YiANDNS 项目 Wiki

DNS Gateway · 易安云网联 DNS 网关

v1.0.3 · 发行版代号 "Marhaba"

本文档依据 v1.0.3 源码与实现整理，覆盖架构、管理页面、加密 DNS、YiANLink 协议、调度数学模型、配置参考、运维监控与开发者指南。

目录

目录 2
1. 项目概述 5
1.1 项目简介与定位 5
1.2 核心特性总览 5
1.3 术语表 6
1.4 文档阅读指南 7
2. 快速开始 8
2.1 系统要求与环境准备 8
2.2 安装部署步骤 8
2.3 首次运行与初始化 8
2.4 最小可用配置示例 9
3. 系统架构 11
3.1 整体架构图 11
3.2 核心模块划分与职责 11
3.3 请求处理流程（端到端链路） 12
3.4 技术栈与依赖 13
4. 管理页面（管理后台） 14
4.1 功能总览 14
4.2 各页面/模块说明 14
4.3 常见操作指引 15
4.4 权限与账户体系 16
4.5 API 接口文档（管理后台 REST API） 17
5. 加密 DNS 协议支持（DoT / DoH / DoQ） 19
5.1 协议简介与对比 19
5.2 各协议的配置方法（端口、证书、监听设置） 19
5.3 TLS/证书管理（证书申请、更新、自签证书说明） 20
5.4 性能表现与调优建议 20
5.5 客户端接入示例 21
6. YiANLink 协议（自研） 22
6.1 协议设计背景与目标 22
6.2 协议规范 22

6.2.1 报文格式与数据结构 22
6.2.2 握手与连接建立流程 22
6.2.3 加密与认证机制 22
6.2.4 传输层承载方式 23
6.3 协议状态机与生命周期 23
6.4 与标准协议（DoT/DoH/DoQ）的协同与互操作 23
6.5 协议版本与兼容性说明 23
6.6 错误码与异常处理 23
6.7 协议实现参考（客户端示例代码） 23
7. 调度系统与数学模型 25
7.1 调度目标与设计思路 25
7.2 数学公式详解 25
7.2.1 公式定义与符号说明（变量表） 25
7.2.2 各公式的物理含义与推导 27
7.2.3 公式应用场景（何时触发、输入输出） 28
7.3 调度决策流程 29
7.4 参数调优指南 29
7.5 调度效果示例（实例演算） 30
8. 配置参考手册 31
8.1 配置文件结构总览 31
8.2 全部配置项详解 31
8.3 典型配置示例 35
9. 运维与监控 37
9.1 日志体系与排查方法 37
9.2 监控指标说明 37
9.3 常见问题排查（FAQ / 故障处理手册） 38
9.4 升级与回滚流程 39
10. 开发者文档 40
10.1 代码仓库结构说明 40
10.2 本地开发环境搭建 40
10.3 构建、测试与发布流程 40
10.4 贡献指南 41
11. 安全说明 42

11.1 安全模型与威胁假设 42
11.2 已知限制与最佳实践 42
附录 44
附录 A. YiANLink 报文格式速查表 44
附录 B. 数学公式推导完整过程 44
附录 C. CLI 命令 / API 速查表 45
附录 D. 版本更新日志（Changelog） 46
附录 E. 参考资料 47

1. 项目概述

1.1 项目简介与定位

YiANDNS（易安云网联 DNS 网关，工程名 **DNS Gateway**）是一个面向家庭与企业局域网的企业级 DNS 服务系统：它接收局域网客户端的 DNS 查询，先经过滤引擎拦截广告、追踪与恶意域名，再经缓存加速，最后转发至可配置的上游 DNS（明文 UDP/TCP 或加密 DoT/DoH/DoQ，以及自研 YiANLink 私有协议），并以 Web 控制台提供全程可视化管理。

它解决的核心问题：

- **统一管控**：为整个局域网提供单一 DNS 出口，集中实施广告拦截、域名黑白名单、重写与按客户端分组的差异化策略；
- **加密出站**：在明文 DNS 易被劫持、窥探的环境下，提供 DoT/DoH/DoQ 加密上游与加密服务端能力；
- **解析质量与容灾**：多上游智能调度（负载均衡 / 趋势预测 / 并行请求）+ 熔断健康检查，自动规避劣化节点。

工程形态为单个可执行 fat jar，基于 Java 21（虚拟线程），100% 纯 Java、零 native 依赖，一次打包可在任意操作系统（含 Alpine/musl）与任意架构（x86/ARM64，如树莓派）直接运行。当前版本 **v1.0.3**，发行版代号 **"Marhaba"**（覆盖 1.0.0 → 1.2.20 区间）。

1.2 核心特性总览

特性域	能力概览
管理页面	Vue 3 + Element Plus + ECharts 控制台：仪表盘、查询日志、管控规则、系统设置；PBKDF2 口令 + Bearer 会话鉴权
多协议支持	明文 UDP/TCP（默认 53）；DoT（RFC 7858，853）；DoH（RFC 8484，443，HTTP/2 与 HTTP/1.1）；DoQ（RFC 9250，853，纯 Java QUIC）；自研 YiANLink（TCP 6833）
智能调度	上游查询方式三选项：负载均衡（轮询）、趋势预测（代价能量函数 + Softmax + SWRR 动态权重）、并行请求（并发全部取最快）；Split DNS 域名分流；熔断健康检查与自动降级
过滤引擎	AdGuard 风格规则（hosts 行、Adblock 双斜线拦截、@@ 例外、$important/$client=/$dnstype= 修饰符、正则）、域名后缀树索引、按客户端 IP/CIDR 分组档案、远程规则源自动拉取与原子热替换
缓存	默认「频率感知动态 TTL」引擎（按域名访问热度浮动 TTL、负缓存、统计）；JVM 参数 `-Ddns.cache.engine=caffeine` 可回退 Caffeine W-TinyLFU
查询日志与统计	JSONL 按天轮转落盘 + 内存环形缓冲；24h/7d 统计

	窗口、TopN 域名、拦截率
容量自保护	预测性容量控制：基于泰勒展开余项的虚拟线程并发闸门，洪峰下快速拒绝防雪崩
可观测性	内存指标注册表（计数器/延迟）、按协议计数、活跃连接数、JVM/系统状态采样（纯 JDK API）
插件扩展	ServiceLoader SPI：classpath 内嵌 + plugins/ 目录外置；请求/响应过滤器、上游/缓存/配置/指标 Provider 扩展点
跨平台	零 native：DoT/DoH 走 JDK SSLContext + Netty，DoQ 走纯 Java QUIC（Kwik），无 BoringSSL/openssl

1.3 术语表

术语	定义
DoT	DNS over TLS（RFC 7858）：DNS 报文经 TLS 承载于 TCP 853 端口，报文前加 2 字节长度前缀
DoH	DNS over HTTPS（RFC 8484）：DNS 报文作为 HTTP 请求体/参数经 HTTPS 传输（application/dns-message）
DoQ	DNS over QUIC（RFC 9250）：DNS 报文经 QUIC 流传输，ALPN 标识 doq，天然多路复用
YiANLink	本项目自研的私有应用层加密传输协议：AES-256-GCM + PSK 预共享密钥 + TCP 长连接，仅搬运加密 DNS 报文
PSK	Pre-Shared Key 预共享密钥：YiANLink 的 256 位对称密钥，Base64 编码后配置
频率感知缓存	本项目默认缓存引擎（FrequencyAwareCache）：按域名访问热度（指数衰减频次）动态浮动缓存 TTL，热点更久、冷域名更快过期
SWRR	Smooth Weighted Round Robin 平滑加权轮询（Nginx 同款算法）：按权重比例平滑选路，避免高权重节点扎堆
代价能量 J	趋势预测调度的核心函数：综合 RTT 水平、恶化趋势、加速度抖动与离散度的标量代价，J 越小节点越优
Softmax	将各节点代价映射为归一化权重的概率化方法：$W_i = \exp(-0.5 \cdot J_i) / \Sigma \exp(-0.5 \cdot J_k)$
EMA	指数移动平均，用于平滑 RTT 等时序指标
ALPN	应用层协议协商：TLS 握手阶段协商 HTTP/2（h2）、doq 等应用协议
SNI	服务器名称指示：TLS ClientHello 中携带的目标域名

