### 版本修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本记录 | 版本编号 | 作者 | 说明 |
|  | V1.0 | @ | 针对需求#29928，首次实现DNS探测功能方案。 |
|  |  |  |  |

### 需求场景描述

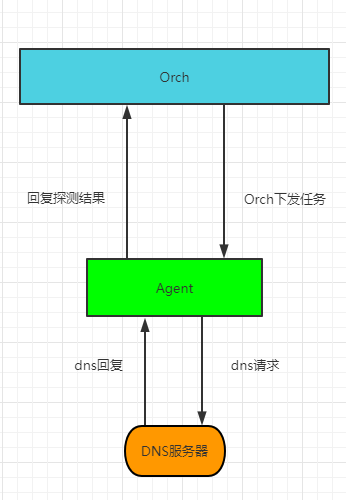
国电投需要通过CPE来探测数据中心的DNS解析情况，希望控制台Probe Template支持DNS探测，DNS解析探测时，需要指定以CPE的lan接口地址为源地址，确保DNS流量可以进隧道，通过数据中心DNS服务器去解析。

该需求主要供国电投运维管理团队使用，手动填写dns ip（一般是数据中心dns服务器）。

结果除了解析情况和dst ip以外，最好能加上发起请求的源ip和CPE命名，这样就可以直观的看到是哪家单位的哪一段地址请求的解析结果；也需要对解析的质量进行分析，需要包括类似ping的时间、延迟、抖动、丢包和target ip等信息。

### **功能设计描述**

功能逻辑：

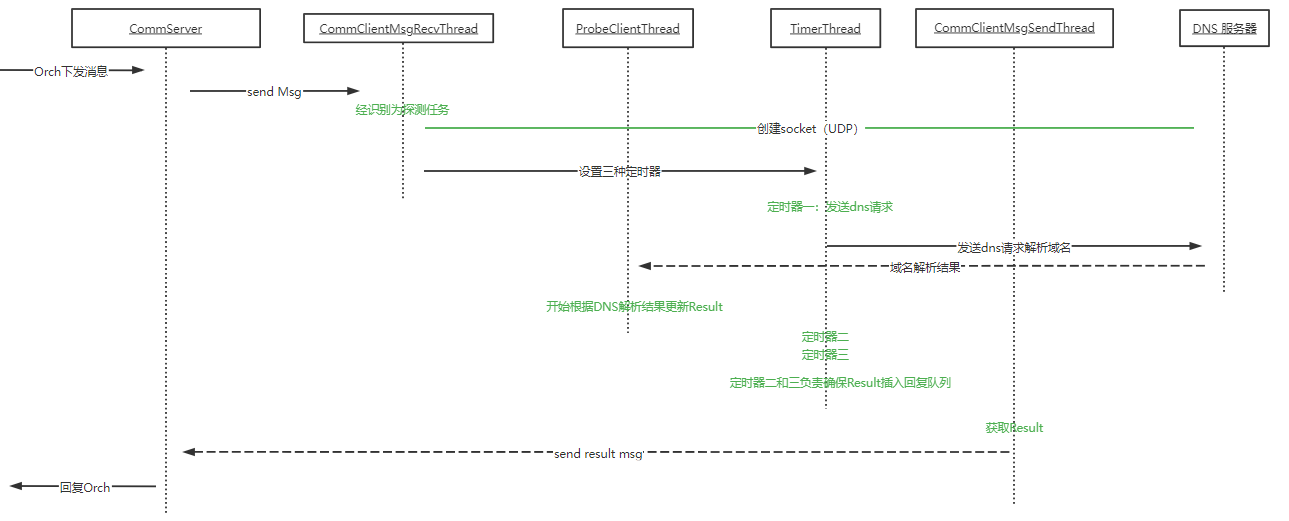


1. 功能整体逻辑简述

* Orch平台向用户提供DNS探测任务的Probe Template
* 用户在Orch端填写需求字段，dns ip、domain及lan tag（lan侧ip）等
* Orch处理后下发到Agent
* Agent则主要负责与DNS服务器交互并处理信息回复Orch
* Orch将收到消息解析为视图展示给用户

1. 功能细节逻辑描述

**功能逻辑时序图：**



**Orch部分：**

**Orch前端界面**

Name： 探测任务名

LAN TAG： LAN TAG

DNS IP DNS server（后续可拓展多个）

Domain 域名（后续可拓展多个）

packet interval(ms) 包间隔时间

total packet 发送的包次数

deadline(s) 任务结束时间

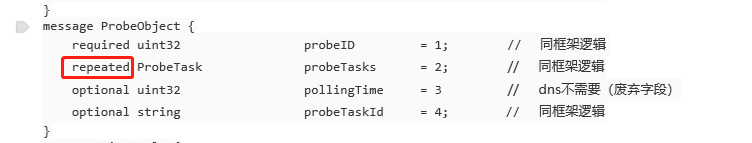
timeout(s) 包失效时间

若为周期性任务，则多出参数：

Interval Time（minte） 任务的间隔时间

**Orch后端功能点：**

* 校验功能，lan tag（src ip）和dns ip需要校验ipv4或ipv6，当用户给到不一致的ip格式，需要报错提醒。
* 每个lan tag只保留一个ip（同框架逻辑，为第一个）
* 当需要拓展时，下发多个dns ip和domain时，需要将ProbeObject根据组合给到相应数量ProbeTask，例如填写4个dns ip，2个domain，则需要下发一个ProbeObject对应8个probeTask，该部分同框架逻辑，如下图。



