外，报文应被丢弃。

应为 Sync 报文的接收生成满足 7.3.4 要求的〈syncEventIngressTimestamp〉。

接收到的 Sync 报文应在接收后尽快处理。

如果接收到的 Sync 报文 sourcePortIdentity 字段的成员与接收时钟数据集的 parentDS.parentPortIdentity 相应成员相同，则报文来自当前主时钟。

当 Sync 报文被接收并满足以下所有条件时：

- 接收 Sync 报文的端口处于 SLAVE 或 UNCALIBRATED 状态；
- 接收到的 Sync 报文 flagField 的 twoStepFlag 比特为 FALSE（表示将不接收 Follow_Up 报文）；
- Sync 报文来自当前主时钟；

则：

- 应根据 11.6.2 调整 Sync 报文 correctionField 用于校正不对称性；
- 应基于接收到的 Sync 报文内容和〈syncEventIngressTimestamp〉，根据 12.2 将本地时钟同步；

当 Sync 报文被接收并满足以下所有条件时：

- 接收 Sync 报文的端口处于 SLAVE 或 UNCALIBRATED 状态；

- 接收到的 Sync 报文 flagField 的 twoStepFlag 位为 TRUE(表示将接收 Follow_Up 报文);
 - Sync 报文来自当前主时钟;
- 则:
- 应根据 11.6.2 调整 Sync 报文 correctionField 用于校正不对称性。

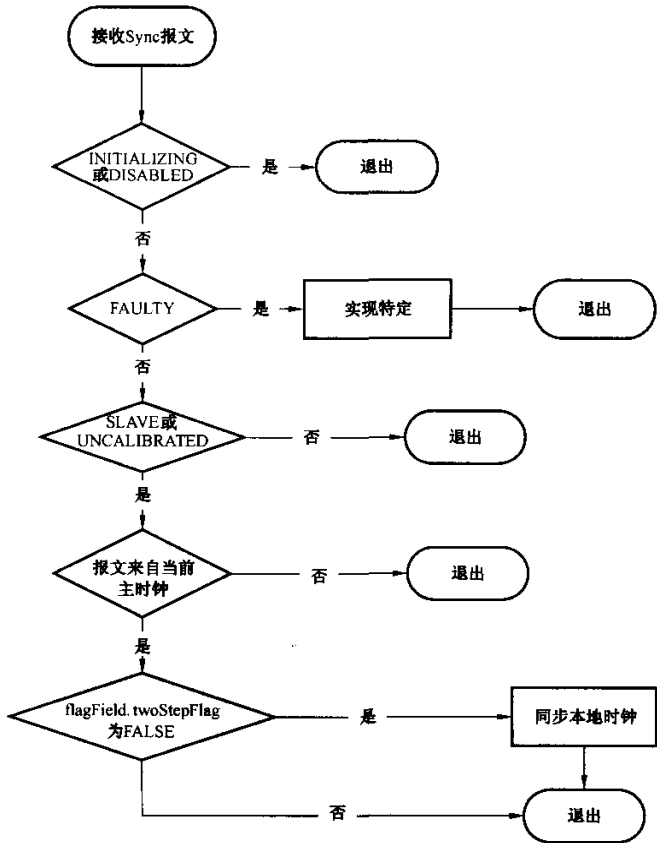


图 30 Sync 报文接收逻辑

9.5.5 来自其他时钟的 Follow_Up 报文的接收

处理 Follow_Up 报文的逻辑见图 31。图中表示的状态是指接收 Follow_Up 报文的端口的当前状态。

如果端口在 INITIALIZING 或 DISABLED 状态接收 Follow_Up 报文,则报文应被丢弃。如果端口在 FAULTY 状态接收 Follow_Up 报文,则除出于实现特定目的以满足 9.2 要求外,报文应被丢弃。

如果接收到的 Follow_Up 报文 sourcePortIdentity 字段的成员与先前 Sync 报文的 sourcePortIdentity 字段的相应成员相同,并且接收到的 Follow_Up 报文的 sequenceId 字段与相同先前的 Sync 报文的 sequenceId 字段相匹配,则该 Follow_Up 报文和该 Sync 报文是相关联的。

如果相关 Sync 报文的 sourcePortIdentity 字段的成员与接收时钟数据集的 parentDS.parentPortIdentity 成员的相应成员相同,则报文是从当前主时钟发出的。

Follow_Up 报文应在接收后尽快处理。

当相关 Sync 报文被接收并满足以下所有条件时:

- 接收 Follow_Up 报文的端口处于 SLAVE 或 UNCALIBRATED 状态;
- Sync 报文来自当前主时钟;

——Follow_Up 报文与该 Sync 报文相关联；

则应根据 12.2,并基于接收到的 Follow_Up 报文、相关 Sync 报文和相关 Sync 报文的〈syncEventIngressTimestamp〉,将本地时钟同步。

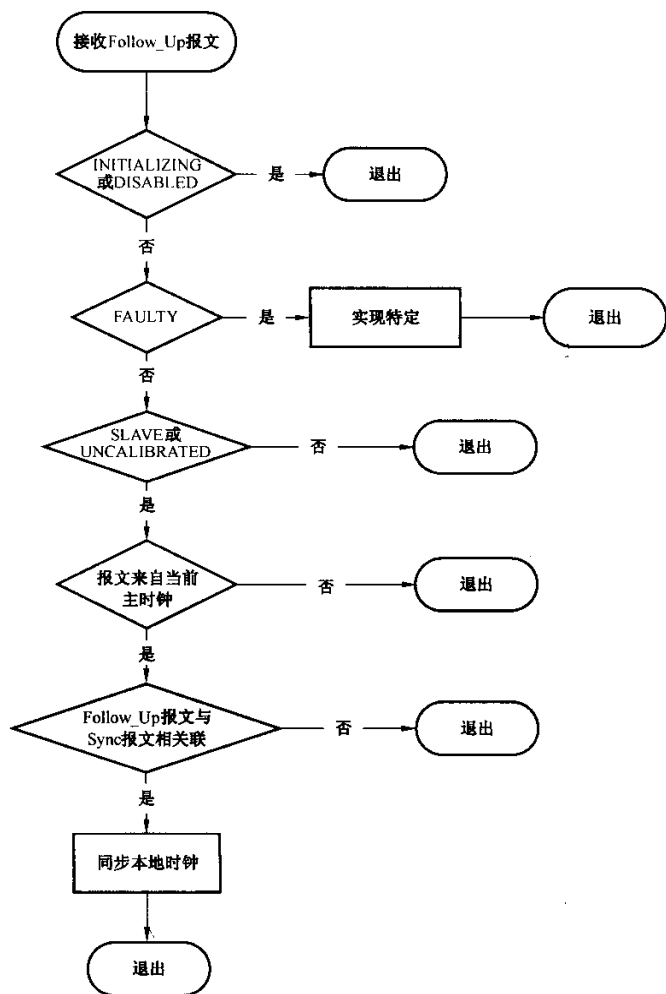


图 31 Follow_Up 报文接收逻辑

9.5.6 来自其他时钟的 Delay_Req 报文的接收

处理 Delay_Req 报文的逻辑见图 32。图中表示的状态是指接收 Delay_Req 报文的端口的当前状态。

如果端口在 INITIALIZING 或 DISABLED 状态接收 Delay_Req 报文,则报文应被丢弃。如果端口在 FAULTY 状态接收 Delay_Req 报文,则除出于实现特定目的以满足 9.2 要求外,报文应被丢弃。

除非本标准另有规定,如果接收 Delay_Req 报文的端口不处于 MASTER 状态,则报文应被丢弃。

当接收 Delay_Req 报文时,应生成满足 7.3.4 要求的〈delayReqEventIngressTimestamp〉。

如果接收 Delay_Req 报文的端口处于 MASTER 状态,该端口应传输符合 11.3 和 9.5.12 条件的 Delay_Resp 报文。

Delay_Req 报文应在接收后尽快处理。

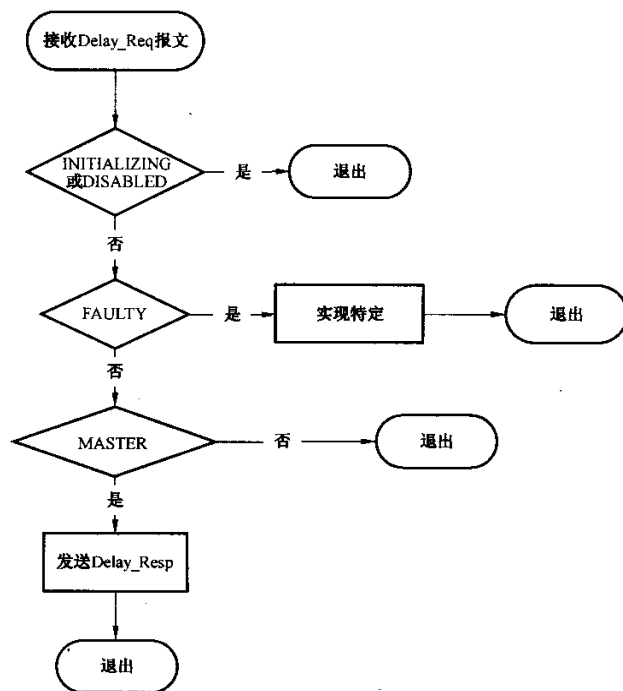


图 32 Delay_Req 报文接收逻辑

9.5.7 来自其他时钟的 Delay_Resp 报文的接收

处理 Delay_Resp 报文的逻辑见图 33。图中表示的状态是指接收 Delay_Resp 报文的端口的当前状态。

如果端口在 INITIALIZING 或 DISABLED 状态接收 Delay_Resp 报文，则报文应被丢弃。如果端口在 FAULTY 状态接收 Delay_Resp 报文，则除出于实现特定目的以满足 9.2 要求外，报文应被丢弃。

如果接收到的 Delay_Resp 报文的 requestingSourcePortIdentity 字段的成员与该接收时钟先前发送的 Delay_Req 报文的 sourcePortIdentity 字段的相应成员相同，并且接收到的 Delay_Resp 报文的 requestingSequenceId 字段与相同先前的 Delay_Req 报文的 sequenceId 字段相匹配，则该 Delay_Resp 报文与该 Delay_Req 报文是相关联的。

如果接收到的 Delay_Resp 报文 sourcePortIdentity 字段的成员与接收时钟数据集的 parentDS.parentPortIdentity 成员的相应成员相同，则 Delay_Resp 报文是从当前主时钟发出的。

Delay_Resp 报文应在接收后尽快处理。

当端口发送相关联的 Delay_Req 报文之后接收 Delay_Resp 报文，并满足以下所有条件：

- 接收 Delay_Resp 报文的端口处于 SLAVE 或 UNCALIBRATED 状态；
- Delay_Resp 来自当前主时钟；
- Delay_Resp 报文与发送的 Delay_Req 报文相关联，

则接收时钟应：

- a) 基于接收到的 Delay_Resp 报文和相关 Delay_Req 报文的内容，执行 11.3 要求的延时请求-响应机制动作；
- b) 将数据集 portDS.logMinDelayReqInterval 成员的值更新为 Delay_Resp 报文的 logMessageInterval 成员的值。

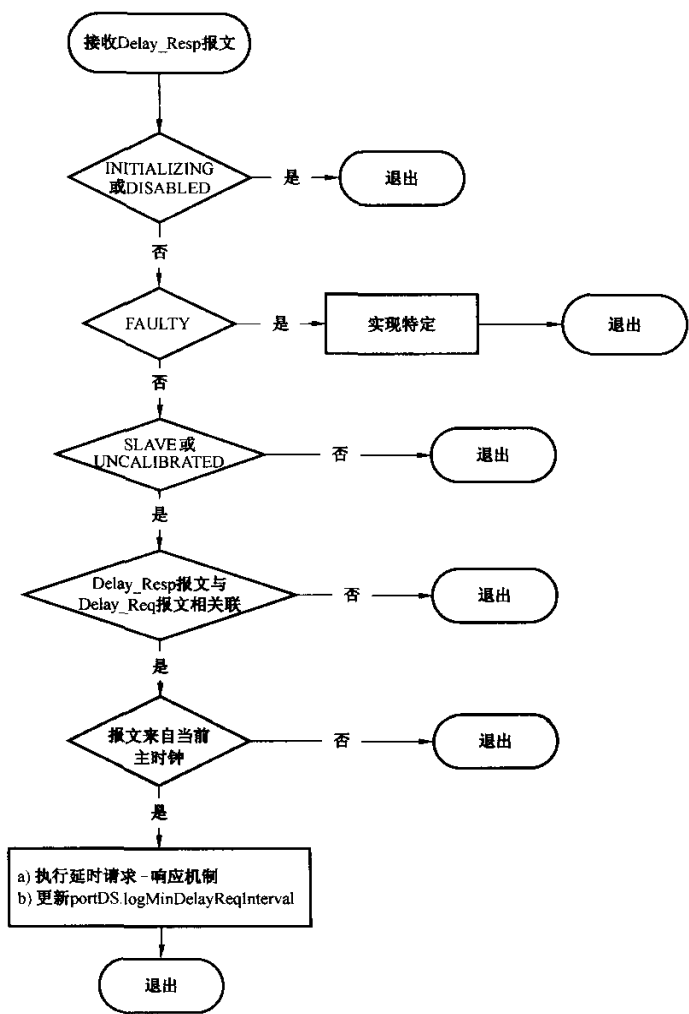


图 33 Delay_Resp 报文接收逻辑

9.5.8 Announce 报文的发送

除非本标准另有规定，除本条要求外，端口不应发送 Announce 报文。

处于 MASTER 状态的端口应周期性发送 Announce 报文。

Announce 报文应通过多播(见 7.3.1)发送，以 2 为底报文传输间隔(单位为秒(s))的平均值的对数，等于发送时钟数据集 portDS.logAnnounceInterval 的值。一个节点应以 90%置信度，在由 portDS.logAnnounceInterval 计算出的时间间隔的±30%内发出报文。

根据第 16 章，可选地，Announce 报文可通过单播传输，在协商好的传输间隔发送。

注：用于为每个单播地址保存该数据的节点容量，可能限制协商的单播合约个数(按照第 16 章)。在一些环境中，多播传输可能是不可行的。只要保持协议的运行，允许使用单播传输，见 7.3.1。

9.5.9 Sync 报文的发送

9.5.9.1 通用规范

除非本标准另有说明，除 9.5.9.2~9.5.9.4 要求外，端口不应发送 Sync 报文。

9.5.9.2 通用要求

处于 MASTER 状态的端口应周期性发送 Sync 报文。Sync 报文应通过多播(见 7.3.1)发送，以 2

为底的报文传输间隔(单位为秒)平均值的对数,等于发送时钟数据集 portDS.logSyncInterval 的值。一个节点应在由 portDS.logSyncInterval 计算出的时间间隔的 $\pm 30\%$ 内发出报文,置信度为 90%。

根据第 16 章,可选地,Sync 报文可通过单播传输,在协商好的传输间隔发送。

注:用于为每个单播地址保存该数据的节点容量,可能限制协商的单播合约个数(按照第 16 章)。在一些环境中,多播传输可能是不可行的。只要保持协议的运行,允许使用单播传输,见 7.3.1。

发送的 Sync 报文字段应符合第 13 章要求。

当发送 Sync 报文时,应生成满足 7.3.4 要求的(syncEventEgressTimestamp)。

9.5.9.3 单步时钟

Sync 报文的 originTimestamp 字段应是(syncEventEgressTimestamp)除去任何小数纳秒部分的估计值,误差在 ± 1 s 内。

Sync 报文的 originTimestamp 字段应为除去任何小数 ns 部分的(syncEventEgressTimestamp)。Sync 报文 correctionField 和 originTimestamp 字段的和,应为包括任何小数 ns 部分的(syncEventEgressTimestamp)的值。

9.5.9.4 双步时钟

Sync 报文 originTimestamp 字段应为 0,或误差在 ± 1 s 内的(syncEventEgressTimestamp)的估计值。

Sync 报文的 correctionField 应设为 0。

双步时钟应既发送 Sync 报文,又发送 Follow_Up 报文。

端口应将 Sync 报文的 sequenceId 值作为 Follow_Up 报文的 sequenceId 字段的输入。获得相关联 Follow_Up 报文的 preciseOriginTimestamp 和 correctionField 字段值的机制应被启动。

9.5.10 Follow_Up 报文的发送

除非本标准另有说明,仅当 9.5.9.4 要求时,端口才发出 Follow_Up 报文。要求发送 Follow_Up 报文的 Sync 报文是相关联的 Sync 报文。

相关联 Sync 报文发送后应尽快发送 Follow_Up 报文,并且应在向相同目的地址发送后续 Sync 报文之前完成发送。

Follow_Up 报文 sequenceId 字段应为相关联 Sync 报文 sequenceId 字段的值。

Follow_Up 报文的 preciseOriginTimestamp 字段应为相关联 Sync 报文的(syncEventEgressTimestamp)除去任何小数 ns 部分的估计值,误差在 ± 1 s 内。

Follow_Up 报文的 preciseOriginTimestamp 字段应为相关联 Sync 报文的除去任何小数 ns 部分的(syncEventEgressTimestamp)的估计值。将 Follow_Up 报文的 correctionField 与相关 Sync 报文的 correctionField 的和,加入到 Follow_Up 报文的 preciseOriginTimestamp 字段,应为相关 Sync 报文的包括任何小数 ns 部分的(syncEventEgressTimestamp)的精确值。

根据 9.5.9.2,如果 Follow_Up 报文与一个可选单播 Sync 报文相关联,则 Follow_Up 报文也应作为单播报文,被发送到与相关联 Sync 报文相同的单播地址。这些单播 Follow_Up 报文应满足本条的所有其他要求。

9.5.11 Delay_Req 报文的发送

9.5.11.1 通用要求

只有当满足以下所有条件时,时钟才应在一个端口发出 Delay_Req 报文:

- 端口处于 SLAVE 或 UNCALIBRATED 状态;
- 配置设备执行延时请求-响应机制,见 8.2.5.4.4;
- 允许设备按照 9.5.11.2 的时间要求进行这种操作。

Delay_Req 报文通过广播发送,除非:

- a) 使用第 16 章的可选单播内容;

b) 或者行规规定。

本 Delay_Req 报文字段应符合 11.3 要求。

注 1: 如果时钟使用的 PTP 行规规定了仅谐振 (syntonize-only), 则不要求时钟发送 Delay_Req 报文。在该情况下, 不要求维护延时请求状态信息。

注 2: 对于使用对等延时机制的时钟, 见 9.5.13。

9.5.11.2 时间要求

在一个端口发送 Delay_Req 报文应有如下限制:

——当要求时, 可发送初始 Delay_Req 报文。

——后续 Delay_Req 报文的发送, 应使得报文发送间隔 (单位为 s) 的平均值不小于 $2^{\text{portDS.logMinDelayReqInterval}}$ s, 具有 90% 置信度。

注: portDS.logMinDelayReqInterval 的值为最近接收到的 Delay_Resp 报文的 logMessageInterval 字段, 该 Delay_Resp 报文是对端口发出的 Delay_Req 报文的响应。

当满足 Delay_Req 报文平均发送间隔规范, Delay_Req 报文的发送时间应使用以下时间选项之一:

——发送时间的选择应使得连续 Delay_Req 报文的时间间隔按照随机分布。除非在适用 PTP 行规中另有规定, 随机分布应为 $0 \sim 2^{\text{logMinDelayReqInterval}+1}$ 秒之间的均匀随机分布。应为每个发送的报文计算一个新发送间隔的随机值。该分布的粒度是实现特定的, 但应不大于 $2^{\text{logSyncInterval}-4}$ s。当采用多播通信模型时应使用该选项, 单播模型也可使用, 见 7.3.1。例如, 对于 $\text{logMinDelayReqInterval} = \text{logSyncInterval}$ (见 7.7.2.4), 分布的最小值为 0, 最大值为 $\{2^{\text{logSyncInterval}+1}\} = 2 \times \text{syncInterval}$; 即分布的最大值为 2 倍 syncInterval, 由此而得到的平均发送间隔为 1 倍 syncInterval。

——当接收 Sync 报文后, Delay_Req 报文应尽快发送。单播模型 (见 7.3.1) 可使用该选项。当采用多播模型时, 除非在适用 PTP 行规中另有规定, 都不应使用改选项。

9.5.12 Delay_Resp 报文的发送

除非本标准中另有规定, 当端口与 Delay_Req 报文的接收相关联并满足 9.5.11 其他要求时, 端口应发出 Delay_Resp 报文。Delay_Req 报文的接收应满足 9.5.6。

要求发送 Delay_Resp 报文的 Delay_Req 报文是相关联的 Delay_Req 报文。

当满足以下所有条件时, 时钟应在一个端口上发送 Delay_Resp 报文:

——端口处于 MASTER 状态, 或者可选单播合约 (见第 16 章) 要求;

——配置设备执行延时请求-响应机制, 见 8.2.5.4.4;

——发送是由相关联 Delay_Req 报文的接收而引发的, 见 9.5.6。

如果相关联 Delay_Rep 报文是通过多播发送的, 则 Delay_Resp 报文也应通过多播发送。

如果相关联 Delay_Rep 报文是通过单播发送的, 则 Delay_Resp 报文也应通过单播发送。

本 Delay_Resp 报文字段应符合 11.3 要求。

Delay_Resp 报文的 receiveTimestamp 字段应为相关联 Delay_Rep 报文的 (delayReqEventIngress-Timestamp)。

在发送 Delay_Resp 报文之前, 按顺序端口应:

a) 为 logMinDelayReqInterval 字段计算更新值, 见 7.7.2.4;

b) 使用该计算值更新发送 Delay_Resp 报文的端口的数据集 portDS.logMinDelayReqInterval 成员的值, 见 8.2.5.3.2;

c) 按照 11.3.2 规定, 确保报文字段的值为当前值;

d) 将表 24 中规定的值插入到报文的 logMessageInterval 字段。

接收相关联 Delay_Rep 报文后, 应尽快发送 Delay_Resp 报文。

9.5.13 Pdelay_Req 报文的发送

9.5.13.1 通用要求

只有当满足以下所有条件时,时钟在端口上发出 Pdelay_Req 报文:

- 配置设备执行对等延时机制,见 8.2.5.4.4;
- 允许设备按照 9.5.13.2 的时间要求进行操作。

本 Pdelay_Req 报文的字段应符合 11.4.3 要求。

注:如果时钟使用的 PTP 行规定了仅谐振(synthesize-only),则不要求时钟启动对等延时机制或延时请求-响应机制。在该情况下,不要求维护对等延时机制状态信息。

9.5.13.2 时间要求

从请求端口发送 Pdelay_Req 报文应遵循如下限制:

- 如要求,可发送初始的 Pdelay_Req 报文;
- 后续 Pdelay_Req 报文的发送,应使得以 2 为底的报文发送间隔(单位为 s)的平均值的对数,不小于由发送时钟数据集 portDS.logMinPdelayReqInterval 成员的值计算出的时间间隔。

9.5.14 Pdelay_Resp 报文的发送

接收相关联的 Pdelay_Req 报文后应尽快发送 Pdelay_Resp 报文。

9.5.15 Pdelay_Resp_Follow_Up 报文的发送

发送相关联的 Pdelay_Resp 报文后应尽快发送 Pdelay_Resp_Follow_Up 报文。

9.6 本地时钟的改变

由于 1)本地时钟内部特性的变化(如本地振荡器干扰);或 2)同 PTP 外部交互的变化,如从与本地时钟直接同步的 GPS 接收器接收到的信息变化,可导致数据集的更新。

所有导致的数据集变化,对于任何其他数据集访问动作而言,包括 9.2.6.8 规定的动作,都应是原子性的。

10 透明时钟的 PTP

10.1 端到端和点到点透明时钟的通用要求

所有透明时钟都应按照网络寻址规则转发所有的非 PTP 报文。

对于所有透明时钟,PTP 版本 1 的所有报文:

- 应按照网络寻址规则被转发;
- 透明时钟不应应对 PTP 版本 1 的报文进行任何驻留时间或路径延时的校正;
- 不禁止对 PTP 版本 1 报文的驻留时间或路径延时的校正。这超出本标准范围。

所有透明时钟应根据第 12 章谐振到最高级时钟。

应支持到多个域的谐振。

对于实现谐振的时钟,术语“主谐振域”被定义为 domainNumber 0,或者如果缺省为配置到新域,则被定义为配置的缺省域。

如果透明时钟实现了谐振,应按照以下动作:

- a) 缺省地,它应谐振到主谐振域的主时钟;
- b) 主谐振域可被配置为非 domainNumber 0 的域;
- c) 如果实现了谐振到多个域,应维护到每个域的主时钟的谐振;
- d) 透明时钟对事件报文进行的所有驻留时间和链路延时的校正,都应基于驻留时间和链路延时测量的先后顺序:
 - 1) 由谐振到相同域的其他时钟完成测量,域通过进入事件报文的 domainNumber 字段指示;
 - 2) 由谐振到主谐振域的时钟完成测量。

如果透明时钟未实现谐振,由透明时钟进行的对事件报文的所有驻留时间和链路延时的校正,都应基于非谐振自由运行的时钟对驻留时间和链路延时的测量。

10.2 端到端透明时钟要求

所有 PTP 版本 2 的报文都应按照网络寻址规则被转发。

对所有 PTP 版本 2 的事件报文和任何受影响的通用报文的处理,应满足 11.5 的驻留时间校正要求。

端到端透明时钟不应实现 11.4 的对等延时机制。

10.3 点到点透明时钟要求

应实现 11.4 的对等延时机制。

所有 PTP 版本 2 的 Announce、Sync、Follow_Up、Management 和信号报文,应按照网络寻址规则被转发。

对所有 PTP 版本 2 的 Sync 和 Follow_Up 报文的处理,应满足 11.5 的驻留时间校正要求和 11.4.5 的路径延时校正要求。

所有 PTP 版本 2 的 Delay_Req 和 Delay_Resp 报文都应被丢弃。

11 时钟偏移,路径延时,驻留时间和不对称校正

11.1 通用规范

第 11 章规范中给出的机制用于传输事件报文源产生的时间戳和任何所需的校正,以保证事件报文接收方接收到可能的最精确的时间戳。originTimestamp 或 preciseOriginTimestamp 与 correctionField 字段之间的时间信息的实际分发是实现相关的,假设分发应为:执行时间戳字段和 correctionField 计算的接收设备(见下面各章条的规定)获得可能最精确的时间戳。

第 11 章规定了:

- 主从时钟之间的时间偏移计算,即<offsetFromMaster>;
- 延时请求-响应机制;该机制测量一对端口间的<meanPathDelay>,各端口都支持 9.2.5 的状态机;
- 对等延时机制;该机制测量一对端口间的<meanPathDelay>,各端口都支持对等延时机制;
- 对点到点透明时钟中的<meanPathDelay>的校正;
- 基于透明时钟内驻留时间的测量,对透明时钟的事件报文的校正;和
- 路径不对称时间戳的校正。

注 1: 推荐时间戳自身作为时间的最佳可能估计,使得仅需近似时间的简单设备忽略 correctionField。

注 2: 时间戳和 correctionField 间的时间分发范围允许灵活的设备设计。例如,当组合好报文时,一台设备可产生近似的时间,并将产生的所需校正插入到合适的 correctionField。一个不可避免的情况为小数纳秒的表示。小数纳秒不能表示为 Timestamp 数据类型。它在 correctionField 中进行传输,由接收设备综合两部分来得到实际的时间戳。

这些校正机制的示例见附录 C。

11.2 普通时钟和边界时钟的时钟偏移的计算

从时钟同主时钟(边界时钟或者普通时钟作为主时钟)间的时间误差定义如下:

$\langle \text{offsetFromMaster} \rangle = \langle \text{从时钟时间} \rangle - \langle \text{主时钟时间} \rangle$,

其中所有时间是在同一时刻测量的。

$\langle \text{offsetFromMaster} \rangle$ 的值应由从时钟按如下步骤计算:

- 当接收到 Sync 报文时,从时钟应按照 7.3.4 生成一个用于校正内部延时的时间戳<syncEventIngressTimestamp>。如果已知连接进入端口路径的 delayAsymmetry(见 7.4.2),则应进行 11.6 的校正;

b) 如果 Sync 报文的 flagField 的 twoStepFlag 比特为 FALSE,表明将不接收 Follow_Up 报文,则:

$$\langle \text{offsetFromMaster} \rangle = \langle \text{syncEventIngressTimestamp} \rangle - \langle \text{originTimestamp} \rangle - \langle \text{meanPathDelay} \rangle - \text{Sync 报文的 correctionField};$$

c) 如果 Sync 报文的 flagField 的 twoStepFlag 为 TRUE,表明将接收 Follow_Up 报文,则:

$$\langle \text{offsetFromMaster} \rangle = \langle \text{syncEventIngressTimestamp} \rangle - \langle \text{preciseOriginTimestamp} \rangle - \langle \text{meanPathDelay} \rangle - \text{Sync 报文的 correctionField} - \text{Follow_Up 报文的 correctionField}$$

其中:

- $\langle \text{originTimestamp} \rangle$ 应为接收到的 Sync 报文中 originTimestamp 字段的值;
- $\langle \text{preciseOriginTimestamp} \rangle$ 应为接收到的 Follow_Up 报文中 preciseOriginTimestamp 字段的值;
- 如果端口被配置为使用延时请求-响应机制,则 $\langle \text{meanPathDelay} \rangle$ 应按 11.3 规定;
- 如果端口被配置为使用对等延时机制,则 $\langle \text{meanPathDelay} \rangle$ 应按 11.4 规定。

11.3 延时请求-响应机制

11.3.1 延时请求-响应机制通用要求

延时请求-响应机制测量一对 PTP 端口间的 $\langle \text{meanPathDelay} \rangle$,其每个端口都支持 9.2.5 的状态机。延时请求-响应机制使用的报文有 Sync、Delay_Req、Delay_Resp 和可能的 Follow_Up,如图 34 的时序图所示。该机制应在两个时钟的各支持域中独立执行。

图 34 中,Sync 报文的时间戳 t_1 和 t_2 ,以及 Delay_Req 报文的时间戳 t_3 和 t_4 ,应按照 7.3.4.2 定义的方法进行测量。时间戳 t_1 和 t_4 应使用主时钟节点时间进行测量,时间戳 t_2 和 t_3 使用从时钟节点时间进行测量。

如果已知连接进入端口和离开端口的路径的 delayAsymmetry(见 7.4.2),应进行 11.6 的校正。

注: $\langle \text{meanPathDelay} \rangle$ 的标称值可按照下式计算, $\langle \text{meanPathDelay} \rangle = [(t_2 - t_1) + (t_4 - t_3)]/2 = [(t_2 - t_3) + (t_4 - t_1)]/2$ 。

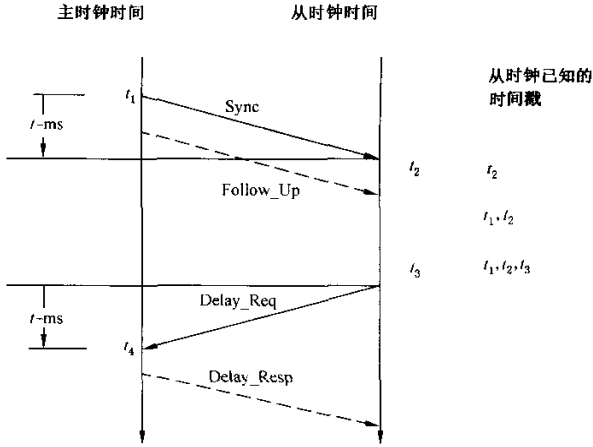


图 34 延时请求-响应路径长度的测量

11.3.2 延时请求-响应机制操作规范

对于每个延时请求-响应测量的实例, $\langle \text{meanPathDelay} \rangle$ 的实际值应按如下进行测量和计算:

- a) 主时钟节点按照 9.5.9 准备并发出一个 Sync 报文。如果节点为双步时钟,还按照 9.5.9.4 准备并发出一个 Follow_Up 报文。
- b) 从时钟节点应:
 - 1) 当接收到主时钟的 Sync 报文,生成时间戳 t_2 ;

- 2) 如果要求不对称校正,则按 11.6.2 修改接收到的 Sync 报文的 correctionField;
- 3) 如果要求基于 9.5.11.2 的时间要求来发送 Delay_Req 报文:
 - i) 准备一个 Delay_Req 报文,并将其 correctionField (见 13.3.2.7) 设为 0。originTimestamp 应设为 0,或者设为 Delay_Req 报文离开时间的估计值,其误差不大于 ± 1 s;
 - ii) 如果要求不对称校正,则按 11.6.3 修改 correctionField;
 - iii) 发送 Delay_Req 报文,生成并保存时间戳 t_3 。
- c) 当接收到 Delay_Req 报文,主时钟节点应:
 - 1) 生成时间戳 t_4 ;
 - 2) 准备 Delay_Resp 报文;
 - 3) 从 Delay_Req 报文复制 sequenceId 字段到 Delay_Resp 报文的 sequenceId 字段;
 - 4) 从 Delay_Req 报文复制 sourcePortIdentity 字段到 Delay_Resp 报文的 sourcePortIdentity 字段;
 - 5) 从 Delay_Req 报文复制 domainNumber 字段到 Delay_Resp 报文的 domainNumber 字段;
 - 6) 将 Delay_Resp 报文的 correctionField 设为 0;
 - 7) 将 Delay_Req 报文的 correctionField 加到 Delay_Resp 报文的 correctionField;
 - 8) 将 Delay_Req 报文的 receiveTimestamp 字段设为时间 t_4 的秒和纳秒部分(s 和 ns);
 - 9) 从 Delay_Req 报文的 correctionField 中减去 t_4 的小数纳秒(ns)部分;
 - 10) 发出 Delay_Req 报文。
- d) 当从时钟接收到 Delay_Resp 报文时:
 - 1) 如果接收到的 Sync 报文指示将不接收 Follow_Up 报文,则 $\langle \text{meanPathDelay} \rangle$ 的计算如下:

$$\langle \text{meanPathDelay} \rangle = [(t_2 - t_3) + (\text{Delay_Resp 报文的 receiveTimestamp} - \text{Sync 报文的 originTimestamp}) - \text{Sync 报文的 correctionField} - \text{Delay_Resp 报文的 correctionField}] / 2。$$
 - 2) 如果接收到的 Sync 报文指示将接收 Follow_Up 报文,则 $\langle \text{meanPathDelay} \rangle$ 的计算如下:

$$\langle \text{meanPathDelay} \rangle = [(t_2 - t_3) + (\text{Delay_Resp 报文的 receiveTimestamp} - \text{Follow_Up 报文的 preciseOriginTimestamp}) - \text{Sync 报文的 correctionField} - \text{Follow_Up 报文的 correctionField} - \text{Delay_Resp 报文的 correctionField}] / 2。$$

注:延时请求-响应路径长度测量通常使用 Delay_Req 报文前最近一次的 Sync 报文与相应的 Follow_Up 报文的时间戳和 correctionField。然而,延时请求-响应测量可使用任何的 Sync 报文和相应的 Follow_Up 报文,尽管这将降低计算出的 $\langle \text{meanPathDelay} \rangle$ 的精度。

11.4 对等延时机制

11.4.1 对等延时机制通用要求

对等延时机制测量端口到端口的传播时间,即两个支持对等延时机制的通信端口间的链路延时。

应在设备的所有端口中执行该测量,包括那些被较低层协议所阻塞(blocked)的端口。支持这一要求的寻址或其他机制是网络特定的,并在本标准的相关附录中进行规定。

该链路延时测量应由每个实现对等机制的端口独立地进行。

注:此要求意味着链路两端的端口已知链路延时。它允许网络重新配置时立即校正路径长度。

在普通时钟和边界时钟里,对等延时机制应与各端口是主时钟或从时钟无关。

对等延时机制使用的报文有:Pdelay_Req、Pdelay_Resp,和可能的 Pdelay_Resp_Follow_Up,如图 35 的时序图所示。

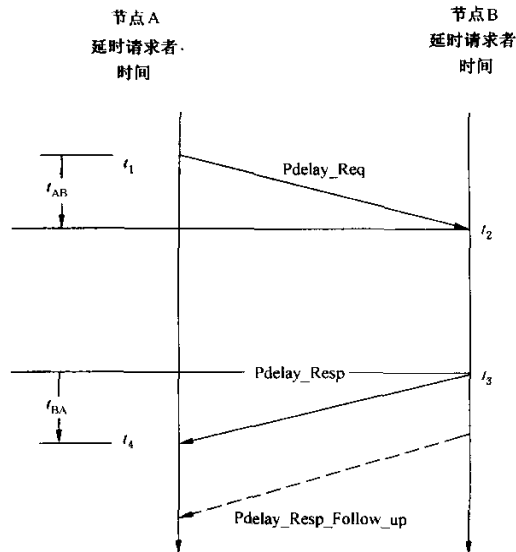


图 35 对等延时链路测量

在图 35 中, Pdelay_Req 报文的时间戳 t_1 和 t_2 , 以及 Pdelay_Resp 报文的时间戳 t_3 和 t_4 按照 7.3.4 的定义进行测量。如果已知连接进入端口和离开端口的链路的 delayAsymmetry(见 7.4.2), 应进行 11.6 的校正。

时间戳 t_1 和 t_4 由节点 A 按如下步骤测量:

- 如果节点 A 为点到点透明时钟, 则所用时标规定见 10.1;
- 如果节点 A 为普通时钟或边界时钟, 则所用时标为时钟域的时标。

时间戳 t_2 和 t_3 由节点 B 按如下步骤测量:

- 如果节点 B 为点到点透明时钟, 则所用时标规定见 10.1;
- 如果节点 B 为普通时钟或边界时钟, 则所用时标为时钟域的时标。

注: $\langle \text{meanPathDelay} \rangle$ 的标称值的计算为: $\langle \text{meanPathDelay} \rangle = [(t_2 - t_1) + (t_4 - t_3)] / 2 = [(t_2 - t_3) + (t_4 - t_1)] / 2$ 。实际值的规定见 11.4.3。

11.4.2 对等延时报文时序

Pdelay_Req 报文从请求端口的发送应有如下限制:

- 当要求时发送初始 Pdelay_Req 报文;
- 应发送后续的 Pdelay_Req 报文, 使得以 2 为底报文传输间隔(以秒(s)为单位)的平均值的对数, 不小于请求时钟数据集的成员 portDS.logMinPdelayReqInterval 的值。

在接收到相关联的 Pdelay_Req 报文后, 应尽快发送 Pdelay_Resp 报文。

在发送相关联的 Pdelay_Resp 报文后, 应尽快发送 Pdelay_Resp_Follow_Up 报文。

11.4.3 对等延时机制操作规范

对于每个对等延时请求-响应测量的实例, meanPathDelay 的实际值应按如下进行测量和计算:

a) 延时请求方, 节点 A:

- 1) 准备 Pdelay_Req 报文, correctionField(见 13.3.2.7)应设为 0;
- i) 如果节点 A 是普通时钟或边界时钟, 则报文头的 domainNumber 字段应设为节点 A

的域。

- ii) 如果节点 A 是谐振的点到点透明时钟,则报文头的 domainNumber 字段应设为正被测量的域,即主谐振域;或者如果节点 A 实现了谐振到多个域,则为备选域之一。
- iii) 如果节点 A 不是谐振的点到点透明时钟,则报文头的 domainNumber 字段应设为 0。
- 2) 如果要求不对称校正,则应按 11.6.4 修改 correctionField。
- 3) originTimestamp 应设为 0,或者设为 Pdelay_Req 报文的离开时间戳 t_1 的估计值,其误差不大于 ± 1 s。
- 4) 应发送 Pdelay_Req 报文,生成并保存时间戳 t_1 。
- b) 如果延时响应方节点 B 是单步时钟,它应:
 - 1) 当接收到 Pdelay_Req 报文时,生成时间戳 t_2 ;
 - 2) 准备 Pdelay_Resp 报文;
 - 3) 从 Pdelay_Req 报文复制 sequenceId 字段到 Pdelay_Resp 报文的 sequenceId 字段;
 - 4) 从 Pdelay_Req 报文复制 sourcePortIdentity 字段到 Pdelay_Resp 报文的 requestingPortIdentity 字段;
 - 5) 从 Pdelay_Req 报文复制 domainNumber 字段到 Pdelay_Resp 报文的 domainNumber 字段;
 - 6) 从 Pdelay_Req 报文复制 correctionField 到 Pdelay_Resp 报文的 correctionField;
 - 7) 然后:
 - i) 将 Pdelay_Resp 报文的 requestReceiptTimestamp 字段设为 0;
 - ii) 发出 Pdelay_Resp 报文,并在发送时生成时间戳 t_3 ;
 - iii) 在生成 t_3 后,当 Pdelay_Resp 报文即将离开响应方时,应将周转时间 $t_3 - t_2$ 加到 Pdelay_Resp 报文的 correctionField,并且对校验和或 Pdelay_Resp 报文的其他内容相关字段做任何需要的校正。

注:如需要并且响应方支持该精度,则 correctionField 的数据类型允许将时间间隔 $t_3 - t_2$ 表示到小数纳秒(ns)。

- c) 如果延时响应方是双步时钟,它应:
 - 1) 当接收到 Pdelay_Req 报文时,生成时间戳 t_2 ;
 - 2) 准备 Pdelay_Resp 报文和 Pdelay_Resp_Follow_Up 报文;
 - 3) 从 Pdelay_Req 报文复制 correctionField 到 Pdelay_Resp_Follow_Up 报文的 correctionField,并将 Pdelay_Resp 报文的 correctionField 设为 0;
 - 4) 从 Pdelay_Req 报文复制 sequenceId 字段到 Pdelay_Resp 报文和 Pdelay_Resp_Follow_Up 报文的 sequenceId 字段;
 - 5) 从 Pdelay_Req 报文复制 sourcePortIdentity 字段到 Pdelay_Resp 报文和 Pdelay_Resp_Follow_Up 报文的 requestingPortIdentity 字段;
 - 6) 从 Pdelay_Req 报文复制 domainNumber 字段到 Pdelay_Resp 报文和 Pdelay_Resp_Follow_Up 报文的 domainNumber 字段;
 - 7) 然后:
 - i) 将 Pdelay_Resp 报文的 requestReceiptTimestamp 字段设为 0;
 - ii) 发出 Pdelay_Resp 报文,并在发送时生成时间戳 t_3 ;
 - iii) 在 Pdelay_Resp_Follow_Up 报文中,将 responseOriginTimestamp 字段设为 0,并将周转时间 $t_3 - t_2$ 加到 correctionField;
 - iv) 发出 Pdelay_Resp_Follow_Up 报文;

8) 或者:

- i) 在 Pdelay_Resp 报文中,将 requestReceiptTimestamp 字段设为时间 t_2 的秒和纳秒(s和ns)部分,并从 correctionField 中减去 t_2 的任何小数纳秒(ns)部分;
- ii) 发出 Pdelay_Resp 报文,并在发送时生成时间戳 t_3 ;
- iii) 在 Pdelay_Resp_Follow_Up 报文中,将 responseOriginTimestamp 字段设为 t_3 的秒(s)和纳秒(ns)部分,并将 t_3 的任何小数纳秒(ns)部分加到 correctionField;
- iv) 发出 Pdelay_Resp_Follow_Up 报文。

d) 延时请求方,节点 A,应:

- 1) 当接收到 Pdelay_Resp 报文时,生成时间戳 t_4 ;
- 2) 如果要求不对称校正,则按 11.6.5 修改 Pdelay_Resp 报文的 correctionField;
- 3) 如果接收到的 Pdelay_Resp 报文的 twoStepFlag 是 FALSE,表明将不接收 Pdelay_Resp_Follow_Up 报文,则计算 $\langle \text{meanPathDelay} \rangle$ 如下:

$$\langle \text{meanPathDelay} \rangle = [(t_4 - t_1) - \text{Pdelay_Resp 的 correctionField}] / 2$$
- 4) 如果接收到的 Pdelay_Resp 报文的 twoStepFlag 是 TRUE,表明将接收 Pdelay_Resp_Follow_Up 报文,则计算 $\langle \text{meanPathDelay} \rangle$ 如下:

$$\langle \text{meanPathDelay} \rangle = [(t_4 - t_1) - (\text{responseOriginTimestamp} - \text{requestReceiptTimestamp}) - \text{Pdelay_Resp 的 correctionField} - \text{Pdelay_Resp_Follow_Up 的 correctionField}] / 2$$

当接收到 Pdelay_Req 报文时,延时响应方节点 B 应尽快发出相关联的 Pdelay_Resp 报文,并符合本条的其他要求,使周转时间 $t_3 - t_2$ 最小。

注:可按 10.1 的谐振来减小由过大的周转时间值引入的误差。

11.4.4 对等延时机使用中的限制

对于每个发送的 Pdelay_Req 报文,延时请求方节点 A 可接收到 0、1 或者多个 Pdelay_Resp 报文。通过观察 Pdelay_Resp 报文不同的 sourcePortIdentity 字段,可检测出多个响应。

注:如果在节点 A 与多个节点 B 设备之间存在端到端透明时钟或普通网桥,或者其他类似的多播和多端口设备,则会出现多个响应。尽管可以区分多个响应,但是在本标准中没有这样的机制,以允许与多个节点 B 设备的每个响应相关联的路径长度,被正确地分配到接收到的 Sync 报文。

在这些情况下应有如下动作:

- a) 当没有接收到 Pdelay_Resp 时,节点 A 应周期性的重传 Pdelay_Req 报文来检查节点 B 的存在。这种情况下的重传率是实现特定的;
- b) 当接收到单个 Pdelay_Resp 时,应按规定执行 11.4 的协议;
- c) 当接收到多个 Pdelay_Resp 时,节点 A 应:
 - 1) 如果是普通时钟或边界时钟,则进入 FAULTY 状态;或如果是点到点透明时钟,则进入故障状态。这种情况下,设备可周期性地重传 Pdelay_Req 报文来检查对该状态的处理。此时的重传率是实现特定的,应丢弃端口接收到的 Sync 和 Follow_Up 报文。
 - 2) 采取实现特定措施来解决此问题。

11.4.5 透明时钟的路径延时校正

11.4.5.1 点到点透明时钟

当接收到 Sync 报文时,点到点透明时钟的端口应:

- a) 如果时钟是单步点到点时钟,在各离开端口发送完 Sync 报文之前,将 $\langle \text{meanPathDelay} \rangle$ 的值加到 Sync 报文的 correctionField。其中 $\langle \text{meanPathDelay} \rangle$ 的值是按照对等延时机制,对连接接收 Sync 报文的进入端口的链路进行测量得到的。
- b) 如果时钟是双步点到点时钟,则在各离开端口上发送 Follow_Up 报文之前,将 $\langle \text{meanPathDelay} \rangle$

的值加到相关联的 Follow_Up 报文的 correctionField。其中〈meanPathDelay〉的值是按照对等延时机制,对连接接收 Sync 报文的进入端口的链路进行测量得到的。应在任何驻留时间的校正之后,将〈meanPathDelay〉加到 correctionField,见 11.5.2.2。

注 1: 为了正确关联 Sync 和 Follow_Up 报文,要求透明时钟维护 Sync 报文的 sequenceId 和 sourcePortIdentity 的记录,来同 Follow_Up 报文的 sourcePortIdentity 和 sequenceId 字段进行比较。

注 2: 如果透明时钟支持该精度,则 correctionField 的数据类型允许将〈meanPathDelay〉表示到小数纳秒(ns)。

11.4.5.2 端到端透明时钟

一个端到端透明时钟不应校正进入端口或离开口端口的路径延时。

11.5 对于 PTP 版本 2 事件的透明时钟驻留时间校正

11.5.1 驻留时间计算

透明时钟应对于所有版本 2 事件报文生成一个进入时间戳(见 7.3.4.2),指示在进入端口接收到事件报文的时间。

透明时钟应对于所有版本 2 事件报文生成一个离开时间戳(见 7.3.4.2),指示在离开口端发送事件报文的时间。

注: 通常,离开时间戳在透明时钟的各离开口有一个不同的值。

所有的时间戳应按 10.1 在域内进行测量。

如果已知连接进入和离开口端口的路径的 delayAsymmetry(见 7.4.2),应进行 11.6 的校正。

对于各离开口,每个这样的事件报文的驻留时间计算如下:

〈residenceTime〉= 离开时间戳 - 进入时间戳。

11.5.2 Sync 报文的驻留时间校正

11.5.2.1 单步透明时钟

当正在发送 Sync 事件报文时,〈residenceTime〉应通过时钟的离开口加到 Sync 事件报文的 correctionField。对于校验和或者报文的其他内容相关字段,离开口应进行任何需要的校正。

注: 如果透明时钟支持该精度,则 correctionField 的数据类型允许将〈residenceTime〉表达到小数纳秒(ns)。

对于任何接收到的 Follow_Up 报文应不进行修改。

11.5.2.2 双步透明时钟

如果接收到的 Sync 报文的 twoStepFlag 是 FALSE,则表明将不接收 Follow_Up 报文,然后:

- a) 应将接收到的 Sync 报文的 twoStepFlag 设为 TRUE,指示随后将有 Follow_Up 报文。尽快在离开口端发送 Sync 报文,并对校验和或者该报文的其他内容相关字段进行任何需要的校正。修正后的 Sync 报文应用于生成离开时间戳,来计算 Sync 报文的驻留时间。
- b) 应在离开口端准备发送 Follow_Up 报文,步骤如下:
 - 1) 应将接收到的 Sync 报文的 originTimestamp 复制到 Follow_Up 报文的 preciseOriginTimestamp 字段;
 - 2) 应将接收到的 Sync 报文的 sequenceId 复制到 Follow_Up 报文的 sequenceId 字段;
 - 3) 应将接收到的 Sync 报文的 sourcePortIdentity 复制到 Follow_Up 报文的 sourcePortIdentity 字段;
 - 4) 应将接收到的 Sync 报文的 domainNumber 复制到 Follow_Up 报文的 domainNumber 字段;
 - 5) 应将接收到的 Sync 报文的 logMessageInterval 复制到 Follow_Up 报文的 logMessageInterval 字段;
 - 6) 应将接收到的 Sync 报文报头头的 flagField 复制到 Follow_Up 报文的 flagField,并且将 twoStepFlag 设为 TRUE;
 - 7) 应将 Follow_Up 报文的 correctionField 设为〈residenceTime〉。对于点到点透明时钟,这应在

校正〈meanPathDelay〉之前完成。见 11.4.5.1。

注：如果透明时钟支持该精度，则 correctionField 的数据类型允许将〈residenceTime〉表达到小数纳秒(ns)。

如果接收到的 Sync 报文的 twoStepFlag 是 TRUE，则表明将接收 Follow_Up 报文，然后：

- a) 应通过离开端口发送接收到的 Sync 报文。该 Sync 报文应用于生成离开时间戳，以计算 Sync 报文的驻留时间。
- b) 在离开端口发送之前，应将〈residenceTime〉加到与 Sync 报文相关联的 Follow_Up 报文的 correctionField。

注：为了正确关联 Sync 和 Follow_Up 报文，要求透明时钟维护 Sync 报文的 sourcePortIdentity 和 sequenceId 字段的记录，以同 Follow_Up 报文的 sourcePortIdentity 和 sequenceId 进行比较。如果透明时钟支持该精度，correctionField 的数据类型允许将〈residenceTime〉表达到小数纳秒(ns)。

11.5.3 Delay_Req 报文的驻留时间校正

11.5.3.1 通用规范

11.5.3 应用于端到端透明时钟。点到点透明时钟不支持 Delay_Req 报文。

11.5.3.2 单步端到端透明时钟

当正在发送 Sync 事件报文时，〈residenceTime〉应被通过时钟的离开端口加到 Delay_Req 报文的 correctionField。对于校验和或者报文的其他内容相关字段，离开端口应进行任何必需的修改。

注：如果透明时钟支持该精度，correctionField 的数据类型允许将〈residenceTime〉表达到小数纳秒(ns)。

对于任何接收到的 Delay_Resp 报文应不进行修改。

11.5.3.3 双步端到端透明时钟

接收到的 Delay_Req 报文应在离开端口发送。该 Delay_Req 报文应用于生成离开时间戳，以计算 Delay_Req 报文的〈residenceTime〉。

在通过离开端口发送 Delay_Resp 报文之前，应将〈residenceTime〉加到与 Delay_Req 报文相关联的 Delay_Resp 报文的 correctionField。

注：为了正确关联 Delay_Req 报文和 Delay_Resp 报文，要求透明时钟维护 Delay_Req 报文的 sourcePortIdentity 和 sequenceId 字段的记录，以同 Delay_Resp 报文的 sourcePortIdentity 和 sequenceId 进行比较。如果透明时钟支持该精度，则 correctionField 的数据类型允许将〈residenceTime〉表达到小数纳秒(ns)。

11.5.4 Pdelay_Req 和 Pdelay_Resp 报文的驻留时间校正

11.5.4.1 概述

11.5.4 应用于端到端透明时钟。Pdelay_Req 和 Pdelay_Resp 报文终止于点到点透明时钟。

注：点到点时钟通常只用在点到点时钟的同构系统中。基于 PTP 的后续版本，11.5.4 的规定允许系统中使用端到端透明时钟。后续版本中可规定如何去实现点到点时钟间的一对多连接的混合系统。

11.5.4.2 单步端到端透明时钟

当正在发送 Pdelay_Req 报文时，Pdelay_Req 报文的〈residenceTime〉应通过时钟的离开端口加到 Pdelay_Req 报文的 correctionField。对于校验和或者报文的其他内容相关字段，离开端口应进行任何必需的修改。

对于 Pdelay_Req 报文的〈residenceTime〉，不应应对任何接收到的 Pdelay_Resp_Follow_Up 或 Pdelay_Resp 报文进行修改。

当正在发送 Pdelay_Resp 报文时，Pdelay_Req 报文的〈residenceTime〉应通过时钟的离开端口加到 Pdelay_Resp 报文的 correctionField。对于校验和或者报文的其他内容相关字段，离开端口应进行任何必需的修改。

对于 Pdelay_Resp 报文的〈residenceTime〉，不应应对任何接收到的 Pdelay_Resp_Follow_Up 报文进行修改。

11.5.4.3 双步端到端透明时钟

测量并保存 Pdelay_Req 报文的〈residenceTime〉,以加入到与 Pdelay_Req 报文相关联的 Pdelay_Resp_Follow_Up 报文的 correctionField 中。

测量并保存与 Pdelay_Req 报文相关联的 Pdelay_Resp 报文的〈residenceTime〉,以加入到与 Pdelay_Req 报文相关联的 Pdelay_Resp_Follow_Up 报文的 correctionField 中。

如果接收到的与 Pdelay_Req 报文相关联的 Pdelay_Resp 报文的 twoStepFlag 是 FALSE,则表明将不接收 Pdelay_Resp_Follow_Up 报文,然后:

- a) 应将接收到的 Pdelay_Resp 报文的 twoStepFlag 设为 TRUE,表明随后将有 Pdelay_Resp_Follow_Up 报文。尽快在离开端口发送 Pdelay_Resp 报文,并且对校验和或者报文的其它内容相关字段进行任何必需的修改。该修改后的 Pdelay_Resp 报文应用于生成离开时间戳,以计算 Pdelay_Resp 报文的〈residenceTime〉。
- b) 应准备发送 Pdelay_Resp_Follow_Up 报文,步骤如下:
 - 1) 将接收到的 Pdelay_Resp 报文的 sequenceId 复制到 Pdelay_Resp_Follow_Up 报文的 sequenceId 字段;
 - 2) 将接收到的 Pdelay_Resp 报文的 requestingPortIdentity 复制到 Pdelay_Resp_Follow_Up 报文的 requestingPortIdentity 字段;
 - 3) 将 responseOriginTimestamp 设为 0;
 - 4) 将接收到的 Pdelay_Resp 报文的 domainNumber 字段复制到 Pdelay_Resp_Follow_Up 报文的 domainNumber 字段;
 - 5) 应将 Pdelay_Resp_Follow_Up 报文的 correctionField 设为 Pdelay_Resp 的〈residenceTime〉同与 Pdelay_Resp 报文相关联的 Pdelay_Req 报文的〈residenceTime〉之和。

如果接收到的 Pdelay_Resp 报文的 twoStepFlag 是 TRUE,则表明将接收 Pdelay_Resp_Follow_Up 报文,然后:

- a) 接收到的 Pdelay_Resp 报文应在离开端口发送。该 Pdelay_Resp 报文应用于生成离开时间戳,以计算 Pdelay_Resp 报文的驻留时间。
- b) 在离开端口发送之前,应将 Pdelay_Resp 的〈residenceTime〉同与 Pdelay_Resp 报文相关联的 Pdelay_Req 报文的〈residenceTime〉之和,加到与 Pdelay_Req 报文相关联的 Pdelay_Resp_Follow_Up 报文的 correctionField。

注 1: 为了正确关联 Pdelay_Req、Pdelay_Resp 和 Pdelay_Resp_Follow_Up 报文,要求透明时钟维护 Pdelay_Req 报文的 sourcePortIdentity 和 sequenceId 字段的记录,以同 Pdelay_Resp 和 Pdelay_Resp_Follow_Up 报文的 requestingPortIdentity 和 sequenceId 进行比较。

注 2: 如果透明时钟支持该精度,则 correctionField 的数据类型允许将〈residenceTime〉表达到小数纳秒(ns)。

11.6 对于 PTP 版本 2 事件报文的不对称校正

11.6.1 通用规范

如果已知连接时钟进入和离开端口的路径的 delayAsymmetry(见 7.4.2),PTP 报文应按如下规定进行校正。

11.6.2 Sync 报文的不对称校正

Sync 报文:

- a) 对于连接离开端口路径的不对称,应不进行校正;
- b) 对于一个边界时钟或者普通时钟,在使用 correctionField 进行任何计算之前,当进入端口进行接收时,应通过把进入路径的 delayAsymmetry 值加到接收到的 Sync 报文的 correctionField,来校正连接进入端口的路径的不对称;

- c) 对于透明时钟,在该透明时钟的离开端口发送 Sync 报文之前,在进入端口进行接收时,应校正连接进入端口的路径的不对称:
 - 1) 如果透明时钟是单步时钟,则在离开端口进行发送之前,应通过把进入路径的 delayAsymmetry 值加到接收到的 Sync 报文的 correctionField 进行校正;
 - 2) 如果透明时钟是双步时钟,则在离开端口发送 Follow_Up 报文之前,应通过把进入路径的 delayAsymmetry 值加到与接收到的 Sync 报文相关联的 Follow_Up 报文的 correctionField 进行校正。

注:该 Follow_Up 报文可由一个端到端透明时钟的上游双步时钟生成,或者由端到端透明时钟自身生成。这要求透明时钟维护 Sync 报文的 sourcePortIdentity 和 sequenceId 字段的记录,以同 Follow_Up 报文的 requesting-PortIdentity 和 sequenceId 进行比较。

11.6.3 Delay_Req 报文的不对称校正

Delay_Req 报文:

- a) 对于连接离开端口路径的不对称,应不进行校正。
- b) 对于一个边界时钟或者普通时钟,在离开端口进行发送之前,应通过从发送的 Delay_Req 报文的 correctionField 中减去离开路径的 delayAsymmetry 值,来修改发送的 Delay_Req 报文的 correctionField。
- c) 对于端到端透明时钟,在进入端口进行接收时,并在离开端口发送后续的 Delay_Req 报文之前:
 - 1) 如果端到端透明时钟是单步时钟,则在离开端口发送之前,应通过从接收到的 Delay_Req 报文的 correctionField 中减去离开路径的 delayAsymmetry 值来进行校正;
 - 2) 如果端到端透明时钟是双步时钟,则在发送 Delay_Resp 报文之前,应通过从与原始 Delay_Req 报文相关联的接收到的 Delay_Resp 报文的 correctionField 中,减去连接 Delay_Req 报文离开端口的离开路径的 delayAsymmetry 值来进行校正。

11.6.4 Pdelay_Req 报文的不对称校正

Pdelay_Req 报文:

- a) 对于连接进入端口路径的不对称,应不校正。
- b) 对于一个边界时钟、普通时钟或者点到点透明时钟,在离开端口进行发送之前,应通过从发送的 Pdelay_Req 报文的 correctionField 中,减去离开路径的 delayAsymmetry 值,来修改发送的 Pdelay_Req 报文的 correctionField。
- c) 对于端到端透明时钟,在进入端口进行接收时,并在离开端口发送后续的 Pdelay_Req 报文之前:
 - 1) 如果端到端透明时钟是单步时钟,则在离开端口进行发送之前,应通过从接收到的 Pdelay_Req 报文的 correctionField 中,减去离开路径的 delayAsymmetry 值来进行校正;
 - 2) 如果端到端透明时钟是双步时钟,则在发送 Pdelay_Resp_Follow_Up 报文之前,应通过从与原始 Pdelay_Req 报文相关联的 Pdelay_Resp_Follow_Up 报文的 correctionField 中,减去连接 Pdelay_Req 报文离开端口的离开路径的 delayAsymmetry 值来进行校正。

注:该 Pdelay_Resp_Follow_Up 报文可由一个端到端透明时钟的上游双步时钟生成,或者由端到端透明时钟自身生成。这要求透明时钟维护 Delay_Req 报文的 sourcePortIdentity 和 sequenceId 字段的记录,以同 Delay_Resp 报文的 requestingPortIdentity 和 sequenceId 进行比较。

11.6.5 Pdelay_Resp 报文的不对称校正

Pdelay_Resp 报文:

- a) 对于连接离开端口的路径的不对称,应不进行校正;

- b) 对于一个边界时钟、普通时钟或者点到点透明时钟,在进入端口进行接收时,并在使用 correctionField 进行任何计算之前,应通过将进入路径的 delayAsymmetry 值加到接收到的 Pdelay_Resp 报文的 correctionField,来校正连接进入端口路径的不对称;
- c) 对于端到端透明时钟,在进入端口进行接收时,并在端到端透明时钟的离开端口发送 Pdelay_Resp 报文之前,应校正连接进入端口路径的不对称:
 - 1) 如果端到端透明时钟是单步时钟,则在离开端口进行发送之前,应通过将进入路径的 delayAsymmetry 值加到接收到的 Pdelay_Resp 报文的 correctionField 来进行校正;
 - 2) 如果端到端透明时钟是双步时钟,则在离开端口发送 Pdelay_Resp_Follow_Up 报文之前,应通过将进入路径的 delayAsymmetry 值加到与发送的 Pdelay_Resp 报文相关联的 Pdelay_Resp_Follow_Up 报文的 correctionField 来进行校正。

注:该 Pdelay_Resp_Follow_Up 报文可由一个端到端透明时钟的上游双步时钟生成,或者由端到端透明时钟自身生成。这要求透明时钟维护 Pdelay_Resp 报文的 requestingPortIdentity 和 sequenceId 字段的记录,以同 Pdelay_Resp_Follow_Up 报文的 requestingPortIdentity 和 sequenceId 进行比较。

12 时钟的同步与谐振

12.1 谐振

12.1.1 通用规范

在一个域内,有一个端口处于 SLAVE 状态的任何时钟,以及任何透明时钟,都应与最高级时钟谐振。

12.1.2 基于 Sync 报文的谐振

时钟 A 可以与另一个时钟 B 谐振,如下:

对于来自时钟 B 的 Sync 和可能的 Follow_Up 报文序列,时钟 A 按如下步骤计算 $\langle \text{correctedMasterEventTimestamp} \rangle$ 和 $\langle \text{syncEventIngressTimestamp} \rangle$:

- a) 当接收到 Sync 报文时,时钟 A 生成并记录一个按 7.3.4 内部延时已校正的时间戳 $\langle \text{syncEventIngressTimestamp} \rangle$ 。如果已知连接进入端口路径的 delayAsymmetry (见 7.4.2),应按 11.6 进行校正。
- b) 如果 Sync 报文的 flagField 的 twoStepFlag 比特是 FALSE,表明将不接收 Follow_Up 报文,则 $\langle \text{correctedMasterEventTimestamp} \rangle = \langle \text{originTimestamp} \rangle + \langle \text{meanPathDelay} \rangle + \text{Sync 报文的 correctionField}$ 。
- c) 如果 Sync 报文的 flagField 的 twoStepFlag 比特是 TRUE,表明将接收 Follow_Up 报文,则 $\langle \text{correctedMasterEventTimestamp} \rangle = \langle \text{preciseOriginTimestamp} \rangle + \langle \text{meanPathDelay} \rangle + \text{Sync 报文的 correctionField} + \text{Follow_Up 报文的 correctionField}$ 。

其中:

- $\langle \text{originTimestamp} \rangle$ 是接收到的 Sync 报文的 originTimestamp 字段值;
- $\langle \text{preciseOriginTimestamp} \rangle$ 是接收到的 Sync 报文的 preciseOriginTimestamp 字段值;
- 如果端口被配置为使用延时请求-响应机制,那么 $\langle \text{meanPathDelay} \rangle$ 应如 11.3 所规定;
- 如果端口被配置为使用对等延时机制,那么 $\langle \text{meanPathDelay} \rangle$ 应为如 11.4 所规定。

然后利用 $\langle \text{syncEventIngressTimestamps} \rangle$ 的序列和 $\langle \text{correctedMasterEventTimestamps} \rangle$ 的序列,来调整时钟 A 的时间变化率,以使其对准到最高级时钟的时间变化率。

注:作为一个示例,主时钟频率和时钟 A 频率的比值,可以估计为在一个接收到的时间戳和若干 syncInterval 后接收到的第二个时间戳之间,时钟 A 经过时间与主时钟经过时间的比值,即:

$$\frac{\langle \text{syncEventIngressTimestamp} \rangle_N - \langle \text{syncEventIngressTimestamp} \rangle_0}{\langle \text{correctedMasterEventTimestamp} \rangle_N - \langle \text{correctedMasterEventTimestamp} \rangle_0}$$

其中 N 是隔开时间戳的 syncInterval 的个数 ($N > 0$)。这样,时钟 A 的频率可以根据这个因子进行调节。

12.1.3 基于其他机制的谐振

在一些网络中,存在对时钟是可访问的、用于谐振两个时钟的物理信号。如果时钟 A 到时钟 B 的谐振与最佳主时钟算法(见 9.3)建立的同步层次结构相匹配,即 Sync 报文是由时钟 B 发送到时钟 A 的,就可以使用这样的信号。这些不在本标准范围内。

12.2 同步

在一个域内,一个端口处于 SLAVE 状态的普通时钟或边界时钟,应同步到同步层次结构中它的主时钟,该同步层次结构由最佳时钟算法建立。同步的特定含义不在本标准范围之内,但会使从时钟按 11.2 计算出的<offsetFromMaster>值的最小化。

13 PTP 报文格式

13.1 概述

本章中的格式和定义规定了发起方如何填充 PTP 报文的字段。本标准的其他部分定义了接收方对它们的处理。

本章的表格中,“八位位组”列表示字段的八位位组长度。“偏移”列表示从报文的 PTP 定义字段开始到该字段第一个八位位组的偏移。

13.2 通用报文格式要求

所有的报文应包括报文头、报文体和报文尾。报文尾长度可以为 0。

保留字段应以所有比特为 0 来发送,并被接收方忽略。

除非另有规定,所有字段值应:

- 是报文的一个固有特性,在本章中规定;
- 或者,基于发起设备的数据集或协议操作,由发起报文的设备,即由发起节点实例化;
- 或者,如果是双步透明时钟,任何生成的 Follow_Up 或 Pdelay_Resp_Follow_Up 报文,除了进行透明时钟操作所要求的调整外,报文看起来应就像是由 Sync 和 Pdelay_Resp 报文的发起方分别生成的;见第 11 章。

字段的数据类型应为各条标题括号中指示的类型。

13.3 报文头

13.3.1 通用报文头规范

所有 PTP 报文的公共报文头的规定见表 18。

表 18 公共报文头

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
保留				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
保留								1	5
flagField								2	6
correctionField								8	8
保留								4	16
sourcePortIdentity 10								10	20
sequenceId								2	30
controlField								1	32
logMessageInterval								1	33

13.3.2 报文头规范

13.3.2.1 transportSpecific (Nibble)

transportSpecific 字段可被低层传输协议使用,并在本标准传输附录中(规范性附录 D~I)该协议的映射规范中定义。

13.3.2.2 messageType (Enumeration4)

messageType 值应指示报文的类型,定义见表 19。

表 19 messageType 字段的值

报文类型	报 文 类	值(十六进制)
Sync	事件	0
Delay_Req	事件	1
Pdelay_Req	事件	2
Pdelay_Resp	事件	3
保留	—	4~7
Follow_Up	通用	8
Delay_Resp	通用	9
Pdelay_Resp_Follow_Up	通用	A
Announce	通用	B
Sigaling	通用	C
Management	通用	D
保留	—	E~F

由 messageType 字段的最高有效位将该字段分为事件报文和通用报文。

注: messageType 后保留的四位位组为 messageType 字段的今后扩展所保留。

13.3.2.3 versionPTP (UInteger4)

versionPTP 字段的值应为发起节点的数据集的 portDS.versionNumber 成员的值。

13.3.2.4 messageLength (UInteger 16)

messageLength 的值应为构成 PTP 报文的八位位组总的个数。所计算的八位位组从报文头的第一个八位位组开始,到所有报文尾的最后八位位组结束,或者如果没有报文尾成员,则到本条定义的报文的最后八位位组结束。

注: messageLength 不包含附录 D 中规定的任何填充比特。

13.3.2.5 domainNumber (UInteger 8)

对于普通时钟或者边界时钟, domainNumber 的值为发起普通时钟或边界时钟数据集的 defaultDS.domainNumber 成员的值。

对于由点到点透明时钟发起的对等延时机制报文,该值应为 11.4.3 中定义的值。

对于管理报文,该值应为 15.4.1.1 中定义的值。

13.3.2.6 flagField (Octet[2])

flagField 数组各比特值的定义见表 20。对于没有在表 20 中定义的报文类型,该值应为 FALSE。

表 20 flagField 的值

八位位组	比特	报文类型	名 称	描 述
0	0	Announce, Sync, Follow_Up, Delay_Resp	alternateMasterFlag	如果发起方的端口处于 MASTER 状态,则该值为 FALSE。将该标志设为 TRUE 的情况见 17.3 和 17.4
0	1	Sync, Pdelay_Resp	twoStepFlag	对于单步时钟, twoStepFlag 值应为 FALSE; 对于双步时钟, twoStepFlag 值应为 TRUE
0	2	ALL	unicastFlag	如果该报文发送到的传输层协议地址为单播地址, 则值为 TRUE; 如果该报文发送到的传输层协议地址为多播地址, 则值为 FALSE
0	5	ALL	PTP profile Specific 1	按备选 PTP 行规定; 否则, 为 FALSE
0	6	ALL	PTP profile Specific 2	按备选 PTP 行规定; 否则, 为 FALSE
0	7	ALL	保留	见注
1	0	Announce	leap61	timePropertiesDS. leap61 的值
1	1	Announce	leap59	timePropertiesDS. leap59 的值
1	2	Announce	currentUtcOffsetValid	timePropertiesDS. currentUtcOffsetValid 的值
1	3	Announce	ptpTimescale	timePropertiesDS. ptpTimescale 的值
1	4	Announce	timeTraceable	timePropertiesDS. timeTraceable 的值
1	5	Announce	frequencyTraceable	timePropertiesDS. frequencyTraceable 的值
注: 保留的比特用于附录 K 的实验性安全机制。				

保留所有未使用的标志。

13.3.2.7 correctionField (Integer64)

correctionField 为以纳秒 (ns) 为单位的测量的校正值再乘以 2^{16} 。比如, 2.5 ns 表示为 0000000000028000₁₆。

除了最高有效位, 字段所有比特全为 1 的值, 应指示校正值太大, 无法表示。

correctionField 的值取决于表 21 中描述的报文类型。

表 21 correctionField 语义

报 文 类 型	CorrectionField 描述
Sync	对小数纳秒、透明时钟的驻留时间(见 11.5.2)、点到点时钟的路径延时(见 11.4.5.1), 以及不对称校正(见 11.6.2)的校正
Delay_Req	对小数纳秒、透明时钟的驻留时间(见 11.5.3)、以及不对称校正(见 11.6.3)的校正
Pdelay_Req	对小数纳秒、透明时钟的驻留时间(见 11.5.4)、以及不对称校正(见 11.6.4)的校正
Pdelay_Resp	对小数纳秒、透明时钟的驻留时间(见 11.5.4)、以及不对称校正(见 11.6.5)的校正
Follow_Up	对小数纳秒、透明时钟的驻留时间(见 11.5.2)、点到点时钟的路径延时(见 11.4.5.1)、以及不对称校正(见 11.6.2)的校正
Delay_Resp	对小数纳秒、透明时钟的驻留时间(见 11.5.3)、以及不对称校正(见 11.6.3)的校正
Pdelay_Resp_Follow_Up	对小数纳秒、透明时钟的驻留时间(见 11.5.4)、以及不对称校正(见 11.6.4 和 11.6.5)的校正
Announce	0
Signaling	0
Management	0

13.3.2.8 sourcePortIdentity (PortIdentity)

字段 sourcePortIdentity 的值应为发起该报文端口的数据集的 portDS.portIdentity 成员。

13.3.2.9 sequenceId (UInteger 16)

除 Follow_Up、Delay_Resp、Pdelay_Resp、Pdelay_Resp_Follow_Up 报文,以及作为另一个管理报文的响应的管理报文外,字段 sequenceId 的值应是由报文的发起方根据 7.3.7 分配。对于这些例外情况,字段 sequenceId 值的定义见表 22 的参考。

表 22 sequenceId 值的例外情况的参考

报 文 类 型	参 考
Follow_Up	9.5.10、11.5.2.2
Delay_Resp	11.3.2
Pdelay_Resp	11.4.3
Pdelay_Resp_Follow_Up	11.4.3
Management	15.4.1.2

13.3.2.10 controlField (UInteger 8)

controlField 的值取决于 messageType 字段定义的报文类型(见 13.3.2.2),并应具有表 23 规定的值。不赞成接收方使用该字段。

注:该字段是为了与按 PTP 版本 1 设计的硬件兼容。

表 23 controlField 枚举

报 文 类 型	controlField 值(十六进制)
Sync	00
Delay_Req	01
Follow_Up	02
Delay_Resp	03
Management	04
其他	05
保留	06~FF

13.3.2.11 logMessageInterval (Integer 8)

字段 logMessageInterval 的值是由报文类型决定的,且其定义见表 24。

表 24 字段 logMessageInterval 的值

报 文 类 型	LogMessageInterval 的值
Announce	数据集成员 portDS.logAnnounceInterval 的值
Sync,Follow_Up	多播报文数据集成员 portDS.logSyncInterval 的值;单播报文为 7F ₁₆
Delay_Resp	多播报文数据集成员 portDS.logMinDelayReqInterval 的值;单播报文为 7F ₁₆
Delay_Req	7F ₁₆
Singaling	7F ₁₆
Management	7F ₁₆
Pdelay_Req	7F ₁₆
Pdelay_Resp	7F ₁₆
Pdelay_Resp_Follow_Up	7F ₁₆

13.4 报文尾

应用层报文的报文尾为一个连续的零或多个数据类型为 TLV 的实体序列。在第 14 章中描述了这些实体的含义。TLV 实体“n+1”的第一个八位位组应紧跟在 TLV 实体“n”的最后一个八位位组之后。TLV 的解释不应取决于它在报文中的位置。节点不应为事件报文附加 TLV 实体。

注：当通过非 PTP 网桥时，将 TLV 实体附加到事件报文上可能改变报文的传输延时。

13.5 Announce 报文

13.5.1 通用 Announce 报文规范

Announce 报文字段的规定见表 25。

表 25 Announce 报文字段

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
报文头(见 13.3)								34	0
originTimestamp								10	34
currentUtcOffset								2	44
保留								1	46
grandmasterPriority 1								1	47
grandmasterClockQuality								4	48
grandmasterPriority 2								1	52
grandmasterIdentity								8	53
stepsRemoved								2	61
timeSource								1	63

13.5.2 Announce 报文字段规范

13.5.2.1 originTimestamp (Timestamp)

当发送 Announce 报文时，originTimestamp 的值应为 0，或者为发起时钟本地时间的估计值，误差不大于±1 s。

13.5.2.2 currentUtcOffset (Integer 8)

currentUtcOffset 的值应是数据集 timePropertiesDS. currentUtcOffset 成员的值。

13.5.2.3 grandmasterPriority 1 (UInteger 8)

grandmasterPriority 1 的值应是数据集 parentDS. grandmasterPriority 1 成员的值。

13.5.2.4 grandmasterClockQuality (ClockQuality)

grandmasterClockQuality 的值应是数据集 parentDS. grandmasterClockQuality 成员的值。

13.5.2.5 grandmasterPriority 2 (UInteger 8)

grandmasterPriority 2 的值应是数据集 parentDS. GrandmasterPriority 2 成员的值。

13.5.2.6 grandmasterIdentity (ClockIdentity)

grandmasterIdentity 的值应是数据集 parentDS. grandmasterIdentity 成员的值。

13.5.2.7 stepsRemoved (UInteger 16)

stepsRemoved 的值应是发出该报文中时的数据集 currentDS. stepsRemoved 成员的值。

13.5.2.8 timeSource (Enumeration 8)

timeSource 的值应是数据集 timePropertiesDS. timeSource 成员的值。

13.6 Sync 和 Delay_Req 报文

13.6.1 通用 Sync 和 Delay_Req 报文规范

Sync 和 Delay_Req 报文字段的规定见表 26。

表 26 Sync 和 Delay_Req 报文的字段

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
报文头(见 13.3)								34	0
originTimestamp								10	34

13.6.2 Sync 和 Delay_Req 报文的字段规范

13.6.2.1 originTimestamp (Timestamp)

字段 originTimestamp 值的规定见 9.5.9 和 11.3。

13.7 Follow_Up 报文

13.7.1 通用 Follow_Up 报文规范

Follow_Up 报文字段的规定见表 27。

表 27 Follow_Up 报文字段

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
报文头(见 13.3)								34	0
preciseOriginTimestamp								10	34

13.7.2 Follow_Up 报文字段规范

13.7.2.1 preciseOriginTimestamp (Timestamp)

preciseOriginTimestamp 值的规定见 9.5.10 和 11.3。

13.8 Delay_Resp 报文

13.8.1 通用 Delay_Resp 报文规范

Delay_Resp 报文字段的规定见表 28。

表 28 Delay_Resp 报文字段

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
报文头(见 13.3)								34	0
receiveTimestamp 10									34
requestingPortIdentity 10									44

13.8.2 Delay_Resp 报文字段规范

13.8.2.1 receiveTimestamp (Timestamp)

receiveTimestamp 值的规定见 9.5.12 和 11.3。

13.8.2.2 requestingPortIdentity (PortIdentity)

requestingPortIdentity 值的规定见 11.3。

13.9 Pdelay_Req 报文

13.9.1 通用 Pdelay_Req 报文规范

Pdelay_Req 报文字段的规定见表 29。

表 29 Pdelay_Req 报文字段

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
报文头(见 13.3)								34	0
originTimestamp								10	34
保留								10	44

注：Pdelay_Req 报文中保留的字段是为了让报文长度与 Pdelay_Resp 报文的长度相匹配。在一些网络和网桥中，不同长度的报文具有不同的传输时间，这样就引入了不对称误差。

13.9.2 Pdelay_Resp 报文字段规范

13.9.2.1 originTimestamp (Timestamp)

字段 originTimestamp 值的规定见 11.4.3。

13.10 Pdelay_Resp 报文

13.10.1 通用 Pdelay_Resp 报文规范

Pdelay_Resp 报文字段的规定见表 30。

表 30 Pdelay_Resp 报文字段

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
报文头(见 13.3)								34	0
requestReceiptTimestamp								10	34
requestingPortIdentity								10	44

13.10.2 Pdelay_Resp 报文字段规范

13.10.2.1 requestReceiptTimestamp (Timestamp)

requestReceiptTimestamp 值的规定见 11.4.3。

13.10.2.2 requestingPortIdentity (PortIdentity)

requestingPortIdentity 值的规定见 11.4.3。

13.11 Pdelay_Resp_Follow_Up 报文

13.11.1 通用 Pdelay_Resp_Follow_Up 报文规范

Pdelay_Resp_Follow_Up 报文字段的规定见表 31。

表 31 Pdelay_Resp_Follow_Up 报文字段

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
报文头(见 13.3)								34	0
responseOriginTimestamp								10	34
requestingPortIdentity								10	44

13.11.2 Pdelay_Resp_Follow_Up 报文字段规范

13.11.2.1 responseOriginTimestamp (Timestamp)

responseOriginTimestamp 值的规定见 11.4.3。

13.11.2.2 requestingPortIdentity (PortIdentity)

requestingPortIdentity 值的规定见 11.4.3。

13.12 信号报文

13.12.1 来自其他节点的信号报文的接收

基于接收到的信号报文所指示的字段,应按照表 32 接受并应用该报文。

表 32 信号报文的接受

targetPortIdentity, clockIdentity 等于:	targetPortIdentity, portNumber 等于:	TLV 规范指示 适用于:	应用动作到:
defaultDS, clockIdentity	全为 1	时钟	时钟
		端口	全部端口
	portDS, portNumber	时钟	时钟
		端口	目标端口
全为 1	全为 1	时钟	时钟
		端口	全部端口
	portDS, portNumber	时钟	时钟
		端口	目标端口

应忽略不被接受的报文。

13.12.2 信号报文的发送

端口应发出一个信号报文,当:

- 接收到的信号报文中的 TLV 要求时;
- 本标准的可选或强制特性要求时;
- 本标准以外的实现特定考虑要求时。

信号报文用来传输一个或多个 TLV 实体序列。信号报文从一个时钟发送到一个或多个其他时钟。

信号报文的公共字段的规定见表 33。

表 33 信号报文字段

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
报文头(见 13.3)								34	0
targetPortIdentity								10	34
一个或多个 TLV								M	44

13.12.2.1 targetPortIdentity (PortIdentity)

targetPortIdentity 的值应为 PortIdentity 类型。targetPortIdentity 的值应是该报文要寻址端口的 PortIdentity(见 7.5.2)。

注: 见 13.12.1。

13.13 管理报文

管理报文的定义见第 15 章。

14 TLV 实体规范

14.1 通用要求

所有 TLV 扩展应具有数据类型 TLV,见 5.3.8。

不能解析一个 TLV 扩展的 PTP 节点应忽略该 TLV 扩展,并尝试解析报文的下一个 TLV。

在第 14 章的表格中,“八位位组”列表示字段的八位位组长度。“TLV 偏移”列表示从 TLV 开始处到字段的第一个八位位组的偏移。

14.1.1 tlvType (Enumeration16)

tlvType 标识 TLV。其值见表 34。

表 34 tlvType 值

tlvType 值	值(十六进制)	在标准中的定义
保留	0000	—
标准 TLV	—	—
MANAGEMENT	0001	15.5.3
MANAGEMENT_ERROR_STATUS	0002	15.5.4
ORGANIZATION_EXTENSION	0003	14.3
可选单播报文协商 TLV	—	16.1
REQUEST_UNICAST_TRANSMISSION	0004	—
GRANT_UNICAST_TRANSMISSION	0005	—
CANCEL_UNICAST_TRANSMISSION	0006	—
ACKNOWLEDGE_CANCEL_UNICAST_TRANSMISSION	0007	—
可选路径溯源机制 TLV	—	16.2
PATH_TRACE	0008	—
可选备选时标 TLV	—	16.3
ALTERNATE_TIME_OFFSET_INDICATOR	0009	—
保留用作标准 TLV	000A~1FFF	—
实验性 TLV	—	14.2
安全 TLV	—	Annex K
AUTHENTICATION	2000	—
AUTHENTICATION_CHALLENGE	2001	—
SECURITY_ASSOCIATION_UPDATE	2002	—
累积频标因子偏移	—	Annex L
CUM_FREQ_SCALE_FACTOR_OFFSET	2003	—
保留用作实验性 TLV	2004~3FFF	—
保留	4000~FFF	—

实验性的 TLV 值应保留,并由 IM/ST 委员会精确网络化时钟工作组进行分配,见 14.2。

14.1.2 lengthField (UInteger 16)

lengthField 的值为 TLV 的 valueField 的八位位组长度,见 5.3.8。

14.1.3 valueField (tlvType 特定)

对本标准定义的各 tlvType,TLV 的 valueField 成员的格式和含义在后续条中定义。

14.2 实验性 TLV

实验性 TLV 目的是便于操作经验的扩展,这些扩展可能成为将来的标准扩展。组织或公司可以向 IM/ST 委员会精确网络化时钟工作组申请实验性 tlvType 的值。实验性 tlvType 的值、提出的 TLV

格式和语义、以及 TLV 负责方的联系方式,将作为公开信息。

实验性 TLV 的值不是持久不变的。他们可以被重新分配:

- 如果 TLV 成为标准的 TLV;
- 如果各方都明确它不再需要;
- 分配数据的 5 年后。

14.3 制造商和标准组织扩展 TLV

14.3.1 概述

制造商和标准组织扩展 TLV 可以由制造商和标准组织分别使用,以根据他们的特殊需要来扩展协议。

14.3.2 TLV 成员规范

所有组织特定 TLV 扩展都应具有如表 35 规定的格式。

表 35 组织特定 TLV 字段

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
organizationId								3	4
organizationSubType								3	7
dataField								N	10

14.3.2.1 tlvType (Enumeration 16)

tlvType 应为 ORGANIZATION_EXTENSION,表示制造商和标准组织定义的扩展。

14.3.2.2 lengthField (UInteger 16)

lengthField 的值为 6+N,其中 N 为偶数,见 5.3.8。

14.3.2.3 organizationId (Octet[3])

该值应是 IEEE⁵⁾ 分配给制造商或标准组织的 OUI 的值。长度为 3 的八位位组数组的赋值顺序为:将数组索引 0 的八位位组成员赋值为 OUI 的最高有效八位位组。由 OUI 标识的组织应保证 organizationalSubType 字段(14.3.2.4)在 organizationId 值定义的范围内是唯一的。

不能识别特定 organizationId 或 organizationalSubType 的 PTP 节点,应丢弃除 lengthField 字段以外的 TLV 内容。

14.3.2.4 organizationSubType (Enumeration 24)

organizationSubType 字段在 organizationId 字段的范围内定义了一个子类型。organizationSubType 的值由 organizationId 标识的制造商或标准组织来分配。

14.3.2.5 数据(organizationSubType 和 organizationId 特定)

数据字段的格式和含义应由 {organizationId,organizationSubType} 的所有者来定义。

15 管理

15.1 概述

管理报文用于访问属性和生成本标准定义的某些事件。

5) OUI 可从 IEEE 注册管理委员会获得,见 <http://standards.ieee.org/regauth/>。这些分配号的指南也可在该网址获得。

15.1.1 管理机制的选择

应使用如下 3 个 PTP 管理机制之一：

- 默认的，15.2 规定的机制；
- 提供 PTP 行规所规定的等价功能的备选管理机制；
- 未规定管理机制。PTP 行规应规定定义好的固定值或/和状态，使得可通过实现特定的方式来寻址所有可组态的变量，见 8.1.2.1.3。

15.2 PTP 管理机制

该管理机制是由第 15 章的剩余部分，以及在本标准可选条中定义的任意管理 TLV 一起定义的。

15.3 管理报文的处理

15.3.1 来自其他节点的管理报文的接收

基于接收到的管理报文所指示的字段，应按照表 36 接受并应用该报文。

表 36 管理报文的接受

targetPortIdentity, clockIdentity 等于：	argetPortIdentity, portNumber 等于：	managementId 适用于(见表 40)	应用动作到
defaultDS, clockIdentity	全为 1	时钟	时钟
		端口	所有端口
	portDS, portNumber	时钟	时钟
		端口	目标端口
全为 1	全为 1	时钟	时钟
		端口	所有端口
	portDS, portNumber	时钟	时钟
		端口	目标端口

注 1：当一个“GET”报文是发送给“所有端口”的，响应节点给每个端口发送一个单独的回复。

应忽略不被接受的管理报文。

注 2：CLOCK_DESCRIPTION TLV 可以用来发现系统中支持 PTP 管理报文的任意时钟。

15.3.2 管理报文的发送

管理报文用来传输单独的管理 TLV 实体。管理报文将信息从一个时钟发送到一个节点管理者，以及从一个节点管理者发送到一个或多个时钟，见 15.3.1。

15.3.3 边界时钟对管理报文的转发

基于端口的状态和接收到的管理报文的 boundaryHops 字段的值，根据如下规则，边界时钟应通过其他端口转发一个端口接收到的多播管理报文：

- 仅转发处于 MASTER、SLAVE、UNCALIBRATED 或 PRE_MASTER 状态的端口所接收到的多播管理报文。
- 如果接收到的 boundaryHops 字段值为 0，则不应转发该管理报文；否则，在转发报文前，边界时钟应将该管理报文的 boundaryHops 字段的值减 1。
- 如果接收到的 boundaryHops 字段的值大于 0，则应仅通过状态为 MASTER、SLAVE、UNCALIBRATED 或 PRE_MASTER 的端口转发管理报文。

15.4 管理报文格式

15.4.1 公共字段

管理报文的公共字段的规定见表 37。

表 37 管理报文字段

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
报文头(见 13.3)								34	0
targetPortIdentity								10	34
startingBoundaryHops								1	44
boundaryHops								1	45
保留				actionField				1	46
保留								1	47
管理 TLV								M	48

15.4.1.1 报文头的 domainNumber

管理报文的公共报文头的 domainNumber(见 13.3)应规定目标域。

15.4.1.2 报文头的 sequenceId

响应管理报文的公共报文头的 sequenceId,应设为接收到的引起响应的管理报文的 sequenceId。否则,sequenceId 见 7.3.7。

15.4.1.3 targetPortIdentity (PortIdentity)

targetPortIdentity 字段是管理报文起作用的端口或节点的 portIdentity。

注: targetPortIdentity 所标识的端口不必是接收管理报文的端口。

如果管理报文是由时钟发送给管理者的,则 targetPortIdentity 字段应设为它作为响应的请求管理报文的 sourcePortIdentity。

15.4.1.4 startingBoundaryHops (UInteger 8)

对于不是作为其他管理报文请求的响应而被发出的报文,startingBoundaryHops 字段的值应是实现相关的。对于作为其他管理报文请求的响应而被发出的管理报文,startingBoundaryHops 的值应是由请求报文的 startingBoundaryHops 和 boundaryHops 字段计算出的值,计算式为(startingBoundaryHops-boundaryHops)。

注:当接收到一个管理报文时,该差值的绝对值指示报文通过的边界时钟的转发次数。

15.4.1.5 boundaryHops (UInteger 8)

boundaryHops 字段的值指示了管理报文的连续转发的剩余次数,其中转发是由接收报文的边界时钟按 15.3.3 完成的。当发出时钟第一次发送时,boundaryHops 的值应和 startingBoundaryHops 字段的值相同。

15.4.1.6 actionField (Enumeration4)

actionField 的值应指示当接收表 38 定义的报文时所采取的动作。

表 38 actionField 的值

动 作	采取的动作	值(十六进制)
GET	管理报文应包含单个管理 TLV。TLV 的 managementId 字段指示需要取回的特定信息。managementId 标识的数据当前值应在一个管理 TLV 中返回,并将其 actionField 的值设为 RESPONSE。如果发生错误,则应返回一个管理错误状况 TLV,并将其 actionField 的值设为 RESPONSE	0

表 38 (续)

动 作	采取的动作	值(十六进制)
SET	管理报文应包含单个管理 TLV。TLV 中的数据应用来更新 managementId 字段标识的数据的当前值。试图设置一个静态值或不可组态值,应返回一个管理错误状况 TLV,见 15.5.4。如果更新成功,则应返回管理报文,并将其 actionField 的值设为 RESPONSE。如果发生错误,则应返回管理错误状况 TLV,并将其 actionField 的值设为 RESPONSE。如果 managementId 字段标识的数据由几个字段组成,则更新应被视为原子性的 actionField,并且对任何项更新的失败都应认为是 SET 执行中的错误。具有可组态和不可组态数据相混合的数据定义的 TLV 是不允许的	1
RESPONSE	TLV 中的数据应为管理报文的 managementId 字段标识的当前数据值,该管理报文具有 GET 或 SET 的 actionField。managementId 的值应同请求报文中 managementId 的值相同。如果 GET 或 SET 的 actionField 所要求的 actionField 不能完全执行,则响应为一个管理错误状况 TLV,见 15.5.4	2
COMMAND	managementId 字段所指示的事件应被引发。该命令的结果应通过一个将 actionField 设为 ACKNOWLEDGE 的管理报文进行确认	3
ACKNOWLEDGE	确认管理报文是命令管理报文的响应。managementId 的值应同命令报文中的 managementId 值相同。如果命令不能执行,确认报文应为一个管理错误状况 TLV	4
保留	---	5-F

15.4.1.7 managementTLV

管理报文应用 0 个或 1 个 TLV 作为报文尾。

15.5 管理 TLV

15.5.1 管理 TLV 简介

15.5.1.1 概述

15.5 详述了管理 TLV 和管理错误状况 TLV 的结构。

有两种管理 TLV 的形式:处理数据集或单独数据集成员的管理 TLV;引发事件的管理 TLV。

15.5.1.1.1 数据集的管理

PTP 定义的可组态属性,不论是在第 8 章的数据集中维护或是以实现特定方式维护,都由 actionField 的值为 GET 和 SET 的管理报文分别读出和更新。这些报文的 TLV 数据结构仅包含可组态的变量。

PTP 定义的静态、动态和可组态属性,不论是在第 8 章的数据集中维护或是以实现特定方式维护,都由 actionField 的值为 GET 的管理报文读出。SET 可以不被用于这些报文。

15.5.1.1.2 事件的管理

对于引发事件的 TLV,管理报文的 actionField 的值为 COMMAND。对于每个 TLV 的 managementId,都定义或参考了事件和引发语义。

15.5.2 管理 TLV 的字段格式

管理 TLV 应具有表 39 规定的格式。

表 39 管理 TLV 字段

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
managementId								2	4
dataField								N	6

15.5.2.1 tlvType (Enumeration16)

tlvType 应为 MANAGEMENT。

15.5.2.2 lengthField (UInteger 16)

lengthField 的值为 2+N,其中 N 是偶数,见 5.3.8。

15.5.2.3 managementId (Enumeration16)

managementId 字段值的定义见表 40。对每个 managementId 值的 TLV 语义应按本标准 15.5、第 16 章和第 17 章进行定义。

表 40 允许动作列中的条目指示了管理报文公共字段中 actionField 字段的允许值,见 15.4.1.6。当接收具有不允许的 actionField 值的管理报文时,应采取以下动作:

- 丢弃管理 TLV 的内容;
- 返回一个管理错误状况 TLV(见 15.5.4),NOT_SUPPORTED。

表 40 managementId 值

managementId 名称	managementId 值(十六进制)	允许动作	应用到
可应用到所有节点类型	0000~1FFF	—	—
NULL_MANAGEMENT	0000	GET、SET、COMMAND	端口
CLOCK_DESCRIPTION	0001	GET	端口
USER_DESCRIPTION	0002	GET、SET	时钟
SAVE_IN_NON_VOLATILE_STORAGE	0003	COMMAND	时钟
RESET_NON_VOLATILE_STORAGE	0004	COMMAND	时钟
INITIALIZE	0005	COMMAND	时钟
FAULT_LOG	0006	GET	时钟
FAULT_LOG_RESET	0007	COMMAND	时钟
保留	0008~1FFF	—	—
可应用到普通时钟和边界时钟	2000~3FFF	—	—
DEFAULT_DATA_SET	2000	GET	时钟
CURRENT_DATA_SET	2001	GET	时钟
PARENT_DATA_SET	2002	GET	时钟
TIME_PROPERTIES_DATA_SET	2003	GET	时钟