NXDOMAIN	权威应答"域名不存在"（rcode 3）；负缓存即对该结果短期缓存
负缓存	对 NXDOMAIN 结果按 negativeTtlSeconds 缓存，减轻对已删域名重复查询的放大
W-TinyLFU	Caffeine 的准入+淘汰混合算法，以 Count-Min Sketch 频率估计实现高命中率容量淘汰；-`Ddns.cache.engine=caffeine` 回退引擎采用
Split DNS	域名分流：命中的域名（精确/通配符）绕过智能调度，直连指定专属上游
熔断	上游节点连续失败达阈值后进入 open 状态被禁用，cooldown 后 half-open 试探恢复
半开连接	一端异常消失（未发 FIN/RST）而服务端仍持有的 TCP 连接；YiANLink 依赖 60s 读超时回收
fat jar	maven-shade-plugin 产出的单文件可执行 jar，内含全部模块与第三方依赖
reactor	Maven 多模块构建中的一个构建单元；本项目 12 reactor = 1 父 POM + 11 代码模块
预测闸门	预测性容量控制组件：虚拟线程 + Semaphore 动态许可，基于泰勒预测准入/拒绝查询

1.4 文档阅读指南

读者	推荐阅读路径	目标
使用者（家庭/小团队）	第 2 章 → 4.1–4.3 → 5.5	部署上线、会用控制台、客户端接入
运维工程师	第 8 章 → 第 9 章 → 5.2–5.4 → 11.2	配置调优、证书管理、监控排障、安全加固
后端开发者	第 3 章 → 第 7 章 → 第 10 章 → 附录 A/B	理解架构与数学模型、参与开发
安全审计	第 11 章 → 5.3 → 6.2/6.6 → 4.4	威胁模型、密码学使用、已知限制
协议对接方（自研客户端）	第 6 章 → 附录 A → 7 章	实现 YiANLink 客户端、理解调度行为

---

2. 快速开始

2.1 系统要求与环境准备

项目	要求
运行时	JDK 21 及以上（OpenJDK / IBM Semeru / 华为毕昇均可；构建以 `--release 21` 编译）
内存	≥ 512MB（建议 1GB；默认缓存 10 万条目约数十 MB 堆）
磁盘	≥ 100MB（含查询日志按天轮转的空间，按保留时长评估）
操作系统	任意（Windows / Linux / macOS，含 Alpine/musl 容器；x86 与 ARM64）
网络	数据面 UDP+TCP 53（可改）；管理面 TCP 8080（建议仅内网/本机）；可选 853/443/6833
权限	Linux 监听 53 需要 root 或 `setcap 'cap_net_bind_service=+ep' <jdk>/bin/java`；Windows 需放行防火墙
可选	证书文件（启用 DoT/DoH/DoQ 时必需）；YiANLink PSK（启用 YiANLink 时必需）

2.2 安装部署步骤

路径 A：下载发行版（推荐）

1. 前往项目托管平台（Gitee）的 Releases 页面，下载 `dns-proxy-1.0.3.jar`；

2. 放置到任意部署目录（如 `/opt/yiandns/`），该目录将成为数据目录基准（`data/`、`TLS/` 相对解析）。

> *仓库 `.gitignore` 排除了 `target/` 与 `*.jar`，源码不含编译产物：`git clone` 后必须自行构建。*

路径 B：从源码构建

前置条件：JDK 21+（配置 `JAVA_HOME`）、Maven 3.6+（全局 `mvn`）。

```
# Windows（项目根目录下）
powershell -File "build.ps1" clean package
# 产物：dns-proxy\target\dns-proxy-1.0.3.jar
# Linux / macOS
mvn -s maven-settings.xml -B clean package
```

构建完成后建议执行端到端冒烟（随机端口拉起 UDP+TCP，9 项断言）：

```
powershell -File "smoke.ps1"
```

2.3 首次运行与初始化

```
# 使用示例配置启动（DNS 监听 0.0.0.0:53，Web 管理 127.0.0.1:8080）
java -jar "dns-proxy-1.0.3.jar" "config\config.example.yaml"
```

- 配置文件解析顺序：命令行参数路径 → 当前目录 `config.yaml` / `config.yml` / `config.json` → 均不存在时使用内置默认配置；
- 配置分层加载：内置默认值 → 用户配置文件深合并 → 运行期覆盖层（Web 保存路径），未写出的字段回退默认值；
- 启动横幅会打印版本（如 `DNS 服务启动完成 (v1.0.3)`）、各协议监听地址、缓存引擎（默认显示 `启用[频率感知]`）与预测闸门参数。

首次访问 Web 控制台：浏览器打开 `http://127.0.0.1:8080`，默认账号口令 `admin` / `admin`。

> ⚠ ***安全警告：****默认凭据为公开知识。将服务暴露到公网或非受信网络之前，务必修改 `web.adminPassword`，并将 `web.bindAddress` 限制为可信网段。*

验证 DNS 服务：

```
nslookup baidu.com 127.0.0.1
```

优雅关闭：前台运行按 `Ctrl+C` 触发 shutdown hook，依次停止：Web 管理 → DoQ/DoH/DoT → UDP/TCP 监听 → 排空在途请求（默认 500ms）→ 插件 → 上游连接池 → 查询日志缓冲落盘。

退出码：

退出码	常量	含义
0	EXIT_OK	正常退出
1	EXIT_STARTUP_FAILURE	启动失败（上游全无效、证书缺失、未预期异常等）
2	EXIT_CONFIG_ERROR	配置校验失败（控制台逐条打印字段路径与原因）
3	EXIT_PORT_IN_USE	DNS/Web 监听端口被占用

容错策略：过滤规则文件缺失仅告警跳过；单个插件启动失败告警后继续；查询日志目录不可写自动降级内存模式；上游运行期不可达由熔断 + failover 兜底，不阻塞启动。

（可选）Linux systemd 服务示例：

```
# /etc/systemd/system/yiandns.service
[Unit]
Description=YiANDNS DNS Gateway
After=network.target

[Service]
ExecStart=/usr/bin/java -jar /opt/yiandns/dns-proxy-1.0.3.jar /opt/yiandns/config.yaml
WorkingDirectory=/opt/yiandns
SuccessExitStatus=143
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

2.4 最小可用配置示例