注：拓展多个dns ip或domain时，区别主要体现在Orch将ProbeObject分解为对应数量的ProbeTask，而最小处理单位ProbeTask不变（即一个dns ip，一个domain，一个src ip）。

注：LAN TAG若不填，会导致返回结果没有srcAddr，Agent部分有详细介绍。

**Proto部分（已同步gitlab）：**

下发字段

message ProbeConfigItem {  
 optional uint32 txPktInterval = 1; // 对应packet interval  
 optional uint32 rxPktInterval = 2; // dns不需要  
 optional uint32 txTotalPkt = 3; // 对应total packet

optional uint32 rxTotalPkt = 4; // dns不需要

optional uint32 rxPktLength = 5; // dns不需要  
 optional uint32 txPktLength = 6; // dns不需要  
 optional uint32 rxBandwidth = 7; // dns不需要  
 optional uint32 txBandwidth = 8; // dns不需要  
 optional uint32 durationTime = 9; // 对应deadline  
 optional uint32 traceTtl = 10; // dns不需要  
 optional string dstDomain = 11; // 对应domain  
 optional uint32 perPktTimeout = 12; // 对应timeout  
 }

message ProbeTask {  
 required uint32 probeProtocol = 1; // LW\_PROBE\_PROTOCOL，为103

required uint32 frequency = 2; // dns不需要

required FiveTuple match = 3; // 五元组部分

required ProbeConfigItem configure = 4; // dns配置部分  
 optional uint32 taskFrequency = 5; // 对应是否周期性任务，0为one-time

optional uint32 vpnId = 6; // 同框架逻辑

}

message ProbeObject {  
 required uint32 probeID = 1; // 同框架逻辑   
 repeated ProbeTask probeTasks = 2; // 同框架逻辑  
 optional uint32 pollingTime = 3; // dns不需要（废弃字段）  
 optional string probeTaskId = 4; // 同框架逻辑  
 }

message FiveTuple {

repeated uint32 ipProto = 1; // dns不需要

repeated IpType srcIps = 2; // 对应lan tag，为一个

repeated IpType dstIps = 3; // 对应DNS IP，为一个

repeated Range srcPorts = 4; // dns不需要

repeated Range dstPorts = 5; // dns不需要

repeated Ipv6Type srcIpv6s = 6; // 对应lan tag（ipv6类型），为一个

repeated Ipv6Type dstIpv6s = 7; // 对应DNS IP （ipv6类型），为一个

}

回复字段

message ResRecord {

required uint32 success = 1; // 0-success; 1-failed;

optional uint32 errorCode = 2; // 1-Format error; 2-Server failure; 3-Name Error; 4-Not Implemented; 5-Refused

repeated NetAddr dstAddr = 3; // resolution result of Domain name

}

message DNSResult {

optional string cpeName = 1; // name of CPE

optional NetAddr srcAddr = 2; // src Ip for CPE

repeated ResRecord resRecords = 3;

optional uint32 latency = 4; // avg latency; us

optional uint32 jitter = 5; // avg jitter; us

optional uint32 pktLoss = 6; // thousandth

}

message ProbeTaskResult {

required uint32 probeProtocol = 1; // LW\_PROBE\_PROTOCOL 为103

required uint32 errorCode = 2; //每一次发包对应结果

optional TransResult transResult = 3; //dns不需要

optional SIPResult sipResult = 4; //dns不需要

required bool isTaskEnd = 5; //是否由失效时间触发，包括最后一次包的timeout和任务的deadline

optional TraceResult traceResult = 6; //dns不需要

optional PingResult pingResult = 7; //dns不需要

optional DNSResult dnsResult = 8; //dns回复部分

}

message ProbeResult {

required uint32 probeID = 1; //同框架逻辑

repeated ProbeTaskResult taskResult = 2; //task部分

optional string probeTaskId = 3; //同框架逻辑

optional uint64 fireTime = 4; //同框架逻辑

}

ResRecord中errorCode解释

当值为 0 时，解析成功

当值为 1 时，表示报文格式错误(Format error)，服务器不能理解请求的报文;

当值为 2 时，表示域名服务器失败(Server failure)，因为服务器的原因导致没办法处理这个请求;

当值为 3 时，表示名字错误(Name Error)，只有对授权域名解析服务器有意义，指出解析的域名不存在

当值为 4 时，表示查询类型不支持(Not Implemented)，即域名服务器不支持查询类型;