表 40 (续)

managementId 名称	managementId 值(十六进制)	允许动作	应用到
PORT_DATA_SET	2004	GET	端口
Priority 1	2005	GET、SET	时钟
Priority 2	2006	GET、SET	时钟
DOMAIN	2007	GET、SET	时钟
SLAVE_ONLY	2008	GET、SET	时钟
LOG_ ANNOUNCE_INTERVAL	2009	GET、SET	端口
ANNOUNCE_RECEIPT_TIMEOUT	200A	GET、SET	端口
LOG_ SYNC_INTERVAL	200B	GET、SET	端口
VERSION_NUMBER	200C	GET、SET	端口
ENABLE_PORT	200D	COMMAND	端口
DISABLE_PORT	200E	COMMAND	端口
TIME	200F	GET、SET	时钟
CLOCK_ACCURACY	2010	GET、SET	时钟
UTC_PROPERTIES	2011	GET、SET	时钟
TRACEABILITY_PROPERTIES	2012	GET、SET	时钟
TIMESCALE_PROPERTIES	2013	GET、SET	时钟
UNICAST_NEGOTIATION_ENABLE	2014	GET、SET	端口
PATH_TRACE_LIST	2015	GET	时钟
PATH_TRACE_ENABLE	2016	GET、SET	时钟
GRANDMASTER_CLUSTER_TABLE	2017	GET、SET	时钟
UNICAST_MASTER_TABLE	2018	GET、SET	端口
UNICAST_MASTER_MAX_TABLE_SIZE	2019	GET	端口
ACCEPTABLE_MASTER_TABLE	201A	GET、SET	时钟
ACCEPTABLE_MASTER_TABLE_ENABLED	201B	GET、SET	端口
ACCEPTABLE_MASTER_MAX_TABLE_SIZE	201C	GET	时钟
ALTERNATE_MASTER	201D	GET、SET	端口
ALTERNATE_TIME_OFFSET_ENABLE	201E	GET、SET	时钟
ALTERNATE_TIME_OFFSET_NAME	201F	GET、SET	时钟
ALTERNATE_TIME_OFFSET_MAX_KEY	2020	GET	时钟
ALTERNATE_TIME_OFFSET_PROPERTIES	2021	GET、SET	时钟
保留	2020~3FFF	—	—
可应用到透明时钟	4000~5FFF	—	—
TRANSPARENT_CLOCK_DEFAULT_DATA_SET	4000	GET	时钟
TRANSPARENT_CLOCK_PORT_DATA_SET	4001	GET	端口
PRIMARY_DOMAIN	4002	GET、SET	时钟

表 40 (续)

managementId 名称	managementId 值(十六进制)	允许动作	应用到
保留	4003~5FFF	—	—
可应用到普通时钟、边界时钟和透明时钟	6000~7FFF	—	—
DELAY_MECHANISM	6000	GET、SET	端口
LOG_MIN_PDELAY_REQ_INTERVAL	6001	GET、SET	端口
保留	6002~BFFF	—	—
此范围用于实现特定标识符	C000~DFFF	—	—
此范围是由备选 PTP 行规分配的	E000~FFFF	—	—
保留	FFFF	—	—

注：managementId 的实现特定范围是由制造商分配的，来对他们自己的设备定义唯一的管理功能。不期望互操作性，并且用户必须保证这样的 TLV 指向合适的设备。

15.5.3 每个 managementId 的管理 TLV 数据字段规范

15.5.3.1 可用于所有时钟的 TLV 数据字段

15.5.3.1.1 NULL_MANAGEMENT

管理 TLV 数据字段的长度为 0。接收该 TLV 应不导致影响数据集或状态的动作发生。NULL_管理报文的接收应符合 actionField 的要求，见 15.4.1.6。

注：NULL_管理报文典型用于通过执行管理程序且不改变协议运作来对实现进行测试。例如，可发送这样的报文，以测试接收到的管理报文是否正在被记录到实现特定的事件日志中。

15.5.3.1.2 CLOCK_DESCRIPTION

包含该 TLV 的管理报文字段的 targetPortIdentity、portNumber 成员(见 15.4.1.3)应指示 physicalAddress、protocolAddress 和 profileIdentity 适用的端口。

该 TLV 的所有其他字段应用于整个节点，并应由一个指向节点任意端口的询问返回。

该数据字段的规定见表 41。

表 41 CLOCK_DESCRIPTION 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
ClockType								2	0
PhysicalLayerProtocol								L	2
PhysicalAddressLength								2	2+L
PhysicalAddress								S	4+L
ProtocolAddress								N	4+L+S
ManufacturerIdentity								3	4+L+S+N
保留								1	4+L+S+N+3
ProductDescription								P	4+L+S+N+4
RevisionData								Q	4+L+S+N+4+P
UserDescription								R	4+L+S+N+4+P+Q
ProfileIdentity								6	4+L+S+N+4+P+Q+R
填充								M	4+L+S+N+4+P+Q+R

15.5.3.1.2.1 ClockType (Boolean[16])

ClockType 值应指示表 42 定义的 PTP 节点的类型。ClockType 字段中比特的 TRUE 值指示该描述适用于节点。

注：几个元素可能为 TRUE，例如，一个普通时钟和一个端到端透明时钟的组合。这是一个静态值，且不属于第 8 章的时钟数据集部分。

表 42 ClockType 规范

数组索引(十六进制)	描 述
0	该节点实现了一个普通时钟
1	该节点实现了一个边界时钟
2	该节点实现了一个点到点透明时钟
3	该节点实现了一个端到端透明时钟
4	该节点实现了一管理节点
5~F	保留

15.5.3.1.2.2 PhysicalLayerProtocol (PTPText)

PhysicalLayerProtocol 的值应指示定义 PhysicalAddress 成员的物理层协议。这是一个静态值，而非第 8 章的时钟数据集部分。

该字段中的符号个数最多应为 32 个，见 5.3.9。

15.5.3.1.2.3 PhysicalAddressLength (UInteger 16)

PhysicalAddressLength 的值是 PhysicalAddress 字段的八位位组长度。范围是 1~16 个八位位组。这是一个静态值，而非第 8 章的时钟数据集部分。

15.5.3.1.2.4 PhysicalAddress (Octet[PhysicalAddressLength])

PhysicalAddress 的值应是由字段的 targetPortIdentity.portNumber 成员指示的端口的物理地址，例如，IEEE 802.3 终端站的 MAC 地址。如果对于一个没有物理地址的特定网络技术，该值长度应为 0。

这是一个静态值，而非第 8 章的时钟数据集部分。

15.5.3.1.2.5 protocolAddress (PortAddress)

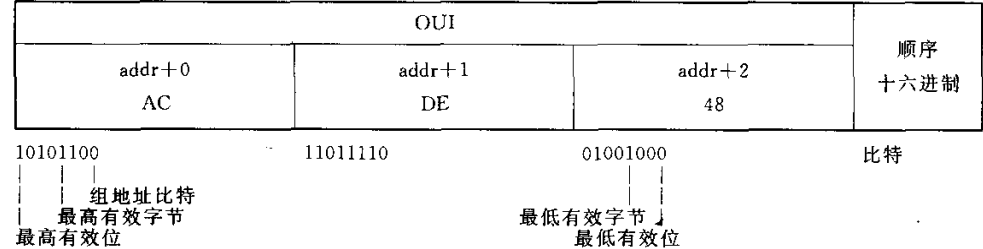
protocolAddress 的值应是由字段的 targetPortIdentity.portNumber 成员指示的端口的协议地址。这是一个静态值(不能使用协议对其修改)，而非第 8 章的时钟数据集部分。

15.5.3.1.2.6 manufacturerIdentity (Octet [3])

manufacturerIdentity 的值为节点制造商所拥有的一个 OUI。

示例：

公司 X 的 OUI 为 ACDE48(十六进制)。公司 X 的 manufacturerIdentity 的字节和比特表示如下。



这是一个静态值，而非第 8 章的时钟数据集部分。

15.5.3.1.2.7 productDescription (PTPText)

productDescription 字段应依次指示：

- a) 节点制造商的名称,manufacturerName,后面跟一个分号(;);
 - b) 节点的模型编号,modelNumber,后面跟一个分号(;);
 - c) 此模式实例的标识符,instanceIdentifier,如 MAC 地址或序列号。
- productDescription.textField 字段中(见 5.3.9)符号个数最多应为 64 个。
- 这是一个静态值,而非第 8 章的时钟数据集部分。

manufacturerName、modelName 和 instanceIdentifier 字符串的内容和含义都是由制造商决定的。

15.5.3.1.2.8 revisionData (PTPText)

该值应指示对节点硬件(HW)、固件(FW)和软件(SW)的修改。该信息应是用分号隔开的文本字段,依次为 HW;FW;SW。不适用的元素应用长度为 0 的文本字段指示。这是一个静态值,而非第 8 章的时钟数据集部分。

RevisionData.textField 字段的符号个数最多应为 32 个。

15.5.3.1.2.9 userDescription (PTPText)

userDescription 字段应依次指示:

- a) 用户定义的设备名称,如 Sensor-1,后跟一个分号(;);
- b) 用户定义的设备物理位置,如 Rack-2 Shelf-3。

任何字段都可以缺少,如(;Rack-2 Shelf-3)或者(Sensor-1)。缺省地,不要求任何内容。这是一个可组态的值,而非第 8 章的时钟数据集部分。

在 userDescription.textField 字段(见 5.3.9)的符号最多应为 128 个。

15.5.3.1.2.10 profileIdentity (Octet[6])

profileIdentity 的值应标识端口实现的 PTP 规范,该端口由字段的 targetPortIdentity.portNumber 成员指示。

profileIdentity 的值应由 19.3.3 定义的行规的制定者来分配。

这是一个静态值,而非第 8 章的时钟数据集部分。

示例:

附录 J 的 J.4.1 定义“使用对等延时机制的缺省 PTP 行规”的 profileIdentity 为 6 个八位位组的字段 001B19000200₁₆。该 profileIdentity 的字节和比特表示如下。

OUI			行规索引			字段
addr+0	addr+1	addr+2	addr+3	addr+4	addr+5	顺序
00	1B	19	00	02	00	十六进制
00000000	00011011	000110011	00000000	00000010	00000000	比特
组地址比特 最高有效字节 最高有效位			最低有效字节 最低有效位			

15.5.3.1.2.11 填充(Octet[M])

填充字段应是一个长度为 M 的八位位组数组,其中 M 为 1 或 0。如果 M 等于 1,则八位位组中所有比特都应 0,见 15.5.2.2。

15.5.3.1.3 USER_DESCRIPTION

该数据字段的规定见表 43。

表 43 USER_DESCRIPTION 管理 TLV 数据字段

比特							八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0	
userDescription							L	0
填充							M	L

15.5.3.1.3.1 userDescription(PTPText)

该 TLV 用于对 CLOCK_DESCRIPTION TLV 返回的用户描述的值进行组态。

userDescription 字段应依次指示：

- a) 用户定义的设备名称,如 Sensor-1,后跟一个分号(;)；
- b) 用户定义的设备物理位置,如 Rack-2 Shelf-3。

任何字段都可以缺少,如(;Rack-2 Shelf-3)或者(Sensor-1)。缺省不要求任何内容。

在 userDescription.textField 字段(见 5.3.9)的符号个数最多应为 128 个。

15.5.3.1.3.2 填充(Octet[M])

填充字段应是一个长度 M 的八位位组数组,其中 M 为 1 或者 0。如果 M 为 1,则八位位组中所有比特应为 0,见 15.5.2.2。

15.5.3.1.4 SAVE_IN_NON_VOLATILE_STORAGE

该数据字段长度为 0。该 TLV 的接收应导致将适用的动态和可组态数据集成员的当前值复制到非易失读写内存,见 8.1.3.5。

15.5.3.1.5 RESET_NON_VOLATILE_STORAGE

该数据字段长度为 0。该 TLV 的接收应导致将非易失读写内存的内容重新设置到适用的动态和可组态数据集成员的初始值,见 8.1.3.5。

15.5.3.1.6 INITIALIZE

该数据字段的规定见表 44。

表 44 INITIALIZE 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
initializationKey								2	0

15.5.3.1.6.1 initializationKey(Enumeration16)

报文的 initializationKey 字段的值定义应如表 45 规定。

表 45 initializationKey 枚举

INITIALIZATION_KEY	值(十六进制)	定义
INITIALIZE_EVENT	0000	在普通时钟和边界时钟,带有该值的 INITIALIZE 报文应引发 INITIALIZE 事件(见 9.2.6.3)。在其他节点,这应引起实现特定的初始化程序的执行
保留	0001~7FFF	保留。接收带有该 initializationKey 的 INITIALIZE 报文不应引发任何动作
实现特定	8000~FFFF	结果是实现特定的

15.5.3.1.7 FAULT_LOG

FAULT_LOG TLV 的数据字段应如表 47 规定。FAULT_LOG TLV 返回一个故障记录列表。每个故障记录由 FaultRecord 结构来规定,见 5.3.10。

FaultRecord.faultTime 的值应指示故障发生的时间,该时间由节点的本地时钟指示。对于时间戳中的字段,全为 1 的值应指示发生的时间是无效的。

FaultRecord.severityCode 成员的值应从表 46 中的枚举中选择。

表 46 故障日志 severityCode 枚举

值(十六进制)	Fault Record, Severity Code description
00	Emergency; 系统不可用
01	Alert; 需要立即动作
02	Critical; 危急状况
03	Error; 错误状况
04	Warning; 警告状况
05	Notice; 正常且重要状况
06	Informational; 信息报文
07	Debug; 调试级报文
08~FF	保留

FaultRecord, faultName、FaultRecord, faultValue 和 FaultRecord, faultDescription 成员的值是实现特定的,并且可长度为 0;即,PTPText 结构的 lengthField 成员为 0。

FaultRecord, faultName 应为在实现中故障的唯一名称。

FaultRecord, faultValue 应为与故障相关的任意值,它对故障诊断是必要的。

FaultRecord, faultDescription 应为故障的任意补充描述。

故障日志的大小是实现特定的。到接收 FAULT_LOG_RESET TLV(见 15.5.3.1.8)之前,故障日志是由时钟维护的。

表 47 FAULT_LOG 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
numberOfFaultRecords								2	0
faultRecord								N	2
填充								M	2+N

15.5.3.1.7.1 numberOfFaultRecords (UInteger 16)

numberOfFaultRecords 字段的值应是要返回的故障记录的个数。

15.5.3.1.7.2 faultRecord (FaultRecord[numberOf FaultRecords])

faultRecord 字段的值应为一个故障记录数组。

15.5.3.1.7.3 填充(Octet[M])

填充字段应是一个长度为 M 的八位位组数组,其中 M 为 1 或者 0。如果 M 为 1,则八位位组中所有比特应为 0,见 15.5.2.2。

15.5.3.1.8 FAULT_LOG_RESET

该 TLV 不包含数据。

FAULT_LOG_RESET 命令应导致 FAULT_LOG 被清空。

15.5.3.2 适用于普通时钟和边界时钟的 TLV 数据字段

15.5.3.2.1 TIME

该 TLV 用于设置时间。时间源于最高级时钟,并由 PTP 分发给域中的其他时钟。

最高级时钟的时间通常是通过同一个主参考(如 GPS)进行交互来确定的,并依据本标准范围内的

方法。当通过 actionField 为 SET 的管理报文将该 TLV 发送给一个非最高级时钟的节点时,节点应返回一个管理错误状况 TLV。当 actionField 为 GET 时,节点应返回时间的当前值。

注:如果时间是在非最高级时钟的时钟内设置的,它应在接收到下一个 Sync 报文时被覆盖,因此它仅作为瞬态值存在。

当发送给一个可设置的最高级时钟,应采用关于 GET 和 SET 的通常规则,见 15.4.1.6。

数据字段规定见表 48。

表 48 TIME 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
currentTime								10	0

15.5.3.2.1.1 currentTime (Timestamp)

currentTime 的值为要设置给目标时钟的值。

注:由于该 TLV 是通过网络传输的,设置时间的精度受到网络波动的限制。因为这个原因,不规定设置时间精确到小数纳秒。多数情况下,实际精度是毫秒级或者更低,取决于构成数据字段的信息源和网络特性。

15.5.3.2.2 CLOCK_ACCURACY

该 TLV 可用于设置目标时钟的精度。

注:最高级时钟的精度和时间通常是通过同一个主参考或应用特定的时间源(如 GPS)进行交互来确定的,且依据本标准范围外的方法。如果最高级时钟的时间是通过 TIME TLV 设置的,则精度也应被相应设置。因为在 BMC 算法的执行中考虑 clockAccuracy 属性,所以通过该 TLV 对任意时钟的 clockAccuracy 属性进行设置,可能导致下一次 BMC 算法执行时最高级时钟的改变。

当发送给一个可设置的最高级时钟时,应采用关于 GET 和 SET 的通常规则,见 15.4.1.6。

数据字段的规定见表 49。

表 49 CLOCK_ACCURACY 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
clockAccuracy								1	0
保留								1	1

15.5.3.2.2.1 clockAccuracy (Enumeration 8)

clockAccuracy 的值应为数据集 defaultDS. clockQuality. clockAccuracy 成员的值。该值应从 clockAccuracy 枚举中选取,见表 6。

15.5.3.2.3 ENABLE_PORT

该 TLV 不包含数据。

在一个普通时钟和边界时钟,ENABLE_PORT 报文的接收应引发 DESIGNATED_ ENABLED 事件,见 9.2.6.4。

15.5.3.2.4 DISABLE_PORT

该 TLV 不包含数据。

在一个普通时钟和边界时钟,DISABLE_PORT 报文的接收应引发 DESIGNATED_ DISABLE 事件,见 9.2.6.5。

15.5.3.3 适用于普通时钟和边界时钟的 defaultDS 数据集的 TLV 数据字段

15.5.3.3.1 DEFAULT_DATA_SET

该数据字段的规定见表 50。

表 50 DEFAULT_DATA_SET 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	SO	TSC	1	0
保留								1	1
numberPorts								2	2
Priority 1								1	4
clockQuality								4	5
Priority 2								1	9
clockIdentity								8	10
domainNumber								1	18
保留								1	19

15.5.3.3.1.1 TSC (Boolean)

TSC 的值应为数据集的 defaultDS. twoStepFlag 成员的值。

15.5.3.3.1.2 SO (Boolean)

SO 的值应为数据集的 defaultDS. slaveOnly 成员的值。

15.5.3.3.1.3 numberPorts (UInteger 16)

numberPorts 的值应为数据集的 defaultDS. numberPorts 成员的值。

15.5.3.3.1.4 priority 1 (UInteger 8)

priority 1 的值应为数据集的 defaultDS. priority 1 成员的值。

15.5.3.3.1.5 clockQuality (clockQuality)

clockQuality 的值应为数据集的 defaultDS. clockQuality 成员的值。

15.5.3.3.1.6 priority 2 (UInteger 8)

Priority 2 的值应为数据集的 defaultDS. Priority 2 成员的值。

15.5.3.3.1.7 clockIdentity (clockIdentity)

clockIdentity 的值应为数据集的 defaultDS. clockIdentity 成员的值。

15.5.3.3.1.8 domainNumber (UInteger 8)

domainNumber 的值应为数据集的 defaultDS. domainNumber 成员的值。

15.5.3.3.2 Priority 1

该数据字段的规定见表 51。

表 51 Priority 1 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
Priority 1								1	0
保留								1	1

15.5.3.3.2.1 Priority 1 (UInteger 8)

Priority 1 的值应为数据集的 defaultDS. Priority 1 成员的值。

15.5.3.3.3 Priority 2

该数据字段的规定见表 52。

表 52 Priority 2 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
Priority 2								1	0
保留								1	1

15.5.3.3.3.1 Priority 2 (UInteger 8)

Priority 2 的值应为数据集的 defaultDS. Priority 2 成员的值。

15.5.3.3.4 DOMAIN

该数据字段的规定见表 53。

表 53 DOMAIN 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
domainNumber								1	0
保留								1	1

15.5.3.3.4.1 domainNumber (UInteger 8)

domainNumber 的值应为数据集的 defaultDS. domainNumber 成员的值。

15.5.3.3.5 SLAVE_ONLY

该数据字段的规定见表 54。

表 54 SLAVE_ONLY 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	SO	1	0
保留								1	1

15.5.3.3.5.1 SO (Boolean)

SO 的值应为数据集的 defaultDS. slaveOnly 成员的值。

15.5.3.4 适用于普通时钟和边界时钟的 currentDS 数据集的 TLV 数据字段

15.5.3.4.1 CURRENT_DATA_SET

该数据字段的规定见表 55。

表 55 CURRENT_DATA_SET 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
stepsRemoved								2	0
offsetFromMaster								8	2
meanPathDelay								8	10

15.5.3.4.1.1 stepsRemoved (UInteger 16)

stepsRemoved 的值应为数据集的 currentDS. stepsRemoved 成员的值。

15.5.3.4.1.2 offsetFromMaster (TimeInterval)

offsetFromMaster 的值应为数据集的 currentDS. offsetFromMaster 成员的值。

15.5.3.4.1.3 meanPathDelay (TimeInterval)

meanPathDelay 的值应为数据集的 currentDS. meanPathDelay 成员的值。

15.5.3.5 适用于普通时钟和边界时钟的 parentDS 数据集的 TLV 数据字段

15.5.3.5.1 PARENT_DATA_SET ,

该数据字段的规定见表 56。

表 56 PARENT_DATA_SET 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
parentPortIdentity								10	0
0	0	0	0	0	0	0	PS	1	10
保留								1	11
observedParentOffsetScaledLogVariance								2	12
observed ParentClockPhaseChangeRate								4	14
grandmasterPriority 1								1	18
grandmasterClockQuality								4	19
grandmasterPriority 2								1	23
grandmasterIdentity								8	24

15.5.3.5.1.1 parentPortIdentity (PortIdentity)

parentPortIdentity 的值应为数据集的 parentDS. parentPortIdentity 成员的值。

15.5.3.5.1.2 PS (Boolean)

PS 的值应为数据集的 parentDS. PS 成员的值。

15.5.3.5.1.3 observedParentOffsetScaledLogVariance (UInteger 16)

observedParentOffsetScaledLogVariance 的值应为数据集的 parentDS. observedParentOffsetScaledLogVariance 成员的值。

15.5.3.5.1.4 observedParentClockPhaseChangeRate (UInteger 32)

observedParentClockPhaseChangeRate 的值应为数据集的 parentDS. observedParentClockPhaseChangeRate 成员的值。

15.5.3.5.1.5 grandmasterPriority 1 (UInteger 8)

grandmasterPriority 1 的值应为数据集的 parentDS. grandmasterPriority 1 成员的值。

15.5.3.5.1.6 grandmasterClockQuality (ClockQuality)

grandmasterClockQuality 的值应为数据集的 parentDS. grandmasterClockQuality 成员的值。

15.5.3.5.1.7 grandmasterPriority 2 (UInteger 8)

grandmasterPriority 2 的值应为数据集的 parentDS. grandmasterPriority 2 成员的值。

15.5.3.5.1.8 grandmasterIdentity (ClockIdentity)

grandmasterIdentity 的值应为数据集的 parentDS. grandmasterIdentity 成员的值。

15.5.3.6 适用于普通时钟和边界时钟的 timePropertiesDS 数据集的 TLV 数据字段

15.5.3.6.1 TIME_PROPERTIES_DATA_SET

该数据字段的规定见表 57。

表 57 TIME_PROPERTIES_DATA_SET 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
currentUtcOffset								2	0
0	0	FTRA	TTRA	PTP	UTCv	LI-59	LI-61	1	2
timeSource								1	3

15.5.3.6.1.1 currentUtcOffset (UInteger 16)

currentUtcOffset 的值应为数据集的 timePropertiesDS. currentUtcOffset 成员的值。

15.5.3.6.1.2 LI-61 (Boolean)

LI-61 的值应为数据集的 timePropertiesDS. leap61 成员的值。

15.5.3.6.1.3 LI-59 (Boolean)

LI-59 的值应为数据集的 timePropertiesDS. Leap59 成员的值。

15.5.3.6.1.4 UTCv (Boolean)

UTCv 的值应为数据集的 timePropertiesDS. currentUtcOffsetValid 成员的值。

15.5.3.6.1.5 PTP (Boolean)

PTP 的值应为数据集的 timePropertiesDS. ptpTimescale 成员的值。

15.5.3.6.1.6 TTRA (Boolean)

TTRA 的值应为数据集的 timePropertiesDS. timeTraceable 成员的值。

15.5.3.6.1.7 FTRA (Boolean)

FTRA 的值应为数据集的 timePropertiesDS. frequencyTraceable 成员的值。

15.5.3.6.1.8 timeSource (Enumeration 8)

timeSource 的值应为数据集的 timePropertiesDS. timeSource 成员的值。

15.5.3.6.2 UTC_PROPERTIES

该数据字段的规定见表 58。

表 58 UTC_PROPERTIES 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
CurrentUtcOffset								2	0
0	0	0	0	0	UTCv	LI-59	LI-61	1	2
保留								1	3

15.5.3.6.2.1 currentUtcOffset (UInteger 16)

currentUtcOffset 的值应为数据集的 timePropertiesDS. currentUtcOffset 成员的值。

15.5.3.6.2.2 LI-61 (Boolean)

LI-61 的值应为数据集的 timePropertiesDS. leap61 成员的值。

15.5.3.6.2.3 LI-59 (Boolean)

LI-59 的值应为数据集的 timePropertiesDS. Leap59 成员的值。

15.5.3.6.2.4 UTCv (Boolean)

UTCv 的值应为数据集的 timePropertiesDS. currentUtcOffsetValid 成员的值。

15.5.3.6.3 TRACEABILITY_PROPERTIES

该数据字段的规定见表 59。

表 59 TRACEABILITY_PROPERTIES 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
0	0	FTRA	TTRA	0	0	0	0	1	0
保留								1	1

15.5.3.6.3.1 TTRA (Boolean)

TTRA 的值应为数据集的 timePropertiesDS.timeTraceable 成员的值。

15.5.3.6.3.2 FTRA (Boolean)

FTRA 的值应为数据集的 timePropertiesDS.frequencyTraceable 成员的值。

15.5.3.6.4 TIMESCALE_PROPERTIES

该数据字段的规定见表 60。

表 60 TIMESCALE_PROPERTIES 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
0	0	0	0	PTP	0	0	0	1	0
保留								1	1

15.5.3.6.4.1 PTP (Boolean)

PTP 的值应为数据集的 timePropertiesDS.ptpTimescale 成员的值。

15.5.3.6.4.2 timeSource (Enumeration 8)

timeSource 的值应为数据集的 timePropertiesDS.timeSource 成员的值。

15.5.3.7 适用于普通时钟和边界时钟的 portDS 数据集的 TLV 数据字段

15.5.3.7.1 PORT_DATA_SET

该数据字段的规定见表 61。

表 61 PORT_DATA_SET 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
portIdentity								10	0
portState								1	10
logMinDelayReqInterval								1	11
peerMeanPathDelay								8	12
logAnnounceInterval								1	20
announceReceiptTimeout								1	21
logSyncInterval								1	22
delayMechanism								1	23
logMinPdelayReqInterval								1	24
保留				versionNumber				1	25

15.5.3.7.1.1 portIdentity (portIdentity)

portIdentity 的值应为数据集的 portDS.portIdentity 成员的值。

15.5.3.7.1.2 portState (Enumeration 8)

portState 的值应为数据集的 portDS.portState 成员的值。

15.5.3.7.1.3 logMinDelayReqInterval (Integer 8)

logMinDelayReqInterval 的值应为数据集的 portDS. logMinDelayReqInterval 成员的值。

15.5.3.7.1.4 peerMeanPathDelay (TimeInterval)

peerMeanPathDelay 的值应为数据集的 portDS. peerMeanPathDelay 成员的值。

15.5.3.7.1.5 logAnnounceInterval (Integer 8)

logAnnounceInterval 的值应为数据集的 portDS. logAnnounceInterval 成员的值。

15.5.3.7.1.6 announceReceiptTimeout (UInteger 8)

announceReceiptTimeout 的值应为数据集的 portDS. announceReceiptTimeout 成员的值。

15.5.3.7.1.7 logSyncInterval (Integer 8)

logSyncInterval 的值应为数据集的 portDS. logSyncInterval 成员的值。

15.5.3.7.1.8 delayMechanism (Enumeration 8)

delayMechanism 的值应为数据集的 portDS. delayMechanism 成员的值。

15.5.3.7.1.9 logMinPdelayReqInterval (Integer 8)

logMinPdelayReqInterval 的值应为数据集的 portDS. logMinPdelayReqInterval 成员的值。

15.5.3.7.1.10 versionNumber (UInteger4)

versionNumber 的值应为数据集的 portDS. versionNumber 成员的值。

15.5.3.7.2 LOG_ANNOUNCE_INTERVAL

该数据字段的规定见表 62。

表 62 LOG_ANNOUNCE_INTERVAL 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
logAnnounceInterval								1	0
保留								1	1

15.5.3.7.2.1 logAnnounceInterval (Integer 8)

logAnnounceInterval 的值应为数据集的 portDS. logAnnounceInterval 成员的值。

15.5.3.7.3 ANNOUNCE_RECEIPT_TIMEOUT

该数据字段的规定见表 63。

表 63 ANNOUNCE_RECEIPT_TIMEOUT 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
announceReceiptTimeout								1	0
保留								1	1

15.5.3.7.3.1 announceReceiptTimeout (UInteger 8)

announceReceiptTimeout 的值应为数据集的 portDS. announceReceiptTimeout 成员的值。

15.5.3.7.4 LOG_SYNC_INTERVAL

该数据字段的规定见表 64。

表 64 LOG_SYNC_INTERVAL 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
logSyncInterval								1	0
保留								1	1

15.5.3.7.4.1 logSyncInterval (Integer 8)

logSyncInterval 的值应为数据集的 portDS.logSyncInterval 成员的值。

15.5.3.7.5 DELAY_MECHANISM

DELAY_MECHANISM 数据字段的值应为表 65 规定的 portDS.delayMechanism 的值。

表 65 DELAY_MECHANISM 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
delayMechanism								1	0
保留								1	1

15.5.3.7.5.1 delayMechanism (Enumeration 8)

对于普通时钟和边界时钟, delayMechanism 的值应为数据集的 portDS.delayMechanism 成员的值。对于透明时钟, 如果实现, 该值应为数据集 transparentClockDefaultDS 的 transparentClockDefaultDS.delayMechanism 成员; 否则, 该值应从该值的实现特定存储器中获得。

15.5.3.7.6 LOG_MIN_PDELAY_REQ_INTERVAL

该数据字段的规定见表 66。

表 66 LOG_MIN_PDELAY_REQ_INTERVAL 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
logMinPdelayReqInterval								1	0
保留								1	1

15.5.3.7.6.1 logMinPdelayReqInterval (Integer 8)

logMinPdelayReqInterval 的值应为数据集的 portDS.logMinPdelayReqInterval 成员的值。

15.5.3.7.7 VERSION_NUMBER

该数据字段的规定见表 67; 见 7.5.5。

表 67 VERSION_NUMBER 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
保留				Version Number				1	0
保留								1	1

15.5.3.7.7.1 versionNumber (UInteger4)

versionNumber 的值应为数据集的 portDS.versionNumber 成员的值。

15.5.3.8 适用于普通时钟和边界时钟的 defaultDS 数据集的 TLV 数据字段

15.5.3.8.1 TRANSPARENT_CLOCK_DEFAULT_DATA_SET

该数据字段的规定见表 68。

表 68 TRANSPARENT_CLOCK_DEFAULT_DATA_SET 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
clockIdentity								8	0
numberPorts								2	8
delayMechanism								1	10
primaryDomain								1	11

15.5.3.8.1.1 clockIdentity (clockIdentity)

clockIdentity 的值应为数据集的 defaultDS. clockIdentity 成员的值。

15.5.3.8.1.2 numberPorts (UInteger 16)

numberPorts 的值应为数据集的 defaultDS. numberPorts 成员的值。

15.5.3.8.1.3 delayMechanism (Enumeration 8)

delayMechanism 的值应为数据集的 defaultDS. delayMechanism 成员的值。

15.5.3.8.1.4 primaryDomain (UInteger 8)

primaryDomain 的值应为数据集的 defaultDS. primaryDomain 成员的值。

15.5.3.9 DELAY_MECHANISM

使用与可用于普通时钟或边界时钟 portDS 数据集的相同 TLV, 见 15.5.3.7.5。

15.5.3.9.1 PRIMARY_DOMAIN

该数据字段的规定见表 69。

表 69 PRIMARY_DOMAIN 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
primaryDomain								1	0
保留								1	1

15.5.3.9.1.1 primaryDomain (UInteger 8)

如果实现, 则 primaryDomain 的值应为数据集 transparentClockDefaultDS 的成员 transparentClockDefaultDS. primaryDomain 的值; 否则, 该值应从该值的实现特定存储器获得。

15.5.3.10 适用于透明时钟的 transparentClockDefaultDS 数据集的 TLV 数据字段

15.5.3.10.1 TRANSPARENT_CLOCK_PORT_DATA_SET

该数据字段的规定见表 70。

表 70 TRANSPARENT_CLOCK_PORT_DATA_SET 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
portIdentity								10	0
0	0	0	0	0	0	0	FLT	1	10
logMinPdelayReqInterval								1	11
peerMeanPathDelay								8	12

15.5.3.10.1.1 portIdentity (portIdentity)

portIdentity 的值应为数据集的 transparentClockDefaultDS. portIdentity 成员的值。

15.5.3.10.1.2 FLT (Boolean)

FLT 的值应为数据集的 transparentClockDefaultDS. FLT 成员的值。

15.5.3.10.1.3 logMinPdelayReqInterval (Integer 8)

logMinPdelayReqInterval 的值应为数据集的 transparentClockDefaultDS. logMinPdelayReqInterval 成员的值。

15.5.3.10.1.4 peerMeanPathDelay (Integer 8)

peerMeanPathDelay 的值应为数据集的 transparentClockDefaultDS. peerMeanPathDelay 成员的值。

15.5.3.10.2 LOG_MIN_PDELAY_REQ_INTERVAL

使用与适用于普通时钟或边界时钟的 portDS 数据集相同的 TLV, 见 15.5.3.7.6。

15.5.4 MANAGEMENT_ERROR_STATUS 的 TLV

15.5.4.1 概述

该 TLV 在响应或确认管理报文中返回,见 15.4.1.6。MANAGEMENT_ERROR_STATUS 的格式见表 71。

表 71 MANAGEMENT_ERROR_STATUS 的 TLV 格式

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
managementErrorId								2	4
managementId								2	6
保留								4	8
displayData								N	12
填充								M	12+N

15.5.4.2 tlvType

tlvType 的值应为 MANAGEMENT_ERROR_STATUS。

15.5.4.3 lengthField

lengthField 应为 8+N+M,其中 N 为 displayData 字段的长度,M 为填充字段的长度。

15.5.4.4 managementErrorId (Enumeration16)

managementErrorId 的值应从表 72 定义的枚举中选取。

表 72 managementErrorId 枚举

managementErrorId	规 范	值 (十六进制)
保留	—	0000
RESPONSE_TOO_BIG	要求的操作不适合放在一个单独的响应报文中	0001
NO_SUCH_ID	无法识别 managementId	0002
WRONG_LENGTH	可以识别 managementId,但数据长度有误	0003
WRONG_VALUE	managementId 和长度都正确,但一个或多个值有误	0004
NOT_SETABLE	SET 命令中的一些变量没有被更新,因为它们是不可组态的	0005
NOT_SUPPORTED	该节点不支持所要求的操作	0006
保留	—	0007~BFFF
实现特定	此范围被用于实现特定的错误	C000~DFFF
PTP profile defined	此范围是由备选 PTP 行规分配的	E000~FFFD
GENERAL_ERROR	发生的错误不能由其他 managementErrorId 值所覆盖	FFFE
保留	—	FFFF

15.5.4.5 managementId (Enumeration16)

managementId 字段的值定义见表 40。managementId 字段包含的 managementId 应与发生错误的管理 TLV 的 managementId 相对应。

15.5.4.6 displayData (PTPText)

这是一个可选文本字段,提供了可读的错误解释。

displayData.textField 字段(见 5.3.9)中的符号个数最多应为 50 个。

15.5.4.7 填充(Octet[M])

填充字段应是一个长度为 M 的八位位组数组,其中 M 为 1 或 0。如果 M 为 1,则八位位组中所有比特都为 0,见 15.5.2.2。

16 通用可选特征

16.1 单播报文协商(可选)

16.1.1 通用单播协商端口操作规范

一个端口(请求方)可以通过发送一个 REQUEST_UNICAST_TRANSMISSION TLV 实体,以请求另一个端口(授权者)发送单播 Announce、Sync、Delay_Resp 或者 Pdelay_Resp 报文。

单播发送请求的产生、授权、确认、取消可以不考虑 PTP 端口状态,但是在普通时钟或者边界时钟的任何端口处于 INITIALIZING、FAULTY 或 DISABLED 状态下,或者透明时钟的任何端口处于故障情况下,这些操作不应发生。

接收到一个 REQUEST_UNICAST_TRANSMISSION TLV 实体的端口应以一个 GRANT_UNICAST_TRANSMISSION TLV 实体进行响应。被发送的 GRANT_UNICAST_TRANSMISSION TLV 实体授权或拒绝请求。

通过发送一个 CANCEL_UNICAST_TRANSMISSION TLV 实体,获得授权的端口(被授权者)可以通知授权者,它不再需要所授权的服务。授权者在接收到 CANCEL_UNICAST_TRANSMISSION TLV 时,总是以一个 ACKNOWLEDGE_CANCEL_UNICAST_TRANSMISSION TLV 进行响应,并可以立即停止提供所指示的服务。

通过发送一个 CANCEL_UNICAST_TRANSMISSION TLV 实体,授权者可以通知被授权者不能再提供所授权的服务。

被授权者在接收到 CANCEL_UNICAST_TRANSMISSION TLV 时,总是以一个 ACKNOWLEDGE_CANCEL_UNICAST_TRANSMISSION TLV 进行响应,并应立即停止使用所指示的服务。授权者应该继续提供已授权的服务,直到接收到一个 ACKNOWLEDGE_CANCEL_UNICAST_TRANSMISSION TLV 或者已经发送了实现特定数量的 CANCEL_UNICAST_TRANSMISSION TLV 为止。

当授权是关于 Announce 或 Sync 报文的,授权者应以一个平均报文间隔来发送报文,该间隔近似等于被授权的报文间隔。除非授权者或者被授权者取消该授权,从授权报文发送时刻开始,这种发送应至少持续该授权期间。

当授权是关于 Delay_Resp 报文的,并且被授权者接收到 Delay_Req 报文的平均报文间隔不小于所授权的报文间隔时,授权者应以 Delay_Resp 报文来响应接收到的每个 Delay_Req 报文。除非授权者或者被授权者取消了该授权,从授权报文发送时刻开始,这种操作应至少持续整个授权期间。如果接收 Delay_Req 报文的平均间隔小于授权的报文间隔,授权者可以忽略多余的 Delay_Req 报文。

在发送报文中,报文间隔应在授权的 $\pm 30\%$ 的授权报文间隔内,置信度为 90%。

对每种报文类型,在任何时刻只有一个授权是活动的。如果接收了一个授权给特定 messageId 发送的授权报文,将取消任何之前与授权者达成的协议。

如果在两个端口间为特定报文类型协商了一个单播合约,那么这两个端口间的任何该类型的多播报文将被忽略。

当为发送 Delay_Resp 报文协商了一个单播合约时,则与 Delay_Resp 报文相关联的 Delay_Req 报文也应是单播的。

16.1.2 单播协商使能

通过 UNICAST_NEGOTIATION_ENABLE 的管理报文可以使能或禁止单播协商机制。该机制缺省情况下是禁止的,除非在 PTP 行规中另有规定。

如果禁止该机制,节点应响应 UNICAST_NEGOTIATION_ENABLE TLV 实体。一个被禁节点不应:

- 响应 REQUEST_UNICAST_TRANSMISSION TLV 实体;
- 发送 REQUEST_UNICAST_TRANSMISSION TLV 实体;
- 发送 GRANT_UNICAST_TRANSMISSION TLV 实体。

16.1.3 授权端口操作

如果请求方在当前协议到期之前发出一个新的发送请求,若资源允许,则授权者应以一个授权(至少与之前授权未超期部分一样长)来响应该请求。

如果授权端口认为报文间隔和授权持续时间的组合是不合理的,则该端口应优先减少它的授权持续时间而不是降低速率。

16.1.4 单播 TLV

16.1.4.1 REQUEST_UNICAST_TRANSMISSION TLV 规范

REQUEST_UNICAST_TRANSMISSION TLV 格式的规定见表 73。

表 73 REQUEST_UNICAST_TRANSMISSION TLV 格式

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
messageType				保留				1	4
logInterMessagePeriod								1	5
durationField								4	6

16.1.4.1.1 tlvType

tlvType 的值应是 REQUEST_UNICAST_TRANSMISSION。

16.1.4.1.2 lengthField

lengthField 的值是 6。

16.1.4.1.3 messageType (Enumeration4)

messageType 的值应指示请求发送的单播报文的报文类型。枚举的编码与报文头中 messageType 字段中所使用的相同,见 13.3.2.2。总应拒绝除 Announce、Sync、Delay_Resp 或者 Pdelay_Resp 报文外的单播报文请求。如果一个双步时钟授权单播发送 Sync 或者 Pdelay_Resp 报文,则相应的 Follow_Up 和 Pdelay_Resp_Follow_Up 报文也应采用单播发送。

16.1.4.1.4 logInterMessagePeriod (Integer 8)

logInterMessagePeriod 的值应是以 2 为底,被请求的单播报文间所请求的平均间隔(以秒为单位)的对数。

16.1.4.1.5 durationField (UInteger 32)

durationField 的值应是发送被请求报文所需的秒数。

16.1.4.2 GRANT_UNICAST_TRANSMISSION TLV 规范

GRANT_UNICAST_TRANSMISSION TLV 格式的规定见表 74。

表 74 GRANT_UNICAST_TRANSMISSION TLV 格式

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
messageType				保留				1	4

表 74 (续)

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
logInterMessagePeriod								1	5
durationField								4	6
保留								1	10
0	0	0	0	0	0	0	R	1	11

16.1.4.2.1 tlvType

tlvType 的值应是 GRANT_UNICAST_TRANSMISSION。

16.1.4.2.2 lengthField

lengthField 的值是 8。

16.1.4.2.3 messageType (Enumeration4)

messageType 的值应指示授权发送的单播报文的报文类型。枚举的编码与报文头中 messageType 所使用的一致,见 13.3.2.2。该值应与 REQUEST_UNICAST_TRANSMISSION TLV 请求中 messageType 字段相同。

16.1.4.2.4 logInterMessagePeriod (Integer 8)

logInterMessagePeriod 的值应是以 2 为底,被请求的单播报文间的授权平均间隔(以秒为单位)的对数。

16.1.4.2.5 durationField (UInteger 32)

durationField 的值应是发送报文所需的秒数。0 值表示请求已被拒绝。

16.1.4.2.6 R(请求续订)(Boolean)

当请求端口重复它的请求,授权端口认为授权可以被续订时,R 值应为 TRUE。

16.1.4.3 CANCEL_UNICAST_TRANSMISSION TLV 规范

CANCEL_UNICAST_TRANSMISSION TLV 格式的规定见表 75。

表 75 CANCEL_UNICAST_TRANSMISSION TLV 格式

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
messageType				保留				1	4
保留								1	5

16.1.4.3.1 tlvType

tlvType 的值应是 CANCEL_UNICAST_TRANSMISSION。

16.1.4.3.2 lengthField

lengthField 的值为 2。

16.1.4.3.3 messageType (Enumeration4)

messageType 的值应指示要取消发送的单播报文的类型。枚举的编码与报文头中 messageType 字段所使用的相同,见 13.3.2.2。

16.1.4.4 ACKNOWLEDGE_CANCEL_UNICAST_TRANSMISSION TLV 规范

ACKNOWLEDGE_CANCEL_UNICAST_TRANSMISSION TLV 格式的规定见表 76。

表 76 ACKNOWLEDGE_CANCEL_UNICAST_TRANSMISSION TLV 格式

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
messageType				保留				1	4
保留								1	5

16.1.4.4.1 tlvType

tlvType 的值应为 ACKNOWLEDGE_CANCEL_UNICAST_TRANSMISSION。

16.1.4.4.2 lengthField

lengthField 的值为 2。

16.1.4.4.3 messageType (Enumeration4)

messageType 的值应指示正在被确认的取消发送单播报文的类型。该值应与 CANCEL_UNICAST_TRANSMISSION TLV 中 messageType 字段相同,本报文是对 CANCEL_UNICAST_TRANSMISSION TLV 的确认。

16.1.4.5 UNICAST_NEGOTIATION_ENABLE

该报文用于使能或禁止单播协商机制。

一个单播协商机制被使能的节点,在接收到一个 EN 字段值为 FALSE 的 UNICAST_NEGOTIATION_ENABLE TLV 时,应通过使用 16.1.4.3 中定义的 CANCEL_UNICAST_TRANSMISSION TLV 来取消所有的协商授权。直到 CANCEL_UNICAST_TRANSMISSION TLV 已被发送到所有的被授权者,该节点应在对任何 UNICAST_NEGOTIATION_ENABLE TLV 的响应中报告,协商机制已被使能。

UNICAST_NEGOTIATION_ENABLE 管理 TLV 的格式定义见表 77。

表 77 UNICAST_NEGOTIATION_ENABLE 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	EN	1	0
保留								1	1

16.1.4.5.1 EN (Boolean)

EN 为 TRUE 表示单播发送机制可运行;反之,表示该机制不能运行。

16.2 路径追踪(可选)

16.2.1 概述

16.2 规定了一种机制:用 TLV 追踪 PTP Announce 报文经过时间系统的路由。这是一个可在边界时钟上实现的可选机制。

当接收到 TLV 时,一个边界时钟扫描该 TLV 的 pathSequence 成员,以检查它自身的 clockIdentity 是否存在,这将指示一个环路的存在。边界时钟把它的 clockIdentity 附加到 TLV 的 pathSequence 成员的尾端,并把 TLV 附加到正要离开的 Announce 报文。

该机制的主要用途之一是检测在边界时钟的环路中无休止地循环的 Announce 报文,即所谓的欺诈帧。如果检测到这样的环路,接收到的 Announce 报文应被丢弃。这样的环路通过在底层网络上执行生成树协议而被移除,见 6.2。

注:9.3.2.5 的机制提供了一种保护措施,防止生成树协议运行中的失败或瞬态引起的欺诈帧。

通常,边界时钟可以从多个源接收 Announce 报文;当前父时钟以及任意数量的外部主时钟。应只为来自当前父时钟的 Announce 报文维护<pathTraceList>(16.2.3)。将该机制应用到来自当前父时钟的除 Announce 报文外的报文超出了本标准的范围。

16.2.2 路径追踪使能

应维护一个实现特定的使能控制。如果被使能,则路径追踪机制应是可操作的。如果被禁止,则除了处理 PATH_TRACE_ENABLE 管理 TLV 外,路径追踪机制是不活动的。缺省地,路径追踪机制应是禁止的,除非在 PTP 行规中另有规定。

16.2.3 路径追踪列表

应维护类型 ClockIdentity 的成员的一个实现特定的列表<pathTraceList>,其初始化值应为空列表。

16.2.4 状态变化

无论何时时钟更新基于判定代码 M1 或 M2 的数据集(见 9.3.5),<pathTraceList>都应被初始化为空列表。

16.2.5 Announce 报文的接收

下列附加规范须应用于对接收到的 Announce 报文的处理,见 9.5.3。接收 Announce 报文的边界时钟的一个端口应:

- a) 扫描存在的任何 PATH_TRACE TLV,比较其 clockIdentity 字段与 defaultDS 数据集的 clockIdentity 字段是否相等;
- b) 如果 TLV 存在并找到了匹配者,则报文应被丢弃;
- c) 如果 TLV 存在且未找到匹配者,则时钟应将 TLV 的 pathSequence 成员复制到<pathTraceList>,见 16.2.3。

16.2.6 Announce 报文的发送

下列附加规范用于 Announce 报文的发送,见 9.5.8。

发送 Announce 报文的端口应给将一个 PATH_TRACE TLV 附加到报文。PATH_TRACE TLV 的数据字段的值应为<pathTraceList>(见 16.2.3),在列表的尾部附加时钟的 clockIdentity。如果得到的 Announce 报文大小超出了网络技术所允许的最大帧长度,则不应附加 PATH_TRACE TLV。