```
server:
  bindAddress: 0.0.0.0
  udpPort: 53
  tcpPort: 53
upstreams:
  - address: 223.5.5.5
    protocol: UDP
upstream:
  strategy: ROUND_ROBIN
```

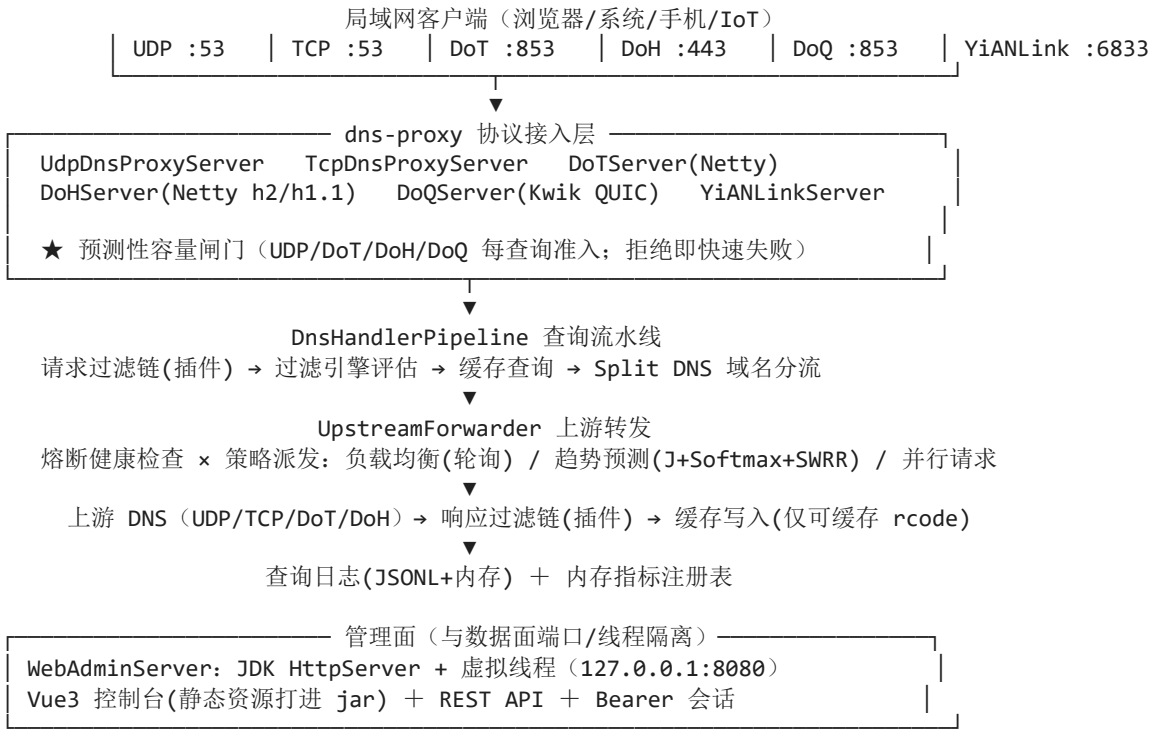
```
  cache:
    enabled: true
  web:
    enabled: true
    bindAddress: 127.0.0.1
    port: 8080
    adminUsername: admin
    adminPassword: "改成强口令"
```

将以上内容保存为 `config.yaml` 放在 jar 同目录，直接 `java -jar dns-proxy-1.0.3.jar` 即可运行。

---

3. 系统架构

3.1 整体架构图



数据流向要点：管理面故障不影响查询路径；DNS 数据面所有"每查询创建虚拟线程"的入口统一经预测闸门准入；TCP 与 YiANLink 为"每连接一线程 + 连接内串行查询"模型，不进入按查询计数的闸门（连接级治理独立）。

3.2 核心模块划分与职责

Maven reactor 共 12 个构建项目 = 1 个父 POM（依赖管理与聚合）+ 11 个代码模块：

模块	职责
dns-server-api	DNS 消息/交换/编解码接口层（DnsRequest/DnsResponse/DnsExchange/DnsCodec 契约）
dns-core	dnsjava 编解码落地（DnsjavaCodec）、应答工厂（NOERROR/NXDOMAIN/REFUSED/CNAME/A/AAAA）
plugin-api	插件扩展点：DnsPlugin、请求/响应过滤器（sealed 决策）、上游/缓存/配置/指标 Provider
config	YAML/JSON 分层配置加载（ConfigNode 深合并）、字段级校验（ConfigError 聚合）、热更新监听、UpstreamPolicy
logging	查询日志：JSONL 按天轮转 / 内存环形缓冲、异步刷盘、24h 桶统计聚合
metrics	内存指标注册表（计数器/延迟/仪表、QPS tick）

filter-engine	AdGuard 风格规则解析、域名后缀树、黑名单/白名单/重写匹配、按客户端分组引擎（UserAwareFilterEngine）
upstream-resolver	UDP/TCP/DoT/DoH 上游传输、三策略选择器、趋势预测调度包（scheduler：代价能量+SWRR）、熔断健康检查、域名分流转发、上游地址解析（UpstreamAddressParser）
cache	频率感知动态 TTL 缓存（默认引擎：访问频次浮动 TTL、负缓存、统计）；-`Ddns.cache.engine=caffeine` 回退 Caffeine W-TinyLFU
dns-proxy	UDP/TCP/DoT/DoH/DoQ/YiANLink 服务器、查询流水线、预测闸门（concurrent 包）、启动入口 DnsGatewayApplication
web-admin	Web 管理 REST API（JDK HttpServer + 虚拟线程、PBKDF2 口令、Bearer 会话、Vue3 控制台静态资源）

3.3 请求处理流程（端到端链路）

启动装配（12 步，失败自动逆序回滚）：

1. 配置加载与强校验（DnsConfigManager.load，校验失败 exit 2）

2. 内存指标注册表

3. 查询日志服务（文件失败降级内存 + WARN）

4. 插件装载（classpath + plugins/，单个失败告警跳过）

5. 动态规则服务 RuleService（本地 TSV + 远程规则源）

6. 过滤引擎（UserAwareFilterEngine + 规则变更热重建）

7. 缓存（默认频率感知引擎；`caffeine` 显式回退；enabled=false 时禁用）

8. 上游转发器（候选构建 + 策略选择器 + 趋势预测调度器按需启动）

9. 查询流水线 DnsHandlerPipeline（过滤→缓存→上游 编排）

10. UDP/TCP 监听启动

11. 安全传输 DoT/DoH/DoQ（证书加载一次共享 + hostname 校验）与 YiANLink 监听；预测闸门全局单例

12. Web 管理面启动

查询链路（以 UDP 为例）：

1. 接收线程收到数据报，解析 DNS wire；

2. 预测闸门 `executeTask：tryAcquire` 失败 → 不创建线程，直接在 accept 线程内回 SERVFAIL（DoH 回 HTTP 200 + DNS SERVFAIL；DoQ 按 RFC 9250 §4.3 重置流）；

3. 虚拟线程内：解码 → 请求过滤链（插件）→ 过滤引擎评估（顺序：important 黑名单 → 白名单/例外 → 重写 → 普通黑名单 → 放行）；

4. 缓存查询（键 = 规范化域名|qtype）：命中重写 requestId 并标记 fromCache 直接返回；

5. 未命中 → 域名分流匹配：命中则直连专属上游（独立超时 2000ms，超时 SERVFAIL，熔断自动降级回默认引擎）；未命中走上游策略派发；

6. UpstreamForwarder 按策略（轮询/SWRR 权重序/并发全部）逐候选尝试；SERVFAIL/REFUSED/FORMERR/NOTIMP 视为可重试切换下一候选，NXDOMAIN/NOERROR 为终态；传输异常记入熔断计数；

7. 响应过滤链（插件）→ 可缓存 rcode（仅 NOERROR/NXDOMAIN）按 minTtl 写入缓存；CNAME 重写应答在返回 CNAME 的同时解析目标域名并附带其 A/AAAA 记录（解析失败降级为只回 CNAME）；

8. 查询日志（异步）+ 指标（dns.query.*、dns.protocol{...}、延迟）→ 回包（UDP 原路回写；TCP/DoT 按帧回写；DoH 200 application/dns-message；DoQ 同流回写后 FIN）。

3.4 技术栈与依赖

依赖	版本	用途
Java	21（虚拟线程、record、sealed）	运行时与语言基线
Maven	3.6+（maven-shade-plugin 3.6.0）	构建与 fat jar
dnsjava	3.6.1	DNS 编解码与记录模型
Netty	4.1.115.Final	DoT（NIO+TLS）与 DoH（HTTP/2 + HTTP/1.1）服务端传输；TLS 走 JDK SSLContext，无 tcnative
Kwik	0.8.6（包名 net.luminis.quic）	纯 Java QUIC 实现，承载 DoQ
Caffeine	3.1.8	回退缓存引擎（-Ddns.cache.engine=caffeine；W-TinyLFU）
SnakeYAML	2.3（SafeConstructor）	YAML 配置解析
Jackson	2.18.2	JSON 配置解析与 REST JSON 序列化
BouncyCastle	bcpkix-jdk18on 1.78.1	PEM 私钥解析（PKCS#8/PKCS#1/SEC1）
JUnit	5.11.3	单元测试
Spring Boot	无	项目不含 Spring，Web 层为 JDK HttpServer + 虚拟线程

---

4. 管理页面（管理后台）

4.1 功能总览

控制台入口 `http://<web.bindAddress>:<web.port>/index.html`（默认 `http://127.0.0.1:8080`）。顶部导航：

页面	功能
仪表盘（Dashboard）	当前 QPS / 累计查询 / 拦截次数 / 缓存命中率四卡片 + 请求域名 Top5、拦截域名 Top5、协议占比、上游次数与时延四图表（ECharts）
查询日志	最近查询记录（域名/类型/结果/来源协议/客户端/上游/耗时/命中规则）、详情、统计与 TopN、清空日志与统计
管控规则	黑名单 / 白名单 / 域名分流 / 重写 / 自定义 五类规则独立编辑、保存即时热生效；远程规则源管理（自动/手动更新）
设置	基本设置（系统状态、数据存储、系统操作、日志设置）、DNS 服务（上游设置、缓存设置、上游查询方式）、DNS 协议（UDP/TCP、加密 DNS、YiANLink）、上游服务器管理
登录/账户	单管理员登录、退出登录、401 自动登出

所有管理 API（除登录与静态页外）均需 `Authorization: Bearer <token>`；前端 401 时自动清除本地 token 并退回登录页。

4.2 各页面/模块说明

4.2.1 登录页：账号 + 口令登录（`POST /api/auth/login`，匿名）。成功后 token 存于浏览器 localStorage（键 `dns_admin_token`，有效期 8 小时）。

4.2.2 仪表盘：四张统计卡（大数字 + 迷你趋势线）与四张图表，数据来自 `GET /api/metrics`、`GET /api/stats?topN=5`、`GET /api/status`，定时轮询刷新；后端无数据时展示真实空态（绝不使用模拟数据）。

4.2.3 查询日志：来自内存环形缓冲（上限 logging.maxEntries）；支持按条数拉取最近记录、单条详情、统计窗口（statsRetentionHours）内的 TopN 域名；提供"清空查询日志 / 清空统计"维护操作（`/api/maintenance/*`）。

4.2.4 管控规则页（核心）：

- 五个规则页签各自独立保存、保存后即时热重建过滤引擎（原子热替换，查询无锁）：
  - **黑名单**：拦截域名（Adblock 双斜线拦截、hosts 0.0.0.0 行、纯域名、`*.`通配 等）；
  - **白名单**：`@@` 例外规则，优先于黑名单；
  - **域名分流**：精确/通配符域名 → 指定专属上游（Split DNS，详见 8.2 domainRouting 段）；保存时校验禁止指回本网关自身监听地址（防回环）；

  - **重写**：源域名 → 目标 IP/IPv6/CNAME（CNAME 自动环路检测）；
  - **CNAME 重写联动解析**：应答在返回 CNAME 的同时解析目标域名并附带其 A/AAAA 记录（目标解析失败时仅返回 CNAME），客户端无需二次查询即可获得 IP；
  - **自定义**：混合组，按行自动识别（@@ → 白名单、带目标 → 重写、其余 → 黑名单）。
- **远程规则源**：每类列表可挂 HTTP(S) 订阅源（名称/URL/规则数/上次更新 四列表格，每页 10 条分页）；按间隔并发拉取，自动识别 Adblock/hosts/纯域名/混合格式；支持手动"更新列表"与关闭自动更新（intervalSeconds=0）；列表内不放本地输入框，本地规则在页签输入区维护。

4.2.5 设置页：

- **基本设置**：