当值为 5 时，表示拒绝(Refused)，一般是服务器由于设置的策略拒绝给出应答，如服务器不希望对某些请求者给出应答。

**Agent部分：**

**定时器**

定时器一回调请求函数，目的是发送请求给dns服务器，主要负责封装报文进行发包。

定时器二回调LastPktTimerOut函数，当最后一个包的失效时间到达触发，填充探测质量等信息。

定时器三回调探测任务失效函数，当任务失效时间到达触发，填充探测质量等信息。

定时器四回调TaskStartAgain，负责周期探测。

**Result部分**

当DNS服务器有回复给到Agent时，失效时间内的回复会更新Result（ProbeClientThread负责），随后由CommClientMsgSendThread获取到Result回复Orch，该部分同框架逻辑。

探测质量计算规则

Latency，∑（收包时间戳-发包时间戳）/包数量，即记录每次的延迟并取平均值。

Jitter，∑（两次延迟差绝对值）/(包数量-1)，即记录每次抖动并取平均值。

pktLoss，（丢包数量/应收包数量）‰，记录失效包数量千分比。**与流传输计算存在区别，需注意。**

**DNS交互**

参考

* + 严格按照rfc标准制定报文结构：https://www.ietf.org/rfc/rfc1035.txt
  + DNS理解参考：https://network.51cto.com/article/641655.html
  + 代码Demo：<https://www.binarytides.com/dns-query-code-in-c-with-linux-sockets/>

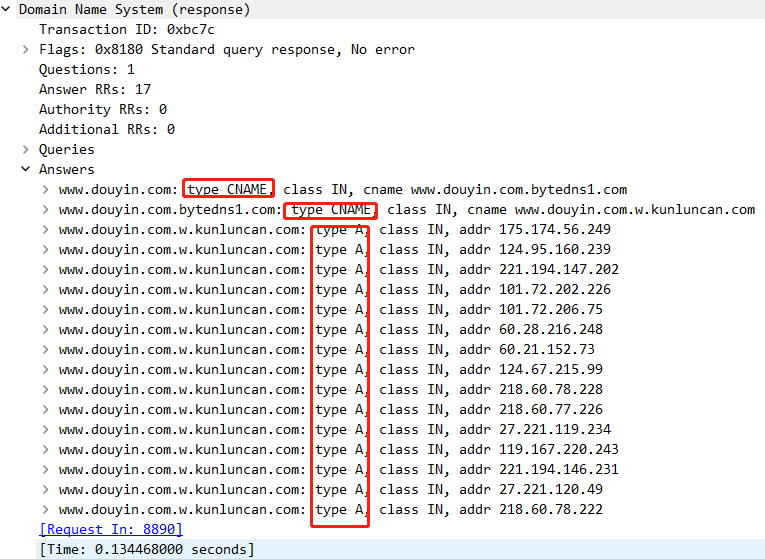
举例

假设有一个发包次数为5，发包间隔时间为1s，每个包失效时间为2s，这次任务结束时间为20s。发送的lan tag中ip为172.168.171.x，domain为www.douyin.com，dns ip为8.8.8.8和114.114.114.114的DNS探测任务。

Orch部分会将任务以ProbeObject下发，并将1（src ip）\*1（domain）\*2（dns ip）种情况，封装到子字段ProbeTask。即最终会有一个包含2个ProbeTask的ProbeObject下发。

Agent收到消息后会遍历其中的ProbeTask队列，按照下发内容执行探测任务，并解析dns服务器答复情况，封装成包含2个ProbeTaskResult的ProbeResult返回给Orch。

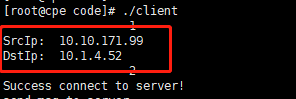
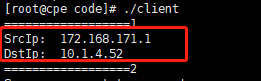
Agent解析时，DNS服务器会给到以下回复结构：



随后Agent会封装成DNSResult结构，最终产生一个包含2个ProbeTaskResult的ProbeResult答复Orch。其中与dns服务器一次交互关键信息（包含15个ip结果等信息）会放在DNSResult中。

**指定src ip**

创建socket时，Agent作为客户端请求DNS服务器，bind能够设置发起端ip，能够保证从固定网口发出，如果不设定，经测试为wan口的IP，而非lan口IP（对应orch不下发lan tag，此时返回时将不返回src ip字段）。左图是客户端不bind，获取socket的发起端ip，右图是设定后的情况。

**IPV4和IPV6**

该部分主要注意协议一致，即src ip和dst ip必须格式一致。下发时由Orch校验ip格式，保证每次执行任务可行。

编码部分主要注意socket参数。

### **会议纪要**

* 该次会议结果主要得出初始方案，确定方案问题。结果存放钉钉#29928群

关于#29928设计方案讨论\_2022.4.20.pdf

关于#29928设计方案讨论\_2022.4.20.mp4

* 经讨论总结三种实现方案，以供讨论执行方案。结果存放钉钉#29928群

#29928三种设计方案\_2022.4.24.pdf

* 该次会议为陈旭介绍probe部分实现原因，及确定#29928方案。记录存放钉钉#29928群

29928dns探测讨论.mp4