16.2.7 PATH_TRACE TLV 规范

PATH_TRACE TLV 格式的规定见表 78。

表 78 PATH_TRACE TLV 格式

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
pathSequence								8N	4

16.2.7.1 tlvType

tlvType 的值应为 PATH_TRACE。

16.2.7.2 lengthField

lengthField 的值应为 8 N。

16.2.7.3 pathSequence (ClockIdentity[N])

pathSequence 的值应为时钟标识符的一个列表。

16.2.8 PATH_TRACE_LIST 管理报文

该管理报文的 TLV 可用于从一个普通时钟或边界时钟取回当前<pathTraceList>(见 16.2.3)。数

据字段的规定见表 79。

表 79 PATH_TRACE_LIST 管理 TLV 数据字段

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
pathSequence								8N	0

16.2.8.1 pathSequence (ClockIdentity[N])

pathSequence 的值是<pathTraceList>中时钟标识符的列表,见 16.2.3。

16.2.9 PATH_TRACE_ENABLE 管理报文

该管理报文可用于使能或禁止路径追踪机制。PATH_TRACE_ENABLE TLV 数据字段的规定见表 80。

表 80 PATH_TRACE_ENABLE 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	EN	1	0
保留								1	1

16.2.9.1 EN (Boolean)

EN 的值为 TRUE 指示路径追踪机制是可运行的;反之,指示该机制是不可运行的。

16.3 备选时标(可选)

16.3.1 概述

通过发送一个 ALTERNATE_TIME_OFFSET_INDICATOR TLV 实体,最高级时钟可以指示备选时间与它的节点时间之间的偏移。

可以维护多个备选时标。设计为支持该选项的节点是最高级时钟时,应维护资源以支持一个实现特定的备选时标个数。每个支持的备选时标可以由该选项的 TLV 中的 keyField 字段值标识,并由 ALTERNATE_TIME_OFFSET_INDICATOR TLV 中的 displayName 字段描述。键值应为从 0 开始到<maxKey>值结束的连续值。可以使能或禁止每个支持的备选时标。

注:不要求来自不同制造商的节点发送的键值对应到相同的备选时标。

一个节点应在所有 Announce 报文中为每个被使能的备选时标发送一个 ALTERNATE_TIME_OFFSET_INDICATOR TLV。如果一个备选时标被禁止,则节点不应发送该 ALTERNATE_TIME_OFFSET_INDICATOR TLV 实体。

一个备选时标可以具有不连续性(例如,在夏令时间的开始和结束时刻)。

备选时间偏移指示器不应用于指示 UTC 相对 PTP 时标的偏移或即将发生的偏移的变化。

如果一个不连续(跳跃)即将发生,节点应在不连续即将发生前所发送的至少 portDS. announceReceiptTimeout+1 个 Announce 报文的连续序列中指示该情况。该不连续的时间和大小应按表 81 中规定的 jumpSeconds 和 timeOfNextJump 字段来指示。

如果一个接收到的 ALTERNATE_TIME_OFFSET_INDICATOR TLV 实体的 jumpSeconds 字段是非零值,则指示即将发生一个不连续;如果接收节点的时间大于接收到的 TLV 中 timeOfNextJump 字段的值,则该节点应忽略这个 TLV。

备选时标机制的属性可以用 ALTERNATE_TIME_OFFSET_ENABLE、ALTERNATE_TIME_OFFSET_NAME 和 ALTERNATE_TIME_OFFSET_PROPERTIES 管理 TLV 来管理。

16.3.2 边界时钟转发

不是最高级时钟且实现了备选时标选项的边界时钟,应在它发送的任何 Announce 报文中,转发包

含在所有 ALTERNATE_TIME_OFFSET_INDICATOR TLV 实体中的信息,而这些 TLV 实体是包含在该边界时钟最近从其主时钟接收到的 Announce 报文中的。

注:从时钟和它最高级时钟间的系统中的所有边界时钟都应实现备选时标选项。

16.3.3 ALTERNATE_TIME_OFFSET_INDICATOR TLV 规范

16.3.3.1 概述

ALTERNATE_TIME_OFFSET_INDICATOR TLV 格式定义见表 81。

表 81 ALTERNATE_TIME_OFFSET_INDICATOR TLV 格式

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
keyField								1	4
currentOffset								4	5
jumpSeconds								4	9
timeOfNextJump								6	13
displayName								L	19
填充								M	19+L

16.3.3.2 tlvType

tlvType 的值应是 ALTERNATE_TIME_OFFSET_INDICATOR。

16.3.3.3 keyField (UInteger 8)

keyField 的值应指示本 TLV 实体中报告的备选时标。

16.3.3.4 currentOffset (Integer 32)

currentOffset 的值是备选时间对节点时间的偏移,以秒为单位。备选时间是该值与节点时间的和。

16.3.3.5 jumpSeconds (Integer 32)

jumpSeconds 的值应是备选时间的下一个不连续时间的长度,以秒为单位。零值表示无不连续。正值表示不连续将引起备选时间的 currentOffset 增加。

16.3.3.6 timeOfNextJump (UInteger 48)

timeOfNextJump 的值应是下一个不连续将发生时的发送节点时间的秒值部分。不连续发生在 timeOfNextJump 值指示的秒的开始时刻。

16.3.3.7 displayName (PTPText)

displayName 的值是备选时标的文本名称。

注:应使用通用首字母缩略词,例如:NTP、PT、PST、PDT 分别表示网络时间协议、太平洋时间、太平洋标准时间以及太平洋夏令时间。

displayName.textField 字段(见 5.3.9)的最大符号数应为 10。

16.3.3.8 填充(Octet[M])

填充字段是长度为 M 的八位位组数组,其中 M 为 1 或 0。如果 M 为 1,则八位位组中所有比特应为 0,见 15.5.2.2。

16.3.4 ALTERNATE_TIME_OFFSET_ENABLE 管理报文

ALTERNATE_TIME_OFFSET_ENABLE 管理 TLV 允许在一个时钟中使能或禁止指示的备选时标。

该数据的维护是实现特定的。

ALTERNATE_TIME_OFFSET_ENABLE 管理 TLV 数据格式规定见表 82。

表 82 ALTERNATE_TIME_OFFSET_ENABLE 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
keyField								1	0
0	0	0	0	0	0	0	EN	1	1

16.3.4.1 keyField (UInteger 8)

keyField 的值应指示被该 TLV 实体使能或禁止的备选时标。值 FF₁₆ 应指示最高级时钟维护的所有备选时标被使能或禁止。如果该值不与一个被维护的备选时标相关,则应忽略内容并返回一个 MANAGEMENT_ERROR_STATUS TLV。

16.3.4.2 EN (Boolean)

如果 EN 为 TRUE,应将 keyField 值指示的时标的 ALTERNATE_TIMESCALE_OFFSET_INDICATOR TLV 附加到 Announce 报文。如果 EN 为 FALSE,不应附加该 TLV。

16.3.5 ALTERNATE_TIME_OFFSET_NAME TLV 规范(可选)

ALTERNATE_TIME_OFFSET_NAME 管理 TLV 允许用一个备选时标的时标偏移描述属性配置时钟。该数据的维护是实现特定的。

ALTERNATE_TIME_OFFSET_NAME 管理 TLV 数据格式规定见表 83。

表 83 ALTERNATE_TIME_OFFSET_NAME 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
keyField								1	0
displayName								L	1
填充								M	1+L

16.3.5.1 keyField (UInteger 8)

keyField 的值应指示该 TLV 实体更新或查询的备选时标。

如果值是 FF₁₆ 或者与任何被维护的备选时标不相关,该 TLV 应被忽略并返回 MANAGEMENT_ERROR_STATUS TLV。

16.3.5.2 displayName (PTPText)

displayName 的值应为备选时标的文本名,见 16.3.3.7。

16.3.5.3 填充(Octet[M])

填充字段是长度为 M 的八位位组数组,其中 M 为 1 或 0。如果 M 为 1,八位位组中所有比特应为 0,见 15.5.2.2。

16.3.6 ALTERNATE_TIME_OFFSET_MAX_KEY 管理 TLV

ALTERNATE_TIME_OFFSET_MAX_KEY 管理 TLV 允许一个管理节点确定维护的备选时标个数。

该数据的维护是实现特定的。

ALTERNATE_TIME_OFFSET_MAX_KEY 管理 TLV 数据格式规定见表 84。

表 84 ALTERNATE_TIME_OFFSET_MAX_KEY 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
maxKey								1	0
保留								1	1

16.3.6.1 maxKey (UInteger 8)

maxKey 的值应指示最大键值。

16.3.7 ALTERNATE_TIME_OFFSET_PROPERTIES 管理 TLV(可选)

ALTERNATE_TIME_OFFSET_PROPERTIES 管理 TLV 允许用备选时标的时标偏移属性配置时钟。

如果接收到该 TLV 并且报文的 actionField 为 SET, 则 currentOffset、jumpSeconds、timeOfNextJump 的更新应是原子性的。如果这些值中的任一个更新失败, 将返回 MANAGEMENT_ERROR_STATUS TLV。

该数据的维护是实现特定的。

ALTERNATE_TIME_OFFSET_PROPERTIES 管理 TLV 数据格式定义见表 85。

表 85 ALTERNATE_TIME_OFFSET_PROPERTIES 管理 TLV 数据字段

比特								八位位组	TLV 偏移
7	6	5	4	3	2	1	0		
keyField								1	0
currentOffset								4	1
jumpSeconds								4	5
timeOfNextJump								6	9
保留								1	15

16.3.7.1 keyField (UInteger 8)

keyField 的值应指示该 TLV 实体更新或查询的备选时标。

如果值是 FF₁₆或与任何被维护的备选时标不相关, 该 TLV 应被忽略并返回 MANAGEMENT_ERROR_STATUS TLV。

16.3.7.2 currentOffset (Integer 32)

currentOffset 的值是备选时间的偏移, 见 16.3.3.4。

16.3.7.3 jumpSeconds (Integer 32)

jumpSeconds 的值应是下一个不连续的大小, 见 16.3.3.5。

16.3.7.4 timeOfNextJump (UInteger 48)

timeOfNextJump 的值应是下一个不连续将发生时的时间, 见 16.3.3.6。

17 状态配置选项

17.1 概述

本章规定附加可选特征, 可用来与最佳主时钟算法联合使用以增强性能或者加强对端口状态选择的更多控制。

许多应用要求一个或多个下列性能:

- 通信网络中中断的自动恢复;
- 时钟失效的自动恢复;
- 对端口状态选择的明确控制。

本标准提供几种设计特征以满足这些要求。

普通时钟与边界时钟(第 9 章)中的最佳主时钟算法以及状态机的运行, 保证系统中存在的最佳时钟是最高级时钟的主-从层次结构的建立。这提供了从网络失效以及单个时钟失效中自动恢复的功能。恢复的速度取决于 announceInterval 以及网络拓扑。

点到点透明时钟(第10章)提供网络重新配置情况下的快速恢复。由于对等延时机制(见11.4)测量重新配置情况下所有链路上的路径延时,所以所需的路径延时校正信息立即可用。

注:这些选项应小心使用。例如,如果有些连接到通信路径的时钟被配置为使用可接受主时钟表,有些没有,则可能超过一个以上的时钟自认为是最佳主时钟。如果用不兼容的可接受主时钟表配置连接到通信路径的时钟,可能超过一个以上的端口自认为是最佳主时钟。可能发生与任何这些选项的配置机制不一致的类似误配置。

17.2 选项的数据类型

17.2.1 概述

17.2的数据类型规范可用于第17章的任何实现的选项,如下:

- PortAddressQueryTable;
- AcceptableMaster;
- AcceptableMasterTable;
- 第5章的所有规范适用于17.2定义的数据类型。

17.2.2 PortAddressQueryTable

PortAddressQueryTable类型表示端口地址以及查询间隔的列表。

```
struct PortAddressQueryTable
{
    UInteger 16 maxTableSize;
    Integer 8 logQueryInterval;
    UInteger 16 actualTableSize;
    PortAddress[actualTableSize] portAddress;
};
```

maxTableSize是实现特定的,应是actualTableSize的最大允许值。

17.2.3 AcceptableMaster

```
struct AcceptableMaster
{
    PortAddress acceptableAddress;
    UInteger 8 alternatePriority 1;
};
```

17.2.4 AcceptableMasterTable

```
struct AcceptableMasterTable
{
    UInteger 16 maxTableSize;
    UInteger 16 actualTableSize;
    AcceptableMaster[actualTableSize] acceptableMaster;
};
```

maxTableSize是实现特定的,应是actualTableSize的最大允许值。

17.3 最高级时钟簇(可选)

17.3.1 通用规范

最高级时钟簇选项要求在最高级时钟簇的成员间单播传输。当使用最佳主时钟算法的正常操作时,从一个最高级时钟的失效中恢复所需的时间取决于announceInterval。如果从一个最高级时钟切换到另一个最高级时钟的要求时间与多播announceInterval不兼容,17.3的机制可用来减少选择一个新最高级时钟的所需时间。如果实现了该选项,单播协商选项(见16.1)也应实现。

2~5个普通时钟或边界时钟可指定为一个最高级时钟簇。为正确运行,簇中的每个时钟应将defaultDS.Priority 1的值配置为:簇中所有成员的defaultDS.Priority 1的值小于域中所有其他时钟的

defaultDS, Priority 1 的值。

注：尽管设计用于增强最高级时钟的切换性能，在有些拓扑中，该机制对将主-从层次结构中处于最高级时钟下的主时钟指定为簇可能有用。用于这种目的的主时钟簇的配置不在本标准的范围内。

17.3.2 最高级时钟簇的运行

簇中的每个时钟应：

- a) 用数据类型 PortAddressQueryTable(见 17.2.2)，维护一个已配置的潜在最高级时钟的表 <grandmasterClusterTable>。表的每个 portAddress 成员都应保存最高级时钟簇成员的一个端口协议地址，以使能成员间的单播传输；
- b) 用单播报文协商选项(见 16.1)周期性地请求来自 <grandmasterClusterTable> portAddress 成员所列的所有端口的单播 Announce 报文；
- c) 如果端口正在依据本条发送一个单播 Announce 报文，并处于 MASTER 状态，将 flagField.alternateMasterFlag 设为 FALSE，否则将其设为 TRUE；
- d) 将 <grandmasterClusterTable> 的 logQueryInterval 成员的值插入到 REQUEST_UNICAST_TRANSMISSION TLV 的 logInterMessageInterval 字段；
- e) 在每个协商的单播发送超时前请求一个续订；
- f) 不管 alternateMasterFlag 的值，使用接收到的来自簇成员的单播 Announce 报文，执行最佳主时钟算法以确定它的每个端口的 portState。

如果 <grandmasterClusterTable> 为空 (actualTableSize 是 0)，除了处理 GRANDMASTER_CLUSTER_TABLE 管理 TLV，该选项应是不活动的。除非在 PTP 行规中另有规定，actualTableSize 的缺省值为 0。

该信息的维护是实现特定的，不属于第 8 章节点的数据集的部分。

17.3.3 GRANDMASTER_CLUSTER_TABLE 管理 TLV 数据字段规范

17.3.3.1 通用规范

tableSize 为 0 的 GRANDMASTER_CLUSTER_TABLE TLV 的接收将引起接收时钟把所有成员从 <grandmasterClusterTable> 中的端口地址的数组中清除。

如果 GRANDMASTER_CLUSTER_TABLE TLV 的 tableSize 成员非零，且无接收节点的端口地址包含在该 TLV 的 grandmasterClusterMembers 的列表中，则管理报文应被拒绝，并且 <grandmasterClusterTable> 不应更新。对于这种情况，managementErrorId 应为 WRONG_VALUE。

如果该 TLV 的 grandmasterClusterMembers 中的 PortAddress 数组不能完全存储，则不应更改 <grandmasterClusterTable> 并拒绝管理报文。对这种情况，如果是由于长度不匹配，managementErrorId 应为 WRONG_LENGTH，其他失败应为 GENERAL_ERROR。

另外，一旦接收到 managementId 为 GRANDMASTER_CLUSTER_TABLE 以及 actionField 值为 SET 的管理报文，时钟应用该管理报文的 grandmasterClusterMembers 替代 <grandmasterClusterTable> 的当前 portAddress 成员。标识接收时钟的成员不应进入 <grandmasterClusterTable> 中。如果任何成员更新失败，则应返回一个 managementErrorId 为 GENERAL_ERROR 的 MANAGEMENT_ERROR_STATUS TLV。

17.3.4 GRANDMASTER_CLUSTER_TABLE 管理 TLV

管理 TLV 的数据字段规定见表 86。

表 86 GRANDMASTER_CLUSTER_TABLE 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
logQueryInterval								1	0
tableSize								1	1
grandmasterClusterMembers								L	2
填充								M	2+L

17.3.4.1 logQueryInterval (Integer 8)

logQueryInterval 的值应是以 2 为底,簇成员的单播 Announce 报文间的平均间隔(以秒为单位)的对数。

17.3.4.2 tableSize (UInteger 8)

tableSize 的值应是 grandmasterClusterMembers 数组中的条目个数。条目个数的最大值应为 5。

17.3.4.3 grandmasterClusterMembers (PortAddress[tableSize])

grandmasterClusterMembers 数组的 PortAddress 元素应传送 17.3.2 的〈grandmasterClusterTable〉的成员的相应的协议地址。

17.3.4.4 填充(Octet[M])

填充字段是长度为 M 的八位位组数组,其中 M 为 1 或 0。如果 M 为 1,八位位组中所有比特应为 0,见 15.5.2.2。

17.4 备选主时钟(可选)

17.4.1 概述

该选项允许不是当前最佳主时钟的备选主时钟与从时钟端口交换 PTP 时间信息,并且从时钟端口获取它本身与每个备选主时钟间的传输路径特征。当最佳主时钟失效时,允许一个从时钟以一个小相位偏移切换到一个备选主时钟。

17.4.2 备选主时钟的报文发送

一个端口应按表 87 的限制发送多播 Announce 报文。依据 17.4 的条款发送 Announce 报文的端口应将 alternateMasterFlag(见 13.3.2.6)设为 TRUE。这些报文应以 logAnnounceInterval 定义的间隔发送,见 8.2.5.4.1。

一个端口应按表 87 的限制发送多播 Sync 报文,如果是一个双步时钟还应发送 Follow_Up 报文。依据 17.4 的条款发送 Sync 或 Follow_Up 报文的端口应将 alternateMasterFlag 设为 TRUE。这些报文应以表 87 中〈logAlternateMulticastSyncInterval〉定义的间隔发送。

注:不想使用来自备选主时钟信息的从时钟节点仅仅忽略 alternateMasterFlag 为 TRUE 的所有报文。

端口应维护表 87 中规定的可配置属性。该信息的维护是实现特定的,不属于第 8 章中节点的数据集部分。

表 87 备选主时钟属性

名称	类型	描述
〈numberOfAlternateMasters〉	UInteger 8	不处于 MASTER 状态的端口 A,当其他端口的数量小于〈numberOfAlternateMasters〉,并且这些端口: <ul style="list-style-type: none"> ● 当前正在发送合格的(见 9.3.2.5)flagField.alternateMasterFlag 为 TRUE 的 Announce 报文,这些报文正在被端口 A 接收; ● 将通过最佳主时钟算法优先于端口 A 被选为最佳主时钟; 端口 A 应发送多播 Announce 报文。〈numberOfAlternateMasters〉的缺省值应为 0
〈transmitAlternateMulticastSync〉	Boolean	如果为 TRUE 且端口当前正在发送 alternateMasterFlag 为 TRUE 的多播 Announce 报文,则端口也应发送多播 Sync 报文,如果是一个双步时钟,还应发送 Follow_Up 报文
〈logAlternateMulticastSyncInterval〉	Integer 8	以 2 为底的,依据 17.4 发送的 Sync 报文间的平均间隔(以秒为单位)的对数

注:〈numberOfAlternateMasters〉的缺省值导致多播 Announce 报文仅在端口处于 MASTER 状态时才被发送。

17.4.3 ALTERNATE_MASTER 管理 TLV 数据字段

表 87 中的备选主时钟属性可以用 managementId 为 ALTERNATE_MASTER 的一个管理报文更新。

如果 actionField 值为 SET 时接收到 ALTERNATE_MASTER 管理 TLV, <logAlternateMulticastSyncInterval>和<numberOfAlternateMasters>的更新应是原子性的。如果两者有一个更新失败,应返回一个 MANAGEMENT_ERROR_STATUS TLV。

ALTERNATE_MASTER 管理 TLV 数据格式规定见表 88。

表 88 ALTERNATE_MASTER 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	S	1	0
logAlternateMulticastSyncInterval								1	1
numberOfAlternateMasters								1	2
保留								1	1

17.4.3.1 S (Boolean)

S 的值应为<transmitAlternateMulticastSync>的值。

17.4.3.2 logAlternateMulticastSyncInterval (Integer 8)

logAlternateMulticastSyncInterval 的值应为<logAlternateMulticastSyncInterval>的值。

17.4.3.3 numberOfAlternateMasters (UInteger 8)

numberOfAlternateMasters 的值应为<numberOfAlternateMasters>的值。

17.5 单播发现(可选)

17.5.1 概述

单播发现选项允许将 PTP 用于不提供多播的网络(例如,多个 IP 网络)。从时钟端口的配置通过潜在的主时钟地址实现。从时钟可请求这些主时钟发送单播 Announce、Sync、Delay_Resp 报文给它。如果实现了该选项,则单播协商选项(见 16.1)也应实现。

17.5.2 单播发现的运行

一个普通时钟或边界时钟应维护一个已配置的主时钟表<unicastMasterTable>。该表应具有数据类型 PortAddressQueryTable, 见 17.2.2。表的每个 portAddress 成员,都保存该节点试图与之建立通信的一个远程端口的协议地址。

节点应使用单播报文协商选项(见 16.1)周期性地请求来自<unicastMasterTable>所列的所有端口的单播 Announce 报文。如果一个请求没有被一个端口授权,则该请求应在表的 logQueryInterval 成员所指示的延时后重复。

端口应维护数据类型 PortAddressQueryTable 规定的可配置属性。这些属性的维护是实现特定的。

如果<unicastMasterTable>为空(<unicastMasterTable>. actualTableSize 为 0),则除了对 UNICAST_MASTER_TABLE 和 UNICAST_MASTER_MAX_TABLE_SIZE 管理 TLV 的处理,该选项应不起作用。

<unicastMasterTable>. actualTableSize 的缺省值为 0,除非在 PTP 行规中另有规定。

<unicastMasterTable>. logQueryInterval 的值应为以 2 为底,节点请求单播 Announce 报文的平均请求间隔(以秒为单位)的对数。

<unicastMasterTable>. portAddress 成员应传送<unicast Master Table>中每个成员的相应协议地址。

17.5.3 UNICAST_MASTER_TABLE 管理 TLV 数据字段

UNICAST_MASTER_TABLE 管理 TLV 的数据字段规定见表 89。

表 89 UNICAST_MASTER_TABLE 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
logQueryInterval								1	0
tableSize								2	1
unicastMasterTable								L	3
填充								M	3+L

如果 actionField 为 SET 时接收到该 TLV, 则 logQueryInterval、tableSize、unicastMasterTable 的更新应是原子性的。如果这些值的任何一个更新失败, 则应返回一个 MANAGEMENT_ERROR_STATUS TLV。

tableSize 字段值为 0 时 UNICAST_MASTER_TABLE TLV 的接收应引起接收端口从〈unicastMasterTable〉中清除所有的〈unicastMasterTable〉. portAddress 成员。

17.5.3.1 logQueryInterval (Integer 8)

logQueryInterval 的值应是〈unicastMasterTable〉. logQueryInterval 的值。

17.5.3.2 tableSize (UInteger 16)

tableSize 的值应是〈unicastMasterTable〉. actualTableSize 的值且不大于〈unicastMasterTable〉. maxTableSize 的值。

17.5.3.3 unicastMasterTable (PortAddress[tableSize])

unicastMasterTable 数组的 PortAddress 成员应传送 17.5.2 中〈unicastMasterTable〉每个成员的相应协议地址。

17.5.3.4 填充(Octet[M])

填充字段是长度为 M 的八位位组数组, 其中 M 为 1 或 0。如果 M 为 1, 则八位位组中所有比特应为 0, 见 15.5.2.2。

17.5.4 UNICAST_MASTER_MAX_TABLE_SIZE 管理 TLV 数据字段

UNICAST_MASTER_MAX_TABLE_SIZE 管理 TLV 数据字段的规定见表 90。

表 90 UNICAST_MASTER_MAX_TABLE_SIZE 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
maxTableSize								2	0

17.5.4.1 maxTableSize (UInteger 16)

maxTableSize 的值应是〈unicastMasterTable〉的〈unicastMasterTable〉. maxTableSize 的值。

17.6 可接受主时钟表(可选)

17.6.1 概述

可接受主时钟表选项允许将从时钟端口配置为拒绝与不在可接受主时钟列表中的时钟同步。

注: 这可用于排除与可疑的欺骗或伪装主时钟的同步。

17.6.2 可接受主时钟表的操作

普通时钟或边界时钟应维护一个已配置好的表, 即 AcceptableMasterTable 类型的〈acceptableMasterTable〉, 并为每个端口维护一个可配置 Boolean 值〈acceptableMasterTableEnabled〉。

〈acceptableMasterTableEnabled〉的缺省值为 FALSE, 除非在 PTP 行规中另有规定。

〈acceptableMasterTable〉. actualTableSize 的缺省值为 0,除非在 PTP 行规中另有规定。

〈acceptableMasterTable〉. acceptableMaster 成员应传送〈acceptableMasterTable〉中每个成员的相应协议地址以及 alternatePriority 1 值。

该信息的维护是实现特定的,不属于第 8 章节点的数据集部分。

每个端口上的可接受主时钟表选项的操作规定见表 91。

表 91 可接受主时钟表选项的操作

〈acceptableMasterTableEnabled〉	操作 规 范
FALSE	〈acceptableMasterTable〉不用于该端口。协议正常运行。该端口应处理 ACCEPTABLE_MASTER_TABLE_ENABLED 管理 TLV
TRUE	<p>由最佳主时钟算法(见 9.3)确定的 E_{best} 所指示的端口应是 AcceptableMasterTable 的一个成员。</p> <p>如果正在从一个以上的〈acceptableMasterTable〉成员接收合格的 Announce 报文(见 9.3.2.5),则应使用 9.3.4 的数据集比较算法来选择 E_{best},以从该表的成员中选择作为主时钟的端口。</p> <p>如果端口的表的 AcceptableMaster 成员的 alternatePriority 1 成员是 0,则 alternatePriority 1 成员对 E_{best} 的计算不起作用。如果 alternatePriority 1 成员的值大于 0,则为了计算 E_{best},应使用该表的 alternatePriority 1 成员的值取代来自远程端口的 Announce 报文中的 Priority 1 的值</p>

17.6.3 ACCEPTABLE_MASTER_TABLE 管理 TLV 数据字段

ACCEPTABLE_MASTER_TABLE 管理 TLV 数据字段的规定见表 92。

如果 actionField 为 SET 时接收到该 TLV,则 tableSizeacceptable、MasterTable 的更新应是原子性的。如果这些值中任一个更新失败,则将返回一个 MANAGEMENT_ERROR_STATUS TLV。

表 92 ACCEPTABLE_MASTER_TABLE 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
tableSize								2	0
acceptableMasterTable								L	2
填充								M	2+L

17.6.3.1 tableSize (Integer16)

tableSize 的值应是〈acceptableMasterTable〉. actualTableSize 的值,且不应大于〈acceptableMasterTable〉. maxTableSize 的值。

17.6.3.2 acceptableMasterTable (acceptableMaster[tableSize])

acceptableMasterTable 数组的 acceptableMaster 成员应传送 17.6.2 中〈acceptableMasterTable〉的每个成员的相应协议地址以及 alternatePriority 1 值。

17.6.3.3 填充 (Octet[M])

填充字段是长度为 M 的八位位组数组,其中 M 为 1 或 0。如果 M 为 1,则八位位组中所有比特应为 0,见 15.5.2.2。

17.6.4 ACCEPTABLE_MASTER_MAX_TABLE_SIZE 管理 TLV 数据字段

ACCEPTABLE_MASTER_MAX_TABLE_SIZE 管理 TLV 数据字段的规定见表 93。

表 93 ACCEPTABLE_MASTER_MAX_TABLE_SIZE 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
maxTableSize								2	0

17.6.4.1 maxTableSize (UInteger 16)

maxTableSize 的值应是<acceptableMasterTable>的<acceptableMasterTable>.maxTableSize 的值。

17.6.5 ACCEPTABLE_MASTER_TABLE_ENABLED 管理 TLV 数据字段

该管理 TLV 数据字段的规定见表 94。

表 94 ACCEPTABLE_MASTER_TABLE_ENABLED 管理 TLV 数据字段

比特								八位位组	TLV 数据偏移
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	EN	1	0
保留								1	1

17.6.5.1 EN (Boolean)

EN 的值应是<acceptableMasterTableEnabled>的值。

18 兼容性要求

18.1 版本 2 与未来版本的兼容性

当节点接收到 versionNumber 大于 2(见 7.5.5)的 PTP 报文时,节点应丢弃该报文。

18.2 版本 1 与版本 2 之间的兼容性

不要求节点支持版本 1 与版本 2 间的转换。如果支持该转换,则应按 18.3 和 18.4 实现。

依据 PTP 版本 1 实现的 PTP 节点,与依据 PTP 版本 2 实现的节点应通过一个为该目的而设计的边界时钟进行通信。这样的一个边界时钟:

- a) 与版本 1 节点的通信是通过实现版本 1 报文发送和协议要求的端口来实现的;
- b) 与版本 2 节点的通信是通过实现版本 2 报文发送和协议要求的端口来实现的;
- c) 两种协议版本运行差别的内部解决,包括报文格式的转换以及属性值差别的解决,见 18.3、18.4、18.5 中的规定。

透明时钟转换设备不在本标准的范围内。

转换设备应只在 portDS.versionNumber 设为 1 的 PTP 端口上发送版本 1 报文。portDS.versionNumber 设为 2 的 PTP 端口上接收到的版本 1 报文应被忽略。

注 1: 在通信路径内使用的版本自动检测不在本标准范围之内。

注 2: 版本 1 与版本 2 端口间转发的多播管理报文不在本标准范围之内。

在版本 1 与版本 2 边界时钟或普通时钟节点间的转换的特定要求和限制见 18.3、18.4、18.5。

18.3 报文格式和数据类型

18.3.1 域

转换设备应:

- a) 映射版本 1 的子域值_DFLT 到版本 2 的 domainNumber 值 0,反之亦然;
- b) 映射版本 1 的子域值_ALT1 到版本 2 的 domainNumber 值 1,反之亦然;
- c) 映射版本 1 的子域值_ALT2 到版本 2 的 domainNumber 值 2,反之亦然;
- d) 映射版本 1 的子域值_ALT3 到版本 2 的 domainNumber 值 3,反之亦然。

非以上规定的域的映射不在本标准范围内。

18.3.2 版本 1 的 Stratum 与版本 2 的 clockClass

转换设备应将版本 1 的 Stratum 值映射到版本 2 的 clockClass 的值,见表 95。

表 95 版本 1 的 Stratum 到版本 2 的 clockClass

版本 1 的 Stratum	版本 2 的 clockClass
0	6
1	9
2	10
3	248
4	251
255	255

除了表 98 规定的外,转换设备应将版本 2 的 clockClass 的值映射到版本 1 的 Stratum 的值,见表 96。

表 96 版本 2 的 clockClass 到版本 1 的 Stratum

版本 2 的 clockClass	版本 1 的 Stratum
6	0
7	0
9	1
10	2
13~248	3
251	4
255	255

18.3.3 版本 1 的 grandmasterIsPreferred 和版本 2 的 Priority 1

转换设备应将版本 1 的 grandmasterIsPreferred 字段映射到版本 2 的 Priority 1 字段,见表 97。

表 97 版本 1 的 grandmasterIsPreferred 字段到版本 2 的 Priority 1 字段的转换

grandmasterIsPreferred	Priority 1
0	128
1	127

转换设备应将版本 2 的 Priority 1 字段映射到版本 1 的 grandmasterIsPreferred 和 grandmasterClockStratum 字段,见表 98。

表 98 版本 2 的 Priority 1 字段到版本 1 的转换

版本 2	版本 1	
Priority 1	grandmasterIsPreferred	grandmasterClockStratum
0~126	1	0
127	1	按表 96
128	0	按表 96
>128	0	255

18.3.4 版本 1 的 clockIdentifier 和版本 2 的 clockAccuracy 以及 timeSource

版本 1 的 clockIdentifier 同版本 2 的 clockAccuracy 和 timeSource 的映射不保留这些属性的语义。

转换设备应将版本 1 的 clockIdentifier 映射到版本 2 的 clockAccuracy 的值,见表 99。版本 2 的 timeSource 属性值应按表 99 所示设置。

表 99 版本 1 的 clockIdentifier 到版本 2 的 clockAccuracy 的映射

版本 1 的 clockIdentifier	版本 2 的 clockAccuracy	版本 2 的 timeSource
ATOM	22 ₁₆	ATOMIC_CLOCK
GPS	22 ₁₆	GPS
NTP	2F ₁₆	NTP
HAND	30 ₁₆	HAND_SET
INIT	FD ₁₆	OTHER
DFLT	FE ₁₆	INTERNAL_OSCILLATOR

转换设备应将版本 2 的 clockAccuracy 映射到版本 1 的 clockIdentifier 值,见表 100。当转换到版本 1 时,版本 2 的 timeSource 值应不予考虑。

表 100 版本 2 的 clockAccuracy 到版本 1 的 clockIdentifier 的映射

版本 2 的 clockAccuracy	版本 1 的 clockIdentifier
20~22 ₁₆	ATOM
23~2F ₁₆	NTP
30 ₁₆	HAND
31~FD ₁₆	INIT
FE ₁₆	DFLT

18.3.5 版本 1 的 grandmasterIsBoundaryClock 字段与版本 2 的 Priority 2 字段

转换设备应将版本 1 的 grandmasterIsBoundaryClock 字段映射到版本 2 的 Priority 2 字段,见表 101。

表 101 版本 1 的 grandmasterIsBoundaryClock 字段到版本 2 的 Priority 2 字段的转换

grandmasterIsBoundaryClock	Priority 2
FALSE	128
TRUE	127

转换设备应将版本 2 的 Priority 2 字段映射到版本 1 的 grandmasterIsBoundaryClock 字段,见表 102。

表 102 版本 2 的 Priority 2 字段到版本 1 的 grandmasterIsBoundaryClock 字段的转换

V2 Priority 2	V1 grandmasterIsBoundaryClock
0~127	TRUE
128~255	FALSE

18.3.6 版本 1 的 control 和版本 2 的 messageType 字段

转换设备对版本 1 的 control 和版本 2 的 messageType 字段间的映射见表 103。

表 103 版本 1 的 control 和版本 2 的 messageType 字段的映射

报文	报文类	版本 2 的 messageType 的值	版本 1 的 control 的值	版本 1 的 messageType
Sync	事件	0 ₁₆	00 ₁₆	01 ₁₆
Delay_Req	事件	1 ₁₆	01 ₁₆	01 ₁₆
Pdelay_Req	事件	2 ₁₆	N/A	N/A

表 103 (续)

报文	报文类	版本 2 的 messageType 的值	版本 1 的 control 的值	版本 1 的 messageType
Pdelay_Resp	事件	3 ₁₆	N/A	N/A
保留	—	4~7 ₁₆	N/A	N/A
Follow_up	通用	8 ₁₆	02 ₁₆	02 ₁₆
Delay_Resp	通用	9 ₁₆	03 ₁₆	02 ₁₆
Pdelay_Resp_Follow_Up	通用	A ₁₆	N/A	N/A
Announce	通用	B ₁₆	N/A	N/A
Signaling	通用	C ₁₆	N/A	N/A
Management	通用	D ₁₆	04 ₁₆	02 ₁₆
保留	—	E~F ₁₆	N/A	N/A
保留	—	N/A	05~FF ₁₆	所有其他值

18.3.7 版本 2 的 sourcePortIdentity 和版本 1 的 sourceCommunicationTechnology、sourceUuid、sourcePortId

将版本 1 的 sourceCommunicationTechnology(其值 1 用于所有一致的设备),映射到表 4 中的版本 2 的通信技术“版本 1 设备”,反之亦然。

注: 尽管版本 1 表 2 枚举了 sourceCommunicationTechnology 的许多值,但是在版本 1 中仅定义了到以太网(版本 1 附录 D)的传输映射,对应于 sourceCommunicationTechnology 的值 1。

将版本 1 的 sourceUuid 映射到 sourcePortIdentity 字段中 clockIdentity 成员的八位位组 2~7,见 7.5.2.2.3。

版本 2 的 sourcePortIdentity 的 clockIdentity 成员:

- 如果依据 7.5.2.2.3;将八位位组 2~7 映射到 clockUuid;将通信协议映射到版本 1 中为相同协议定义的 communicationId;
- 如果依据 7.5.2.2.2;版本 1 的 communicationId 应为 0。应将 EUI-64 版本 2 的 clockIdentity 的 6 个最低有效八位位组映射到版本 1 的 clockUuid 字段的 6 个八位位组。

将版本 1 的 sourcePortId 映射到版本 2 的 sourcePortIdentity 字段的 PortNumber 成员,反之亦然。

18.3.8 版本 2 的 grandmasterIdentity 和版本 1 的 grandmasterCommunicationTechnology、grandmasterClockUUID、grandmasterPortId

映射规则与 18.3.7 中的映射规则相同。

18.3.9 版本 2 的 parentPortIdentity 和版本 1 的 parentCommunicationTechnology、parentClockUuid、parentPortId

映射规则与 18.3.7 中的映射规则相同。

18.3.10 公共报文头的 FlagField

版本 1 到版本 2 的公共报文头标志的转换规定见表 104。

表 104 flagField 从版本 1 到版本 2 的转换

版本 2 的 flagField 比特	设为版本 1 属性值
currentUtcOffsetValid	如果 Stratum=1 或 2 且标识符不是 INIT 或 DFLT,为 TRUE;否则,为 FALSE
leap61	PTP_LI_61
leap59	PTP_LI_59
alternateMasterFlag	FALSE

表 104 (续)

版本 2 的 flagField 比特	设为版本 1 属性值
twoStepFlag	PTP_ASSIT
timeTraceable	如果 Stratum=1 或 2 且标识符不是 INIT 或 DFLT,为 TRUE。否则,为 FALSE
frequencyTraceable	如果 Stratum=1 或 2 且标识符不是 INIT 或 DFLT,为 TRUE。否则,为 FALSE
ptpTimeScale	如果标识符不是 INIT 或 DFLT,为 TRUE。否则,为 FALSE
unicastFlag	FALSE
所有其他 flagField	设为 FALSE

版本 2 到版本 1 的公共报文头标志的转换规定见表 105。

表 105 flagField 从版本 2 到版本 1 的转换

版本 1 标志	设为版本 2 的 flagField 比特值
忽略该信息	currentUtcOffsetValid
PTP_LL61	leap61
PTP_LL59	leap59
如果 ALTERNATE_MASTER 为 TRUE,版本 2 的报文不应被传输到版本 1 区域	alternateMasterFlag
PTP_ASSIST	twoStepFlag
如果 unicastFlag 为 TRUE,版本 2 的报文不应被传输到版本 1 区域	unicastFlag
PTP_SYNC_BURST	FALSE
PARENT_STATS	FALSE
PTP_EXT_SYNC	FALSE
PTP_BOUNDARY_CLOCK	FALSE
忽略该信息	所有其他标志

18.3.11 版本 2 的 logMessageInterval 和版本 1 的 Sync 报文的 syncInterval

在转换到版本 1 时忽略版本 2 的 Announce、Delay_Req、Delay_Resp 报文的 logMessageInterval。

将版本 2 的 Sync 或 Follow_Up 报文的 logMessageInterval 转换为版本 1 的 Sync 报文的 syncInterval。

将版本 1 的 Sync 报文的 syncInterval 转换为版本 2 的 Announce、Sync、Follow_Up 报文的 logMessageInterval。

将版本 1 的 Sync 报文的 syncInterval+5 转换为版本 2 的 Delay_Req 和 Delay_Resp 的 logMessageInterval。

注：通过将版本 1 的 syncInterval 加 5 来获得版本 2 的 Delay_Req 和 Delay_Resp 的 logMessageInterval 是必要的，以满足 7.7.2.4 的要求。

18.3.12 版本 2 的类型 ClockQuality 字段和版本 1 的 Stratum、标识符和方差字段

版本 2 的 grandmasterClockQuality 字段在版本 1 的对应字段映射如下：

- 版本 2 的 clockClass 成员到 grandmasterClockStratum 的映射是依据 18.3.2；
- 版本 2 的 clockAccuracy 成员到 grandmasterClockIdentifier 的映射是依据 18.3.4；
- 版本 2 的 offsetScaledLogVariance 成员在用 $8\,000_{16}$ 校正偏移后直接映射到 grandmasterClockVariance,见 7.6.3.3。

18.3.13 版本 2 的类型 Timestamp 字段与版本 1 的 epochNumber 和类型 TimeRepresentation 字段

版本 2 的 nanosecondsField 成员(UInteger 32)映射到版本 1 的 nanoseconds 成员(Integer 32)。

版本 1 的 nanoseconds 成员如果为正,则映射到版本 2 的 nanosecondsField 成员;如果为负,由于版本 2 不允许负的时间戳,则将产生一个错误。

版本 2 的 secondsField 字段(UInteger 48)的最低有效 32 比特直接映射到版本 1 的 seconds 字段,反之亦然。

版本 2 的 secondsField 字段(UInteger 48)的最高有效 16 比特直接映射到版本 1 的 epochNumber,反之亦然。

18.3.14 与版本 1 无对应部分的版本 2 的字段

对表 106 中每个版本 2 的字段,转换设备应采取规定的动作。

表 106 与版本 1 无对应部分的版本 2 的字段

版本 2 的字段	报文	版本 2 到版本 1	版本 1 到版本 2
transportSpecific	公共报文头	忽略该信息	依据适用附录 D~附录 I 设置
messageLength	公共报文头	忽略该信息	依据 13.3.2.4 设置
correctionField	公共报文头	不在本标准范围内	设为 0

18.3.15 与版本 2 无对应部分的版本 1 的字段

对表 107 中每个版本 1 的字段,在转换到版本 2 时,转换设备应采取规定的动作。

表 107 与版本 2 无对应部分的版本 1 的字段

版本 1 的字段	报文	动作
versionNetwork	公共报文头	忽略该信息
messageType		
utcReasonable	Sync 和 Delay_Req	忽略该信息
localClockVariance		
localClockStratum		
localClockIdentifier		
associatedSequenceId	Follow_Up	映射到版本 2 中的公共报文头的 sequenceId
requestingSourceSequenceId	Delay_Resp	映射到版本 2 中的公共报文头的 sequenceId
managementMessageKey	Management	每个管理报文的格式和语义在版本 1 和版本 2 间映射
parameterLength		
messageParameters		

18.4 命名变化

表 108 给出了语义保持不变的情况下同样的参量在版本 1 和版本 2 中名称的对应。

注:语义已发生变化的参量包含在前面的章节中。

表 108 名称对应

版本 1 名称	版本 2 名称
estimatedMasterVariance	portDS, observedParentOffsetScaledLogVariance
estimatedMasterDrift	portDS, observedParentClockPhaseChangeRate

18.5 对混合版本 1 和版本 2 系统的限制

第 18 章的转换规范允许符合本条规定限制的版本 1 和版本 2 的混合系统,版本 1 和版本 2 的混合系统应按图 36 所示进行配置,并遵从表 109 的任意一行的限制。版本 1 的 Stratum 3 时钟不应用于版本 1 和版本 2 的混合系统的实现中。

也可能有其他的系统配置和限制,但不在本标准范围内。

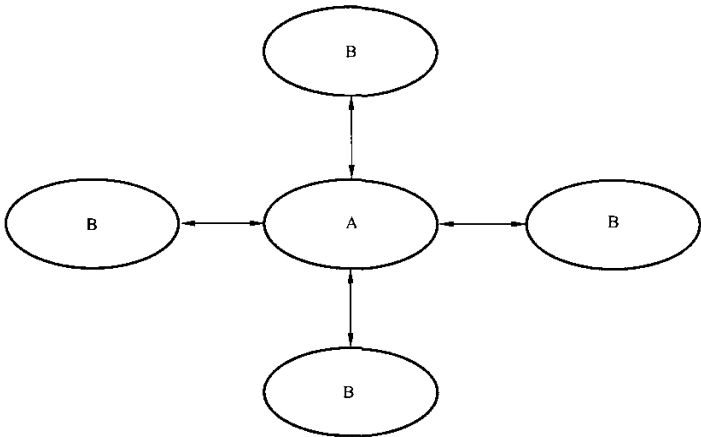


图 36 允许的混合系统配置

图 36 示出区域 A 实现了一个 PTP 版本,且与一个或多个实现了第二个 PTP 版本的区域相连接。

表 109 混合系统限制

区域 A 特征	区域 B 特征	限制
版本 2 的时钟	版本 1 的时钟	系统中无 clockClass 6 或 7 的时钟,或 Stratum 1 或 Stratum 2 的时钟
版本 2 的时钟	版本 1 的时钟	区域 A 至少包含一个 clockClass 6 或 7 的时钟,并且时钟 Priority 1 <128
版本 2 的时钟	版本 1 的时钟	任意 Stratum 1 或 2 的时钟包含在一个单一 B 区域
版本 1 的时钟	版本 2 的时钟	所有区域 B 的时钟的 Priority 1 >128
版本 1 的时钟	版本 2 的时钟	所有区域 B 的时钟的 Priority 1 >127 且有一个区域 A 的时钟的 grandmasterIsPreferred=TRUE
版本 1 的时钟	版本 2 的时钟	任意 clockClass 6 或 7 的时钟以及 Priority 1 <128 的任意时钟均包含在一个单一 B 区域中

19 一致性

19.1 一致性目标

本章中一致性要求的基本原则是:

- 提高依据本标准建立的系统的互操作水平;
- 用可能最广泛的应用范围来鼓励 PTP 组件制造商;
- 为持续的技术改进和演变提供机会。

19.2 PTP 一致性要求

19.2.1 通用一致性规范

一致性要求是针对节点规定的。

节点应符合本标准的所有条款,除了:

- 明确标有“可选的”;
- 对只发布频率且不要求测量路径延时的应用,一个备选 PTP 行规可规定不实现或激活 11.3 和 11.4 的路径延时机制。

对每个实现的选项,节点应符合规定该选项的条款。

19.2.2 传输一致性规范

使用在本标准附录中定义映射的传输协议的节点应符合该附录。

使用没在本标准中定义映射的传输协议来传输 PTP 包时,应由对该传输具有管理权限的标准化组织或其指定机构定义并发布的映射来定义该传输。规定该映射的出版物应由一个 PTP 行规引用。

注:定义传输映射的组织须为传输保留一个枚举值,见 7.4.1。

19.2.3 PTP 行规一致性规范

声称兼容性的节点应规定至少一个它所遵守的 PTP 行规。在缺乏一个合适的备选 PTP 行规时,应使用两个缺省 PTP 行规中的一个。缺省 PTP 行规的规定见附录 J。

如果一个特定的属性或选项在所选的 PTP 行规中未作规定,则节点应符合规定相同路径延时机制的缺省 PTP 行规中规定的值或选择。

所有的 PTP 设备都应支持一个缺省 PTP 行规。

19.3 PTP 行规

19.3.1 PTP 行规概述

19.3.1.1 概述

PTP 行规的目的是允许组织对 PTP 的属性值和可选特征规定特定的选择。当使用相同的传输协议时,这些特定的选择交互工作并获得满足特定应用要求的性能。

一个 PTP 行规是一组要求选项、禁止选项以及可配置属性的范围和缺省。行规规范应与 19.2.1 和 19.2.2 中的规范一致。

19.3.1.2 PTP 行规推荐

一个 PTP 行规应定义:

- 要实现的最佳主时钟算法选项(见 9.3.1);
- 要实现的配置管理选项(见 15.1.1);
- 要实现的路径延时机制,延时请求-响应(见 11.3)或对等延时(见 11.4);
- PTP 所有可配置的属性和数据集成员的范围及缺省值;
- 要求的、允许的或禁止的传输机制;
- 要求的、允许的或禁止的节点类型;
- 要求的、允许的或禁止的选项。

一个 PTP 行规只能通过以下方式扩展本标准:

- a) 使用 14.3 的 TLV 机制;
- b) 可选最佳主时钟算法规范,见 9.3.1;
- c) 可选管理机制规范,见 15.1.1;
- d) 19.2.2 的规定;
- e) 7.3.1 的规定。

19.3.2 特定的 PTP 行规

可以开发 PTP 行规的外部组织包括:

- a) 在行业中拥有管理权限的公认的标准化组织,例如:IEC、IEEE、IETF、ANSI 或 ITU;
- b) 行业贸易协会或其他类似的在行业中拥有行业标准权威的公认的组织;
- c) 其他合适的组织。

PTP 行规开发组织应咨询 IM/ST 委员会精确网络化时钟同步工作组进行技术审查。

19.3.3 PTP 行规规范

PTP 行规应由行规文档起始处印刷的以下文本标识,如图 37 所示。

在图 37 中指示的各条定义如下:

- profileName:该字段应是行规制定组织指定的行规文本标题,见 J.3.1 的示例。
- profileVersion:该字段应是行规制定组织指定的行规版本,版本号有两个字段组成:一个 primaryVersion (UInteger 16) 和一个 revisionNumber (UInteger 8)。ProfileVersion 应印刷为“版本 primaryVersion, revisionNumber”。
- profileIdentifier:该字段应为一个 EUI-48。EUI-48 的 OUI 部分由行规制定组织拥有。该组织应保证对组织规定的每个行规和版本 profileIdentifier 是唯一的。EUI-48 剩余的八位位组应依次为 primaryVersion 和 revisionNumber。见 J.3.1 的示例。
- organizationName:该字段应是制定行规以及拥有行规标识符 OUI 的组织的文本名称。
- sourceIdentification:该字段应是一个 URL、e-mail 或常规邮件地址,可向其发送关于行规的查询或复制请求。

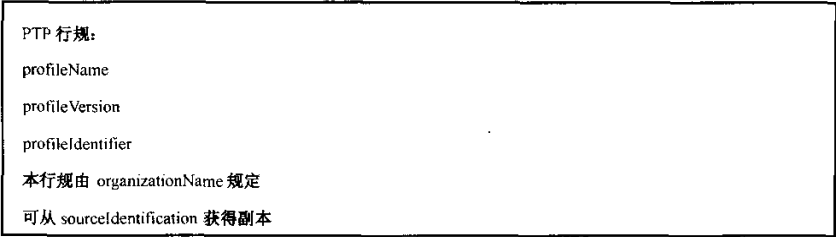


图 37 行规印刷形式

附 录 A
(资料性附录)
使用 PTP

A.1 概述

在分布式系统中,PTP 为精确时钟同步提供了简单的算法。当设计这样的系统时,需要解决下列问题:

物理设计问题:

- 时钟是如何物理分布的?
- 应用什么网络技术?