  - 系统状态（3 秒局部刷新）：运行时间、线程数、Load1/5/15（Windows 取不到显示 N/A）、堆/栈/直接内存、YoungGC/OldGC 次数与耗时、最近 GC 停顿、当前系统（os.name/version/arch）与 Java 版本三行（version/vendor/runtime/vm）；全部指标仅用 JDK 自带 API 采集；
  - 数据存储：`storage.dataDir`（查询日志等数据落盘基准，相对路径以 jar 目录解析）；
  - 系统操作：重启服务 / 关闭服务（异步 202，约 300ms 后执行；重启完整重装配且登录会话失效）；
  - 日志设置：保留时长（小时）、统计窗口、存储类型（JSONL/MEMORY）、级别。
- **DNS 服务**：
  - 上游设置：上游列表增删改（address 支持 host / host:port / 完整 URL + protocol UDP/TCP/DOT/DOH）、**上游查询方式**下拉（1 负载均衡 / 2 趋势预测 / 3 并行请求，热更新）、状态重置开关 resetStateOnUpdate 与"重置上游状态"按钮；
  - 缓存设置：开关、maxEntries、defaultTtlSeconds、negativeTtlSeconds、minTtlSeconds、maxTtlSeconds。
- **DNS 协议**：
  - UDP/TCP 开关与端口；
  - 加密 DNS（DoT/DoH/DoQ）：各协议开关与端口、证书路径（certFile/keyFile）、对外域名 hostname、最低 TLS 版本（DoH）、legacyRsaCompat 兼容开关；
  - YiANLink：开关、端口（默认 6833）、PSK（支持随机生成、Base64 32 字节校验）；保存写入顶层 `yianlink` 段并提示重启生效。
- **上游服务器**：上游节点管理与连接状态展示。

4.3 常见操作指引

界面截图请结合实际部署版本对照补充；以下步骤以默认布局为准。

操作一：修改上游并切换查询方式（热生效，无需重启）

1. 进入 设置 → DNS 服务 → 上游设置；

2. 增/删/改上游条目（地址支持 host / host:port / 完整 URL，如 `https://doh.pub/dns-query` 或 `tls://dot.pub`；协议 UDP/TCP/DOT/DOH，地址前缀须与协议一致）；

3. "上游查询方式"选择 负载均衡 / 趋势预测 / 并行请求 之一；

4. 点击保存：配置整体原子校验 → 通过后写回启动配置文件 → 运行期立即切换。失败时整批不生效并提示校验错误。

操作二：添加拦截规则

1. 进入 管控规则 → 黑名单；

2. 输入区粘贴规则（如 Adblock 双斜线拦截 `||ads.example.com^`、hosts 行 `0.0.0.0 tracker.net`、通配 `*.bad.io`）；

3. 保存后立即生效，仪表盘"拦截次数"随之增长；命中规则显示在查询日志的 matchedRule 字段。

操作三：部署证书并启用 DoH

1. 将 PEM 全链证书与私钥放置于运行目录 `TLS/fullchain.pem`、`TLS/privkey.pem`；

2. 设置 → DNS 协议 → 加密 DNS：勾选 DoH，端口 443，路径 `/dns-query`；填写 server.hostname（如 `dns.example.com`，启动时校验证书 SAN/CN 覆盖该域名）；

3. 保存并重启服务；横幅会输出 `DoH → https://dns.example.com/dns-query` 客户端提示；

4. 客户端验证：浏览器/Firefox DoH 填入上述 URL，或 `kdig +https dns.example.com example.com`。

操作四：开启 YiANLink 私有通道

1. 设置 → DNS 协议 → YiANLink：打开开关，端口保持 6833；

2. PSK 点击"随机生成"（或粘贴 256 位 Base64 密钥），妥善下发给受控客户端；

3. 保存（提示重启生效）→ 重启服务 → 横幅出现 YiANLink 监听行；

4. 客户端按第 6 章协议实现接入；查询日志"来源协议"显示 YiANLink。

操作五：配置域名分流（Split DNS）

1. 管控规则 → 域名分流；

2. 添加条目：域名模式（`www.a.com` 精确 或 `*.b.com` 通配，通配仅匹配子域不含自身）→ 专属上游（如 `udp://119.29.29.29:5353`）→ 可选超时（默认 2000ms）；

3. 保存热生效；命中的查询绕过智能调度直连专属上游，上游不可达时自动降级回默认引擎（熔断保护）。

操作六：修改管理员口令

1. 编辑配置文件 `web.adminPassword`（或通过 设置 保存）→ 重启服务；旧会话全部失效，用新口令重新登录。

4.4 权限与账户体系

项	说明
账户模型	单管理员（web.adminUsername / web.adminPassword），无多用户与 RBAC

口令存储	PBKDF2WithHmacSHA256，12 万迭代 + 随机盐，存储格式 `pbkdf2$iter$saltB64$hashB64`
校验	MessageDigest.isEqual 常数时间比较，抗时序侧信道
会话	登录返回 32 字节 SecureRandom token，有效期 8 小时；logout 即注销
传输鉴权	除登录与静态页外所有 REST API 要求 `Authorization: Bearer <token>`；401 自动登出
网络隔离	web.bindAddress 默认 127.0.0.1，与 DNS 数据面端口分离；管理面线程池独立，故障不影响查询
敏感信息	`GET /api/config` 对敏感字段脱敏返回

4.5 API 接口文档（管理后台 REST API）

除 `POST /api/auth/login` 与静态页外，均需 Bearer 鉴权。

方法	路径	说明
POST	/api/auth/login	匿名登录，返回 token（8 小时有效）
POST	/api/auth/logout	注销当前会话
GET	/api/auth/me	当前会话信息
GET	/api/status	DNS/Web 端口与运行状态、缓存容量等
GET	/api/connections	当前活跃连接列表（TCP/DoT/DoH/YiANLink）
GET	/api/queries?limit=N	最近查询日志
GET	/api/queries/detail	查询日志详情
GET	/api/stats?topN=N	查询统计与 Top 域名
GET	/api/rules?group=	读取某列表规则原文、计数与上游源状态（group=blocklist/allowlist/rewrite/custom/domainRouting）
POST	/api/rules	追加规则文本（body: content + group），持久化至 data/filters/user-rules.tsv 并异步热重建
PUT	/api/rules	用文本整体替换某个规则组
GET	/api/rules/sources	查看 HTTP(S) 远程规则源及下载状态
POST	/api/rules/sources	添加远程规则源（name + url + group + intervalSeconds + enabled）
PUT	/api/rules/sources	按列表整体同步某组的上游源（intervalSeconds=0 表示关闭自动更新）
DELETE	/api/rules/sources?id=	删除远程规则源并清除其缓存
POST	/api/rules/sources/refresh[?id=或?group=]	立即刷新一个/某组/全部远程规则源
GET	/api/config	配置树（敏感字段脱敏）
PUT	/api/config	增量 patch 深合并更新；整体原子校验，通过后写回启动配置文件，重启保留
GET	/api/metrics	指标计数器与平均延迟

GET	/api/system/info	系统状态采样（运行时间/负载/内存/GC/线程/OS/Java）
POST	/api/system/restart	异步重启 DNS/Web 服务（重读配置并重新绑定端口）
POST	/api/system/shutdown	异步关闭服务并退出进程
POST	/api/maintenance/clear-query-log	清空查询日志
POST	/api/maintenance/clear-stats	清空统计
POST	/api/upstream/reset	手动重置上游熔断与 RTT 状态（配合 upstream.resetStateOnUpdate）
GET	/api/plugins	已加载插件列表
GET	/index.html	管理页（匿名，SPA 静态资源 /assets/* 同为匿名）

配置保存语义：`PUT /api/config` 先做整体校验，通过后按扩展名（YAML/JSON）**原子写回启动时使用的配置文件**，再切换运行期配置；写盘失败则本次保存整体不生效（不会出现"重启后设置丢失"）。多配置文件合并启动时写回第一个源文件。

---

5. 加密 DNS 协议支持（DoT / DoH / DoQ）

5.1 协议简介与对比

维度	DoT	DoH	DoQ
规范	RFC 7858	RFC 8484	RFC 9250
传输	TLS over TCP	HTTPS（本项目 h2 与 HTTP/1.1 双支持）	QUIC（ALPN doq，TLS 1.3 内建）
默认端口	853	443	853
报文封装	2 字节大端长度前缀	HTTP 请求体（POST）或 ?dns=base64url（GET）	每 QUIC 双向流一查询，2 字节长度前缀
连接模型	长连接多查询	keep-alive / h2 多路复用	流级并发，无队头阻塞
典型场景	系统级 DNS（Android 私有 DNS、systemd-resolved）	浏览器（Firefox/Chrome）、应用内嵌	追求低时延与并发的客户端
本项目实现	Netty NIO + JDK SSLContext；**固定纯 TLS 1.2 且彻底禁用 ALPN**；30s 空闲关闭	Netty HTTP/2（Http2FrameCodec/MultiplexHandler）+ HTTP/1.1 兜底管线；ALPN 白名单不匹配明确拒绝	Kwik 纯 Java QUIC；ServerConnector + doq 协议工厂；可与 DoT 同端口共存（853）
证书	TLS/fullchain.pem + privkey.pem（三协议共享）	同左	同左；**私钥仅支持 RSA（PKCS#8）**

补充：DoT/DoQ 端口校验规则——dot/doh 与 tcpPort/udpPort 重叠报错、doq 与 udpPort 重叠报错、dot 与 doh 同端口报错（**dot+doq 同 853 合法**，RFC 默认共享）、doh.path 须匹配 `^/[a-zA-Z0-9_\-/]*$`。

5.2 各协议的配置方法（端口、证书、监听设置）