逻辑问题:

- 系统是单一时钟组,还是时钟被分为几个逻辑组,每个组有自己的时间观念。

组件问题:

- 时钟需同步到怎样的精度?
- 什么是系统的时钟源? 它应溯源到 UTC 吗?

本地实现问题:

- 如何满足时序要求?
- 共享通信网络其他应用如何影响 PTP?
- 精度要求如何影响实现?
- 本地振荡器的设计问题是什么?

系统实现问题:

- 系统如何划分?
- 使用哪些选项?
- 使用哪些行规?

性能问题:

- 网络延时和波动如何影响时间精度?
- 时钟振荡器的稳定性如何影响时间精度?

一致性测试问题:

- 有助于一致性和性能测试的特征;
- 有助于校准设备时间的特征。

A.2~A.9 分别阐述每个主题。

A.2 物理设计

时钟通过网络彼此通信。典型地,网络技术的选择是基于主要应用。PTP 工作在任何基于数据包的环境中。尽管有可能设计单播 PTP 组件和系统,但仍将 PTP 设计成工作在多播的环境。以太网是实现 PTP 的理想网络,该附录的其余部分将以以太网为例。

任何网络都要受到距离、允许节点数和流量的限制。如果要同步的时钟分布超出网络技术的范围,则可以将系统设计为分隔的“时间岛”,同步这些时间岛超出 PTP 范围。

例如,如果系统由相距几英里的两个紧凑的区域组成,在每个区域内可使用 PTP,则区域间的同步由另一种技术如 GPS 提供。

在一个区域内,距离、流量和节点个数问题通常通过特定网络组件解决。对于以太网,局部化的节

点一般通过网桥通信。对于更大、更复杂的系统,用路由器将系统分为几个仅使用网桥的区域。通常,用这些设备分隔出的每个层次,会给报文在节点间的传输时间引入额外的统计延时和延时波动。

PTP 被设计为使延时和延时波动的影响最小。为达到最好 PTP 性能,在具有最严格的时钟同步要求的时钟之间,网络拓扑应将分隔设备的数量限制为最小。

边界时钟能改进由路由器进行分隔的网络性能,或代替普通网桥。透明时钟也能代替普通网桥,尤其在许多设备连接成线性拓扑的环境中。

没有实现 PTP 的网桥可能引入明显的时间抖动和路径不对称。尽管在实现 PTP 的系统中可能包含这样的网桥,但不应使用这些网桥,除非应用可以容忍由它们的时间抖动和路径不对称引入的时间误差,或者通过适当的滤波算法将时间误差减小。

A.3 逻辑设计

大多数应用包含要实现同步的一组时钟。在这种情况下,所有时钟可以放置在单个域中。如果使用在本标准中规定的缺省值和适用的 PTP 行规,通常不必要组态这些时钟。

如果应用需要几组时钟,每组时钟保持一个不同的自相容时基,那么可以应用下列两种方案的一种:

- 如果应用的其余部分被分在同一个组,也可能用分离的非通信网络;在那种情况下,每组能使用缺省的域。网络路由器通常就是用于这个目的。
- 如果各个组必须共享同一个网络,那么每个组可能被分配到不同的域中。按期望逻辑地划分时钟。在每个时钟上的处理负荷可能受影响,也可能不受影响,这取决于到网络底层物理地址的映射。

除了将 PTP 节点分配到一个域中,在缺省的情况下,PTP 定义一个自由管理系统。在一个域内,只要组件使用推荐的多播通信模型,PTP 节点可被增加或被移除,而不必修改地址列表。增加或移除 PTP 节点可能使一个不同的时钟成为系统的最高级时钟。当系统自动重新校准新延时模式到新的最高级时钟,这可能在时基中引起一个瞬变。

本标准为用户提供了几个组态选项,这些用户要求对主时钟的选择或者决定系统性能的不同时序和其他属性实施更多控制。例如,使用 Priority 1 的属性,允许系统设计者为最高级时钟的选择,按优先级顺序指定最多 254 个设备。

A.4 组件问题

选择 PTP 系统组件的基本问题是要求的同步精度。

- 应选择支持给定精度要求的协议特性的时钟;
 - 网络组件和物理设计决策也影响前面章节所述精度。
- 合理设计的以太网 PTP 系统能容易的实现亚微秒级精度。

第二个问题是建立 PTP 系统历元的技术。在每个域中,历元由根据最佳主时钟算法选择的最高级时钟定义。

如果 TAI 或者 UTC 可溯源的时间是必须的,那么最高级时钟必须维护一个 PTP 时基。

在一个域中,如果最高级时钟的 clockClass 的最小值是 6,7,52 或 187,则时基是 PTP。从 PTP 时标,可使用 PTP 分发的 currentUtcOffset 值计算 UTC。断电后这样系统可维护也可不维护该历元;见 7.6.2.4。

在一个域中,如果最高级时钟的 clockClass 的最小值是 13,14,58,193,216 或者更大,则时基是 ARB 或者是由用户建立的一个时基;见 7.6.2.4。这样的系统断电后可维护也可不维护历元。

主时钟失效可能是由于它的时间或频率不正确。该问题的检测和恢复不在本标准范围内。某些信息,如在 parentDS 数据集中维护的父时钟统计数据,可用来帮助检测一个“false-ticking”(伪标)主时

钟。建议实现者考虑来自尽可能多的时钟信息,并根据时钟固有稳定性来衡量来自每个时钟的信息。

DISABLE_PORT 管理报文可用来帮助从“false-ticking”主时钟的恢复。注意停用或降级一个主时钟有副作用(尤其当它是一个边界时钟),因此决定是否那样做还要考虑除了它的计时质量以外的因素。这个决定超出本标准的范围。

A.5 本地实现问题

A.5.1 概述

本条为 PTP 普通时钟、边界时钟和透明时钟的实现提供了一些指南。尽管实现不在本标准的范围内,但实现必须考虑在通过 PTP(或其他协议)进行同步的时钟顶部建立的服务,不得降低时钟精度。

A.5.2 时序问题

实现必须满足报文处理和时序要求,也必须满足操作基于 PTP 报文信息同步本地时钟的伺服系统所必需的任何时序要求。

实现必须确保有足够的计算和内存资源可用以满足这些要求。实现也必须确保执行 PTP 所需的资源比其他共享这些资源的应用具有足够高的优先级,以满足 PTP 和伺服系统的时序要求。在一个实现中,PTP 任务应该被分配最高的优先级,与分配给协议栈和其他操作系统资源的优先级类似。

在每个 syncInterval 中,PTP 实现通常需要一个短时的资源占有。系统 syncInterval 的选择必须与所有系统组件中的可用资源一致。

其他应用对网络资源的使用会影响 PTP 精度,见 A.5.3。

A.5.3 精度问题

A.5.3.1 概述

PTP 系统可达到的精度受以下因素限制:

- 时钟协议栈的延时波动;
- 延时的不对称;
- 网络组件的延时波动;
- 时间戳的精度;
- 稳定性问题。

A.5.3.2 协议栈延时波动

最简单的 PTP 实现像普通应用一样运行在网络协议栈的顶部。时间戳在应用级生成。协议栈的延时波动导致这些时间戳的偏差。典型地,取决于操作系统,这些偏差处于几百微秒到毫秒(μs 到 ms)的范围。

实现可在中断级而不在应用级生成时间戳。这种情况下典型地,延时波动可被降低到几十微秒(μs),这取决于其他应用对其他中断的使用以及网络的通信模式。

使用硬件辅助技术可最大程度降低由协议栈延时波动引起的偏差。这些硬件辅助技术在协议栈的物理层生成时间戳。典型地,在该点上的延时波动可在纳秒(ns)范围内。例如,在以太网系统中,这些偏差是由 PHY 芯片的锁相特性引入的,该 PHY 芯片从输入数据流中修复时钟和数据的同步。该延时波动的影响可通过时钟伺服算法的合理设计来减低。

A.5.3.3 网络组件延时波动

网络组件会在报文的传播时间中引入波动。这直接影响 currentDS、offsetFromMaster 和 currentDS、meanPathDelay 值的精度。

网桥和路由器会受到存储-转发延时波动的影响。典型的以太网网桥具有通过高速背板或交换结构进行通信的输入和输出缓冲器。每个端口典型地直接连接到一个终端设备或其他以太网网桥。产生延时波动的最主要因素是输出缓冲和输出排队。如果输出子网总是可用的,这个延时波动典型地在纳秒范围内,并且可通过平均技术将其减小。到一个包含时钟的节点的密集通信可增大由于其输出缓冲

而导致的延时波动。该增大的延时波动更不易被降低。正确的 PTP 系统设计必须考虑该影响并采取措施来降低影响。

大多数网桥和路由器支持通信优先级划分。高优先级通信受到的传输时间波动较小。只要有可能,PTP 事件报文与其他数据相比,应按高优先级发送。为每种传输协议推荐的特定优先级,见附录 D~附录 I。

A.5.4 时间戳精度

PTP 要求的生成时间戳的时钟分辨率必须与期望精度一致。注意该分辨率与 PTP 方差相关,见 7.6.3。

A.5.5 稳定性问题

如上所述,引入到 `currentDS.offsetFromMaster` 和 `currentDS.meanPathDelay` 成员计算的延时波动,可通过对本地时钟任何同步伺服算法的适当设计而减小。必须在平均次数(采样数)和对延时波动以外影响(如振荡器的稳定性)的响应速度之间进行权衡。

本地时钟的基本时间稳定性必须与要求的 `syncInterval` 和精度规范一致。在比平均值算法间隔小的时间间隔中,用来减小延时波动的算法不校正本地时钟的漂移。伺服算法不能校正 `syncInterval` 内发生的随机漂移。

在高精度情况下,要满足驱动本地时钟的本地振荡器的稳定性规范十分困难。在成本与稳定性之间进行权衡。典型地,本地晶振器是石英晶体。石英晶体的频率典型地由于热量、机械和老化影响而漂移。这些影响中,热量影响是大多数应用中最难解决的。

例如,一个典型无补偿的晶体的热规格是每摄氏度 1 PPM。2 s 的 `syncInterval` 内温升 1 °C 会产生大约 2 μ s 的偏差。因此,几十纳秒范围内的精度意味着,通过更好的晶体热规格、减小的 `syncInterval`,和更好的热管理的一些组合来降低两个数量级的热漂移。

随着计算和网络带宽要求的相应增加,PTP 允许根据所选的 PTP 行规将 `syncInterval` 降低到小数秒。

1 PPM/摄氏度以下的晶体热规格越来越难以达到。温度环境的控制必须认真管理,尤其在高精度实现中。典型地,很长的平均时间需要耐高温晶体或使用更稳定的振荡器。PTP 系统的短时热漂移和典型平均时间,经常通过周围设备的散热、节点内的冷却方式、增加振荡器的热质量和类似技术来管理。见 Sullivan, Allan et al[B24]对时钟特性的详细讨论。

A.6 系统实现问题

PTP 系统是在一起运行的以满足应用需求的 PTP 组件集合。可互操作 PTP 系统的协议按本标准规定运行,节点的选择和配置使得协议成功地构建一个主-从时钟层次结构,且在一个端口处于 SLAVE 状态的节点能够与一个端口处于 MASTER 状态的节点同步。一个最佳的 PTP 系统是可互操作的、可管理的,且可满足应用同步要求的系统。确保由一致的 PTP 节点组成的系统是最佳系统,这对于系统集成者而言是一个问题。以下推荐有益于构建可互操作的系统。

- 在整个域中使用单一的传输,或将域划分成数个区域,每个区域使用单一传输,区域间使用边界时钟连接。
- 在整个系统中使用单一的管理方法。无论是本标准的管理报文机制还是 PTP 行规中的可选管理机制都是可接受的。
- 在整个域中使用相同的最佳主时钟算法选择。即使使用边界时钟连接,不保证其几个区域使用不同最佳时钟算法选择的域具有可互操作性。可使用本标准定义的最佳时钟算法,或 PTP 行规中规定的可选算法。
- 在整个域中使用相同的状态配置选项的选择(第 17 章)。如果使用了状态配置选项,系统集成者有责任确保所选的配置产生一个可互操作的系统。即使使用边界时钟连接,不保证其几个

区域使用不同配置选项选择的域具有可互操作性。

- 在整个域中使用单一的路径延时机制(见 11.3 和 11.4),或将域划分成几个区域,每个区域使用单一的路径延时机制。区域间使用一个或多个边界时钟连接。
- 在整个域中使用可互操作的属性和可组态数据集的值集,或者将域划分成几个区域,每个区域使用单一的可互操作集。区域间使用一个或多个边界时钟连接。
- 对系统中所有节点的每个属性和可组态数据集成员,使用相同的缺省值。
- 对系统中所有节点的每个属性,使用相同的要求最大范围值和要求最小范围值。
- 为了使选项按设计运行并避免可互操作性问题,一些选项必须在系统中的每个节点都存在并激活,例如实验性安全选项。其他选项在实现它们的节点子集上是有效的,即使系统中的其他节点不支持这些选项。而且,这样节点的存在不影响没有实现这些选项的节点,例如单播选项。
- 在整个域中只使用实现相同 PTP 行规的节点,或者将域划分为几个区域,每个区域使用同样的 PTP 行规。区域间使用边界时钟连接,该边界时钟具有解决 PTP 行规差异的能力。不保证两个 PTP 行规规范可用于设计能解决差异的边界时钟。例如,不可能确保使用本标准的最佳主时钟算法实现自配置的区域,能够与使用主—从层次结构配置的区域实现互操作。
- 在整个域中只使用实现 PTP 相同版本的节点,或者将域划分成几个区域,每个区域使用同样的版本(版本 1、版本 2 或者将来的版本)。区域间使用一个边界时钟连接。

A.7 性能

为实现最佳时钟同步性能,应满足下列条件:

- a) 主时钟和从时钟之间网络延时应对称。
- b) 时钟的时间戳机制或协议路径可能存在不对称延时。如果这些不对称不可忽视,则它们应得到正确的解决,见 11.6。
- c) 主时钟和从时钟之间的网络延时,在 Delay_Req 报文间时间间隔内应为常量。
- d) 由于网络组件和时钟协议栈引起的延时波动应通过以下两个技术来减小:
 - 1) 对于一个给定时钟的实现,PTP 中所用的时间戳应尽量靠近物理层产生。只有报文确实被发送后才能产生最精确时间戳的情况下,实际值在主时钟发出的 Follow_Up 报文或者在点到点透明时钟发出的 Pdelay_Resp_Follow_Up 报文中传输。
 - 2) 由协议栈和网络组件引入的、且不能通过边界时钟或透明时钟隔离的残余延时波动,可以通过求平均值来减小。平均值算法超出本标准的范围。
- e) 实现协议的时钟的计算能力必须足够大,并且时钟数量必须足够小,才能满足时间限制。例如,边界时钟和普通时钟的实现者,需考虑处理与节点通信的从时钟发出的 Delay_Req 报文所需的资源。由于资源限制而不能处理这些报文可能导致错过路径延时测量而使同步性能降低。当选择节点和设计系统时,用户需意识到这些限制。
- f) 时钟振荡器必须有足够的固有稳定性和精度,见 A.5.4 和 A.5.5。

注:对于帮助生成这些时间戳的机制,参见 Eidson, et al. [B4], IETF RFC1589[B8] 和 IETF RFC2783[B13]。

A.8 有助于一致性测试的推荐

为帮助:

- 测试 PTP 系统的性能;
- 校准 PTP 设备;
- 校验一致性。

所有 PTP 普通时钟和边界时钟应每秒钟提供一个脉冲(PPS)信号,并且脉冲的上升沿与时钟秒计

时部分的每次增加相一致。如果实现了但不一致,则设备规范应包括 1 PPS 信号与秒增加事件时间的偏移。该信号可以是一个可访问的内部测试点,而不必作为嵌入时钟的设备(如传感器)的可见外部输出。

A.9 在单播网络或具有非 PTP 网桥和路由器网络中的实现推荐

A.9.1 概述

在具有不支持 PTP 标准的网桥和路由器的网络中,引入 PTP 主时钟和从时钟。此外,许多网络不支持多播传输。

单播通信模式能用来解决这些问题的一些。只要保留协议行为,7.3.1 允许使用单播模型。

本条描述当使用单播通信模型时在备选 PTP 行规中必须规定的问题,以便完成的实现能够在满足宽范围时间要求的网络中工作。一些广域网的要求,如安全性和恢复性,超出本探讨的范围。

A.9.2 单播模型中的边界时钟和透明时钟

在多播模型中,普通时钟和边界时钟自动创建一个同步层次结构而不必事先知道网络拓扑。除管理报文外,保证边界时钟终止所有 PTP 报文。而且,如果只存在 PTP 网桥、路由器、透明时钟和普通时钟,保证对等延时机制所使用的报文在相邻点到点时钟终止,从而确保机制正常运行。然而,在单播模型中,上述条件不适用。

为了保留协议行为,当使用单播模型时,必须实现下列功能:

- 正确创建同步层次结构;
- 正确交换同步所用的时间报文和相关联通用报文;
- 正确运行测量路径延时的对等延时或延时请求-响应机制;
- 配置时钟的管理机制。

实现这些功能的一个方法是,对所有普通时钟、边界时钟和点到点透明时钟进行预先配置,使得从每个端口可看见相邻时钟的单播协议地址。上述的一个例外是,如果使用 16.1 的单播选项,对于使用延时请求-响应机制的只能作为从时钟的时钟,其地址不必在其他时钟上预先配置。如果使用对等延时机制,配置必须确保从每个端口只能看见一个点到点时钟,见 11.4.4。

即使端口之间有网桥、路由器或透明时钟,时钟端口也可能是相邻的。如果端口之间是一个边界时钟,那么时钟端口不是相邻的。当网络重配置时,相邻关系可能改变;在这种情况下,两个端口可通过单播穿过边界时钟进行通信。如果一种可学习拓扑改变的机制可用,时钟能停止非相邻节点之间的所有单播通信,以生成优化的同步层次结构并较好地利用网络资源。如果这样的学习机制不可用,取决于网络拓扑,可使用端到端透明时钟来代替边界时钟或点到点透明时钟。在所有情况下,为实现协议的正确运行,实现必须提供一种可打破转发环路的机制。

A.9.3 单播选项

第 17 章的配置选项可用来为每个端口配置所需的单播协议地址。

16.1 的单播选项可用来为 Announce、Sync、Delay_Resp、Pdelay_Resp 报文和任何相关通用报文建立单播通信。

可选地,可通过管理程序创建两个节点间的单播合约。这些合约包括单播地址信息和 Sync、Announce 和 Delay_Req 报文各自规定的数据包速率。

16.2 的路径追踪选项可用于定义一种打破转发环路的机制。

因为 15.2 的管理机制依赖多播模型的使用和边界时钟的转发,用来配置时钟的可选管理机制必须按 15.1.1 允许的进一步规范。

A.9.4 单播一致性

A.9.4.1 概述

19.2.3 规定要声明一致性,一个节点除必须符合 PTP 标准外,还必须符合一个 PTP 行规。该行

规必须说明与附录 J 中缺省 PTP 行规规范的任何差异。A.9.4.2 包括了一些规范的示例,这些规范是实现单播模型以满足 A.9.1 和 A.9.2 所讨论的要求所必需的。没有讨论的可能是可选的最佳主时钟算法和可选的基于单播的管理机制。

A.9.4.2 PTP 选项和属性值

须支持 16.1 和 17.5 中定义的单播选项,且缺省是可运行的。允许 15.5.4.1.7 和第 17 章的所有其他选项,但缺省地须是非激活的。

如 7.3.1 允许的,缺省使用单播通信模型式。如果也实现了多播通信型,缺省地必须是非激活的。多播通信是一个推荐选项,为了在这些网络中扩展将来的多播支持,并且允许与支持 PTP 缺省行规的设备实现互操作。

单播报文的时序由单播协商 REQUEST_UNICAST_TRANSMISSION TLV 中的 logInterMessagePeriod 字段的值决定。

REQUEST_UNICAST_TRANSMISSION TLV 的 logInterMessagePeriod 字段的推荐值如下:

- 请求单播 Announce 报文:logInterMessagePeriod 的缺省初始化值是 1(每 2 次 1 s),可组态范围是-3(每 8 次 1 s)~3(每 8 次 1 s);
- 请求单播 Sync 报文:logInterMessagePeriod 的缺省初始化值是-4(每 16 次 1 s),可组态范围是-7(每 128 次 1 s)~1(每 2 次 1 s);
- 请求单播 Delay_Resp 报文:logInterMessagePeriod 的缺省初始化值是-4(每 16 次 1 s),可组态范围是-7(每 128 次 1 s)~6(每 64 次 1 s)。

每个 REQUEST_UNICAST_TRANSMISSION TLV 的 durationField 值的缺省初始化为 300 (300 s),可组态范围是 10~1 000。

这些缺省值和组态范围值的维护和配置是实现特定。

在实现 GRANT_UNICAST_TRANSMISSION TLV 的机制中,只要请求在可组态范围内,准入值应与接收到的 REQUEST_UNICAST_TRANSMISSION TLV 中所请求的值相同。

注:由于传输可能是不可靠的,请求端口在一个实现特定的超时以后,如果没有接收到准入 TLV,应重复该请求。

对于接收连续服务,请求方应在准入周期结束之前重新发出该请求。如果没有接收到准入值,推荐的提前量应包含足够的空余时间来至少重发请求两次以上。

defaultDS, announceReceiptTimeout, defaultDS. Priority 1, defaultDS. Priority 2, defaultDS. slaveOnly 和 τ 的值与 J.3 中的规定一致。

物理要求与 J.3 中的规定一致。

附 录 B

(资料性附录)

PTP 中的时间戳和历元

B.1 总则

本附录中许多主题的详细讨论见 Allan 等[B1]、Allan 等[B2]、IETF RFC 1305(1992)[B7]、ISO 8601:2004[B17]、<http://tycho.usno.navy.mil/time.html>[B18]和 Sullivan 等[B24]。

在一个域内部,时间特性由该域的最高级时钟决定。

最高级时钟决定:

- a) 时间增长速率。最高级时钟频率精度由任意两个事件之间决定的时间间隔的精度进行测量的,正如最高级时钟测量的,对应于使用符合国际定义时刻的时钟测量的同一时间间隔。国际定义时刻(SI)是定义 TAI 时标的时间测量,该 TAI 时标由巴黎附近的国际度量衡办公署维护。

- b) 时标的原点或历元

PTP 最高级时钟可用的可能时标和历元如下:

——PTP 时标:由 timePropertiesDS. ptpTimescale 值为 TRUE 指示。该历元是 PTP 历元。

——ARB 时标:由 timePropertiesDS. ptpTimescale 值为 FALSE 指示。该历元是实现特定的。

B.2 UTC、TAI 和 PTP 历元

TAI 和 UTC 是基于如同在旋转水平面上实现的 SI 时刻的时间国际标准。UTC 由一组原子时钟实现,并形成公共使用的其他时标的守时基础。

UTC 是大多数工程和商业关心的时标。UTC 表示法由 ISO 8601:2004[B17]规定为:用 YYYY-MM-DD 表示日期,用 hh:mm:ss 表示每天的时间。UTC 时间增长速率与 TAI 速率是相同。UTC 与 TAI 的时差是个恒定偏移。该偏移通过加上或减去闰秒来校正。TAI 连续地增长,而由于的闰秒的引入 UTC 则为不连续地增长。

从 UTC 时间 1972 年 1 月 1 日 0 时开始(校正的儒略日(MJD)41317.0)⁶⁾,世界的标准时间系统启用闰秒实现,以允许仅用整数秒校正 UTC 时刻和 TAI,它们两个都用 d、h、min 和 s 来表示。这天的 TAI 减 UTC 为 10 s。在 1972 年 1 月 1 日以前,UTC 和 TAI 之间偏移的校正以秒的小数部分实现。

应用在 UTC 而不是 TAI 中的闰秒校正法,推荐在六月或十二月的最后一天的 23:59:59 这一时刻后进行。第一个这样的校正,单个正闰秒校正是在 UTC 时间 1972 年 6 月 30 日 23:59:59 后进行的,且在该时刻后 UTC 落后 TAI 1 s。

注:在 UTC 时间 2006 年 1 月 1 日 0 时,TAI-UTC=+33 s。

在计算机系统中,通用基于 POSIX 时间转换算法典型地用于生成 TAI 和 UTC 正确的 ISO 8601:2004[B17]印刷表示法。UTC 落后 TAI 固定闰秒数。

设置 PTP 历元,以使 POSIX 算法到 PTP 时标时间戳的直接应用,将 PTP 时间戳转换为 ISO 8601:2004[B17]中的 TAI 表示法。PTP 也分发 Announce 报文 currentUtcOffset 字段中的闰秒

6) 儒略日(JD)是后面跟有格林威治平正午经过天数的小数部分的儒略日数字(JDN)。JDN 是包含原点的一天,JD=0 为公元前 4713 年 1 月 1 日格林威治平正午。修正儒略日(MJD)是小于 2400000.5 的儒略日,其原点转移到 1858 年 11 月 17 日午夜。例如,1900 年 1 月 1 日 0 时,JD=2415020.5 且 MJD=15020。

〈currentLeapSeconds〉的当前数量。在应用 POSIX 算法之前,用 PTP 时间戳减去〈currentLeapSeconds〉可得 ISO 8601:2004 中的 UTC 表示。相反地,应用 POSIX 逆算法并加上〈currentLeapSeconds〉,可将 ISO 8601:2004 中 UTC 格式转换到生成 PTP 时间戳要求的格式。

例如,应用于 PTP 时间戳值为 8 s 的 POSIX 算法得到 00:00:08 1970:01:01(TAI 时间 1970 年 1 月 1 日午夜以后 8 s)。此时,currentUtcOffset 的值大约是 8 s。PTP 时间戳值 8 减去 8 s 得到 0。应用于值为 0 的 POSIX 算法生成 00:00:00 1970:01:01(UTC 时间 1970 年 1 月 1 日第一秒的开始),其为 UTC 的期望值。见 7.2.2,PTP 历元比该时间提前约 8 s。因此,POSIX 算法到 PTP 时标时间戳的直接应用得到该时间的 TAI 印制格式。

B.3 标准时间源

有两个标准时间源在 PTP 系统的实现中特别关注,在该 PTP 系统中 UTC 可溯源时间为应用要求的。

第一个时间源是实现 NTP 协议的系统集,广泛用于同步校园内和世界范围内的计算机系统。维护与 NTP 客户端同步的 NTP 服务器集。这些服务器本身与可溯源到国际标准的时钟服务器同步。来自 NTP 系统的 UTC 时钟精度通常在毫秒级。NTP 提供当前时间、当前闰秒数(仅在 NTP 第 4 版中支持),以及标记引入闰秒校正的警告标志,该闰秒放在 UTC 当天最后。NTP 历元生成一个闰秒校正时,NTP 校正 NTP 秒数(换句话说,NTP 时钟在闰秒时刻期间有效地停止,并且一旦插入,闰秒占用的时间间隔被有效地“遗忘”)。NTP 历元是 1900 年 1 月 1 日 0 时。将 1972 年 1 月 1 日 0 时的 NTP 设置为 2272060800.0,以达到与 UTC 一致。通常,NTP 将秒表示为 32 位无符号整型。因此 NTP 每 2^{32} s \approx 136 年翻转一次,第一个这样的翻转大约发生在 2036 年。

第二个关注的系统是美国国防部维护的全球卫星定位系统(GPS)。GPS 系统的 UTC 时间精度通常在 10 ns~100 ns 的范围内。GPS 系统传输表示时间为{GPS Weeks,GPS SecondsInLastWeek},即从 GPS 历元开始的周数和从当前周开始的秒数。可计算出 GPS 秒,即从 GPS 历元开始的秒数。GPS 提供当前时间,闰秒的当前数量,和标记引入闰秒校正的警告标志。通过 GPS 时间,UTC 和 TAI 时间可用 GPS 传输中的信息进行计算。GPS 历元开始于 1980 年 1 月 6 日 0 时(MJD 44 244)。GPS 周以卫星传播模 1 024 周=19.7 年来表示。第一个这样的翻转发生在 1999 年 8 月 15 日和 8 月 22 日周之间。相信一些但不是全部的商业系统已正确处理该翻转。

可以方便地使用这些系统的任意一个来给 clockClass 为 6 的时钟提供时间。在每个系统中对于关键时刻,讨论的时标和时间示例之间的关系见表 B.1。表 B.1 中,PTP 秒涉及 PTP 时标分发时间秒的部分,并且注意参考的是 TAI 时间 1970 年 1 月 1 日。

表 B.1 时标间的关系

始 于	到	公 式
NTP 秒	PTP 秒	PTP 秒=NTP 秒-2 208 988 800+currentUtcOffset
PTP 秒	NTP 秒	NTP 秒=PTP 秒+2 208 988 800-currentUtcOffset
GPS 秒=(GPS 周 \times 7 \times 86 400)+ GPSSecondsInLastWeek (GPS 周数必须包含 1 024 \times 翻转次数)	PTP 秒	PTP 秒=GPS 秒+315 964 819
PTP 秒	GPS 秒	GPS 秒= PTP 秒-315 964 819

附录 C

(资料性附录)

驻留时间和不对称校正实例

C.1 概述

本附录提供了几个实例来阐明时间报文的交换和透明时钟内驻留时间、路径延时及不对称的校正应用。

本附录给出时钟间的事件报文传输时间和透明时钟内的驻留时间,且不假定它们在两方向上是相同的。

在每个案例中,图中仅包括时间报文的关键字段。图中左边的时钟总是假设为主时钟,或在测量链路延时的情况下,为点到点响应方。

代表每个时钟的框中显示的时间,表示设备中的本地时间。在图中时间表示为:秒:纳秒。纳秒的小数部分。例如,144:7.3 是 144.000 000 007 3 s。值得注意的是,报文的时间戳字段不能保留纳秒的小数部分。纳秒的小数部分只能保存在 correctionField,在图中表示为纳秒.纳秒小数部分。

给出设备中时钟的关系,使得如果需要能计算出相对于主时钟的时间。该关系用 $\text{Time} = \text{Tm} + \text{offset}$ 的形式表示,其中 Tm 是主时钟时间,offset 是相应时钟和相应主时钟之间的偏移。在报文框显示的时间是应由发送报文时钟输入的时间。例如,在图 C.1 中,当 Sync 报文由主时钟发送,通过端到端透明时钟到达从站的过程中,Sync 报文的进入时间戳和离开时间戳为:144:7.3, (114:7.3+0.65+100.3), (144:7.3+0.65+100.3+207.4) 和 (144:7.3+0.65+207.4+0.5+25.2)=144:241.05。第一个离开主时钟的离开时间戳 t_1 。第二个是透明时钟的进入时间戳,通过计算 t_1 和 0.65 的和得到,0.65 是主时钟和透明时钟之间的传输时间;100.3 是主时钟的本地时间和透明时钟的本地时钟之间的假设偏移量。该计算会得到透明时钟产生的进入时间戳。其他项的计算是类似的。Delay_Req 报文的进入和离开时间戳用相同的方法计算。值得注意的是,相对于从时钟的从时钟离开时间是 144:651.1,相对于主时钟是 144:651.1-25.2=144:625.9。

从站或请求时钟中的计算表示结合时间戳、correctionField 和不对称校正来计算偏移和路径延时的过程。这些值总是与图中所用的假设值做比较来阐明协议的运行。

图中的字母标注参考标准的具体条款,以指示被说明的要点。

除图 C.1 外,所有的示例都图示对驻留时间和不对称的校正。

注:为了清楚表示,本附录中的十六进制数字用 0x 开始的数字表示。

C.2 用延时请求-响应机制计算

C.2.1 给出驻留时间校正的主时钟、端到端透明时钟和所有单步从时钟

图 C.1 给出了在由一个端到端的透明时钟分离的两个普通时钟、一个主时钟和一个从时钟组成的系统中,meanPathDelay 和 offsetFromMaster 的测量和计算。系统中所有的时钟都是单步时钟。在图 C.1 中,没有给出不对称校正。

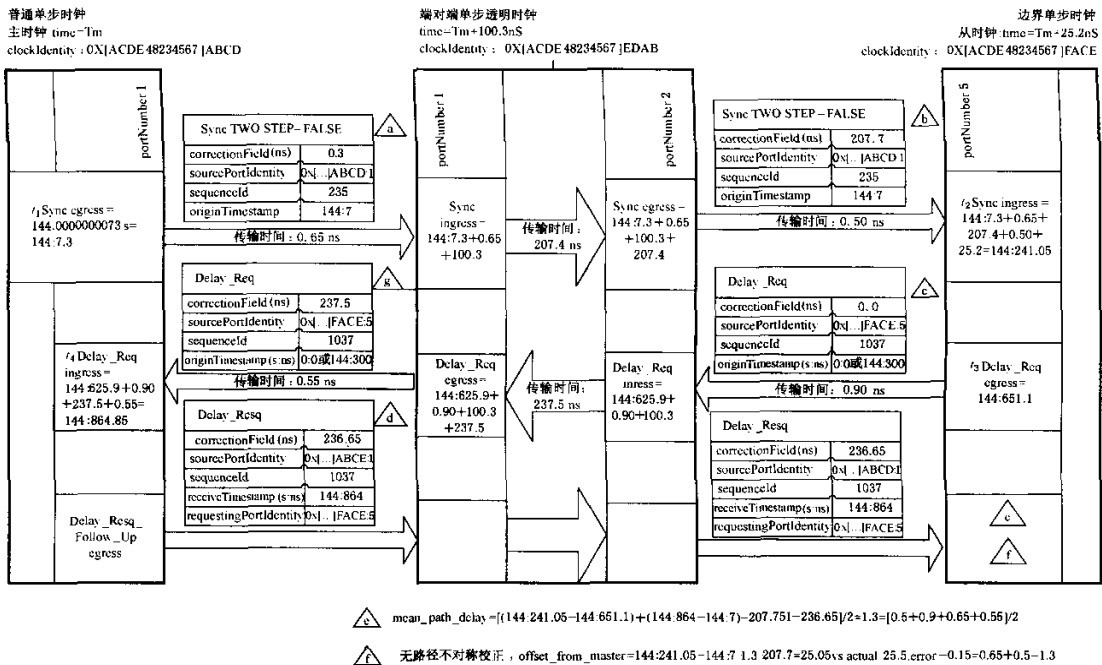


图 C.1 不带不对称校正的主时钟、端到端时钟和单步从时钟

表 C.1 中给出对图 C.1 关键值的解释。

表 C.1 图 C.1 关键值的解释

关键值	参考条款	注 释
a	9.5.9.3 和 11.3.2	时间戳和 correctionField 的和是 t_1
b	11.5.2.1	驻留时间(207.4)已经加到 correctionField
c	11.3.2	时间戳设置为 0 或离开时间戳的估计(144:300)。correctionField 设置为 0
d	11.3.2	注意: requestingPortIdentity 和 sequenceId 是从时钟的对应字段。receiveTimestamp 是不包含纳秒小数部分的 t_4 。correctionField 是 Delay_Req 报文的 correctionField(该值在透明时钟中增加驻留时间(见 11.5.3.2),并减去 t_4 的纳秒小数部分(0.85))
e	11.3.2	第一项是差值($t_2 - t_3$)。第二项是接收时间戳和原始时间戳的差值。最后两项是 correctionField。注意:计算的 meanPathDelay 与图中实际假设的平均路径延时值相当
f	11.2	按顺序排列的项是 t_2 , originTimestamp、meanPathDelay 和 Sync correctionField。注意:由于没有校正传输时间的不对称,计算的偏移误差为-0.15
g	11.5.3.2	驻留时间(237.5)已由透明时钟加到 correctionField

C.2.2 给出驻留时间和不对称计算的主时钟、点到点透明时钟和所有单步从时钟

图 C.2 所示为时钟集,但在该图中,给出了不对称校正。

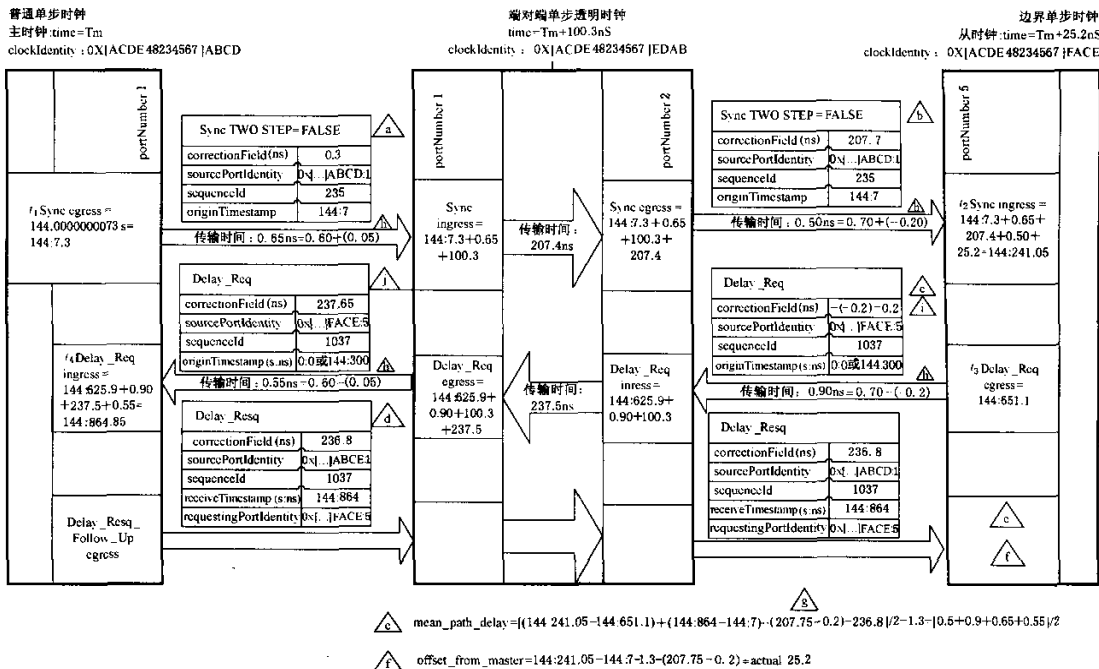


图 C.2 带不对称校正的主时钟、端到端时钟和单步从时钟

在表 C.2 中给出对图 C.2 中关键值的解释。

表 C.2 图 C.2 关键值的解释

关键值	参考条款	注 释
a	9.5.9.3 和 11.3.2	时间戳和 correctionField 的和是 t_1
b	11.5.2.1 和 11.6.2	驻留时间 (207.4) 和进入路径不对称 (0.05) 已经加到 correctionField
c	11.3.2	时间戳设置为 0 或离开时间戳的估计 (144.300)。correctionField 见“i”
d	11.3.2	注意 requestingPortIdentity 和 sequenceId 是从时钟的对应字段。receiveTimestamp 是不包含纳秒小数部分的 t_4 。correctionField 是减去 t_4 的纳秒小数部分 (0.85) 的 Delay_Req 报文的 correctionField
e	11.3.2	第一项是差值 ($t_2 - t_3$)。第二项是接收和源时间戳的差值。最后两个项是在“g”中解释的校正过的 correctionField。注意: 计算的 meanPathDelay 与图中实际假设的路径延时平均值相当
f	11.2	按顺序的项是 t_2 、originTimestamp、meanPathDelay 和 Sync correctionField (按“g”中的解释进行修改)。注意: 由于不对称校正的应用, 计算的偏移十分接近假设值
g	11.6.2	在 Sync 报文的 correctionField 被用于任何计算之前, 将接收 Sync 报文的端口路径的进入不对称 (-0.20) 加到 correctionField 中。因此, 加入到 Sync 报文的 correctionField (207.75) 的该项 (-0.20) 在两个计算中出现
h	7.4.2	传输时间不对称按 7.4.2 中的定义模型化。因此, 在主时钟和透明时钟之间的路径, 平均传输时间是 0.60 ns。主时钟到从时钟的方向, 实际传输时间是 0.65 ns = 0.60 + (0.05), 表示准确不对称值是 0.05。与其他路径相比, 主-从方向较短 (0.7 与 0.9 相比), 产生负不对称值 (-0.2)

表 C.2 (续)

关键值	参考条款	注 释
i	11.6.3	离开路径不对称值(-0.2)已从 correctionField 减去
j	11.6.3 和 11.5.3.2	透明时钟将原始 correctionField 值(0.2)加上驻留时间(237.5),并减去离开路径不对称值(0.05),即 $0.2+237.5-0.05=237.65$

C.2.3 给出驻留时间和不对称计算的双步主时钟、端到端透明时钟和单步从时钟

图 C.3 所示为双步主时钟与单步端到端透明时钟和从时钟之间的相互作用。比较图 C.3 和图 C.2 中 Sync 报文和 Follow_Up 报文的内容,以及 meanPathDelay 和 offsetFromMaster 的计算,可知使用 Follow_Up 报文的影响。

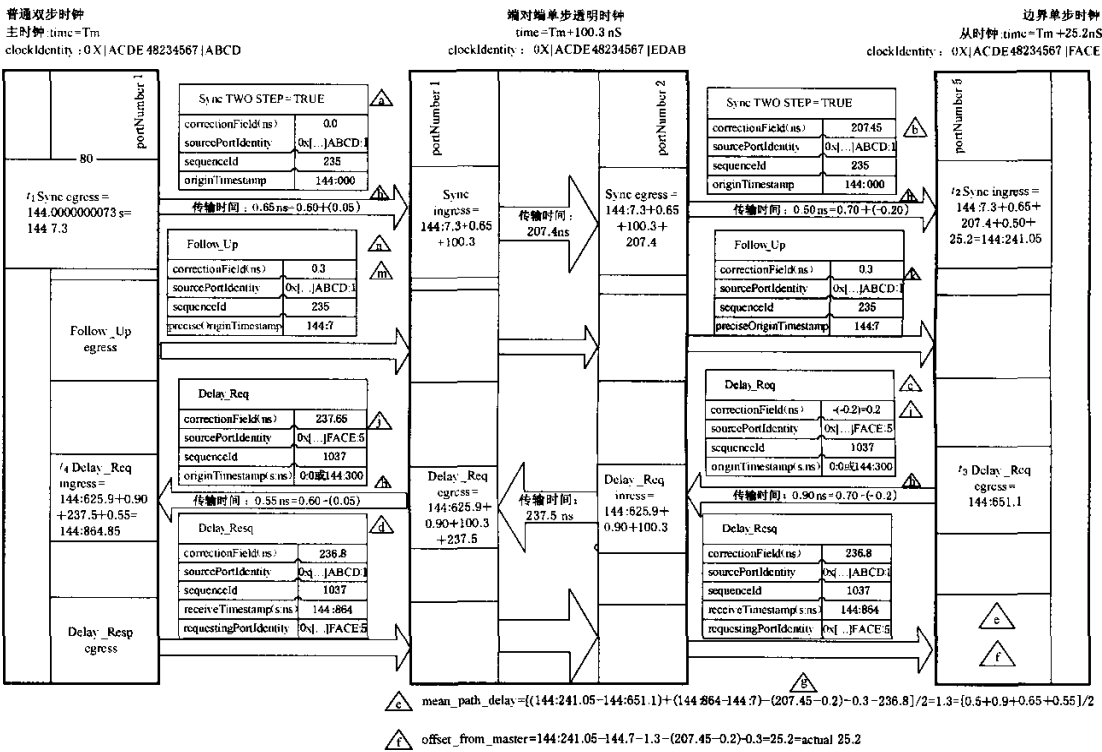


图 C.3 带不对称校正的双步主站、端到端透明时钟和单步从时钟
在表 C.3 中给出对图 C.3 关键值的解释。

表 C.3 图 C.3 关键值的解释

关键值	参考条款	注 释
a	9.5.9.4 和 11.3.2	OriginTimestamp 是 t_1 的估计(144;0),correctionField 为 0
b	11.5.2.1 和 11.6.2	驻留时间(207.4)和进入路径不对称值(0.05)已经加到 correctionField
c	11.3.2	时间戳设置为 0 或离开时间戳的估计(144;300)。correctionField 见“i”
d	11.3.2	注意:requestingPortIdentity 和 sequenceId 是从时钟的对应字段。receiveTimestamp 是不包含纳秒小数部分的 t_4 。correctionField 是减去 t_4 的纳秒小数部分(0.85)的 Delay_Req 报文的 correctionField
e	11.3.2	第一项是差值(t_2-t_3)。第二项是接收时间戳和 preciseOrigin 时间戳的差值。最后三项是按“g”中解释进行修改的 Sync 报文的 correctionField,以及 Follow_Up 和 Delay_Resp 报文的 correctionField。注意:计算的 meanPathDelay 与图中实际假设的平均路径延时相当
f	11.2	按顺序的项是 t_2 、preciseoriginTimestamp、meanPathDelay、Sync correctionField(根据“g”中解释进行修改)和 Follow_Up correctionField。注意:由于不对称校正的应用,计算的偏移十分接近假设值
g	11.6.2	在 Sync 报文中的 correctionField 用于任何计算前,将接收 Sync 报文的端口路径的进入不对称值(-0.20)加到 Sync 报文的 correctionField 中。因此,加到 Sync 报文中 correctionField(207.45)的该项(-0.20)出现在两个计算中
h	7.4.2	传输时间不对称按 7.4.2 中的定义模型化。因此,在主时钟和透明时钟之间的路径,平均传输时间是 0.60 ns。主时钟到从时钟的方向,实际传输时间是 $0.65\text{ ns}=0.60+(0.05)$,表示准确不对称值是 0.05。与其他路径相比,主-从方向路径较短(0.7 与 0.9 相比),产生负不对称值(-0.2)
i	11.6.3	离开路径不对称值(-0.2)已经从 correctionField 减去了
j	11.6.3 和 11.5.3.2	透明时钟将源 correctionField 值(0.2)加上驻留时间(237.5),并减去离开路径不对称值(0.05),即 $0.2+237.5-0.05=237.65$
k	11.5.2.1	Follow_Up 没有校正
m	9.5.10	correctionField 和 preciseOriginTimestamp 的和是离开时间 t_1
n	9.5.10	SequenceId 和 sourcePortIdentity 字段与 Sync 报文的这些对应字段相当