```
server:
  hostname: dns.example.com          # 可选：对外服务域名，启动时校验证书覆盖并输出客户端提示
  tls:                               # 三协议共享证书（任一协议启用时必须存在）
    certFile: TLS/fullchain.pem      # PEM 全链证书
    keyFile: TLS/privkey.pem         # PEM 私钥（PKCS#8 或 PKCS#1；DoQ 仅 RSA）
    minVersion: TLSv1.2              # 仅影响 DoH（TLSv1.2/TLSv1.3）；DoT 固定 TLS1.2；DoQ 固定 TLS1.3
    legacyRsaCompat: false           # true 时额外下发 TLS_RSA_*（兼容老旧客户端；SSLv3/TLSv1/1.1 始终封杀）
  dot:
    enabled: true
    port: 853
  doh:
    enabled: true
    port: 443
    path: /dns-query
  doq:
    enabled: true
```

```
  port: 853                          # 可与 dot 同端口共享
```

- 启用任一协议时证书路径必填；证书加载失败 / 私钥与证书不配对 / hostname 未被覆盖 → 启动失败并给出明确原因（列出证书实际覆盖的域名）；
- 启动横幅输出各协议监听与客户端连接提示（如 `DoT → dns.example.com:853 (DOT)`）；
- 配置可在 设置 → DNS 协议 页修改，保存写回配置文件；协议端口变更涉及重新绑定监听，按界面提示重启生效。

5.3 TLS/证书管理（证书申请、更新、自签证书说明）

证书要求：

- PEM 全链证书（服务器证书 + 中间链按序拼接）+ PEM 私钥；
- 私钥兼容 PKCS#8 / PKCS#1 / SEC1（BouncyCastle 解析）；**DoQ（Kwik）仅支持 RSA 私钥且必须为 PKCS#8**；
- 加载时执行"随机挑战签名 + 证书公钥验签"配对校验，不匹配直接拒绝启动。

获取正规证书（Let's Encrypt 示例）：

```
certbot certonly --standalone -d dns.example.com
# 产物：/etc/letsencrypt/live/dns.example.com/fullchain.pem 与 privkey.pem
# 复制到运行目录 TLS/ 下（或直接在配置中写绝对路径）
```

自签证书（内网场景）：

```
openssl req -x509 -newkey rsa:2048 -nodes -days 3650 \
  -keyout TLS/privkey.pem -out TLS/fullchain.pem \
  -subj "/CN=dns.lan" \
  -addext "subjectAltName=DNS:dns.lan,IP:192.168.1.2"
```

自签证书的客户端须显式信任（导入系统/浏览器信任库，或工具加跳过校验参数）。**注意 SAN 必须包含客户端实际访问的主机名或 IP**，否则主流客户端（JDK HttpClient、浏览器）会因主机名校验失败而拒绝。

更新流程：替换 `TLS/` 下文件 → 重启服务（当前版本启动时加载一次证书）。`legacyRsaCompat` 开启需重启完全生效，关闭立即生效。

hostname 校验：配置 `server.hostname` 后，启动时校验证书 SAN dNSName（支持单级通配 `*.example.com`）覆盖该域名，无 dNSName 时回退 CN；不覆盖则启动失败并列出证书实际域名。横幅同时输出 DoH/DoT/DoQ 的客户端连接提示。

5.4 性能表现与调优建议

基准压测（树莓派 4B / ARM64 / 华为毕昇 JDK 21，1200 并发公网真实压测）：

协议	QPS	成功率
UDP	6599+	95.6%
TCP	3974+	99.2%

压测期间内存与 GC 平稳，无 OOM、停顿极低；同等环境下连接稳定性优于主流同类开源项目。

调优建议：

- 加密协议引入 TLS 握手成本：客户端侧启用会话复用/长连接；服务端已收紧会话缓存（安全加固），不建议再放宽；

- DoT 固定 TLS1.2：老设备兼容优先的取舍；如需 TLS1.3 特性请走 DoH（minVersion=TLSv1.3）或 DoQ；
- 高并发场景关注预测闸门参数（见 7.4）与 `server.threads`；DoH 客户端优先使用 h2 多路复用减少连接数；
- 443 端口与其他 Web 服务共存时建议独立 IP/端口，避免 ALPN/证书冲突；
- 上游侧：加密上游（DOT/DOH）建议与明文上游混配，趋势预测会依据 RTT 自动倾斜。

5.5 客户端接入示例

客户端	接入方式
Windows nslookup（明文）	`nslookup baidu.com 127.0.0.1`
Android（DoT）	设置 → 网络 → 私有 DNS → 填 `dns.example.com`（须证书信任 + SAN 匹配）
systemd-resolved（DoT）	`/etc/systemd/resolved.conf`：`DNS=192.168.1.2` + `DNSOverTLS=yes`
Firefox（DoH）	设置 → 隐私与安全 → DNS over HTTPS → 自定义 URL `https://dns.example.com/dns-query`
Chrome（DoH）	设置 → 安全 → 使用安全 DNS → 自定义提供商填同一 URL
kdig（DoT/DoH/DoQ）	`kdig +tls @dns.example.com -p 853 example.com`；`kdig +https @dns.example.com example.com`；`kdig +quic @dns.example.com -p 853 example.com`
curl（DoH）	`curl --doh-url https://dns.example.com/dns-query https://example.com`
YiANLink 客户端	按第 6 章协议规范自行实现（PSK + TCP 6833）

DoT 客户端失败而 DoQ 正常：多为证书信任或地址不在 SAN（DoQ 工具常默认跳过校验）；服务端不会跳过客户端侧证书校验。

---

6. YiANLink 协议（自研）

6.1 协议设计背景与目标

DoH/DoT 依赖公网证书体系：内网/离线场景签发与轮换繁琐，且 TLS 指纹特征明显、易被识别与劫持。YiANLink 是本项目自研的**私有应用层加密传输协议**，目标：

- **零证书运维**：以 256 位 PSK 预共享密钥替代 PKI，部署即用；
- **抗识别**：自定义二进制帧 + AEAD 加密，流量特征弱；
- **防重放**：帧内时间戳 + 接收窗口（60 秒）校验；
- **职责单一**：在网关内**只做加密传输搬运 DNS 报文**，不做解析/缓存/过滤等 DNS 业务（业务全部仍在统一查询流水线），协议层与业务层完全解耦（YiANLinkHandler 回调式注入）；
- **多路复用防队头阻塞**：客户端使用连接池轮询发送，避免单连接串行阻塞。

6.2 协议规范

6.2.1 报文格式与数据结构

传输帧（TCP 之上，逐帧传输）：

```
[4B Length(大端)] [12B IV] [密文(N)] [16B GCM Tag]
Length = 12 + N + 16（即 IV 之后全部长度）
密文 = AES-256-GCM(PSK, IV, 明文) ，无额外 AAD
```

帧内明文：

```
[8B 时间戳(大端, Unix 毫秒)] [4B StreamId(大端)] [payload]
payload = 完整 DNS 报文（wire 格式，不含 TCP 长度前缀）
```

6.2.2 握手与连接建立流程

无独立握手协商：客户端建立 TCP 连接后直接发送首帧；服务端解帧即验（PSK 对称密钥隐式认证）——

1. TCP 三次握手完成 → 连接进入"就绪"；
2. 客户端发查询帧（随机 IV 加密，时间戳 + StreamId + DNS 查询报文）；
3. 服务端解密 → Tag 校验 → 时间戳窗口校验 → 回调 `onDnsPayloadReceived(payload, clientIp)` 进入统一流水线；
4. 应答完成后以**原 StreamId** 封帧回写；
5. 连接可承载多帧往返（长连接复用）。

6.2.3 加密与认证机制

- 算法：AES-256-GCM（AEAD），密钥即 PSK（配置为 Base64 的 32 字节密钥）；
- IV：每帧 12 字节随机数，**严禁复用**（同一密钥下 IV 重用会破坏 GCM 安全性）；
- 完整性：GCM 16 字节认证标签，任何篡改都会导致解密失败；
- 防重放：明文携带发送时间戳，接收方校验与本机时钟偏差 ≤ 60 秒。

6.2.4 传输层承载方式

TCP 长连接（默认端口 **6833**，可配置；不得与明文 TCP/DoT/DoH 端口冲突）。当前版本不支持 UDP/QUIC 承载；服务端为每连接一条虚拟线程 + 连接内逐帧处理。

6.3 协议状态机与生命周期



- **半开连接治理**：客户端异常消失（不发 FIN）时，服务端依赖 60s 读超时自动回收并归零活跃连接计数（不依赖 FIN/RST，含回归测试保障）；
- open/close 在连接线程内严格配对，保证 `/api/connections` 与监控计数准确；
- 服务端停止时逐连接关闭，遵循 Gateway 统一逆序回滚。

6.4 与标准协议（DoT/DoH/DoQ）的协同与互操作

- 并存监听：YiANLink 与 UDP/TCP/DoT/DoH/DoQ 同时启用互不干扰；
- 统一流水线：YiANLink 查询进入同一 DnsHandlerPipeline，过滤/缓存/域名分流/调度行为与其它协议完全一致；
- 客户端语义：真实 TCP 对端 IP 贯穿全链路（IP 黑白名单、限流、查询日志客户端字段、协议计数 `dns.protocol{YIANLINK}`）；
- 监控契约：计数键 `dns.protocol{YIANLINK}` ↔ 展示名 `YiANLink`；活跃连接并入 totalActiveConnections（yianLinkConnections）。

6.5 协议版本与兼容性说明

- 当前协议版本 v1（帧结构如上，未含版本协商字段）；后续引入 v2 时将增加帧头版本字节并保持 v1 兼容期；
- PSK 更换需服务端与全部客户端同步更新，期间新旧密钥不兼容；
- 端口修改需重启服务生效（前端保存时已提示）。

6.6 错误码与异常处理

异常场景	服务端行为
GCM Tag 校验失败（密钥不符/被篡改）	立即断连（CryptoException → 关闭连接）
时间戳偏差 > 60 秒（重放/时钟漂移）	静默丢弃该帧，不断连、不响应
帧长非法 / 流截断	断连
enabled=true 但 PSK 缺失	启动失败（配置校验）
端口冲突	启动失败（exit 3 语义，PortInUse）
60s 无任何帧（半开/闲置）	读超时回收连接

6.7 协议实现参考（客户端示例代码）

Python 参考（依赖 `cryptography` 库）：

```
import os, socket, struct, time, base64
from cryptography.hazmat.primitives.ciphers.aead import AESGCM

PSK = base64.b64decode("<配置中的 Base64 密钥>")   # 解码后必须 32 字节
key = AESGCM(PSK)
sid = 0

def build_frame(dns_wire: bytes) -> bytes:
    global sid
    sid += 1
    plain = struct.pack(">Q", int(time.time() * 1000)) + struct.pack(">I", sid) + dns_wire
    iv = os.urandom(12)                                # 每帧随机 IV，严禁复用
    ct = key.encrypt(iv, plain, None)
    return struct.pack(">I", len(iv) + len(ct)) + iv + ct

def parse_frame(buf: bytes) -> tuple[int, bytes]:
    total = struct.unpack(">I", buf[:4])[0]
    iv, body = buf[4:16], buf[16:4 + total]
    plain = key.decrypt(iv, body, None)
    ts, resp_sid, payload = plain[:8], plain[8:12], plain[12:]
    assert abs(int.from_bytes(ts, "big") - int(time.time() * 1000)) <= 60_000
    return int.from_bytes(resp_sid, "big"), payload   # 响应以原 StreamId 回包

sock = socket.create_connection(("192.168.1.2", 6833))
sock.sendall(build_frame(dns_query_wire))
# 读取 4B 长度 + 帧体 → parse_frame → 校验 StreamId 与请求匹配
```

服务端对应实现：dns-proxy 的 com.example.dns.proxy.yianlink 包（YiANLinkServer / YiANLinkHandler），业务回调签名 onDnsPayloadReceived(byte[] payload, InetSocketAddress clientIp)（双参版本；旧单参签名已废弃）。

7. 调度系统与数学模型

7.1 调度目标与设计思路

上游调度的目标：**低延迟**（优先 RTT 低的节点）、**高可用**（熔断隔离劣化/故障节点）、**容灾**（失败自动切换与降级）、**公平**（轮询严格均分）。

本项目将"上游查询方式"对外提供三个选项（可在 设置 → DNS 服务 热修改）：

选项	配置值	语义	数学模型
1 负载均衡	ROUND_ROBIN	顺序轮换一台上游；熔断节点垫底兜底，游标只在健康子集上轮转（熔断混合态消除首选偏斜，全健康时与经典轮询逐次一致）	无（确定性轮转）
2 趋势预测	TREND_PREDICTION	依据每节点 RTT 时序动态计算权重并做 SWRR 平滑加权轮询	代价能量函数 J + Softmax + 积分滑窗 + SWRR（本章 7.2.2）
3 并行请求	PARALLEL	并发查询全部健康上游，取首个有效结果；maxUpstreamAttempts 可限制参与数（0=全部）	无（竞速）

节点权重完全由调度器自动计算，**无需（也不支持）手动配置权重**。Split DNS 域名分流命中时绕过上述调度直连专属上游；专属上游独立超时（默认 2000ms）且具备熔断保护，异常时自动降级回默认引擎。

此外，查询并发治理使用**预测性容量控制**（泰勒展开余项 → 虚拟线程并发闸门），其数学模型见 7.2.2 之（二）；缓存 TTL 的动态浮动模型见 7.2.2 之（三）。

7.2 数学公式详解

7.2.1 公式定义与符号说明（变量表）

（一）趋势预测（上游动态权重）

符号	含义	默认/范围
rtt(t)	最近一次成功查询的往返时延（毫秒，nanoTime 打点；失败/超时不写样本）	实测
μ(t)	RTT 的 EMA：$\mu(t)=\alpha\cdot rtt(t)+(1-\alpha)\cdot\mu(t-1)$	α=0.3
v(t)	一阶导（恶化速度）：$v(t)=\mu(t)-\mu(t-1)$	毫秒/采样

a(t)	二阶导（恶化加速度）：a(t)=μ(t)−2μ(t−1)+μ(t−2)	毫秒/采样²
σ(t)	RTT 样本标准差（分母 n−1）	窗口=5
I	积分健康窗口：布尔环形，每秒记录"本周期未被熔断"	N=10
J_i	节点 i 的代价能量（越小越优）	≥0
W_i	节点 i 的发布权重（0–100 整数）	0–100
b	积分健康加成倍率（窗口已满且全健康时 1.2，否则 1）	1/1.2
current[i]	SWRR 当前权重数组	动态
weight[i]	SWRR 每轮使用的发布权重	0–100

（二）预测性容量控制（查询并发闸门）

符号	含义	默认/范围
C(t)	负载 = 当前活跃虚拟线程数 + 任务积压数（积压由 backlogSupplier 注入，当前恒 0）	实测
EMA(t)	C(t) 的指数平滑：EMA(t)=0.3·C(t)+0.7·EMA(t−1)	Δt=1s
v, a, jerk	一阶/二阶/三阶差分	—
predict(t+5)	泰勒预测：EMA + v·5 + 0.5·a·25	5s 前瞻
σ	最近 N=5 采样点标准差（噪声判据）	—
M	最近 N=5 采样点中 jerk 绝对值的最大值，再钳制到 maxJerkLimit	16.0
R_bound	拉格朗日余项上界 (M/6)·125	—
safetyThreshold	安全阈值 max(MAX_VIRTUAL_THREADS − R_bound, MIN_SAFETY_THRESHOLD)	5000 / 1000

（三）频率感知动态 TTL（默认缓存引擎；`-Ddns.cache.engine=caffeine` 回退 Caffeine）

符号	含义	默认/范围
f	域名访问频率（指数衰减频次）：f ← f·exp(−λ·Δt)+1	λ=ln2/300（半衰期 300s）
k	mult(f) 半饱和点	100
mult(f)	TTL 倍率：2/3+(1.75−2/3)·f/(f+k)	[2/3, 1.75]
ttlBase	应答内记录的最小 TTL	—
s	淘汰用访问分数：s ← 0.99·s+1（稳态上限 ≈100）	—

7.2.2 各公式的物理含义与推导

（一）代价能量函数 J（趋势预测）