C.2.4 给出驻留时间和不对称计算的主时钟、端到端透明双步时钟和单步从时钟

图 C.4 所示为一个双步主时钟与一个端到端透明双步时钟和一个单步从时钟之间的相互作用。比较图 C.4、图 C.3 和图 C.2 中 Sync 报文和 Follow_Up 报文的内容,以及 meanPathDelay 与 offset-FromMaster 的计算,可知使用 Follow_Up 报文的影响。

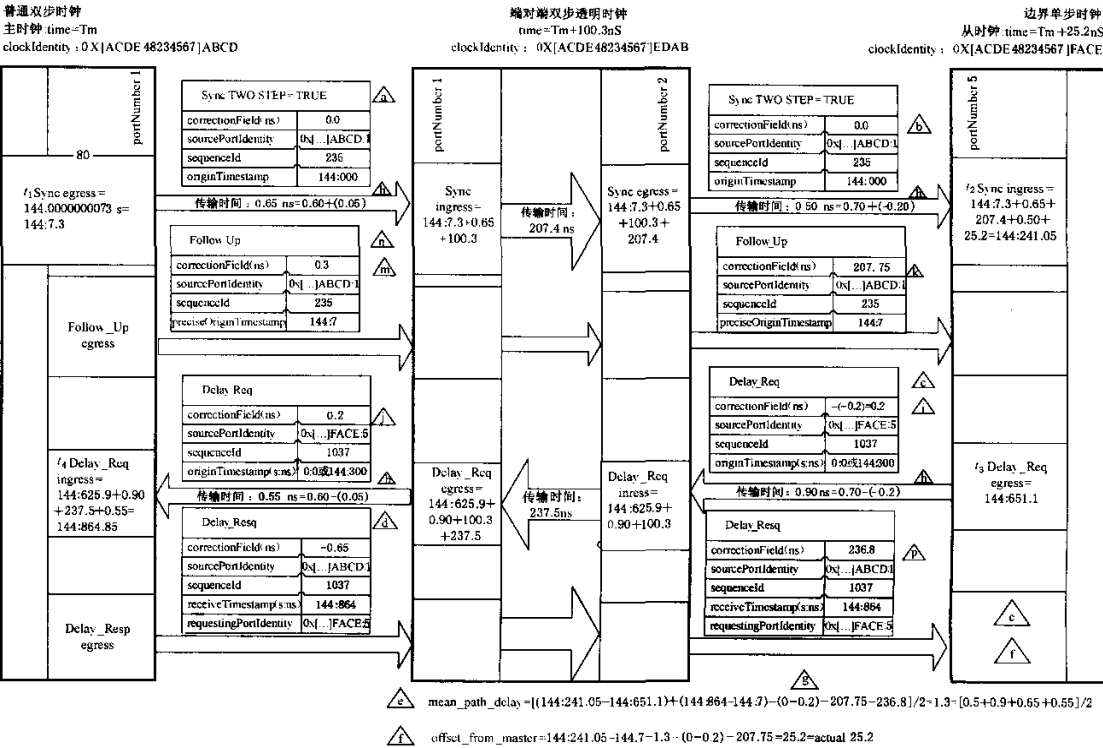


图 C.4 带不对称校正的主时钟、端到端透明双步时钟和单步从时钟

在表 C.4 中给出对图 C.4 关键值的解释。

表 C.4 图 C.4 关键值的解释

关键值	参考条款	注 释
a	9.5.9.4 和 11.3.2	originTimestamp 是 t_1 的估计值(144:0),correctionField 为 0
b	11.5.2.2 和 11.6.2	在 Follow_Up 报文中实现驻留时间(207.4)和进入路径不对称(0.05)的校正,见“k”
c	11.3.2	时间戳设置为 0 或离开时间戳的估计(144:300)。correctionField 见“i”
d	11.3.2	注意,requestingPortIdentity 和 sequenceId 是从时钟的对应字段。receiveTimestamp 是不包含纳秒小数部分的 t_4 。correctionField 等于 Delay_Req 报文的 correctionField(0.2)减去 t_4 的小数纳秒部分(0.85)
e	11.3.2	第一项是差值($t_2 - t_3$)。第二项是接收时间戳和 preciseOrigin 时间戳的差值。最后三项是按“g”的解释已修改的 Sync 报文的 correctionField,以及 Follow_Up 和 Delay_Resp 报文各自的 correctionField。注意:计算的 meanPathDelay 与图中实际假设的平均路径延时相当
f	11.2	按顺序的项是 t_2 、preciseOriginTimestamp、meanPathDelay、Sync 报文的 correctionField (按“g”解释已修改)和 Follow_Up 报文的 correctionField。注意:由于不对称校正的应用,计算的偏移十分接近假设值
g	11.6.2	在 Sync 报文的 correctionField 用到任何计算之前,将接收 Sync 的端口的进入路径不对称(-0.20)加到 Sync 报文的 correctionField。因此,加入到 Sync 报文的 correctionField (0.0)的该项(-0.20)出现在两个计算中

表 C. 4 (续)

关键值	参考条款	注 释
h	7.4.2	传输时间不对称按 7.4.2 的定义模型化。因此,在主时钟和透明时钟之间的路径,平均传输时间是 0.60 ns。主时钟到从时钟的方向,实际传输时间是 0.65 ns=0.60+(0.05),表示准确不对称值是 0.05。与其他路径相比,主-从方向较短(0.7 与 0.9 相比),产生负不对称值(-0.2)
i	11.6.3	离开路径不对称值(-0.2)已从 correctionField 减去
j	11.6.3 和 11.5.3.3	不对 Delay_Req 报文进行校正。驻留时间和离开不对称校正见“p”
k	11.5.2.2 和 11.6.2	将驻留时间(207.4)和进入路径不对称(0.05)校正加到 Follow_Up 报文的 correctionField (0.3)
m	9.5.10	correctionField 和 preciseOriginTimestamp 的和是离开时间 t_1
n	9.5.10	SequenceId 和 sourcePortIdentity 字段与 Sync 报文的对应字段相当
p	11.6.3 和 11.5.3.3	透明时钟将 Delay_Resp 的初始 correctionField 值(-0.65)加上相关 Delay_Req 报文的(237.5)的驻留时间,并减去离开路径不对称(0.05),即-0.65+237.5-0.05=236.8

C.2.5 给出驻留时间和不对称计算的单步主时钟、端到端透明双步时钟和单步从时钟

图 C.5 所示为一个单步主时钟与一个双步端到端透明时钟和一个单步从时钟之间的相互作用。比较图 C.5、图 C.4、图 C.3 和图 C.2 中 Sync 报文和 Follow_Up 报文的内容,以及 meanPathDelay 和 offsetFromMaster 的计算,可知使用 Follow_Up 报文的影响。

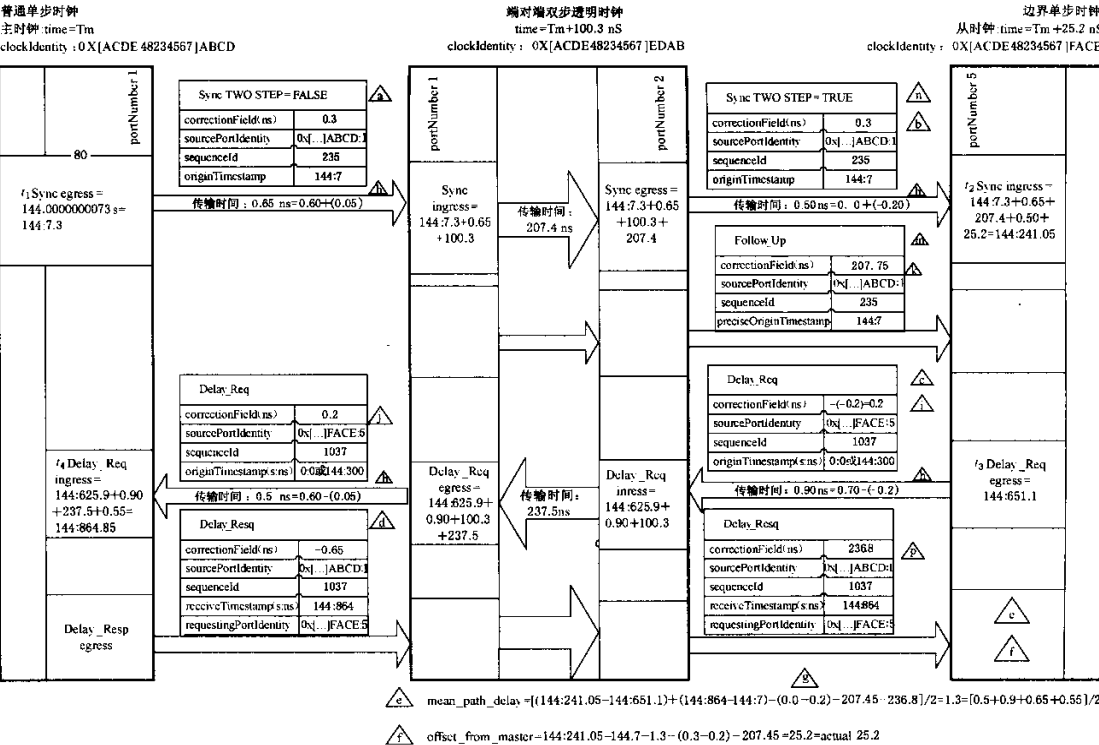


图 C.5 带不对称校正的单步主时钟、双步端到端透明时钟和单步从时钟

在表 C.5 中给出对图 C.5 关键值的解释。

表 C.5 图 C.5 关键值的解释

关键值	参考条款	注 释
a	9.5.9.3 和 11.3.2	时间戳和 correctionField 的和是 t_1
b	11.5.2.2 和 11.6.2	在 Follow_Up 报文中实现驻留时间(207.4)和进入路径不对称值(0.05)的校正,见“k”
c	11.3.2	时间戳设置为 0 或离开时间戳的估计(144,300)。correctionField 见“i”
d	11.3.2	注意: requestingPortIdentity 和 sequenceId 是从时钟的对应字段。receiveTimestamp 是不包含纳秒小数部分的 t_4 。correctionField 是 Delay_Req 报文(0.2)的减去 t_4 的纳秒小数部分(0.85)的 correctionField
e	11.3.2	第一项是差值($t_2 - t_3$)。第二项是接收时间戳和 preciseOrigin 时间戳的差值。最后三项是按“g”中解释已修改的 Sync 报文校正的 correctionField,以及 Follow_Up 和 Delay_Resp 报文各自的 correctionField。注意:计算的 meanPathDelay 与图中实际假设的平均路径延时匹配
f	11.2	按顺序的项是 t_2 、preciseoriginTimestamp、meanPathDelay、Sync 报文的 correctionField (按“g”解释已修改)和 Follow_Up 报文的 correctionField。注意:由于不对称值校正的应用,计算的偏移十分接近假设值
g	11.6.2	在 Sync 报文的 correctionField 用到任何计算之前,将接收 Sync 报文的端口的进入路径不对称 (-0.20) 加到 Sync 报文的 correctionField。因此,加入到 Sync 报文的 correctionField 的该项(-0.20)出现在了两个计算中
h	7.4.2	传输时间不对称按 7.4.2 的定义模型化。因此,对于主时钟和透明时钟之间的路径,平均传输时间是 0.60 ns。在主时钟到从时钟的方向,实际传输时间是 $0.65 \text{ ns} = 0.60 + (0.05)$,表示校正的不对称值是 0.05。与其他路径相比,主-从方向路径较短(0.7 与 0.9 相比),产生负不对称值(-0.2)
i	11.6.3	离开路径不对称值(-0.2)已从 correctionField 减去
j	11.6.3 和 11.5.3.3	对 Delay_Req 报文不进行校正。驻留时间和进入不对称值的校正见“p”
k	11.5.2.2 和 11.6.2	将驻留时间(207.4)加入到 correctionField,随后进入路径不对称(0.05)校正加到 Follow_Up 报文的 correctionField
m	11.5.2.2	将 Sync 报文的 originTimestamp 复制到 preciseOriginTimestamp。将 Sync 报文的 domainNumber(图中未给出)、sourcePortIdentity 和 sequenceId 字段复制到 Follow_Up 报文的对应字段
n	11.5.2.2	twoStepFlag 设置为 TRUE
p	11.6.3 和 11.5.3.3	透明时钟将 Delay_Req 报文的初始 correctionField 值(-0.65)加上相关 Delay_Resp 报文的驻留时间(237.5),并减去离开路径不对称值(0.05),即 $-0.65 + 237.5 - 0.05 = 236.8$

C.3 使用对等延时机制的计算

C.3.1 给出驻留时间和不对称计算的单步对等请求方时钟、端到端透明时钟和对等响应方时钟

图 C.6 所示为一个单步对等请求方时钟与一个单步端到端透明时钟和对等响应方时钟之间的相互作用。

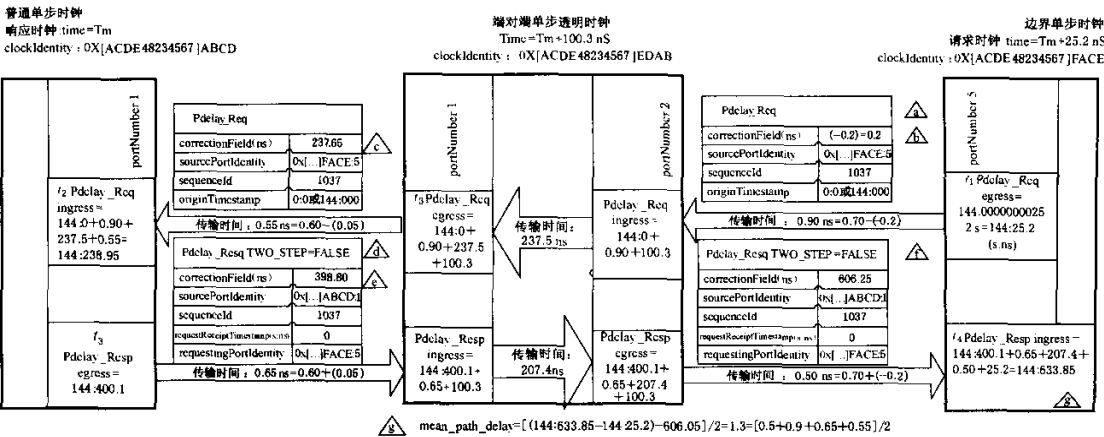


图 C.6 带不对称校正的单步对等响应方时钟、端到端透明时钟和对等请求方时钟
在表 C.6 中给出图 C.6 关键值的解释。

表 C.6 图 C.6 关键值的解释

关键值	参考条款	注 释
a	11.4.3	originTimestamp 是 0 或者离开时间戳 t_1 的估计
b	11.4.3 和 11.6.4	correctionField 等于 $0 - (-0.2) = 0.2$, (-0.2) 是离开路径的不对称值
c	11.5.4.2 和 11.6.4	通过在端到端透明时钟中加上驻留时间,并减去离开路径不对称值来修改 correctionField。 $0.2 + 237.5 - (0.05) = 237.65$
d	11.4.3	requestReceiptTimestamp=0。将 Pdelay_Req 报文的 sourcePortIdentity 和 sequenceId 字段复制到 requestingPortIdentity 和 sequenceId 字段
e	11.4.3	correctionField 等于 Pdelay_Req 报文的 correctionField 和周转时间($t_3 - t_2$)的和,即 $237.65 + (144:400.1 - 144:238.95) = 398.80$
f	11.5.4.2 和 11.6.5	通过在初始 correctionField 上加上驻留时间 207.4 和进入不对称 0.05 来修改 correctionField。 $398.80 + 207.4 + 0.05 = 606.25$
g	11.4.3 和 11.6.5	在计算 meanPathDelay 之前,通过加上进入不对称 0.2 来校正 Pdelay_Resp 报文的 correctionField。即 $606.25 + (-0.2) = 606.05$ 。 $\text{meanPathDelay} = [(t_4 - t_1) - \text{Pdelay_Resp 的 correctionField}] / 2 = [(144:633.85 - 144:25.2) - 606.05] / 2 = 1.3$

C.3.2 给出驻留时间和不对称计算的单步对等请求方时钟、双步端到端透明时钟和单步对等响应方时钟

图 C.7 所示为一个单步对等请求方时钟与双步端到端透明时钟和单步对等响应方时钟相互作用。

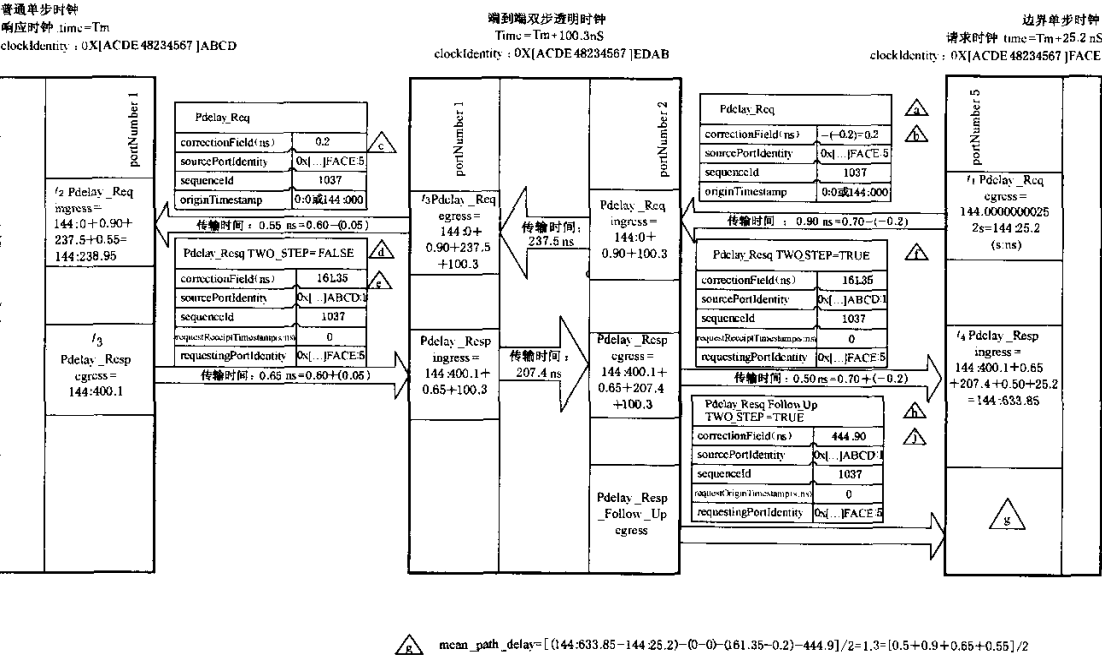


图 C.7 带不对称校正的单步对等响应方时钟、双步端到端透明时钟和单步对等请求方时钟
在表 C.7 中给出图 C.7 关键值的解释。

表 C.7 图 C.7 关键值的解释

关键值	参考条款	注 释
a	11.4.3	OriginTimestamp 是 0 或者离开时间戳 t_1 的估计
b	11.4.3 和 11.6.4	correctionField 等于 $0 - (-0.2) = 0.2$, (-0.2) 是离开路径的不对称
c	11.5.4.2 和 11.6.4	不修改 correctionField。对 Pdelay_Req 的驻留时间和进入路径不对称的校正见“j”
d	11.4.3	requestReceiptTimestamp=0。将 Pdelay_Req 报文的 sourcePortIdentity 和 sequenceId 字段复制到 requestingPortIdentity 和 sequenceId 字段
e	11.4.3	correctionField 是 Pdelay_Req 报文的 correctionField 与周转时间 $(t_3 - t_2)$ 的和, 即 $0.2 + (144:400.1 - 144:238.95) = 161.35$
f	11.5.4.3 和 11.6.5	除设为 TRUE 的 twoStepFlag 以外, 不修改 Pdelay_Resp 的 PTP 字段, 将其设置为 TRUE。Pdelay_Resp 报文的驻留时间和进入路径不对称的校正见“j”
g	11.4.3 和 11.6.5	在计算 meanPathDelay 之前, 通过加上进入不对称 0.2 来修改 Pdelay_Resp 报文的 correctionField。即 $161.35 + (-0.2) = 161.05$ 。meanPathDelay = $[(t_4 - t_1) - (\text{responseOriginTimestamp} - \text{requestReceiptTimestamp}) - \text{Pdelay_Resp 报文的 correctionField} - \text{Pdelay_Resp_Follow_Up 报文的 Pdelay_Resp}] / 2 = [(144:633.85 - 144:25.2) - (0 - 0) - (161.35 - 0.2) - 444.90] / 2 = 1.3$
h	11.5.4.3	将 Pdelay_Resp 报文的 sourcePortIdentity、sequenceId 和 requestingPortIdentity 字段复制到 Pdelay_Resp_Follow_Up 报文的对应字段。TWOSTEP 标志设置为 TRUE。responseOriginTimestamp 设置为 0
j	11.5.4.3, 11.6.4 和 11.6.5	correctionField 等于 Pdelay_Req 与 Pdelay_Resp 报文驻留时间的和, 减去 Pdelay_Req 的离开不对称值, 加上 Pdelay_Resp 报文的进入不对称值, 即 $237.5 + 207.4 - (0.05) + (0.05) = 444.90$

C.3.3 给出驻留时间和不对称计算的单步对等请求方时钟、双步端到端透明时钟、双步对等响应方时钟——选项 1

图 C.8 所示为一个单步对等请求方时钟与双步端到端透明时钟和双步对等响应方时钟之间相互作用。双步响应方时钟用 11.4.3 的第一个选项生成 Pdelay_Resp 和 Pdelay_Resp_Follow_Up 报文。

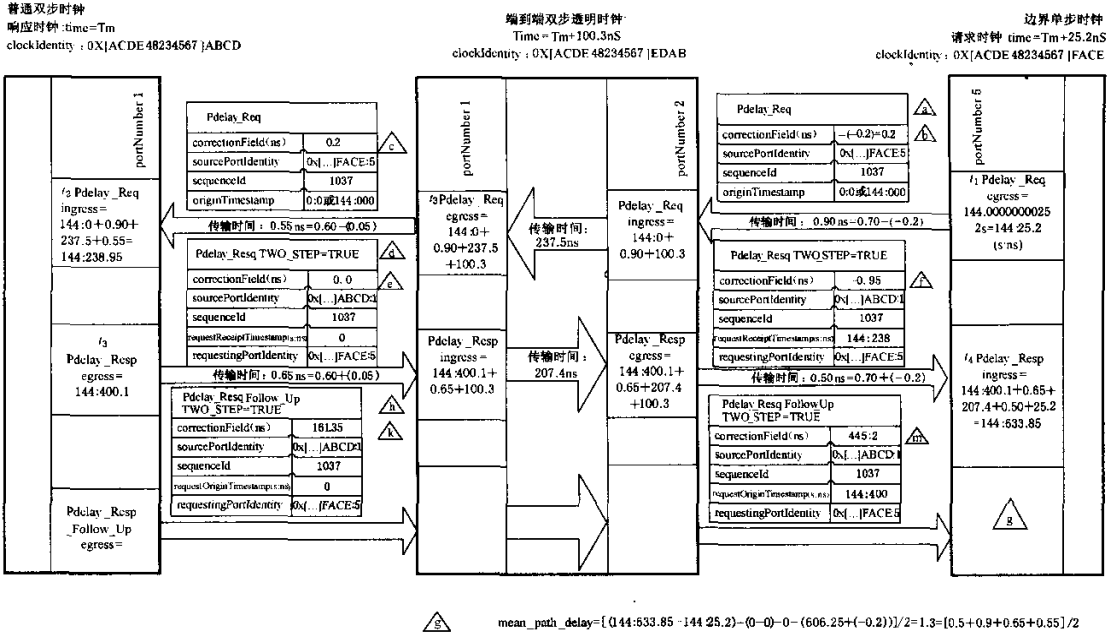


图 C.8 带不对称校正的双步对等响应方时钟、双步端到端透明时钟和单步对等请求方时钟:选项 1
表 C.8 中给出图 C.8 关键值的解释。

表 C.8 图 C.8 关键值的解释

关键值	参考条款	注 释
a	11.4.3	originTimestamp 是 0 或者离开时间戳 t_1 的估计
b	11.4.3 和 11.6.4	correctionField 为 $0 - (-0.2) = 0.2$, 其中 (-0.2) 是离开路径的不对称
c	11.5.4.2 和 11.6.4	不修改 correctionField。Pdelay_Req 的驻留时间和离开路径不对称的校正见“m”
d	11.4.3	requestReceiptTimestamp=0。将 Pdelay_Req 报文的 sourcePortIdentity 和 sequenceId 字段复制到 requestingPortIdentity 和 sequenceId 字段
e	11.4.3	correctionField 设置为 0
f	11.5.4.3 和 11.6.5	不修改 Pdelay_Resp 的 PTP 字段。Pdelay_Resp 报文的驻留时间和进入路径不对称的校正见“m”
g	11.4.3 和 11.6.5	在计算 meanPathDelay 之前,通过加上进入不对称 0.2 来修改 Pdelay_Resp 报文的 correctionField。即 $606.25 + (-0.2) = 606.05$ 。meanPathDelay = $[(t_4 - t_1) - (\text{responseOriginTimestamp} - \text{requestReceiptTimestamp}) - \text{Pdelay_Resp 报文的 correctionField} - \text{Pdelay_Resp_Follow_Up 报文的 correctionField}]/2 = [(144:533.85 - 144:25.2) - (0 - 0) - 0 - 606.05]/2 = 1.3$
h	11.5.4.3	将 Pdelay_Resp 报文的 sourcePortIdentity、sequenceId 和 requestingPortIdentity 字段复制到 Pdelay_Resp_Follow_Up 报文的对应字段。TWOSTEP 标志设置为 TRUE。responseOriginTimestamp 设置为 0

表 C.8 (续)

关键值	参考条款	注 释
k	11.5.4.3,11.6.4和 11.6.5	CorrectionField 是 Pdelay_Req 报文的 correctionField 和周转时间的和,即 $(t_3 - t_2) \cdot 0.2 + (144;400.1 - 144;238.95) = 161.35$
m	11.5.4.3,11.6.4和 11.6.5	correctionField 等于初始 correctionField 与 Pdelay_Req 和 Pdelay_Resp 报文驻留时间的和,减去 Pdelay_Req 离开不对称值,加上 Pdelay_Resp 报文进入不对称值,即 $161.35 + 237.5 + 207.4 - (0.05) + 0.05 = 606.25$

C.3.4 给出驻留时间和不对称计算的单步对等请求方时钟、双步端到端透明时钟和双步对等响应方时钟——选项 2

图 C.9 所示为一个单步对等请求方时钟与双步端到端透明时钟和双步对等响应方时钟的相互作用。双步响应时钟使用 11.4.3 的第 2 个选项生成 Pdelay_Resp 和 Pdelay_Resp_Follow_Up 报文。

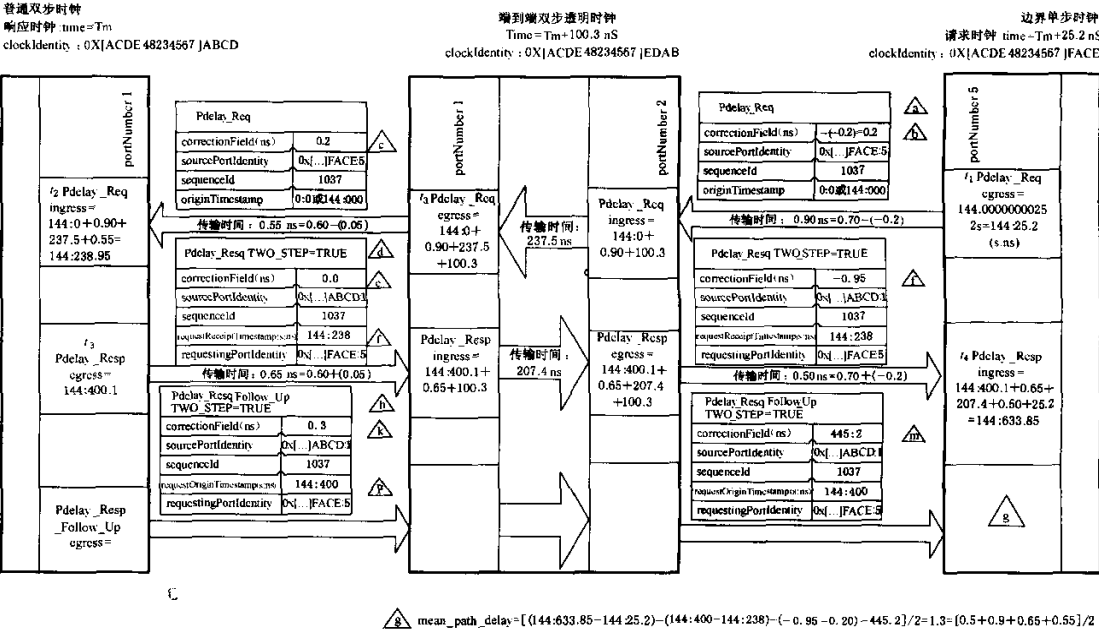


图 C.9 带不对称校正的双步对等响应方时钟、双步端到端透明时钟和单步对等请求方时钟:选项 2
在表 C.9 中给出图 C.9 关键值的解释。

表 C.9 图 C.9 关键值的解释

关键值	参考条款	注 释
a	11.4.3	originTimestamp 是 0 或者离开时间戳 t_1 的估计
b	11.4.3 和 11.6.4	correctionField 是 $0 - (-0.2) = 0.2$, 其中 (-0.2) 是离开路径的不对称
c	11.4.3,11.5.4.3和 11.6.4	不修改 Pdelay_Req 报文的 correctionField
d	11.4.3	将 Pdelay_Req 报文的 sourcePortIdentity 和 sequenceId 字段复制到 requestingPortIdentity 和 sequenceId 字段。TWO-STEP 标志设置为 TRUE
e	11.4.3	correctionField 为 0 减去 t_2 的纳秒小数部分,即 $0 - 0.95 = -0.95$
f	11.4.3,11.5.4.3和 11.6.5	RequestReceiptTimestamp 设置为 t_2 的秒和纳秒部分,144;238

表 C.9 (续)

关键值	参考条款	注 释
g	11.4.3 和 11.6.5	在计算 meanPathDelay 之前,通过加上进入不对称 0.2 来修改 Pdelay_Resp 报文的 correctionField。即 $-(0.95) + (-0.2) = -1.15$ 。meanPathDelay = $[(t_4 - t_1) - (\text{responseOriginTimestamp} - \text{requestReceiptTimestamp}) - \text{Pdelay_Resp}$ 报文的 correctionField - Pdelay_Resp_Follow_Up 报文的 correctionField]/2 = $[(144.633.85 - 144.25.2) - (144.400 - 144.238) - (-0.95 + (-0.20)) - (445.2)]/2 = 1.3$
h	11.4.3	将 Pdelay_Resp 报文的 sourcePortIdentity、sequenceId 和 requestingPortIdentity 字段复制到 Pdelay_Resp_Follow_Up 报文的相同字段。TWO-STEP 标志设置为 TRUE
k	11.4.3 和 11.6.5	correctionField 设置为 Pdelay_Req 报文的 correctionField 与 t_3 的小数纳秒部分的和,即 $0.2 + 0.1 = 0.3$
m	11.5.4.3, 11.6.4 和 11.6.5	correctionField 等于初始 correctionField 与 Pdelay_Req 和 Pdelay_Resp 报文驻留时间的和,减去 Pdelay_Req 离开不对称,加上 Pdelay_Resp 报文进入不对称,即 $0.3 + 237.5 + 207.4 - (0.05) + 0.05 = 445.2$
p	11.4.3	responseOriginTimestamp 设置为 t_3 的秒和纳秒部分,144.400

C.3.5 显示驻留时间和不对称计算的单步对等请求方时钟、单步端到端透明时钟和双步对等响应方时钟——选项 2

图 C.10 所示为一个单步对等请求方时钟与单步端到端透明时钟和双步对等响应方时钟之间相互作用。双步响应方时钟使用 11.4.3 第二个选项生成 Pdelay_Resp 和 Pdelay_Resp_Follow_Up 报文。

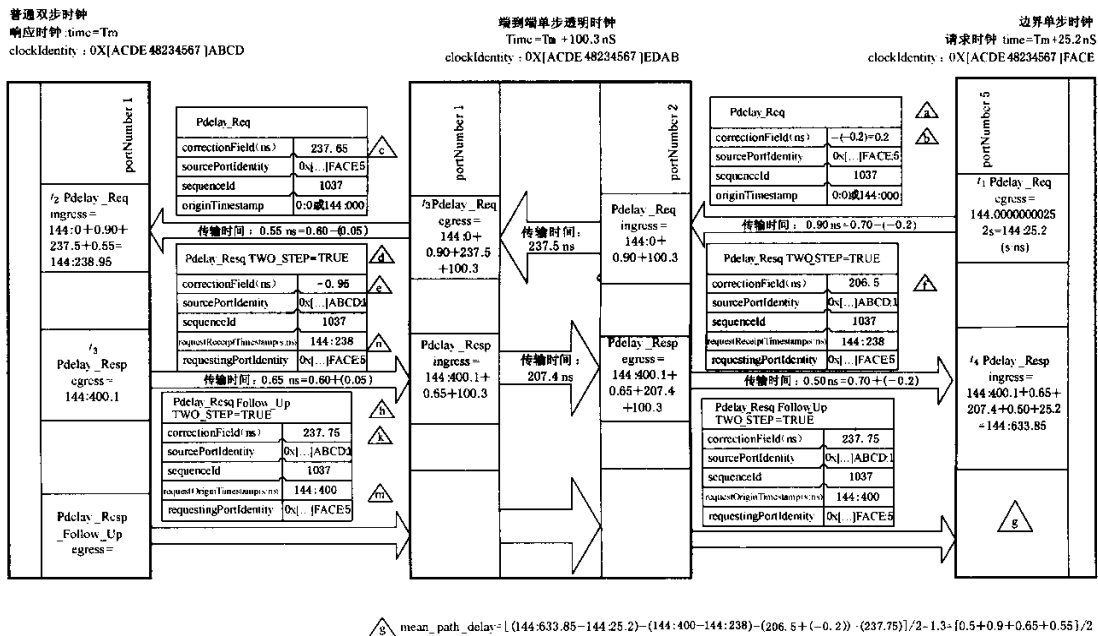


图 C.10 带不对称校正的双步对等响应方时钟、单步端到端透明时钟和单步对等请求方时钟:选项 2
在表 C.10 中给出图 C.10 关键值的解释。

表 C.10 图 C.10 关键值的解释

关键值	参考条款	注 释
a	11.4.3	originTimestamp 为 0 或者离开时间戳 t_1 的估计
b	11.4.3 和 11.6.4	correctionField 等于 $0 - (-0.2) = 0.2$, 其中 (-0.2) 是离开路径的不对称值
c	11.5.4.2 和 11.6.4	correctionField 等于 Pdelay_Req 报文的初始 correctionField 和驻留时间的和, 减去离开路径不对称, 即 $0.2 + 237.5 - 0.05 = 237.65$
d	11.4.3	将 Pdelay_Req 报文的 sourcePortIdentity 和 sequenceId 字段复制到 requestingPortIdentity 和 sequenceId 字段。TWO-STEP 标志设置为 TRUE
e	11.4.3	correctionField 设置为 0 减去 t_2 纳秒的小数部分, $= -0.95$
f	11.5.4.2 和 11.6.5	correctionField 是 Pdelay_Resp 报文的初始 correctionField 与驻留时间的和, 加上 Pdelay_Resp 报文的进入不对称。 $-0.95 + 207.4 + (0.05) = 206.5$
g	11.4.3 和 11.6.5	在计算 meanPathDelay 之前, 通过加上进入不对称 -0.2 来修改 Pdelay_Resp 报文的 correctionField。即 $206.5 + (-0.2) = 206.3$ 。 $meanPathDelay = [(t_4 - t_1) - (responseOriginTimestamp - requestReceiptTimestamp) - Pdelay_Resp \text{ 报文的 } correctionField - Pdelay_Resp_Follow_Up \text{ 报文的 } correctionField] / 2 = [(144; 633.85 - 144; 25.2) - (144; 400 - 144; 238) - (206.5 + (-0.2)) - (237.75)] / 2 = 1.3$
h	11.4.3	将 Pdelay_Resp 报文的 sourcePortIdentity, sequenceId 和 requestingPortIdentity 字段复制到 Pdelay_Resp_Follow_Up 报文的相同字段。TWOSTEP 标志设置为 TRUE
k	11.4.3 和 11.6.5	correctionField 设置为 Pdelay_Req 报文的 correctionField 与 t_3 的纳秒小数部分的和, 即 $237.65 + 0.1 = 237.75$
m	11.4.3	responseOriginTimestamp 设置为 t_3 的秒和纳秒部分, 144; 400
n	11.4.3 和 11.6.5	requestReceiptTimestamp 设置为 t_2 的秒和纳秒部分, 144; 238

C.3.6 给出从主时钟到从时钟传输时间的单步对等主时钟、双步点到点透明时钟和单步对等从时钟

图 C.11 所示为一个单步对等主时钟与双步点到点透明时钟和单步对等从时钟之间相互作用, 以及由主时钟到从时钟的传输时间。

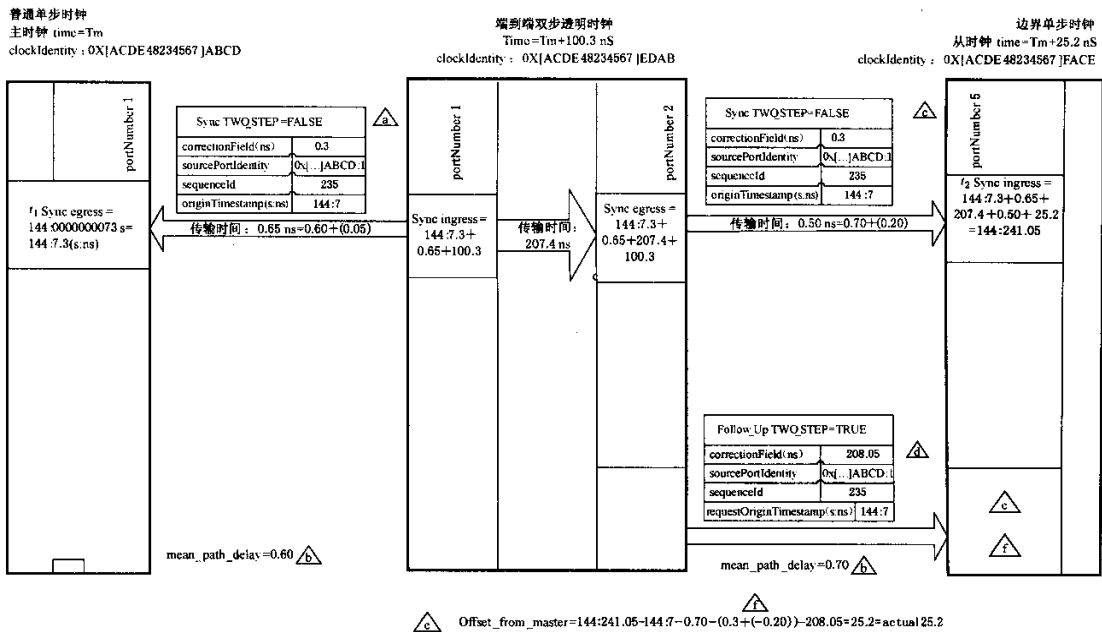


图 C.11 带时间计算的单步对等主时钟、双步点到点透明时钟和单步对等从时钟
在表 C.11 给出图 C.11 关键值的解释。

表 C.11 图 C.11 关键值的解释

关键值	参考条款	注 释
a	9.5.9.3	originTimestamp 是离开时间戳 t_1 的秒和纳秒部分。originTimestamp 和 correctionField 的和是 t_1 的精确值
b	11.4.3	使用对等延时机制的两个链路的 meanPathDelay 值分别确定为 0.60 和 0.70
c	11.5.2.2	在发送 Sync 报文之前, TWO-STEP 标志设置为 TRUE
d	11.5.2.2, 11.4.5.1 和 11.6.2	将 Sync 报文的 sourcePortIdentity 和 sequenceId 字段复制到 sourcePortIdentity 和 sequenceId 字段。PreciseOriginTimestamp 字段设置为 originTimestamp 字段的值。首先将 correctionField 设置为 Sync 报文的驻留时间 207.4, 然后通过 Sync 进入端口路径加上 meanPathDelay 来对该值进行校正, 即 207.4 + 0.6。最后, 通过加入 Sync 进入端口路径不对称对该结果进一步校正, 即 207.4 + 0.6 + 0.05 = 208.05
e	11.2	offsetFromMaster = t_2 - preciseOriginTimestamp - 进入链路的 meanPathDelay - Sync 报文的 correctionField - Follow_Up 报文的 correctionField。 OffsetFromMaster = 144.241.05 - 144.7 - 0.70 - (0.3 + (-0.20)) - 208.05 = 25.2
f	11.6.2	在将 correctionField 用于任何计算“e”(0.3 + (-0.20) = 0.1)之前, 通过加上进入路径不对称来修改接收到 Sync 报文的 correctionField

附录 D
(规范性附录)

PTP 在 IPV4 用户数据包协议上的传输

D.1 概述

本附录规定了在用户数据报协议(UDP,见 IETF RFC 768(1980) [B5])和因特网协议版本 4 (IPv4,见 IETF RFC 791(1981) [B6])上传输报文的特定实现的 PTP 标准部分。本附录规范应适用于所有使用 UDP/IPv4 作为通信服务的 PTP 实现。

PTP 报文的第一个八位位组应紧跟 UDP 报文头的最后一个八位位组。

发送或中间节点可将 UDP 校验和设为 0。

当通过单播发送使用该传输时,透明时钟对 PTP 事件数据包的修改,可使不正确使用 PTP 目的端口的应用出错。

注:下面的 UDP 目的端口是为 PTP 分配的保留值,且不应发生冲突。然而众所周知,一些应用忽略这些分配。正是这些应用易受透明时钟行为的影响。

D.2 UDP 端口号

事件报文的 UDP 目的端口应为 319⁷⁾。

多播通用报文的 UDP 目的端口应为 320。

对一个时钟寻址的单播通用报文的 UDP 目的端口应为 320。

对一个管理者寻址的单播通用报文的 UDP 目的端口应为 PTP 响应报文的 UDP 源端口值。

D.3 IPv4 多播地址

PTP 报文应使用表 D.1 规定的多播报文。

表 D.1 IPv4 多播地址

IANA 分配的名称 ⁸⁾	报文类型	地址
PTP-primary	除对等延时机制报文外的所有报文	224.0.1.129
PTP-pdelay	对等延时机制报文	224.0.0.107

对于发送到 PTP-pdelay 地址的报文,存活时间(TTL)字段应设为 1。

D.4 transportSpecific 字段值

transportSpecific 字段(见 13.3.2.1)应按表 D.2 的规定进行解释。

7) 因特网编码管理局(IANA)为 PTP 分配的专用端口号;见 <http://www.iana.org/assignments/port-numbers>。

8) 因特网编码管理局(IANA)分配的专用多播地址,连同 PTP 一起的 IANA 名称,这些名称出现在 IANA 列表中表示多播地址和名称。

表 D.2 transportSpecific 字段值

比特	名称	含 义
0	hardwareCompatibility	<p>版本 1 的一些硬件实现,在检查时间戳合格之前要辅助时间戳校验输入数据包的长度,并且要求 PTP 事件报文的 UDP 有效载荷长度至少为 124 个八位位组。注意,使用这样的硬件时,应将从该节点发送的所有 Announce 报文和 PTP 事件报文中的比特 0 设为“1”。</p> <p>接收任何比特 0 为“1”的 PTP Announce 报文或事件报文的接收方,应对发送给该接收节点的所有 PTP 事件报文的 UDP 有效载荷进行扩展,使得 UDP 有效载荷长度等于 124 个八位位组。填充的八位位组所有比特应为 0。此填充应附加到所有发送给该接收节点的 PTP 事件报文,并且自该节点最后接收到的比特 0 为“1”的 PTP Announce 报文或事件报文时间开始,此填充要持续 portDS. announceReceiptTimeout 秒的时间。填充应不考虑 PTP 事件报文使用多播还是单播模式传输。</p> <p>如果发送方未请求进行填充,则该比特应以 0 进行发送。除了要求向后兼容一些版本 1 的硬件外,节点应忽略填充的八位位组。</p> <p>见注</p>
1~3	保留	该比特以 0 进行发送,并被接收方忽略
注:本规范可使节点不要求对正在接收的已填充的事件报文进行填充。		

D.5 可选值

对于 PTP 事件报文,服务类型(ToS)字段中区分服务(DS)字段的值应被设为可用的最高流量类选择器代码点。

注:当第二层传输机制允许多优先级时,推荐事件报文使用最高优先级。

D.6 IPv4 选项

不应使用 IPv4 选项。

D.7 协议地址

- 当 networkProtocol 成员值是 UDP/IPv4(见 7.4.1),对数据类型 PortAddress 的任意参量(见 5.3.6):
- addressLength 成员值应为 4。
 - addressField 成员值应用 4 组 2 个十六进制数字表示的端口的 IPv4 地址。例如,用通常文本符号中表示的 IPv4 地址 207.142.131.235,显示为八位位组数组 CF8E83EB₁₆。

附 录 E
(规范性附录)

PTP 在 IPv6 用户数据协议(UDP)上的传输

E.1 概述

本附录规定了在用户数据报协议(UDP,见 IETF RFC 768(1980) [B5])和因特网协议版本 6 (IPv6,见 IETF RFC 2460(1998) [B12])上传输报文的特定实现的 PTP 标准部分。本附录规范应适用于所有使用 UDP/IPv6 作为通信服务的 PTP 实现。

PTP 报文的第一个八位位组应紧跟 UDP 报文头的最后一个八位位组。

发送节点应扩展所有 PTP 报文的 UDP 有效载荷,通过在 PTP 报文末端增加两个八位位组实现。UDP 校验和字段的内容或 UDP 有效载荷的最后两个八位位组,可通过发起方或一个中间节点来修改,以确保对 PTP 字段的任何修改都可保持 UDP 校验和不被损坏。更新 UDP 校验和的修改可通过使用 IETF RFC 1624(1994)[B9]定义的机制来实现。除了用于计算 UDP 校验和外,接收方应忽略 PTP 字段末尾多出的 UDP 字段内容。

E.2 UDP 端口号

事件报文的 UDP 目的端口值应为 319⁹⁾。

多播通用报文的 UDP 目的端口值应为 320。

对一个时钟寻址的单播通用报文的 UDP 目的端口值应为 320。

对一个管理者寻址的单播通用报文的 UDP 目的端口值应为 PTP 响应报文的 UDP 源端口值。

E.3 IPv6 多播地址

Ipv6 多播地址见表 E.1。

表 E.1 IPv6 多播地址

IANA 分配的名称	报文类型	地址(十六进制)
PTP-primary	除对等延时机制报文外的所有报文	FF0X:0:0:0:0:0:181,见注
PTP-pdelay	对等延时机制报文	FF02:0:0:0:0:0:6B
注:在 PTP-primary 地址中,“X”的十六进制值在 IETF RFC 4291(2006)[B14]中有如下定义:		
0 保留		
1 接口本地范围		
2 链路本地范围		
3 保留		
4 管理本地范围		
5 站点本地范围		
6 (未分配)		
7 (未分配)		
8 组织本地范围		
9 (未分配)		
A (未分配)		
B (未分配)		
C (未分配)		
D (未分配)		
E 全球范围		
F 保留		

9) 因特网编码管理局(IANA)为 PTP 分配的专用端口号;见 <http://www.iana.org/assignments/port-numbers>.

对于发送到 PTP-pdelay 地址的报文, Hop Limit(HL)字段应设为 1。

E.4 transportSpecific 字段值

transportSpecific 字段的所有比特值(见 13.3.2.1)应设为 0 传输, 且应被接收方忽略。

E.5 可选值

对于 PTP 事件报文, 流量类(TC)字段中区分服务(DS)字段的值应被设为可用的最高流量类选择器代码点。

注 1: 当第二层传输机制允许多优先级时, 事件报文应使用最高优先级。

注 2: IPv6 扩展报文头的使用超出本标准的范围。

E.6 协议地址

当 networkProtocol 成员值为 UDP/IPv6(见 7.4.1), 对数据类型 PortAddress 的任意参量(见 5.3.6):

——addressLength 成员值应为 16。

——addressField 成员值应是用 16 组两个十六进制数字表示的端口的 IPv6 地址。例如, 按照 IETF RFC 4291 (2006)[B14], 用通常文本符号表示的 IPv6 地址 2001:0DB8:85A3:08D3:1332:8A2E:0270:7225 显示为八位位组数组 20010DB885A308D313328A2E02707225。

附录 F
(规范性附录)

PTP 在 IEEE 802.3/Ethernet 上的传输

F.1 概述

本附录规定了在以太网帧上直接传输报文(见 IEEE Std 802.3;2005)的特定实现的 PTP 标准部分。

PTP 报文的第一个八位位组应占用客户机数据字段的第一个八位位组。

F.2 Ethertype

本附录规范应适用于直接使用 Ethertype 为 88F7₁₆ 的以太网络格式数据包进行通信服务的所有 PTP 实现。

F.3 多播 MAC 地址

缺省地,PTP 报文应使用表 F.1 规定的 MAC 地址。

表 F.1 多播 MAC 地址

报文类型	地址(十六进制)
除对等延时机制报文外的所有报文	01-1B-19-00-00-00
对等延时机制报文	01-80-C2-00-00-0E

00-1B-19 的 OUI 值表示由 IEEE/RAC 分配给本标准的值。01-1B-19-00-00-00 的 MAC 地址值表示在地址空间内从多播地址池中获得的一个多播地址。

01-80-C2-00-00-0E 的 MAC 地址表示从 IEEE Std 802.1AB 管理的多播地址池中获得的一个多播地址。然而,如果在 PTP 行规中定义了这样的应用,则允许将地址 01-1B-19-00-00-00 或地址 01-80-C2-00-00-0E 用于所有 PTP 报文。

为确保对由于(快速/多)生成树协议而被阻塞(blocked)的端口进行对等延时测量,保留地址 01-80-C2-00-00-0E 应被用作 PTP 对等延时机制报文的目标 MAC 地址。

注 1: 按照 IEEE Std 802.1Q;2005 的 8.6.3,目标地址字段包含保留地址的帧不被网桥传递。

注 2: 在 2006 年 7 月 17 日~20 日的会议上,IEEE 802.1 工作组通过了一个提议,其内容包括:“重新分配当前被标识用于 802.1AB 的保留地址,使其作为一个地址可被协议使用,该协议要求将地址范围限制在一个单独 LAN 内”和“IEC 61588 应使用的保留多播地址为 01-80-C2-00-00-0E。”此地址不是专门保留供 PTP 使用,而是一个共享地址。

每个端口上,对等延时测量应使用离开端口的 MAC 地址作为 PTP 对等延时机制报文的源 MAC 地址。

F.4 transportSpecific 字段值

transportSpecific 字段(见 13.3.2.1)应被解释为表 F.2 中定义的 Ethertype 的一个子类型。

如果设备识别出该子类型,则报文被传递给 PTP 层。如果设备未识别出该子类型,则报文被处理为具有不被识别的 Ethertype 的任何其他报文。

表 F.2 以太网传输特定字段

枚举	值(十六进制)	规 范
DEFAULT	0	所有 PTP 第 2 层 Ethernet 发送不被其他的列举值覆盖
ETHERNET_AVB	1	该值被保留用来连接由 IEEE 802.1AVB 任务组开发的 P802.1AS 标准
保留	2~F	保留用于本标准将来版本的分配

F.5 可选值

当 Ethernet 传输机制允许多优先级,事件报文应使用最高优先级。

注:对于以太网,IEEE Std 802.1Q:2005 详述了优先级的实现。

F.6 协议地址

当 networkProtocol 成员值为 IEEE 802.3(见 7.4.1),对数据类型 PortAddress 的任意参量(见 5.3.6):

- addressLength 成员值应为 6。
- addressField 成员值应是以太网报文头的 6 个八位位组源地址字段。

附录 G

(规范性附录)

PTP 在 DeviceNet 上的传输

G.1 协议

本附录规定了 DeviceNet 特定实现的 PTP 标准部分。本附录规范应适应于使用 DeviceNet 作为通信网络的所有 PTP 实现。对于 DeviceNet 的其他信息,见 ODVA(开放式 DeviceNet 制造商协会, <http://www.odva.org>)提供的 DeviceNet 规范。

注: DeviceNet 也包含在 IEC 62026-3:2007 中。

G.2 事件报文时间戳点

事件报文时间戳点(见 7.3.4.1)应为 PTP 事件报文第一个分段数据包的帧结束字段的第六比特下降沿,见图 G.1。

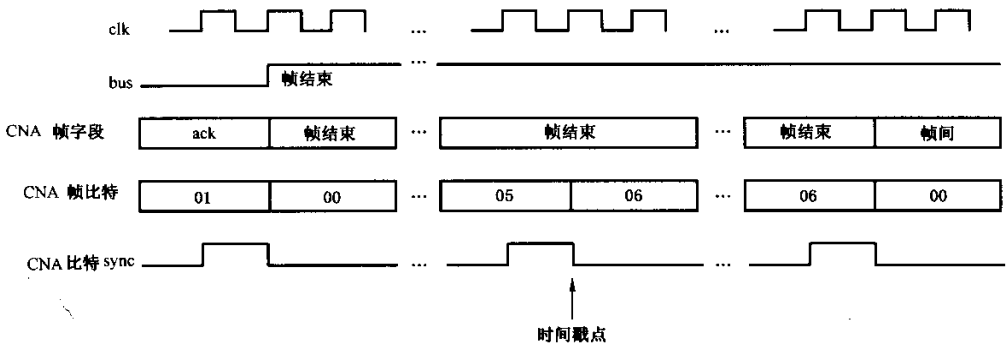


图 G.1 事件报文时间戳点

G.3 clockIdentity

clockIdentity 八位位组[0]~[7](见 7.5.2.2.3)应为节点制造商标识(制造商 ID)和节点相关的惟一序列号的组合。

示例:

X 公司的制造商 ID 是 AC7B₁₆。如果 X 公司希望开发 DeviceNet 设备,则 clockIdentity 的合法值是: 0201AC7BF22235C01₁₆,其中的 4 个八位位组数组 F2235C01₁₆,由 X 公司确保其在 X 公司分配的所有 DeviceNet 编号中是惟一的。

表 G.1 描述了 clockIdentity 的八位位组 2~7 的赋值,八位位组[0]和[1]在 7.5.2.2.3 中规定。

表 G.1 DeviceNet 的 clockIdentity 八位位组 0~7

技术		制造商 ID		序列号				字段
0	1	2	3	4	5	6	7	八位位组顺序十六进制
02	01	AC	7B	F2	23	5C	01	

00000010 00000001 10101100 01111011 11110010 00100011 01011100 00000001 比特

最高有效字节 最低有效字节

最高有效位 最低有效位

G.4 PTP 报文格式

PTP 报文的传输首先传输数据类型的最高有效字节,随后按照字节有效性递减的顺序依次传输。PTP 报文的第一个八位位组应紧跟在 DeviceNet 报文头的最后一个八位位组之后。

这些数据通过多个 DeviceNet 数据包(帧)进行发送,符合标准 DeviceNet 显式报文分组逻辑。所有 PTP 报文在 DeviceNet 上进行分段。所有数据包中都包含 DeviceNet 报文头。

表 G.2 规定了用于所有 PTP 报文数据包的 DeviceNet 报文头。该报文头出现在 PTP 报文的每个 DeviceNet 帧中。

表 G.2 用于所有 PTP 报文数据包的 DeviceNet 报文头

八位位组 0	八位位组 1	八位位组 2	类型 (资料性)	字段名
$h_0 h_1$	$j_0 j_1$	$k_0 k_1$	八位位组 八位位组 八位位组	Fragment=1,XID=0,Source MACID Fragment Type,Fragment Count R/R=1,Service Code=UCMM Service Code

G.5 PTP 的 DeviceNet 寻址

所有 PTP 报文,应通过具有无连接报文管理器(UCMM)能力的设备将其作为无连接响应报文(Message Group 为 3,Message ID 为 5)进行发送,以及通过仅组 2(Group 2 Only)的服务器将其作为无连接响应报文(Message Group 为 2,Message ID 为 3)进行发送。因此,子网上的每个节点都有其自身惟一的多播地址(控制器局域网(CAN)标识符)。所有域使用相同的多播地址。

所有 PTP 报文应将 DeviceNet 报文头中的请求/响应位设为 TRUE。

PTP 多播地址被其他 DeviceNet 功能共享,其中一些功能是点到点报文。为识别一个 PTP 报文,发送节点应将 DeviceNet 报文头中的目的节点字段设为它自己的节点地址。这样,报文进而被 UCMM 服务码标识为 PTP 报文。

事件类报文的 UCMM 服务码字段应为 88(58₁₆),普通类报文的 UCMM 服务码字段应为 89(59₁₆)。

当 networkProtocol 成员值是 DeviceNet 时(见 7.4.1),对数据类型 PortAddress 的任意参量(见 5.3.6):

- addressLength 成员值应为 2。
- addressField 成员值应是 DeviceNet mac ID。

G.6 transportSpecific 字段值

transportSpecific 字段的所有比特(见 13.3.2.1)应设为 0 被发送,且被接收方忽略。

附 录 H

(规范性附录)

PTP 在 ControlNET 上的传输

H.1 协议

本附录规定了 ControlNet 特定实现的 PTP 标准部分。本附录规范应适用于使用 ControlNet 作为通信网络的所有 PTP 实现。关于 ControlNet 的其他信息,见 ControlNet 国际组织¹⁰⁾提供的 ControlNet 规范。

注: ControlNet 也包含在 IEC 61158 中类型 2 中。

H.2 clockIdentity

clockIdentity 八位位组[2]~[7](见 7.5.2.2.3)应为节点制造商 ID 和节点相关的惟一序列号的组合。
示例:

X 公司的制造商 ID 是 AC7B₁₆。如果 X 公司希望开发 ControlNet 设备,则 clockIdentity 的合法值是: 0202AC7BF2235C01₁₆,其中的 4 个八位位组数组 F2235C01₁₆,由 X 公司确保其在 X 公司分配的所有 ControlNet 编号中是惟一的。

表 H.1 描述了 clockIdentity 的八位位组 2~7 的赋值,八位位组[0]和[1]在 7.5.2.2.3 中规定。

表 H.1 ControlNet 的 clockIdentity 八位位组 2~7

技术		制造商 ID		序列号				字段
0	1	2	3	4	5	6	7	八位位组顺序
02	02	AC	7B	F2	23	5C	01	十六进制
00000010	00000010	10101100	01111011	11110010	00100011	01011100	00000001	比特
最高有效字节 最高有效位						最低有效字节 最低有效位		

H.3 PTP 报文格式

PTP 报文的传输首先传输数据类型最高有效字节,随后按照字节有效性递减的顺序依次传输。PTP 报文的第一个八位位组应紧跟在 ControlNetLPacket 报文头的最后一个八位位组之后。

H.4 PTP 的 ControlNet 寻址

PTPL 数据包的目的地址字段应为 255(FF₁₆)(广播)。

对事件报文,PTPLPacket 的固定 Tag 字段应为 141(8D₁₆);对普通报文,PTPLLPacket 的固定 Tag 字段应为 142(8E₁₆)。

当 networkProtocol 成员值是 ControlNet 时(见 7.4.1),对数据类型 PortAddress 的任意参量(见 5.3.6):

- addressLength 成员值应为 2;
- addressField 成员值应是设备的 controlNet 节点号。

H.5 transportSpecific 字段值

transportSpecific 字段的所有比特(见 13.3.2.1)应设为 0 被发送,且被接收方忽略。

10) <http://www.controlnet.org>

附录 I
(规范性附录)

PTP 在 IEC 61158 类型 10 上的传输

I.1 背景

PROFINET(IEC 61158 类型 10)规定了一种现场总线通信系统。关于该现场总线通信系统如何被用来在系统中实现互操作的更多特定信息,在通信行规 IEC 61784-1:2007 和 IEC 61784-2:2007 中给出。

IEC 61784-1:2007 和 IEC 61784-2:2007 规定了通信行规簇(CPF),并且在一个 CPF 内有一个或多个通信行规(CP)。CP 对应于 IEC 61158 的类型。IEC 61784-1:2007 规定了各种现场总线。IEC 61784-2:2007 规定了各种实时以太网现场总线。PROFIBUSTM¹¹⁾ 和 PROFINET 在 CPF3 中规定。CP3/4、CP3/5 和 CP3/6 规定了 IEC 61784-2:2007 中的 PROFINET。

IEC 61158 类型 10 协议在 IEC 61158-6-10:2007 中规定。IEC 61158 类型 10 的服务在 IEC 61158-5-10:2007 中规定。

本附录规定了 IEC 61784-2:2007 的 CP3/4、CP3/5 和 CP3/6(即众所周知的 PROFINET)第 2 层上的协议。这些 CP 参见 IEC 61158-5-10:2007、IEC 61158-6-10:2007 和其他标准。

图 I.1 示出了一个 PTP 区域和一个 IEC 61158 类型 10 区域。使用边界时钟进行两个区域间的协议转换。

本附录协议功能上等同于本标准主要条款和附录中第 2 层上的 PTP 透明时钟和普通时钟的功能。然而,本附录协议具有不同的 PTP 报文编码,以满足 IEC 61158 内 IEC 61784-2:2007 的 CP3/4、CP3/5 和 CP3/6 的编码规范。本附录不适用于 IEC 61784-1:2007 的 CP3/1、CP3/2 和 CP3/3。

注:现有 ASIC 支持 IEC 61784-2:2007 的 CP3/4、CP3/5 和 CP3/6 在第 2 层的 PTP。

本附录的编码应用于满足 IEC 61158 内 IEC 61784-2:2007 的 CP3/4、CP3/5 和 CP3/6 的编码规范要求的实现。

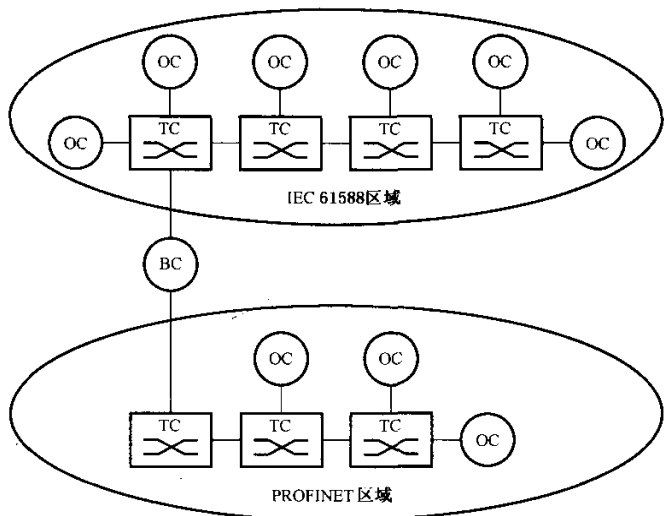


图 I.1 PROFINET 区域与其他域结合

11) PROFIBUSTM 是非盈利组织 PROFIBUS 用户组织(PNO)的商标名。该信息用来方便本标准的用户,不表示对 IEEE 这些产品的认可。如果等效产品能够显示可获得相同结果,则可以使用。

I.2 报文规范

不同报文名的映射见表 I.1。

表 I.1 报文映射

PROFINET 名称	PTP 名称
SyncPDU	Sync
FollowUpPDU	Follow_Up
AnnouncePDU	Announce
未使用	Delay_Req
未使用	Delay_Resp
DelayReqPDU	Pdelay_Req
DelayResPDU	Pdelay_Resp
DelayFuResPDU	Pdelay_Resp_Follow_Up
未使用	Signaling
未使用	Management

PROFINET 报文的编码及所用的首字母缩写词、缩略语和约定应按照 IEC 61158-5-10:2007 和 IEC 61158-6-10:2007 使用。

当 networkProtocol 成员值是 PROFINET 时(见 7.4.1),对数据类型 PortAddress 的任意参量(见 5.3.6):

- addressLength 成员值应为 6;
- addressField 成员值应为以太网报文头的 6 个八位位组源地址。

I.3 IEC 61158 类型 10 的 DLPDU

I.3.1 DLPDU 的抽象句法

表 I.2 给出了根据 IEEE Std 802.3:2005 的 DLPDU 抽象语法概要。

表 I.2 中字段的编码与解码应符合 IEEE Std 802.3:2005 的 DLPDU。

表 I.2 IEEE 802.3DLPDU 语法

DLPDU 名称	DLPDU 结构
DLPDU	Preamble ^a 、StartFrameDelimiter、DestinationAddress、SourceAddress、DLSDU ^b 、DLPDU _ Padding ^c 、FrameCheckSequence
DLSDU	LT、FIDAPDU
FIDAPDU	frameID, SyncPDU ^ AnnouncePDU ^ FollowUpPDU ^ DelayReqPDU ^ DelayResPDU ^ DelayFuResPDU
注: 根据 IEEE Std 802.3:2005, DLPDU 的最小长度为 64 个八位位组(不包括前导码,起始帧定界符)。	
^a 该字段至少包含 7 个八位位组。	
^b DLSDU 最小长度是 2 个八位位组。	
^c 填充的八位位组个数应在 0~46 的范围内,取决于 DLSDU 的长度。填充的值应设置为 0。	

I.3.2 DLPDU 字段目标地址的编码

DLPDU 字段应编码为数据类型 Octec[6]。目标地址字段的值应为 IEEE 802 的 MAC 地址。

对 PROFINET-PDU 中的 PTP,其值应根据表 I.3 设置。

表 I.3 多播 MAC 地址

组 MAC 地址	含 义
01-0E-CF-00-04-00	与 AnnoucePDU 和 FrameID(=FF00 ₁₆)联合用于时钟同步
01-0E-CF-00-04-01	与 AnnoucePDU 和 FrameID(=FF01 ₁₆)联合用于时间同步
01-0E-CF-00-04-XX	与 AnnoucePDU 和 FrameID(=FFxx ₁₆)联合用于同步
01-0E-CF-00-04-1F	与 AnnoucePDU 和 FrameID(=FF1F ₁₆)联合用于同步
01-0E-CF-00-04-20	与有 follow up 的 SyncPDU 和 FrameID(=0020 ₁₆)联合用于时钟同步
01-0E-CF-00-04-21	与有 follow up 的 SyncPDU 和 FrameID(=0021 ₁₆)联合用于时间同步
01-0E-CF-00-04-XX	与有 follow up 的 SyncPDU 和 FrameID(=00xx ₁₆)联合用于同步
01-0E-CF-00-04-3F	与有 follow up 的 SyncPDU 和 FrameID(=003F ₁₆)联合用于同步
01-0E-CF-00-04-40	与 FollowUpPDU 和 FrameID(=FF20 ₁₆)联合用于时钟同步
01-0E-CF-00-04-41	与 FollowUpPDU 和 FrameID(=FF21 ₁₆)联合用于时间同步
01-0E-CF-00-04-XX	与 FollowUpPDU 和 FrameID(=FFxx ₁₆)联合用于同步
01-0E-CF-00-04-5F	与 FollowUpPDU 和 FrameID(=FF3F ₁₆)联合用于同步
01-0E-CF-00-04-80	与 SyncPDU 和 FrameID(=0080 ₁₆)联合用于时钟同步
01-0E-CF-00-04-81	与 SyncPDU 和 FrameID(=0081 ₁₆)联合用于时间同步
01-0E-CF-00-04-XX	与 SyncPDU 和 FrameID(=00xx ₁₆)联合用于同步
01-0E-CF-00-04-9F	与 SyncPDU 和 FrameID(=009F ₁₆)联合用于同步
01-80-C2-00-00-0E 见注 3	与 DelayReqPDU 和 FrameID(=FF40 ₁₆),有 follow up 的 DelayResPDU 和 FrameID(=FF41 ₁₆),DelayFuResPDU 和 FrameID(FF42 ₁₆)和无 follow up 的 DelayResPDU 及 FrameID(FF43 ₁₆)联合用于点到点延时测量
注 1: 八位位组 1 包含单独/组地址比特(LSB) 注 2: 本表中从 01-0E-CF 开始的地址,是基于 PROFIBUS 用户组织 PNO 拥有的 OUI。 注 3: MAC 地址 01-80-C2-00-00-0E 表示从 IEEE Std 802.1AB 管理的多播地址池中获得的一个多播地址。该地址详见 F.3。	

I.3.3 字段 LT 的编码

LT 字段应依据 IEEE Std 802.3:2005(Unsigned 16)的值进行编码。本规范使用符合表 I.4 的值。

表 I.4 LT(长度/类型)

值(十六进制)	含 义
8892	PROFINET

I.3.4 字段 FrameID 的编码

FrameID 字段应编码为数据类型 Unsigned 16,其值依据表 I.5。该字段标识 APDU 的结构和类型。

表 I.5 FrameID

值(十六进制)	含 义	用 途
0000~001F	保留	—
0020	SyncPDU	有 follow_up 的 SyncPDU,用于时钟同步(等时同步应用)
0021	SyncPDU	有 follow_up 的 SyncPDU,用于时间同步
0022~00FF	保留	—
0080	SyncPDU	无 follow_up 的 SyncPDU,用于时钟同步(等时同步应用)
0081	SyncPDU	无 follow_up 的 SyncPDU,用于时间同步
0082~00FF	保留	—
FF00	AnnouncePDU(时钟)	AnnoucePDU 用于时钟同步(等时同步应用)
FF01	AnnouncePDU(时间)	AnnoucePDU 用于时间同步
FF02~FF1F	保留	—
FF20	FollowUpPDU(时钟)	FollowUpPDU 用于时钟同步
FF21	FollowUpPDU(时间)	FollowUpPDU 用于时间同步
FF22~FF3F	保留	—
FF40	DelayReqPDU	DelayReqPDU 用于路径延时测量
FF41	DelayResPDU	DelayResPDU 用于具有 follow_up 的路径延时测量
FF42	DelayFuResPDU	DelayFuResPDU 用于路径延时测量
FF43	DelayResPDU	DelayResPDU 用于无 follow_up 的路径延时测量
FF44~FFFF	保留	—

I.3.5 编码规范

网桥能在边沿转换 2 种格式。不同格式、不同参数和属性名称的映射见表 I.6。flagField(见 13.3.2.6)到 PROFINET 的转换见表 I.7。

表 I.6 参数和属性名称的映射

PROFINET 中的名称	报文类型	PTP 版本 2 中的名称
无对应部分	—	transportSpecific
FrameID	SyncPDU FollowUpPDU AnnouncePDU DelayReqPDU DelayResPDU DelayFuResPDU	messageType
无对应部分	—	VersionPTP
无对应部分	—	messageLength
SubdomainUUID	SyncPDU FollowUpPDU AnnouncePDU DelayReqPDU DelayResPDU DelayFuResPDU	domainNumeber

表 1.6 (续)

PROFINET 中的名称	报文类型	PTP 版本 2 中的名称
按照表 136	SyncPDU	flagField
SequenceId	SyncPDU FollowUpPDU AnnouncePDU DelayReqPDU DelayResPDU DelayFuResPDU	sequenceId
无对应部分	—	controlField
MasterSourceAddress	SyncPDU FollowUpPDU AnnouncePDU	clockIdentity
在 PROFINET 中规定	—	logMessageInterval
Seconds	SyncPDU	seconds(bit 0~31)
NanoSeconds	SyncPDU	Nanoseconds
EpochNumber	SyncPDU	seconds(bit 32~47)
CurrentUTCOffset	SyncPDU	currentUtcOffset
ClockAccuracy	SyncPDU AnnouncePDU	clockAccuracy
ClockClass	SyncPDU AnnouncePDU	clockClass
MasterPriority 1	SyncPDU AnnouncePDU	Priority 1
MasterPriority 2	SyncPDU AnnouncePDU	Priority 2
ClockVariance	SyncPDU AnnouncePDU	offsetScaledLogVariance
无对应部分	—	stepsRemoved
无对应部分	—	grandmasterIdentity
无对应部分	—	parentPortIdentity
RequestSourceAddress	DelayReqPDU DelayResPDU DelayFuResPDU	clockIdentity
RequestPortID	DelayReqPDU DelayResPDU DelayFuResPDU	portNumber

表 I.7 从 PTP 版本 2 到 PROFINET 的 flagField 转换

PROFINET 中的名称	PTP 版本 2 中的名称
最后 1 min 有 61 s	flagField.leap61
最后 1 min 有 59 s	flagField.leap59
由 AnnouncePDU 通知	flagField.alternateMasterFlag
在 FrameID 中编码	flagField.twoStepFlag
时间同步且 ClockStratum=1 或 2 时为 TRUE	flagField.timeTraceable
ClockStratum=1 或 2 时为 TRUE	flagField.frequencyTraceable
时钟同步时为 FALSE (ARP)。 如果标识符不是 INIT 或 DFLT,时间同步时为 TRUE; 否则为 FALSE	flagField.ptpTimescale
时钟同步时为 FALSE(ARP)。 如果标识符不是 INIT 或 DFLT,时间同步时为 TRUE; 否则为 FALSE	flagField.currentUtcOffsetValid
FALSE	flagField.unicastFlag
设为 FALSE	所有其他 flagField

IEC 61158 类型 10 的参数、属性和使用的首字母缩写词、缩略语以及约定的编码,应根据 IEC 61158-5-10:2007 和 IEC 61158-6-10:2007 进行。

附 录 J

(规范性附录)

缺省 PTP 行规

J.1 概述

每个缺省 PTP 行规规定了选项和属性的一个选择。每个选择规定了一个无需用户配置就可工作的系统。

J.2 通用要求

节点应实现规定了缺省值或选项的相应 PTP 行规的所有要求,使得这些缺省值或选项的应用不需要用户配置,即同制造商交付的一样。

J.3 延时请求-响应缺省 PTP 行规

J.3.1 标识

本 PTP 行规的标识值(见 19.3.3)如下:

PTP 行规

具有延时请求-响应机制的缺省 PTP 行规

版本 1.0

行规标识符:00-1B-19-00-01-00

J.3.2 PTP 属性值

所有节点都应支持取值范围,并且属性应具有如下缺省初始化值:

- defaultDS. domainNumber:缺省初始化值应为 0;
- portDS. logAnnounceInterval:缺省初始化值应为 1,可组态取值范围为 0~4;
- portDS. logSyncInterval:缺省初始化值应为 0,可组态取值范围为 -1~+1;
- portDS. logMinPdelayReqInterval:缺省初始化值应为 0,可组态取值范围为 0~5;
- portDS. announceReceiptTimeout:缺省初始化值应为 0,可组态取值范围为 2~10;
- defaultDS. Priority 1:缺省初始化值应为 128;
- defaultDS. Priority 2:缺省初始化值应为 128;
- defaultDS. slaveOnly:如果该参数是可组态的,则缺省值应为 FALSE;
- transparentClockdefaultDS. primaryDomain:缺省初始化值应为 0;
- τ ,见 7.6.3.2:缺省初始化值应为 1.0 s。

每个定义的范围,制造商可自行扩展。

J.3.3 PTP 选项

所有 15.5.4.1.7 和第 17 章的选项都是允许的。缺省地,除非明确地被管理规程激活,这些选项都是非活动的。

节点管理应实现本标准的管理报文机制。

最佳主时钟算法应为 9.3.2 规定的算法。

延时请求-响应机制应为缺省路径延时测量机制。也可实现对等延时机制。

注:每个链路只允许单一机制。边界时钟应在采用不同路径延时机制的链路之间使用。

J.3.4 时钟物理要求

J.3.4.1 频率精度

每个最高级时钟应保持频率偏移不大于 SI 秒的 0.01%。

J.3.4.2 频率调整范围

任何处于 SLAVE 状态的时钟都应能够校正其频率,以匹配满足 J.3.4.1 要求的任何主时钟。

注:频率调整范围应至少为 $\pm 0.025\%$ 。

J.4 点到点缺省 PTP 行规

J.4.1 标识

本 PTP 行规的标识值(见 19.3.3)如下:

PTP 行规

具有对等延时机制的缺省 PTP 行规

版本 2.0

行规标识符:00-1B-19-00-02-00

J.4.2 PTP 属性值

所有节点都应支持取值范围,并且属性应具有如下缺省初始化值:

——defaultDS. domainNumber:缺省初始化值应为 0;

——portDS. logAnnounceInterval:缺省初始化值应为 1,可组态取值范围为 0~4;

——portDS. logSyncInterval:缺省初始化值应为 0,可组态取值范围为-1~+1;

——portDS. logMinPdelayReqInterval:缺省初始化值应为 0,可组态取值范围为 0~5;

——portDS. announceReceiptTimeout:缺省初始化值应为 3,可组态取值范围为 2~10;

——defaultDS. Priority 1:缺省初始化值应为 128;

——defaultDS. Priority 2:缺省初始化值应为 128;

——defaultDS. slaveOnly:如果该参数是可组态的,则缺省值应为 FALSE;

——transparentClockdefaultDS. primaryDomain:缺省初始化值应为 0;

—— τ ,见 7.6.3.2:缺省初始化值应为 1.0 s。

每个定义的范围,制造商可自行扩展。

J.4.3 PTP 选项

所有 15.5.4.1.7 和第 17 章的选项都是允许的。缺省地,除非明确地被管理规程激活,这些选项都是非活动的。

节点管理应实现本标准的管理报文机制。

最佳主时钟算法应为 9.3.2 规定的算法。

对等延时机制应为缺省路径延时测量机制。也可实现延时请求-响应机制。

注:每个链路只允许单一机制。边界时钟应在采用不同路径延时机制的链路之间使用。

J.4.4 时钟物理要求

J.4.4.1 频率精度

每个最高级时钟应保持频率偏移不大于 SI 秒的 0.01%。

J.4.4.2 频率调整范围

任何处于 SLAVE 状态的时钟应能够校正其频率,以匹配满足 J.4.4.1 要求的任何主时钟。

注:频率调整范围应至少为 $\pm 0.025\%$ 。

附录 K

(资料性附录)

安全协议(实验性)

K.1 概述

本附录定义了一个 PTP 的实验性安全扩展,见 14.2。由于本附录不是规范性附录,不使用术语“应”来表示要求,而使用“要求”代替。为了在 PTP 将来版本中本附录成为规范性能正确实现本扩展,建议本扩展的实现者将本附录中的“要求”解释为“应”。

PTP 安全扩展和协议为 PTP 报文提供了组源认证、报文完整性以及重放攻击保护。

PTP 安全协议由以下两个基本机制构成:

- 完整性保护机制,它使用报文认证码来验证接收到的报文是由已认证的源发送的,传输中没有被修改并且是新的(即,不是报文的重放)。重放保护是通过计数器实现的;
- 挑战-响应机制,它用来确认新的源的真实性,维护信任关系的新鲜性。

K.2 协议综述

PTP 安全协议使用对称报文认证码功能。它提供组源认证、报文完整性以及重放保护。安全协议不提供认可。协议支持 HMAC-SHA1-96 和 HMAC-SHA256-128,并且允许将来增加其他报文认证码。要求这些算法的实现与参考文献 IETF RFC2104 [B10]、IETF RFC 2404 [B11]、NIST SHS 标准 [B20]以及 NIST HMAC 标准[B21]一致。

协议中的参与者共享对称密钥。该密钥可由整个域或域的子集共享。可通过手动配置或自动密钥管理协议来完成密钥分发,这两种方式 PTP 安全扩展均支持。密钥分发不在本规范的范围內。

PTP 安全协议中的参与者通过安全联盟(SA)进行通信。一个 SA 包含一个源(源端口,协议地址)、一个目的地(目的端口,协议地址)、一个密钥、一个随机 lifetimeId 以及一个重放计数器。SA 是单向的,它保护从源到目的地的流量。每个节点维护一个输入 SA 表,它用于对输入流量的验证;以及一个输出 SA 表,它用来对输出流量的保护。一个 SA 可以由单个发送方和多个接收方共享。发送方持有 SA 的一个单一副本,每个接收方都有其自己的一个副本。接收方的副本可能同时包含 SA 重放保护计数器的不同值,但它们均比同时存储在 SA 的发送方副本中的重放计数器小。

SA 是由发送方创建的,并与接收方通信。发送方可以选择为每个源和所有的目的地创建一个单一的 SA,即,创建一个 SA,源是其接口单播地址之一,目的地为“所有的”;或者可以选择为每一个源和目的地创建一个单一的 SA,即,创建多个具有相同的接口源地址和不同的多播和单播目的地的 SA。这是实现特定的。如果使用一个单一 SA 对应所有目的地,则重放保护计数器的翻转将快的多,这增加 SA 的更新速率。如果使用多个 SA 对应多个目的地,则输出 SA 表将更大。SA 上的报文接收方不关心发送方使用了哪种 SA 实现方法。

完整性校验值(ICV)是将算法 ID 确定的报文认证码功能应用到整个数据包(该数据包含有安全 AUTHENTICATION TLV)的结果,算法 ID 具有用密钥 ID 标识的适当密钥。只有知道共享密钥的节点可以分别修改报文和它的 ICV;因此,没有密钥的攻击者的任何篡改报文企图,在接收方验证 ICV 失败时均可被检测到。

重放保护机制依赖一个随机寿命 ID(lifetime ID)和重放保护计数器,这些都是 SA 的一部分。无论何时有数据包通过该 SA 时,重放计数器都增加 2。接收方验证数据包中的寿命 ID 与其输入 SA 表中发送方 SA 的寿命 ID 是否匹配,以及数据包中的计数器是否比 SA 中存储的值大。

K.3 通用要求

PTP 安全协议使用 PTP 报文头中的一个标志位来指示报文带有安全 AUTHENTICATION TLV。PTP 报文头的 flagField(见 13.3.2.6)被扩展 1 个比特,定义见表 K.1。

表 K.1 flagField. SECURE 标志

八位位组	比特	报文类型	名称	描述
0	7	所有	SECURE	如果报文尾部是安全 AUTHENTICATION TLV,则为 TRUE; 否则为 FALSE

要求将每个安全传输的 PTP 报文的 flagField. SECURE 标志设为 TRUE,并且包含 K.15 中定义的安全 AUTHENTICATION TLV 扩展。要求该扩展是附加到 PTP 报文的最终 TLV 扩展。

注:为方便安全协议的硬件实现,ICV 字段应是报文的最后字段。

在本附录中,某些情况下规定的动作是“安静地丢弃”一个接收到的报文。这表示该报文不被进一步处理而被丢弃,且没有为处理该报文而分配内部或外部资源。然而,可将事件用一个统计计数器记录,或者采取任何其他不分配新系统资源的类似动作。本定义限制了耗尽系统或网络资源的拒绝服务攻击的能力。

有些 PTP 系统要求支持安全时钟与非安全时钟的混合。例如:一个 PTP 系统由两个支持安全协议的最高级时钟(一个作为后备)、一组支持安全协议的普通时钟,以及一组不支持安全协议的 slave-only 时钟构成。该 PTP 系统的管理者可以允许非安全 slave-only 时钟同步到安全最高级时钟。正如本附录定义的一样,时钟间的所有通信都必须是安全的。因此,尽管从时钟可以解析最高级时钟发送的 Announce 和 Sync 报文,最高级时钟丢弃来自 slave-only 时钟的非安全 Delay_Req 报文。期望在行规或 PTP 下一版本的本附录扩展能引入其他“安全策略”,使得安全时钟和非安全时钟混合使用。像这样扩展的例子包括允许在一个有限报文速率下,一个时钟处理并应答非安全 Delay_Req 和 Pdelay_Req 报文,或者处理并应答非安全管理 Get 请求。

K.4 挑战-响应交换

挑战-响应交换是一个三方互相认证的协议,其中两个节点利用该协议确认其真实性与新鲜性。一个节点只有在成功执行了与另一个节点的挑战-响应后,才信任从该节点接收到的完整性与重放信息。

每个时钟都维护一个输入安全联盟列表。输入 SA 可以是静态的,即提前配置的;或者一旦时钟从一个与当前输入 SA 集中任何成员的 sourcePortIdentity 和协议地址不匹配的端口,接收到一个 PTP 报文时,就立即创建一个输入 SA。

一旦时钟初始化和新 SA 被创建,输入 SA 的信任状态就被设为 UNTRUSTED。信任状态只有在根据 K.10 成功执行了挑战-响应测试后才被设为 TRUSTED。每个输入 SA 维护一个定时器,该定时器测量自上一次接收到一个认证报文后经过的时间。当该定时器超时,则输入 SA 信任状态被设为 UNTRUSTED。如果 SA 不是静态的,则一旦超时该 SA 被丢弃。

当时钟接收到一个来自不信任源的报文时,它启动一个挑战-响应交换。输入报文必须将 flagField. SECURE 标志设为 TRUE,必须包含安全 AUTHENTICATION TLV,且必须通过 K.6 的完整性校验值(ICV)测试。如果接收到的报文不与一个输入 SA 相匹配,只要资源允许就创建一个新的动态输入 SA。输入 SA 维护挑战状态和一个挑战定时器。当发送了一个挑战请求且一个重放处于挂起时,挑战状态被设为 CHALLENGING。只要挑战状态是 CHALLENGING,就没有新的挑战请求通过该 SA 发送。如果未接收到挑战应答,则挑战定时器超时且挑战状态被设为 IDLE。

挑战-响应交换使用 AUTHENTICATION_CHALLENGE TLV。AUTHENTICATION_

CHALLENGE TLV 包含定义以下挑战类型的一个字段:challengeRequest,challengeResponseRequest 和 challengeResponse。AUTHENTICATION_CHALLENGE TLV 必须附加为安全信号报文的第一个 TLV。要求挑战交换中使用的信号报文仅用于安全协议操作。也启动挑战-响应交换来更新 K.5 中定义的安全联盟参数。

K.5 安全联盟更新交换

重放保护要求重放计数器不翻转。输出 SA 维护当前以及下一个随机产生的非零寿命 ID。当输出 SA 的重放计数器翻转时,SA 从当前寿命 ID 转换到下一个寿命 ID,并产生一个新的下一个寿命 ID。

输入 SA 维护当前以及下一个寿命 ID。一旦接收到安全 AUTHENTICATION TLV 中有下一个寿命 ID 的第一个认证报文,输入 SA 转换到该下一寿命 ID。输入 SA 从 AUTHENTICATION TLV 复制重放计数器的值,并启动一个挑战-响应交换来确定新的下一个寿命 ID。

在挑战-响应报文和挑战-响应-请求报文中使用 SECURITY_ASSOCIATION_UPDATE TLV 来传输下一个寿命 ID。如果响应方为每个与请求方通信的输出地址(单播、多播及 p-多播)都维护一个不同的输出 SA,则需要使用多个 TLV。

一旦一个给定密钥即将过期,安全联盟更新交换包括给交换提供下一个密钥 ID 以更新安全联盟密钥。密钥 ID 应经常更换,更换频率足以保护不受由于寿命 ID 重复使用而引起的重放攻击。因此,共享密钥分发机制必须更新共享密钥以维护足够的有效密钥。

输入 SA 维护当前和下一个密钥 ID。一旦接收到安全 AUTHENTICATION TLV 中有下一个密钥 ID 的第一个认证报文,输入 SA 转换到下一个密钥 ID。输入 SA 启动一个挑战-响应交换以确定新的下一个密钥 ID。下一个密钥 ID 用 SECURITY_ASSOCIATION_UPDATE TLV 在挑战-响应报文与挑战-响应-请求报文中传输。

K.6 完整性校验值(ICV)测试

ICV 是应用报文认证码功能的结果,报文认证码功能是由带有密钥 ID 标识的适当密钥的算法 ID 规定的。算法特定的处理规则见 K.15.7。

对每个从输出安全联盟发送的 PTP 报文,安全 AUTHENTICATION TLV 的 ICV 字段计算如下:

- 使用输出 SA 的密钥 ID 规定的密钥作为哈希算法(hash algorithm)要求的密钥值;
- 使用与密钥 ID 相关的、由算法 ID 确定的哈希算法来计算 ICV 值。算法 ID 是从由密钥 ID 索引的密钥列表数据集中找出的;
- 通过使用所选的哈希算法和保密密钥,在所有 PTP 报文字段上计算 ICV 值,计算从公共报文头的第一个八位位组开始,到安全 AUTHENTICATION TLV 的最后一个八位位组结束,包含最后一个八位位组。在该计算前,ICV 字段的所有比特都须设为 0。

计算输入报文的 ICV,并将其与在安全 AUTHENTICATION TLV 中携带的 ICV 进行比较。校验执行如下:

- 如果 AUTHENTICATION TLV 中的 keyId 指定的密钥不是有效的或是未知的,则 ICV 校验失败;
- 如果 AUTHENTICATION TLV 中的 algorithmId 不等于与密钥列表数据集中与 keyId 相关的算法 ID,则 ICV 校验失败;
- 通过算法选择的报文认证码功能与密钥,在所有 PTP 报文字段上计算 ICV 值,计算从公共报文头的第一个八位位组开始,到安全扩展 TLV 的最后一个八位位组结束,并包含安全扩展 TLV 的最后一个八位位组。在该计算前,AUTHENTICATION TLV 的 ICV 字段所有比特

值都须设为 0。如果计算出来的 ICV 与输入报文 AUTHENTICATION TLV 中的 ICV 不匹配,则 ICV 测试失败。

否则,ICV 测试通过。

K.7 安全联盟查找

将接收到的 PTP 报文与输入安全联盟进行匹配,以确定这些报文是否从一个信任源接收到的。如果 PTP 报文头的 sourcePortId 以及源协议地址与输入安全联盟的源端口以及源地址相匹配,并且报文的目的端口(如果规定了)以及目的地址与输入安全联盟的目的端口以及协议地址相匹配,则接收到的报文与该安全联盟相匹配。安全联盟查找指出一个匹配的安全联盟是否存在,并返回该安全联盟的信任状态。

K.8 keyId 检验

本测试验证一个输入报文的 keyId 是否与输入安全联盟值相匹配。将从一个信任源(也通过了 ICV 测试)接收到的 PTP 报文的 keyId 与 SA 的相应值进行比较。

将输入 keyId 与 SA 的 sa.keyId 进行比较。如果两者一样,则测试通过。如果 keyId 与 SA 维护的 sa.nextKeyId 相匹配,则将 SA 的 sa.nextKeyId 复制到 SA 的 sa.keyId,将 sa.nextKeyId 设为 0,并启动一个安全更新交换。与 sa.keyId 或与 sa.nextKeyId 不匹配的报文将被安静地丢弃。

只要输入 SA 的 sa.nextKeyId 为 0,且挑战状态不是 CHALLENGING,则每次 SA 成功接收一个新报文时,输入 SA 联盟均启动一个安全更新交换。

K.9 重放保护机制

重放保护机制依赖这样的事实:sourcePortIdentity、lifetimeId、replayCounter 三元组出现两次的概率极低(几乎为 0)。该概率取决于共享密钥的寿命,密钥交换频率越高则它越低。

lifetimeId 和 replayCounter 字段由输出安全联盟设置。对于通过输出 SA 发送的每个报文,sa.replayCounter 增加 2。当 sa.replayCounter 翻转时,lifetimeId 则由另一个随机的 lifetimeId 代替。sa.replayCounter 和 sa.lifetimeId 的值在安全 AUTHENTICATION TLV 中发送。

重放保护测试是在从一个信任输入 SA 接收到的包上进行的。将输入 lifetimeId 与输入 SA 的 sa.lifetimeId 相比较。如果两者一致,则将输入 replayCounter 字段与输入 SA 的 sa.replayCounter 相比较。如果输入 replayCounter 字段小于或等于输入 SA 的 sa.replayCounter,则报文被安静地丢弃。如果 lifetimeId 字段与输入 SA 维护的 sa.nextLifeTimeId 匹配,则将输入 AUTHENTICATION TLV 中的 replayCounter 字段复制到输入 SA 的 sa.replayCounter,将输入 SA 的 sa.nextLifetimeId 复制到输入 SA 的 sa.lifetimeId,将 sa.nextLifetimeId 设为 0,并启动一个安全更新交换。与 sa.lifetimeId 或与 sa.nextLifetimeId 不匹配的报文将被安静地丢弃。

只要输入 SA 的 sa.nextLifetimeId 为 0,且挑战状态不是 CHALLENGING,则每次 SA 成功接收一个新报文时,输入 SA 均启动一个安全更新交换。

K.10 挑战-响应检验

挑战-响应交换使用随机 nonce 来验证源的真实性与新鲜性。AUTHENTICATION_CHALLENGE TLV 包含一个请求 nonce 和一个应答 nonce。挑战请求和挑战-响应-请求的发送方将请求 nonce 设为一个随机数。挑战-响应或挑战-响应-请求的接收方将它发送的请求中的请求 nonce 与应答进行匹配。如果输入挑战报文中的响应 nonce,与挑战报文的请求 nonce 字段中发送的 nonce 不匹配,则挑战-响应检验失败,挑战报文被安静地丢弃。

挑战报文必须通过 ICV 测试。对于信任源,挑战报文也必须通过重放保护测试。如果这些测试中的任意一个测试失败,则挑战报文被安静地丢弃。

K. 11 共享密钥分发

分发共享密钥来增加和更新系统中每个 PTP 节点的安全密钥列表数据集,不在本标准范围内。

K. 12 密钥产生

密钥产生不在本标准范围内。下面描述了一个可选的密钥产生机制,该机制与 HMAC-SHA1-96 和 HMAC-SHA256-128 一起使用并增加共享密钥数据集。首先描述一般方法。

假设 $H(x)$ 为报文 x 的 n -bit 散列(hash), k 为要产生的 n -bit 密钥, p 为一个口令, ss 为短期(short-term)salt, sl 为长期(long-term)salt。变量 ss 、 sl 、 p 与 x 均为任意字符串。尽管 p 必须保密,但是 ss 和 sl 是公共值。典型地, sl 可以是运行它的网络名或组织名(例如,“物理实验室”), sl 可以是口令周期的日期或一个密钥名(例如,“2007 年 1 月~6 月”)。 $||$ 表示串联,则:

$$\begin{aligned} k &= H(sl || ss || p) \\ \text{for } i &= 1, 2, \dots, 1000 \\ k_i &= H(k_{i-1}) \end{aligned}$$

k 的最终值即(k_{1000})可以按需截取以形成最终密钥。长期 salt(sl)可在所有的站中初始化一次,并在以后不再改变,这防止为不同名称的另一网络所产生的攻击字典被重新用在本网络中。无论何时密钥被改变,短期 salt 都应被改变,这防止为先前密钥而计算的一个攻击字典被重新用来攻击一个新的密钥。

特别地,对 PTP 安全协议,可以采取下列步骤来产生密钥:

对每个 $keyId$:

- 为每个密钥选择一个可变长度的口令;
- 选择一个长期 salt;
- 选择一个短期 salt;
- 将长期 salt、短期 salt 和口令串联成一个单一报文;
- 使用 SHA-1 对报文进行散列计算;
- 对得到的报文散列再进行 1000 次散列计算;
- 通过 SHA-1 得到的输出的 20 个八位位组,就是 K. 13. 2. 3 中定义的 20 个八位位组的共享密钥。

K. 13 安全数据集

K. 13. 1 概述

定义在本条中的安全数据集只用于提供说明,不要求特定的实现。

安全数据集由输入和输出安全联盟列表、密钥列表以及 defaultDS 数据集参数列表构成。除非另有规定,所有成员的缺省值都是 0。

K. 13. 2 密钥列表

实现安全协议的 PTP 节点必须维护一个密钥列表。每个密钥列表条目由 $kl.keyId$ 、 $kl.algorithmId$ 、 $kl.securityKey$ 、 $kl.startTime$ 、 $kl.expirationTime$ 和 $kl.valid$ 比特组成。