```
J = exp(0.01·μ) + 2·max(0, v) + 1.5·|a| + σ/μ
```

项	惩罚对象	物理含义
exp(0.01·μ)	RTT 绝对水平	指数惩罚：μ 每增加 100ms 该项放大约 e 倍，对高时延节点快速降权
2·max(0, v)	上行趋势	只惩罚"正在变慢"，变快不奖励（防低基数操纵）
1.5·|a|	曲率/抖动	恶化加速度越剧烈（无论加速还是震荡）代价越高
σ/μ	相对离散度	变异系数 CV，衡量节点稳定性（尺度无关）

冷启动补全：无样本时 μ=σ=0，σ/μ 在数学上未定义；实现定义该项贡献为 0（无观测即无离散度），其余三项不变。

Softmax 权重：

```
logit_i = −0.5·J_i + ln(b_i)
W_i = exp(logit_i) / Σ_k exp(logit_k)
```

- 熔断节点 W_i 直接置 0（BackoffController 函数式接口适配 UpstreamHealth.isOpen，节点键 host:port authority）；
- 数值实现利用 Softmax 平移不变性（先减最大 logit 再取指数），避免 exp(−0.5·J) 大 J 时整体下溢为 0 导致 0/0；数学结果与原式完全相同；
- 积分加成等价于未归一化权重乘以 b（ln(1.2) 加在 logit 上）；窗口未满或出现熔断打断时 b=1，与原公式逐字一致。

量化发布：W×100 四舍五入写入 AtomicIntegerArray（0–100）；健康节点权重为正但取整为 0 时提升为 1（防量化饿死）；软门限总和不必等于 100（SWRR 不要求）。后台单守护线程（dns-weight-scheduler）固定 1 秒周期执行"采样→计算→发布"，启动先同步发布一次；单轮计算失败保留上一版权重；主请求链路只读该数组（无锁、零分配）。

SWRR 平滑加权轮询：

```
每轮：current[i] += weight[i]（全部可用节点）
选中：current 最大的节点 i*
扣减：current[i*] −= Σ weight
```

- 守恒性：每节点两次 addAndGet，所有 current 之和严格守恒（每次调用净变化 0），长期选中比例严格收敛到权重比；"选最大+扣减"非同一原子操作，瞬时平滑性弱化是有意取舍（查询路径零全局锁）；
- 权重 ≤0 的节点不参与选路且 current 归零（避免恢复瞬间因历史欠账连选）；全部权重为 0 返回 −1，上层回退配置原序，全部候选仍熔断时由 Forwarder 降级为 UPSTREAM_FAILURE。

（二）预测性容量控制（泰勒展开余项 → 并发闸门）

```
EMA(t)    = 0.3·C(t) + 0.7·EMA(t−1)                    （平滑）
predict   = EMA(t) + v(t)·5 + 0.5·a(t)·25              （泰勒 5 秒预测）
噪声降阶   = σ > 3.0 时弃用加速度项：predict = EMA + v·5   （皮亚诺余项）
R_bound   = (M/6)·125，M = min(窗口 max|jerk|, 16.0)    （拉格朗日余项）
```

```
safety   = max(5000 − R_bound, 1000)                    （安全阈值）
目标许可  = clamp(ceil(predict), minPermits, floor(safety))，另加防抖死区 1
```

- 皮亚诺余项：噪声主导（σ>3.0）时二阶项不可信，降阶为一阶预测；
- 拉格朗日余项：三阶导界 M 给出 |R| ≤ M·h³/3! = M·125/6 的误差上界，据此预留安全边际；
- 工程分层：预测计算路径（后台单线程，volatile 发布，严禁 synchronized）/ 容量调整路径（低频）/ 拒绝决策路径（tryAcquire 快速失败，无锁）；
- Semaphore 硬约束：扩容走 release(差值)、缩容必须私有子类 shrink→reducePermits（release 不能创造上限）；先 tryAcquire 成功再创建虚拟线程，finally 归还；
- 接入范围：UDP/DoT/DoH/DoQ 每查询准入（拒绝语义按协议差异化：UDP/DoT 回 SERVFAIL，DoH 回 HTTP 200 + DNS SERVFAIL，DoQ 按 RFC 9250 §4.3 重置流）；TCP/YiANLink 为每连接一线程模型，不接入（连接级治理独立）；拒绝日志每累计 10000 条才打一条 WARN（防日志雪崩）。

（三）频率感知动态 TTL（默认缓存引擎）

```
f ← f·exp(−λ·Δt) + 1      λ = ln2/300（半衰期 300 秒）
mult(f) = 2/3 + (1.75 − 2/3)·f/(f + k)     k = 100（Michaelis-Menten 饱和）
ttlDyn = round(ttlBase · mult(f))
  → 绝对安全 clamp [10, 86400] 秒
  → 用户边界：先按 maxCacheTime 截断，再按 minCacheTime 提升（0 = 不限制）
```

- f=0 → mult=2/3（冷域名缩短 TTL）；f=k → 1.2083；f→∞ → 1.75（热点上限，避免长期不复核上游）；
- 读取按 f·exp(−λ·(now−lastAccess)) 折算当前值；CAS 重试取 max(now, lastAccess) 防时钟倒挂；
- **负缓存分歧**：NXDOMAIN 只做安全 clamp 与 max 截断，绝不被 min 提升（业务语义是允许域名快速复活）；
- 容量淘汰：超限时随机采样 5 条，淘汰衰减分数 s（s←0.99s+1）最低者，不做全表排序；
- 瞬态错误（SERVFAIL/REFUSED/UPSTREAM 系）一律不缓存——该语义与 Caffeine 回退引擎一致。

7.2.3 公式应用场景（何时触发、输入输出）

公式	触发时机	输入	输出	消费者
μ/v/a/σ 更新	每次上游成功应答（网络成功即记录，含 deferrable rcode）	rtt 毫秒	新 Snapshot（CAS 发布）	J 计算
J + Softmax	每秒（调度线程）+ 启动同步一次	全节点 Snapshot + 熔断状态	AtomicIntegerArray 权重	SWRR
SWRR select	每次需选上游的查询	权重快照	首选节点 + 权重降序候选序	Forwarder
域名分流匹配	每次缓存未命中后	查询域名	专属上游或"未命中"	Forwarder
泰勒预测/闸门调整	每秒采样	活跃虚拟线程数 + 积压	最新许可目标（死区内不动）	executeTask
闸门准入	每查询提交	当前许可	通过 /	各传输（差异化拒绝

			RejectedExecution	应答）
f/mult 更新	每次缓存命中/写入	Δt	新 f、动态 TTL	FrequencyAwareCache

7.3 调度决策流程