K. 13. 2. 1 $kl.keyId$ (UInteger 16)

$kl.keyId$ 是密钥的惟一标识符。要求 0 值不用于指示一个有效密钥。

K. 13. 2. 2 $kl.algorithmId$ (UInteger 8)

$kl.algorithmId$ 指示与密钥一起使用的算法。

K. 13. 2. 3 kl. securityKey (Octet[N])

kl. securityKey 字段保留安全密钥。N 值与使用的算法有关。对 SHA-1 和 SHA-256, N=20。

K. 13. 2. 4 kl. startTime (Timestamp)

kl. startTime 指示密钥激活时间。当将有效性比特设为 TRUE 时激活密钥。密钥不应在 kl. startTime 前使用。

K. 13. 2. 5 kl. expirationTime (Timestamp)

kl. expirationTime 指示密钥过期时间。一旦密钥到期,有效性比特设为 FALSE,密钥不再被使用。一个值为 0 的 kl. expirationTime 指示密钥是永久的,因此不会过期。一个永久密钥应分发给所有的时钟。当所有其他时钟与本时钟通信失败时应使用永久密钥,该永久密钥仅限于建立一个安全联盟,然后变为非永久密钥。

K. 13. 2. 6 kl. valid (Boolean)

如果 kl. valid 设为 FALSE,则密钥无效且要求不再被安全协议使用。

K. 13. 3 安全联盟

安全联盟在两个列表中维护,包括输入安全联盟列表和输出安全联盟列表。每个安全联盟由 sa. srcPort, sa. srcAddress, sa. destPort, sa. destAddress, sa. replayCounter, sa. lifetimeId, sa. keyId, sa. nextLifetimeId, sa. nextKeyId, sa. trustState, sa. trustTimer, sa. trustTimeout, sa. challengeState, sa. requestNonce, sa. responseNonce, sa. challengeTimer, sa. challengeTimeout, sa. responseRequired, sa. challengeRequired 和 sa. typeField 组成,定义如下所述。

K. 13. 3. 1 sa. srcPort (PortIdentity)

将 sa. srcPort 与 PTP 报文头的 sourcePortIdentity 相匹配。对输出 SA,它指示时钟的一个端口;对输入 SA,它等于发送报文的时钟的 portIdentity。对输出 SA,sa. srcPort 的 portNumber 成员可设为全 1 的值以指示该 SA 用于时钟的所有端口。对输入 SA,要求 portNumber 成员或 clockIdentity 成员不为全 1 的值。

K. 13. 3. 2 sa. srcAddress (Octet[N])

sa. srcAddress 与 PTP 报文的源协议地址相匹配。对输出 SA,sa. srcAddress 可以设为全 1 值以指示该 SA 匹配所有地址。对输入 SA,要求 sa. srcAddress 不等于全 1 值。接收到的 PTP 报文的源协议地址与输入 SA 的 sa. srcAddress 参数相匹配。发送的 PTP 报文的源协议地址与输出 SA 的 sa. srcAddress 参数相匹配。对 IPv4 封装, N=4;对 IPv6, N=16;对以太网, N=6。

K. 13. 3. 3 sa. destPort (PortIdentity)

对输出 SA,sa. destPort 等于该报文要发送到的端口的 portIdentity;对输入 SA,它等于时钟的其中一个端口的 portIdentity。对输出 SA,可将 sa. destPort 设为全 1 值以指示“所有的时钟和所有的端口”。可将 sa. destPort 的 portNumber 设为全 1 值以指示该 SA 用于“一个特定时钟的所有端口”。例如,将发送到多播地址的 Sync 和 Announce 报文发送给所有的时钟和所有的端口。

K. 13. 3. 4 sa. destAddress (Octet[N])

sa. destAddress 等于 PTP 报文的目的地协议地址。对输出 SA,它是报文要发送到的时钟地址;对输入 SA,它等于时钟的单播地址之一或 PTP 多播地址之一。对输出 SA,sa. destAddress 可设为全 1 值以指示该 SA 匹配所有的地址。对 IPv4 封装, N=4;对 IPv6, N=16;对以太网, N=6。

K. 13. 3. 5 sa. replayCounter (UInteger 32)

对输出 SA,每次 PTP 报文通过该 SA 发送时,sa. replayCounter 增加 2。对输入 SA,sa. replayCounter 保存最后一个成功认证的输入 AUTHENTICATION TLV 的 replayCounter 字段的值。该值用在重放保护机制中。

K. 13. 3. 6 sa. lifetimeId (UInteger 16)

对输出 SA, sa. lifetimeId 是一个随机数, 用于标记通过该 SA 发送的所有包。对输入 SA, 将 sa. lifetimeId 与输入 AUTHENTICATION TLV 中的 lifetimeId 字段进行比较。值为 0 指示仍未设置 sa. lifetimeId。

K. 13. 3. 7 sa. keyId (UInteger 16)

sa. keyId 指示用于 ICV 计算的密钥。对输入 SA, 在挑战-响应交换期间确定它。

K. 13. 3. 8 sa. nextLifetimeId (UInteger 16)

当输出 SA 的 sa. replayCounter 翻转时, 将 sa. nextLifetimeId 的值复制到 sa. lifetimeId, 并产生一个新的随机非零值。输入 SA 的 sa. nextLifetimeId 在挑战-响应交换期间确定。值为 0 表示 sa. nextLifetimeId 还未被设置。

K. 13. 3. 9 sa. nextKeyId (UInteger 16)

sa. nextKeyId 指示当前密钥过期后将使用的密钥。对输入 SA, sa. nextKeyId 在挑战-响应交换期间确定。值为 0 指示 sa. nextKeyId 还未被设置。

K. 13. 3. 10 sa. trustState (Enumeration)

在成功的挑战-响应交换后将输入 SA 的 sa. trustState 设为 TRUSTED, 当超时或有初始化事件时将其设为 UNTRUSTED。不使用输出 SA 的 sa. trustState。

K. 13. 3. 11 sa. trustTimer (UInteger 16)

SA 每次接收到一个成功认证的 PTP 报文时, 输入 SA 的 sa. trustTimer 设为 sa. trustTimeout 值。每个周期 securityEvent 任务将其减 1。当 sa. trustTimer 减到 0 时, 将 SA 的 sa. trustState 设为 UNTRUSTED。输出 SA 的 sa. trustTimer 固定为 0。

K. 13. 3. 12 sa. trustTimeout (UInteger 16)

如果在一个安全事件的 sa. trustTimeout 周期内, 没有通过该 SA 接收到已成功认证的报文, 则输入 SA 超时。输出 SA 的 sa. trustTimeout 固定为 0。

K. 13. 3. 13 sa. challengeState (Enumeration)

输入 SA 的 sa. challengeState 指示 SA 是否正在等待一个挑战请求的应答。可以将其设为 CHALLENGING 或 IDLE 值。输出 SA 的 sa. challengeState 是固定的。

K. 13. 3. 14 sa. challengeTimer (UInteger 16)

一旦 SA 发送了一个挑战-请求或挑战-响应-请求, 则将该输入 SA 的 sa. challengeTimer 设为 sa. challengeTimeout。每个周期 securityEvent 任务将 sa. challengeTimer 减 1。当 sa. challengeTimer 减到 0 时, 该 SA 的 sa. challengeState 设为 IDLE。输出 SA 的 sa. challengeTimer 固定为 0。

K. 13. 3. 15 sa. challengeTimeout (UInteger 16)

在 securityEvent 的 sa. challengeTimeout 周期内, 如果没有接收到一个挑战-响应或挑战-响应-请求, 则将挑战状态改为 IDLE。输出 SA 的 sa. challengeTimeout 固定为 0。

K. 13. 3. 16 sa. requestNonce (UInteger 32)

sa. requestNonce 值是在挑战请求中发送的 requestNonce 字段, 并与响应中的 nonce 比较。它是一个用在挑战-响应交换中的随机产生的非零数。

K. 13. 3. 17 sa. responseNonce (UInteger 32)

sa. responseNonce 是接收到的挑战请求中的 requestNonce, 并在 SA 中维护, 以便于挑战-响应的产生。

K. 13. 3. 18 sa. challengeRequired (Boolean)

当需要发送一个挑战请求以更新输入安全联盟时, 将 sa. challengeRequired 设为 TRUE。

K. 13. 3. 19 sa. responseRequired (Boolean)

当需要发送一个挑战-响应时,将 sa. responseRequired 设为 TRUE。

K. 13. 3. 20 sa. typeField (Enumeration)

sa. typeField 可设为 STATIC 或 DYNAMIC 值。静态 SA 是预先配置的联盟,在时钟初始化期间维护,并存储在非易失内存中。如果没提前设置静态 SA,则需创建动态 SA 来与时钟通信。

下列属性在非易失内存和初始化事件中维护:

- 输出 SA: sa. srcPort、sa. srcAddress、sa. destPort、sa. destAddress 和 sa. keyId。
- 输入 SA: sa. srcPort、sa. srcAddress、sa. destPort、sa. destAddress、sa. trustTimeout、sa. challengeTimeout。

K. 13. 4 defaultDS 数据集安全变量

defaultDS 数据集包括以下两个安全参数: defaultDS. securityEnabled 和 defaultDS. numberSecurityAssociations。

K. 13. 4. 1 defaultDS. securityEnabled (Boolean)

如果将 defaultDS. securityEnabled 设为 TRUE,要求所有的 PTP 通信使用安全协议扩展。

K. 13. 4. 2 defaultDS. numberSecurityAssociations (UInteger 16)

defaultDS. numberSecurityAssociations 是时钟支持的安全联盟的最大个数,包括所有的输入和输出 SA 以及静态和动态 SA。

K. 14 协议操作**K. 14. 1 概述**

本条说明了安全协议的操作,以及对接收到的和发送的 PTP 报文的处理和挑战处理。也描述了处理超时和发送挑战报文的安全事件周期进程。K. 14. 6 详细说明透明时钟处理规则。

K. 14. 2 接收报文处理

图 K. 1 说明当 defaultDS. securityEnabled 设为 TRUE 时,对输入 PTP 报文的处理。只要测试结果是一致的,本条不要求特定的实现。处理步骤如下:

- a) 安静地丢弃接收到的 flagField. SECURE 比特不为 TRUE 的输入报文;
- b) 安静地丢弃接收到的没有附加安全 AUTHENTICATION TLV 的输入报文;
- c) 安静地丢弃没有通过按照 K. 6 的完整性校验验证测试的输入报文;
- d) 根据 K. 7 查找匹配的输入 SA;
- e) 如果没找到匹配的 SA 且有可用的 SA,则产生一个新 SA;如果没有可用的 SA,则安静地丢弃该报文;
- f) 如果 sa. trustState 是 UNTRUSTED:
 - 1) 如果输入报文是一个挑战报文,则按照 K. 14. 3 中的定义继续处理;
 - 2) 否则,将 SA 中的 sa. challengeRequired 比特置位,并丢掉报文;
- g) 如果信任状态为 TRUSTED:
 - 1) 如果依据 K. 8 的 keyId 测试失败,则将报文安静地丢弃;
 - 2) 如果依据 K. 9 的重放保护测试失败,则将报文安静地丢弃;
 - 3) 如果输入报文是一个挑战报文,则按 K. 14. 3 中的定义继续处理;
 - 4) 如果输入 SA 的 sa. nextLifetimeId 或 sa. nextKeyId 为 0,则将 SA 中的 sa. challengeRequired 比特设为 TRUE;
 - 5) 触发信任定时器。

K. 14.3 挑战处理

图 K.2 说明了对输入挑战报文的处理。挑战处理是对 K.14.2 中描述的一般输入报文处理的延续。

输入挑战报文处理包括以下步骤:

- a) 如果报文是一个 challengeRequest, 则将输入 SA 的 sa.responseRequired 成员设为 TRUE, 在该 SA 的 sa.responseNonce 成员中保存 requestNonce 字段, 并退出。
- b) 如果报文是一个 challengeResponse 或 challengeResponseRequest, 且输入 SA 的 sa.challengeState 不为 CHALLENGING(即, 该 SA 不期望一个响应), 或者如果依据 K.10 的挑战-响应校验失败, 则安静地丢弃报文。
- c) 否则(成功的挑战-响应):
 - 1) 将 sa.trustState 设为 TRUSTED;
 - 2) 将 sa.challengeState 设为 IDLE;
 - 3) 从 AUTHENTICATION TLV 将 lifetimeId 字段复制到输入 SA 的 sa.lifetimeId;
 - 4) 从 AUTHENTICATION TLV 将 replayCounter 字段复制到输入 SA 的 sa.replayCounter;
 - 5) 从 SECURITY_ASSOCIATION_UPDATE TLV 将 nextLifetimeId 复制到由 addressType 选择的相关输入 SA 的 sa.nextLifetimeId;
 - 6) 从 SECURITY_ASSOCIATION_UPDATE TLV 将 nextKeyId 复制到由 addressType 选择的相关输入 SA 的 sa.nextKeyId;
 - 7) 触发信任超时。
- d) 如果挑战报文是响应-请求, 将输入 SA 的 sa.responseRequired 比特设为 TRUE, 并将 requestNonce 存储到输入 SA 的 sa.responseNonce 字段。
- e) 丢弃报文并退出。

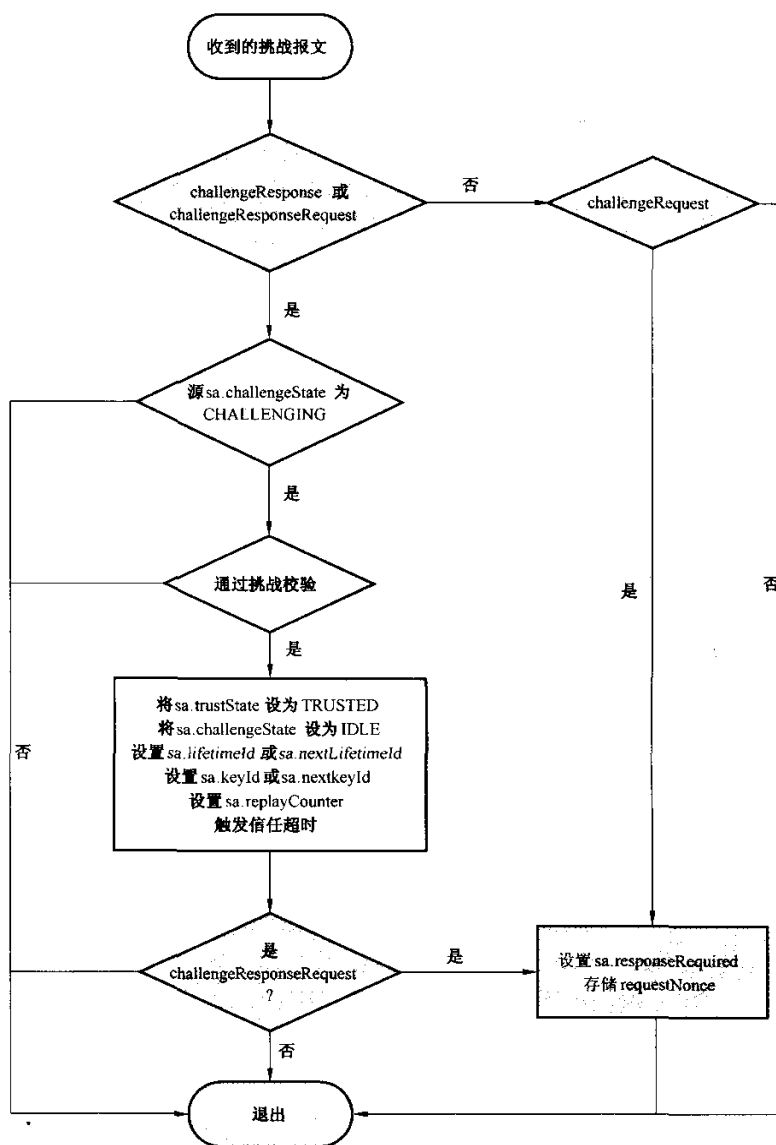


图 K.2 挑战处理

K.14.4 安全传输处理

图 K.3 说明输出 PTP 报文的附加处理。所有报文都附加 AUTHENTICATION TLV。要求按 K.6 计算 ICV。步骤包括：

- a) 将 flagField.SECURE 标志设为 TRUE；
- b) 附加 AUTHENTICATION TLV
 - 1) 从输出 SA 将 sa.replayCounter 复制到 TLV 的 replayCounter 字段；
 - 2) 从输出 SA 将 sa.lifetimeId 复制到 TLV 的 lifetimeId 字段；
 - 3) 从输出 SA 将 sa.keyId 复制到 TLV 的 keyId 字段；
 - 4) 从密钥列表数据集将对应于 sa.keyId 的 kl.algorithmId 复制到 TLV 的 algorithmId 字段。

- c) 计算 ICV,并将其放到 AUTHENTICATION TLV 的 ICV 字段;
- d) 将 SA 的 sa.replayCounter 增加 2;
- e) 如果 SA 的 sa.replayCounter 翻转,则:
 - 1) 将 SA 的 sa.nextLifetimeId 复制到 sa.lifetimeId 字段;
 - 2) 产生一个新的随机非零 sa.nextLifetimeId。确保 sa.nextLifetimeId 值与 sa.lifetimeId 值不同,否则,直到产生了一个不相等的值才产生非零随机值;
 - 3) 将 sa.replayCounter 设为 0。
- f) 发送报文。

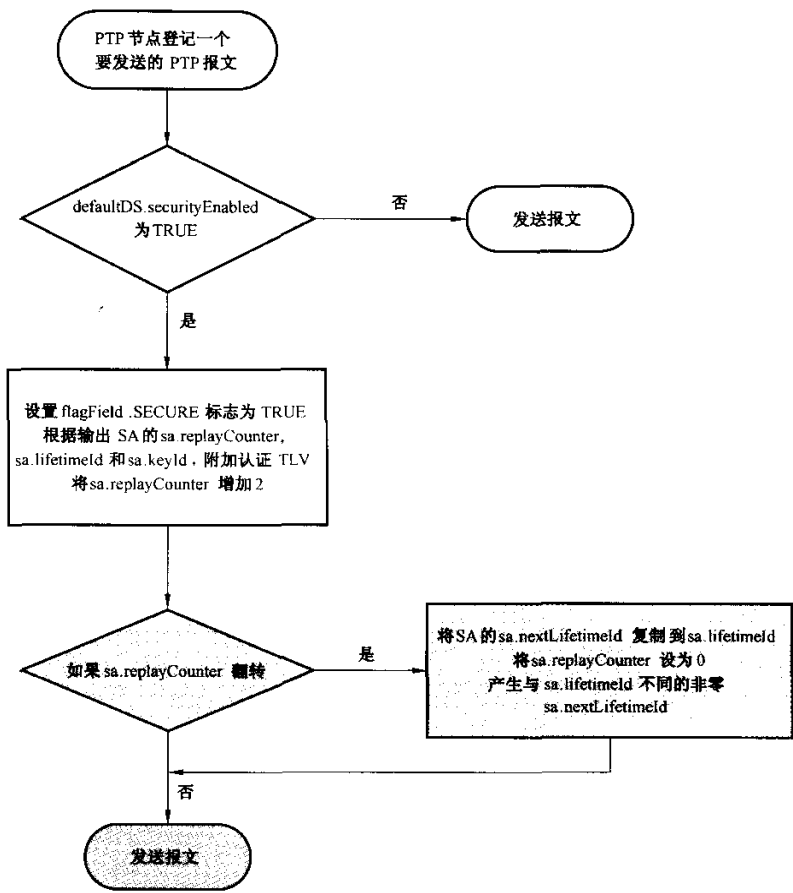


图 K.3 安全传输处理

K. 14.5 安全事件处理

安全事件是处理超时和发送挑战报文的一个周期性事件。应以足够的频率触发安全事件来处理超时。至少应在判定事件进程每次被触发时,触发安全事件进程。要求安全事件进程原子性地执行,见 3.1.2。

图 K.4 说明对每个输入 SA 的安全事件处理步骤。输入 SA 的安全事件处理如下:

- a) 如果 sa.challengeState 是 CHALLENGING,且 sa.challengeTimeout 已经超时,将 sa.challengeState 改为 IDLE。
- b) 如果 sa.challengeState 是 IDLE,且 sa.challengeRequired 和 sa.responseRequired 比特均为 TRUE,则:

- 1) 从所有匹配的输出生成 SA 构建 SECURITY_ASSOCIATION_UPDATE TLV;
 - 2) 产生随机 sa.requestNonce;
 - 3) 构建 AUTHENTICATION_CHALLENGE TLV。将 SA 的 sa.requestNonce 复制到 TLV 的 requestNonce 字段,将 SA 的 sa.responseNonce 复制到 TLV 的 responseNonce;
 - 4) 将 challengeType 设为 challengeResponseRequest,并发送报文;
 - 5) 将 sa.challengeState 设为 CHALLENGING。
- c) 如果 sa.challengeState 是 IDLE,sa.challengeRequired 比特为 TRUE,且 sa.responseRequired 比特为 FALSE,则:
- 1) 产生随机 sa.requestNonce;
 - 2) 构建 AUTHENTICATION_CHALLENGE TLV。将 SA 的 sa.requestNonce 复制到 TLV 的 requestNonce 字段,将 SA 的 sa.responseNonce 复制到 TLV 的 responseNonce;
 - 3) 将 challengeType 设为 challengeRequest,并发送报文;
 - 4) 将 sa.challengeState 设为 CHALLENGING。
- d) 如果 sa.challengeState 为 CHALLENGING,且 sa.responseRequired 比特为 TRUE;或 sa.challengeState 为 IDLE,sa.responseRequired 比特为 TRUE,且 sa.challengeRequired 比特为 FALSE,则:
- 1) 从所有匹配的输出生成 SA 构建 SECURITY_ASSOCIATION_UPDATE TLV;
 - 2) 构建 AUTHENTICATION_CHALLENGE TLV。将 TLV 的 requestNonce 字段设为 0,将 SA 的 sa.responseNonce 复制到 TLV 的 responseNonce;
 - 3) 将 challengeType 设为 challengeResponseRequest,并发送报文。
- e) 如果 sa.trustTimeout 到期,则设置 sa.trustState 为 UNTRUSTED。
- f) 设置 sa.responseRequired 和 sa.challengeRequired 比特为 FALSE。
- g) 退出。

安全事件进程也应处理密钥到期、激活与续订。安全事件不应使密钥超时,除非它可以获取准确时间以允许它确定一个密钥是否即将到期或已经到期。一旦到达密钥的开始时间,安全事件进程应激活密钥。只要安全事件进程不能获取足够准确的时间,所有密钥就被激活。安全事件进程通过 K.5 中定义的安全联盟更新交换,保证输出 SA 在密钥到期前交换新的 sa.keyId。当密钥即将到期时,或由于某些原因密钥需要被替代,则安全事件进程将输出 SA 的 sa.nextKeyId 复制到 sa.keyId 字段,并将 sa.nextKeyId 设为当前密钥到期时将要使用的密钥。

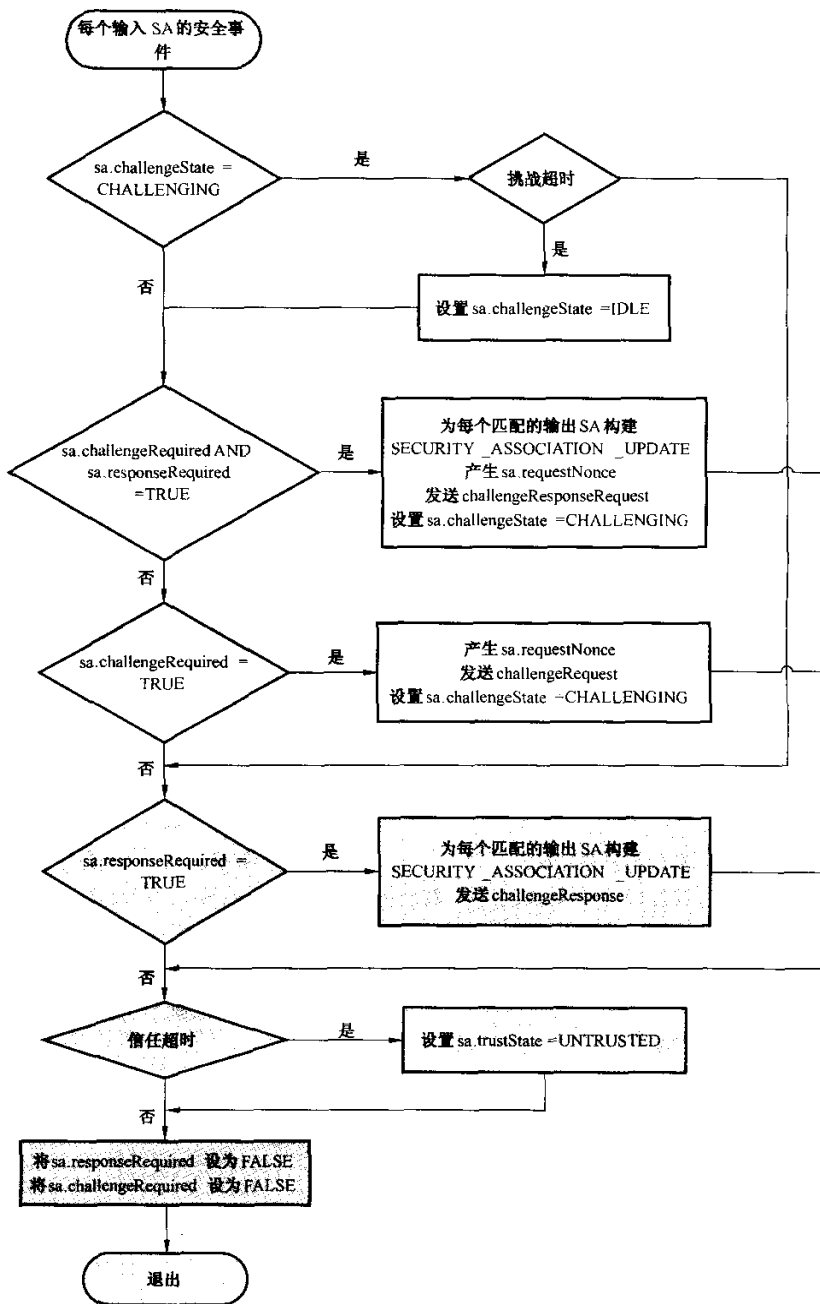


图 K.4 安全事件处理

K. 14.6 安全透明时钟处理规则

本条详细描述要求的附加透明时钟处理规则以支持安全 PTP 通信。PTP 透明时钟或是无安全意识的 (security-unaware), 或是有安全意识的 (security-aware), 或是有安全能力的 (security-capable) 时钟, 定义如下。

无安全意识的透明时钟忽略报文中的 flagField.SECURE 标志的值。被一个无安全意识的透明时钟修改的安全 PTP 报文不能通过 PTP 接收方的 ICV 测试, 并被安静地丢弃。在安全通信路径中, 不可

采用无安全意识的透明时钟。

有安全意识的透明时钟不修改 flagField.SECURE 标志为 TRUE 的 PTP 报文。根据网络的寻址规则转发 PTP 安全报文。有安全意识的透明时钟可以用在安全与非安全 PTP 通信共存的环境中(例如,通过不同的域)。

有安全能力的透明时钟,如果 defaultDS 数据集 defaultDS.securityEnabled 被设为 TRUE 时,可以参与 PTP 安全协议;或如果 defaultDS.securityEnabled 被设为 FALSE 时,可以作为一个有安全意识的时钟。要求有安全能力的透明时钟遵循安全协议处理规则,该规则适用于寻址到该透明时钟的以及由它发出的所有 PTP 报文。要求有安全能力的透明时钟遵循第 10 章与第 11 章中定义的透明时钟处理规则。另外,如果 defaultDS.securityEnabled 被设为 TRUE,要求遵循下列附加处理规则:

- a) 如果 flagField.SECURE 标志为 FALSE,则要求遵循透明时钟的常规的处理规则。下述进一步的规则适用于 flagField.SECURE 标志为 TRUE 的 PTP 报文。
- b) 要求透明时钟对所有 PTP 事件报文执行 K.6 中定义的 ICV 测试。如果 ICV 测试失败,则要求透明时钟安静地丢弃 PTP 报文。
- c) 要求双步透明时钟执行 Delay_Resp 与 Follow_Up 报文的 ICV 测试。如果 ICV 测试失败,则要求透明时钟安静地丢弃 PTP 报文。
- d) 要求按 11.5.2.2 产生 Follow_Up 报文的双步时钟,将 Sync 报文的安全 AUTHENTICATION TLV 复制到 Follow_Up 报文。要求双步时钟将 AUTHENTICATION TLV 的 replayCounter 字段增加 1。
- e) 在透明时钟根据 K.6 中定义的程序完成修改 PTP 报文字段(correctionField、twoStepFlag 等)后,要求它重新计算 PTP 报文的 ICV 值。要求透明时钟用已计算的正确 ICV 更新安全 AUTHENTICATION TLV 的 ICV 字段。

另外,如果 defaultDS 数据集的 defaultDS.securityEnabled 属性为 TRUE,则要求透明时钟不与一个主时钟谐振,除非它与那个时钟达到了完全信任的关系。

不要求透明时钟支持出于安全驻留时间与路径延时校正目的的安全联盟数据集。要求有安全能力的透明时钟支持和维护仅用于管理、谐振与点到点报文收发安全联盟。

K.15 AUTHENTICATION TLV

K.15.1 概述

要求将 AUTHENTICATION TLV 附加到 flagField.SECURE 标志被置位的所有 PTP 报文(见表 K.2)。要求按本条定义来确定 TLV 的字段值。要求发送时将保留字段设为 0,并在接收时将其忽略。

表 K.2 AUTHENTICATION TLV

比特								八位位组
7	6	5	4	3	2	1	0	
tlvType = AUTHENTICATION								2
lengthField								2
lifetimeId								2
replayCounter								4
keyId								2
algorithmId								1
保留								1
填充								M
ICV(完整性校验值)								N

K. 15.2 tlvType

tlvType 的值为 AUTHENTICATION。

K. 15.3 lengthField

TLV 的长度取决于 ICV 和填充长度。对本标准中定义的所有扩展,长度是 26(十进制)。

K. 15.4 lifetimeId (UInteger 16)

lifetimeId 是由 SA 确定的一个固定数。lifetimeId 作为重放保护机制的一部分被校验。要求不将 lifetimeId 设为 0。

K. 15.5 replayCounter (UInteger 32)

对每个通过 SA 发送的包,将 replayCounter 增加 2。replayCounter 作为重放保护机制的一部分被校验。

K. 15.6 keyId (UInteger 16)

密钥标识符(keyId)字段用来选择在许多共享密钥中可能使用哪一个。

K. 15.7 algorithmId (UInteger 8)

要求 algorithmId 的可能值取自表 K. 3 的枚举。

表 K. 3 algorithmId 值

algorithmId	值(十六进制)
NULL	0
HMAC-SHA1-96	1
HMAC-SHA256-96	2
保留	3~80
实现特定的	81~FF

要求所有支持安全扩展的 PTP 节点支持 HMAC1-SHA1-96 算法。

HMAC 处理在 NIST SHS [B20]和 NIST HMAC [B21]中定义。它定义了输出截取与填充过程。SHA1 与 SHA256 都使用大小为 512 比特的块。在散列计算开始前报文用值为零的字节进行填充,以保证被填充的报文是 512 比特的倍数。SHA1 的输出从 160 比特截取为 96 比特,SHA256 的输出从 256 比特截取为 128 比特,以产生 ICV 值。截取选择所产生散列的最左边的比特。

NULL 算法不提供完整性保护,仅出于测试目的而定义。对于 NULL 算法,ICV 字段长度为 0。

K. 15.8 填充(Octet[M])

要求将填充字段的值设为 0,且接收时将其忽略。填充字段长度 M 依据表 K. 4 所列的 algorithmId。

注:填充字段长度的选择使得,对于 IEC 61588 标准本版本中定义的所有 algorithmId,AUTHENTICATION TLV 长度是固定的。固定长度的 AUTHENTICATION TLV 方便硬件实现。然而,未来的 algorithmId 可以定义“和”不为该固定长度的 ICV 和填充长度。

K. 15.9 ICV (Octet[N])

计算 ICV 值的方法在 K. 6 中规定。ICV 长度 N 依据表 K. 4 所列的 algorithmId。

表 K. 4 ICV 和填充长度

algorithmId	ICV 长度(字节)	填充长度(字节)
NULL	0	16
HMAC-SHA1-96	12	4
HMAC-SHA256-128	16	0

K. 16 AUTHENTICATION_CHALLENGE TLV

K. 16.1 概述

AUTHENTICATION_CHALLENGE TLV 用于认证挑战-响应交换。

表 K. 5 说明了该扩展。要求 AUTHENTICATION_CHALLENGE TLV 在信号报文中发送。要求将 AUTHENTICATION_CHALLENGE TLV 附加为报文中的第一个 TLV。

表 K. 5 AUTHENTICATION_CHALLENGE TLV

比特								八位位组
7	6	5	4	3	2	1	0	
tlvType= AUTHENTICATION_CHALLENGE								2
lengthField								2
challengeType								1
保留								1
requestNonce								4
responseNonce								4

K. 16.2 tlvType

tlvType 字段为 AUTHENTICATION_CHALLENGE。

K. 16.3 lengthField

lengthField 值为 14。

K. 16.4 challengeType (UInteger 8)

要求 challengeType 的可能值取自表 K. 6 的枚举。

表 K. 6 challengeType 值

challengeType	值(十六进制)
challengeRequest	0
challengeResponseRequest	1
challengeResponse	2
保留	3~FF

K. 16.5 requestNonce (UInteger 32)

requestNonce 是由挑战-请求或挑战-响应-请求的发送方产生的一个随机数。requestNonce 在挑战-响应报文中应设为 0。

K. 16.6 responseNonce (UInteger 32)

挑战响应方将 challengeRequest 和 challengeResponseRequest 报文中的 requestNonce,复制到挑战-响应-请求和 challengeResponse 中的 responseNonce。ChallengeRequest 报文中的 responseNonce 被设为 0。

K. 17 SECURITY_ASSOCIATION_UPDATE TLV

K. 17.1 概述

SECURITY_ASSOCIATION_UPDATE TLV 用于传输安全联盟的 lifetimeId 值,一旦当前安全联盟的重放计数器发生翻转时则使用该 lifetimeId 值。nextKeyId 在密钥一旦被替代时使用。要求在挑战-响应和挑战-响应-请求的信号报文中发送安全联盟更新 TLV。该 TLV 可用于传递与所有地址

(单播、多播和 p-延时多播)相关的更新的安全联盟;或者如果挑战响应方维护不同的输出安全联盟,则为一个特定地址传递安全联盟更新信息。在同一个报文中可以发送几个 TLV,每个 TLV 为一个特定地址提供更新的 SA。表 K.7 说明了该扩展。要求在发送时将保留字段设为 0,接收时将其忽略。

表 K.7 SECURITY_ASSOCIATION_UPDATE TLV

比特							八位位组
7	6	5	4	3	2	1	
tlvType=SECURITY_ASSOCIATION_UPDATE							2
lengthField							2
addressType							1
保留							1
nextKeyId							2
nextLifetimeId							2

K. 17.2 tlvType

tlvType 字段为 SECURITY_ASSOCIATION_UPDATE。

K. 17.3 lengthField

lengthField 值为 0。

K. 17.4 addressType (UInteger 8)

要求 addressType 的可能值取自表 K.8 的枚举。

表 K.8 addressType 值

addressType	值(十六进制)
所有的	0
多播	1
P-多播	2
单播	3
保留	4~FF

K. 17.5 nextKeyId (UInteger 16)

一旦当前密钥到期或被取代时,nextKeyId 指示输出安全联盟将要使用的密钥。nextKeyId 用来更新输入安全联盟。

K. 17.6 nextLifetimeId (UInteger 16)

一旦当前输出安全联盟的重放计数器发生翻转时,nextLifetimeId 指示输出安全联盟将要使用的 lifetimeId。nextLifetimeId 用来更新输入安全联盟。

附录 L (资料性附录)

累积频标因子偏移的传输(实验性)

L.1 概述

本附录定义了实验性的累积频标因子偏移的 PTP 扩展,见 14.2。由于本附录不是规范性附录,不使用术语“应”来表示要求,而使用“要求”代替。为了在 PTP 将来版本中本附录成为规范性时能正确实现本扩展,建议本扩展的实现者将本附录中的“要求”解释为“应”。

在一些补偿方案中,不是直接调整边界时钟的相位,而是通过减少相位和频率误差的方式来调整频率。边界时钟使用它从主时钟接收到的连续的时间戳来计算频标因子;特别地,使用主时钟发送每个时间戳(originTimestamp 或 preciseOriginTimestamp)的时间,以及它接收每个时间戳的时间。通过使用该信息,相对于主时钟计算频标因子。然而,如果边界时钟也知道相对于其最高级时钟的累积频标因子,就可以减小在边界时钟所累积的相位误差。本条规定了一个可选的 TLV,它可被用于从边界时钟到其从时钟之间传输累积频标因子信息。特别地,TLV 积累了频标因子同 1 之间的累积差。频标因子偏移是指频标因子同 1 之间的差;累积频标因子偏移是指频标因子偏移在多个边界时钟跳的累积。

L.2 使用累积频标因子的频率补偿方案的描述

Balasubramanian 等[B3]详细描述了频率一种补偿方案,该方案使用频标因子(frequencyScaleFactor)。每个相继的边界时钟节点在各个 syncInterval 计算一个频率补偿值,如下:

$$\text{freqCompensationValue}_{k,0} = 1$$

$$\text{freqCompensationValue}_{k,n} = F_{k,n} \times \text{freqCompensationValue}_{k,n-1} \quad \dots\dots\dots (L.1)$$

其中, $F_{k,n}$ 为节点 k 在 syncInterval n 计算出的(frequencyScaleFactor)。调整频率,即补偿频率通过自由运行的本地振荡器频率乘以当前(freqCompensationValue)而得到。 $F_{k,n}$ 的计算见 L.3。

Wang 等[B25]指出,使用本补偿方案的一串边界时钟的相位误差累积可通过 2 种途径大大缩减:(1)在相继的边界时钟对之间同步报文交换,使得在主时钟同步到它的主时钟之后,从时钟立刻同其主时钟交换报文并同步;(2)使用(cumulativeFrequencyScaleFactor)而非(frequencyScaleFactor)来得到补偿的频率。(cumulativeFrequencyScaleFactor),即 $F_{\text{cum},k,n}$,由下式计算(注意,累积是对一串节点而不是时间上的累积):

$$F_{\text{cum},k,n} = F_{\text{cum},k-1,n} \times F_{k,n} \quad \dots\dots\dots (L.2)$$

最高级时钟的(frequencyScaleFactor),即 $F_{0,n}$,等于 1(因为最高级时钟不通过协议同步到另一个时钟)。这样,

$$F_{\text{cum},k,n} = \prod_{i=1}^k F_{i,n} \quad \dots\dots\dots (L.3)$$

(frequencyScaleFactorOffset)被定义为(frequencyScaleFactor)同 1 之间的差,见公式(L.4)。

$$\delta_{k,n} = F_{k,n} - 1$$

$$F_{k,n} = 1 + \delta_{k,n} \quad \dots\dots\dots (L.4)$$

把公式(L.4)带入公式(L.3),并有 $\delta_{k,n} \ll 1$ (即(frequencyScaleFactor)同 1 非常接近,这可通过 L.3 中的公式(L.6)或公式(L.8)来减小),累积频标因子可按公式(L.5)写为:

$$F_{\text{cum},k,n} \cong 1 + \sum_{i=1}^k \delta_{i,n} \quad \dots\dots\dots (L.5)$$

公式(L.5)说明了(cumulativeFrequencyScaleFactor)可通过频标因子偏移的累积和(而不是频标

因子的累积和)获得。累积和的计算量较小,并通过 L. 4 规定的 CUM_FREQ_SCALE_FACTOR_OFFSET TLV 进行传递和累积。

L. 3 累积频标因子偏移的通用规范

如果一个边界时钟和其主时钟实现了此 TLV,则在接收到附在 Sync 或者 Follow_Up 报文上的 TLV 时,要求该边界时钟计算它的频标因子偏移。

如果该 TLV 附在 Sync 报文上,则〈frequencyScaleFactorOffset〉是在接收每个 Sync 报文时进行计算的(由于本条中的所有变量都是参考同一节点,因此为简化而省略节点序号 k),如公式(L. 6)和公式(L. 7)所示:

$$F_n = \frac{(T_{1,n} - T_{1,n-1}) + (T_{1,n} + d_n - T_{2,n})}{T_{2,n} - T_{2,n-1}} \dots\dots\dots (L. 6)$$
$$\delta_n = F_n - 1 \dots\dots\dots (L. 7)$$

其中:

- δ_n 为接收第 n 个 Sync 报文的〈frequencyScaleFactorOffset〉;
- F_n 为接收第 n 个 Sync 报文的〈frequencyScaleFactor〉;
- T_{1,n} 为包含在第 n 个 Sync 报文的 originTimestamp;
- T_{2,n} 为接收第 n 个 Sync 报文的时间戳;
- d_n 为如下两项之和:

- 接收第 n 个 Sync 报文路径上的当前(即第 n 个 Sync 报文的接收时刻)测量的传输时间;
- 第 n 个 Sync 报文的 correctionField。

如果 TLV 附在 Follow_Up 报文上,〈frequencyScaleFactorOffset〉是在接收每个 Follow_Up 报文时进行计算的,如公式(L. 8)和公式(L. 9)所示:

$$F_n = \frac{(T_{1,n} - T_{1,n-1}) + (T_{1,n} + d_n - T_{2,n})}{T_{2,n} - T_{2,n-1}} \dots\dots\dots (L. 8)$$
$$\delta_n = F_n - 1 \dots\dots\dots (L. 9)$$

其中:

- δ_n 为接收第 n 个 Follow_Up 报文的〈frequencyScaleFactorOffset〉;
- F_n 为接收第 n 个 Follow_Up 报文的〈frequencyScaleFactor〉;
- T_{1,n} 为包含在第 n 个 Follow_Up 报文的 preciseOriginTimestamp;
- T_{2,n} 为接收第 n 个 Sync 报文的时间戳;
- d_n 为如下三项之和:

- 接收第 n 个 Sync 报文和相应 Follow_Up 报文路径上的当前(即第 n 个 Sync 报文的接收时刻)测量的传输时间;
- 第 n 个 Sync 报文的 correctionField;
- 对应第 n 个 Sync 报文的 Follow_Up 报文的 correctionField。

边界时钟通过把如上计算出的 δ_n 加到最新从主时钟接收到的 CUM_FREQ_SCALE_FACTOR_OFFSET TLV(见 L. 4)的〈cumulativeFrequencyScaleFactorOffset〉的值,来计算相对于最高级时钟的〈cumulativeFrequencyScaleFactorOffset〉。

如果边界时钟实现了此 TLV,但其主时钟未实现此 TLV,则要求边界时钟将其计算的〈cumulativeFrequencyScaleFactorOffset〉设为如上计算的 δ_n。

注:如果一个边界时钟未实现此 TLV,则它仍可使用补偿方案,该补偿方案不使用〈cumulativeFrequencyScaleFactor〉,即 F_n。这种情况下,边界时钟忽略任何它从其主时钟接收到的 CUM_FREQ_SCALE_FACTOR_OFFSET TLV,并且不发送 CUM_FREQ_SCALE_FACTOR_OFFSET TLV 到其从时钟。

如果边界时钟实现了 CUM_FREQ_SCALE_FACTOR_OFFSET TLV,则该 TLV 应在每一个

Follow_Up 报文中发送。仅当通信路径中的所有时钟都支持该扩展,并将 Delay_Req 报文填充为相同长度来避免不对称影响时,CUM_FREQ_SCALE_FACTOR_OFFSET TLV 可在 Sync 报文中发送。

L. 4 CUM_FREQ_SCALE_FACTOR_OFFSET TLV 规范

要求 CUM_FREQ_SCALE_FACTOR_OFFSET TLV 格式的规定见表 L. 1。

表 L. 1 CUM_FREQ_SCALE_FACTOR_OFFSET TLV 格式

比特								八位位组	偏移
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
cumulativeFrequencyScaleFactorOffset								4	4

L. 4. 1 tlvType

要求 tlvType 的值为 CUM_FREQ_SCALE_FACTOR_OFFSET。

L. 4. 2 cumulativeFrequencyScaleFactorOffset

cumulativeFrequencyScaleFactorOffset 的值是最新计算出的边界时钟(边界时钟发送了包含该 TLV 的 Sync 报文)相对于最高级时钟的 \langle cumulativeFrequencyScaleFactorOffset \rangle ,该值乘以 2^{41} 后以 2 的补码形式用 32 位带符号整型数表示。乘以 2^{41} 后,截取 \langle cumulativeFrequencyOffset \rangle 的任何剩余小数部分,即正的 \langle cumulativeFrequencyScaleFactorOffset \rangle 由它的最小值代替,负的 \langle cumulativeFrequencyScaleFactorOffset \rangle 由它的最大值代替。

参 考 文 献

- [B1] Allan D W, Ashby N, Hodge C. "Fine-tuning time in the Space Age," IEEE Spectrum, Mar. 1998.
- [B2] Allan D W, Ashby N, Hodge C C. "The science of timekeeping," Hewlett Packard Application Note 1289, 1997.
- [B3] Balasubramanian S, Harris K R, Moldovansky A. "A frequency compensated clock for precision synchronization using IEEE 1588 protocol and its application to ethernet," Workshop on IEEE 1588, Gaithersburg, MD, 2003.
- [B4] Eidson J C, et al. "Method for recognizing events and synchronizing clocks," U. S. Patent 5,566, 180, Oct. 15, 1996.
- [B5] IETF RFC 768 (1980) "User Datagram Protocol," Postel, J., Aug. 1980.
- [B6] IETF RFC 791 (1981) "Internet Protocol," Postel, J., Sept. 1981.
- [B7] IETF RFC 1305 (1992) "Network Time Protocol (Version 3)," Mills, D. L., Mar. 1992.
- [B8] IETF RFC 1589 (1994) "A Kernel Model for Precision Timekeeping," Mills, D. L., Mar. 1994.
- [B9] IETF RFC 1624 (1994) "Computation of the Internet Checksum via Incremental Update," Rijnsinghani, A., ed., May 1994.
- [B10] IETF RFC 2104 (1997) "HMAC: Keyed-Hashing for Message Authentication," Krawczyk, H., Bellare, M., and Canetti, R., Feb. 1997.
- [B11] IETF RFC 2404 (1998) "The Use of HMAC-SHA-1-96 within ESP and AH," Madson, C., Nov. 1998.
- [B12] IETF RFC 2460 (1998) "Internet Protocol, Version 6 (IPv6) Specification," Deering, S. and Hinden, R., Dec. 1998.
- [B13] IETF RFC 2783 (2000) "Pulse-Per-Second API for UNIX-like Operating Systems," Mogul, J. and Stone, J., Mar. 2000.
- [B14] IETF RFC 4291 (2006) "IP Version 6 Addressing Architecture," Hinden, R. and Deering, S., Feb. 2006.
- [B15] International Vocabulary of Basic and General Terms in Metrology (VIM), BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, OIML, 2nd ed., 1993, definition 6. 10.
- [B16] ISO/IEC 9945:2003 Information technology Portable Operating System Interface (POSIX®).
- [B17] ISO 8601:2004 Data elements and interchange formats Information interchange Representation of dates and times.
- [B18] Items and pointers on (<http://tycho.usno.navy.mil/time.html>), a Web page maintained by the U. S. Naval Observatory.
- [B19] ITU-T Recommendation G. 810, Definitions and Terminology for Synchronization Networks, ITU-T, Geneva, Aug. 1996, Corregendum 1, Nov. 2001.
- [B20] National Institute of Standards and Technology, Secure Hash Signature Standard (SHS) (FIPS PUB 180-2).
- [B21] National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC) (FIPS PUB 180-2).
- [B22] Perlman R, Interconnections, Bridges and Routers, Reading, MA: Addison-Wesley, 1992.
- [B23] Service de la Rotation Terrestre, Observatoire de Paris, 61, Av. De l'Observatoire 75014 PARIS

(France).

[B24] Sullivan D B, Allan D W, Howe D A, Walls F L eds. "Characterization of clocks and oscillators," NIST Technical Note 1337, Mar. 1990.

[B25] Wang S, Cho J, Garner G M. "Improvements to boundary clock based time synchronization through cascaded switches," 2006 Conference on IEEE 1588, Gaithersburg, MD, Oct. 2-4, 2006.