```
查询到达（缓存未命中）
    │
    ├─ 域名分流命中？ ──是──► 直连专属上游（独立超时 2000ms；熔断/超时 → 降级回默认引擎）
    │
    └─ 否 → 按上游查询方式派发：
        ├─ 负载均衡：RoundRobinSelector 在健康子集上轮转（熔断节点恒垫底兜底）
        ├─ 趋势预测：DynamicWeightedUpstreamSelector → SWRR 首选 + 其余按权重降序（零权重沉底）
        │            全 0 / 全熔断 → 配置原序兜底
        └─ 并行请求：并发全部健康上游，首个有效应答胜出
    │
    ▼
逐候选尝试：health.allows() 跳过熔断节点
    ├─ 应答 NOERROR / NXDOMAIN → 终态返回
    ├─ SERVFAIL / REFUSED / FORMERR / NOTIMP → 视为可重试，切换下一候选
    └─ 传输异常/超时 → 熔断计数 +1（不写 RTT 样本），切换下一候选
    │
    ▼
全部候选失败 → 降级应答（UPSTREAM 错误码/SERVFAIL）
```

熔断状态实时生效（health.allows()），而权重数组每秒发布一次——熔断后最多 1 秒内 SWRR 仍可能把该节点排首位，由 Forwarder 的健康检查兜底跳过，不影响正确性。

7.4 参数调优指南

参数	默认值	可调性	调优效果
上游查询方式 strategy	ROUND_ROBIN	Web 热更新	三策略切换见 7.1；热点波动大选趋势预测，上游数量多且要求最低时延选并行请求
缓存引擎 dns.cache.engine	频率感知（默认）	JVM 系统属性（仅启动时读取，切换须重启）	未设置/frequency=频率感知；caffeine=回退 W-TinyLFU；未知值告警后仍按频率感知处理
resetStateOnUpdate	false	Web 热更新	true 时保存上游配置即重置熔断/RTT 状态（排查期用）；默认 false 保留状态
maxUpstreamAttempts	0（全部）	配置/代码	限制并行请求参与者数量，控制放大系数
域名分流 timeoutMs	2000	每条目可配	专属上游独立超时；0=默认值
闸门 maxVirtualThreads / minSafetyThreshold / maxJerkLimit	5000 / 1000 / 16.0	Gateway 装配常量	上调允许更高并发，下调提前拒绝保护小内存实例
α（EMA）	0.3	源码常量	增大响应更快但抖动更大
σ 窗口 / 积分窗口 N	5 / 10	源码常量	窗口越大越平滑、恢复越稳

积分加成 bonus	1.2	源码常量	连续健康节点的权重倾斜强度
发布周期	1s	源码常量	缩短降低熔断感知延迟，增加计算开销
f 半衰期 / k / mult 上限	300s / 100 / 1.75	源码常量	决定热点 TTL 放大速度与上限
缓存 TTL 边界	clamp[10,86400]；min/max 用户边界 0	配置可调	minTtlSeconds/maxTtlSeconds 控制实际缓存时长

说明：公式常量（α、窗口、bonus、闸门数学参数）为内置设计值，调整需修改源码；运维层面可调项以配置为准。公式语义不允许在工程加固中被改变。

7.5 调度效果示例（实例演算）

例 1：三节点稳态权重

设 A/B/C 稳定 RTT = 30 / 60 / 120ms（σ=0、v=0、a=0）：

```
J_A = e^0.30 ≈ 1.350    J_B = e^0.60 ≈ 1.822    J_C = e^1.20 ≈ 3.320
logit = −0.5·J：−0.675 / −0.911 / −1.660
Softmax（减最大 logit 后归一）：W_A ≈ 46.2%，W_B ≈ 36.5%，W_C ≈ 17.3%
量化发布 → 46 / 36 / 17（总和 99，软门限合法）
```

SWRR 长期选中比例即 46:36:17；随后 A 出现恶化（v>0）→ J_A 抬升 → 权重自动回落，实现"趋势"感知。

例 2：熔断隔离

B 连续失败达阈值 → isOpen → W_B=0 → SWRR 中 current[B] 归零、永不首选；恢复后 half-open 试探成功，下一发布周期权重逐步恢复（期间被健康检查兜底跳过，无错误应答泄漏）。

例 3：预测闸门演算

C(t) 序列 100 → 110 → 120（EMA 初值 100）：

```
EMA1 = 0.3·110 + 0.7·100 = 103
EMA2 = 0.3·120 + 0.7·103 = 108.1
v = 5.1，a = 108.1 − 2·103 + 100 = 2.1
predict(5s) = 108.1 + 5.1·5 + 0.5·2.1·25 = 159.85 → 目标许可 160
```

---

8. 配置参考手册

8.1 配置文件结构总览

配置文件支持 YAML（config.yaml / config.yml）与 JSON（config.json）；分层模型：**内置默认 → 用户文件（深合并）→ 运行期覆盖层**，未写出的字段回退默认值。顶层段：

顶层段	内容
server	DNS 监听与安全传输（bindAddress/端口/线程/hostname/tls/dot/doh/doq）
yianlink	YiANLink 私有协议（enabled/port/psk）
upstreams	上游列表（address + protocol）
upstream	上游查询方式（strategy、resetStateOnUpdate）
domainRouting	Split DNS 域名分流列表
filter	过滤规则路径与按客户端分组
cache	缓存开关/容量/TTL
storage	数据存储根目录
logging	查询日志与运行日志级别
web	Web 管理面
plugins	插件私有配置（插件经 subset("plugins.<name>") 读取）

配置校验：非法值聚合为 ConfigError 列表逐条报错（exit 2）；Web 保存走"整体校验 → 原子写回启动配置文件 → 切换运行期配置"，校验失败整体不应用。变更通知按顶层段比对（无变更不通知）。

8.2 全部配置项详解

server（DNS 监听与安全传输）

键	类型	默认值	说明
server.bindAddress	string	0.0.0.0	DNS 监听地址
server.udpPort	int	53	UDP 端口（0 不允许）
server.udpEnabled	bool	true	UDP 监听开关（false 不启动 UDP；TCP/DoT/DoH 至少启用一种）
server.tcpPort	int	53	TCP 端口（0 = 禁用 TCP）
server.threads	int	8	工作线程数
server.hostname	string	""	对外服务域名（可选）：启动时校验证书 SAN/CN 覆盖；横幅输出

			客户端提示
server.tls.certFile	string	TLS/fullchain.pem	PEM 全链证书（DoT/DoH/DoQ 共享；任一协议启用时必须存在）
server.tls.keyFile	string	TLS/privkey.pem	PEM 私钥（PKCS#8/PKCS#1；DoQ 仅 RSA）
server.tls.minVersion	string	TLSv1.2	DoH 最低 TLS 版本（TLSv1.2/TLSv1.3）；DoT 固定 TLS1.2；DoQ 固定 TLS1.3；TLSv1/1.1 永不支持
server.tls.legacyRsaCompat	bool	false	兼容老旧客户端：开启后额外下发 TLS_RSA_*；开启需重启生效，关闭立即生效
server.dot.enabled / port	bool/int	false / 853	DoT 服务端开关与端口
server.doh.enabled / port / path	bool/int/string	false / 443 / /dns-query	DoH 服务端；path 须匹配 ^/[a-zA-Z0-9_\-/]*$
server.doq.enabled / port	bool/int	false / 853	DoQ 服务端；可与 DoT 同端口共存

yianlink（YiANLink 私有协议）

键	类型	默认值	说明
yianlink.enabled	bool	false	启用后监听 TCP；需重启生效
yianlink.port	int	6833	不得与明文 TCP / DoT / DoH 端口冲突
yianlink.psk	string	""	256 位 Base64 预共享密钥（解码后 32 字节）；enabled=true 时必填

upstreams / upstream（上游与查询方式）

键	类型	默认值	说明
upstreams[].address	string	—	支持 host / host:port / 完整 URL（如 `https://doh.pub/dns-query`、`tls://dot.pub`）；DoH 可带路径；至少一条
upstreams[].protocol	string	UDP	UDP / TCP / DOT / DOH（默认端口 53/53/853/443）；地址 scheme 前缀须与 protocol 一致，否则告警跳过该条目
upstream.strategy	string	ROUND_ROBIN	ROUND_ROBIN 负载均衡 / TREND_PREDICTION 趋势预测 / PARALLEL 并行请求；旧值

			（SEQUENTIAL/RANDOM/PRIORITY/FAILOVER）自动归一 ROUND_ROBIN
upstream.resetStateOnUpdate	bool	false	保存上游配置时是否重置熔断与 RTT 状态（默认尽量保留）

上游地址语法（网关与域名分流共用 UpstreamAddressParser）：

```
[scheme://][userinfo@]host[:port][/path]
- scheme 仅用于推导默认端口与 DoH 路径，协议以 protocol 配置为准；
- scheme 与 protocol 冲突 → 启动报错；http:// 明文 DoH 直接拒绝；
- DoT 的 host 兼作 TLS SNI；DoH 保留 path 与 query；
- 例：https://doh.pub/dns-query（DOH）、tls://dot.pub（DOT，等价 dot.pub:853）。
```

domainRouting（Split DNS 域名分流）

键	类型	默认值	说明
domainRouting[].domainPattern	string	—	精确域名（www.a.com）或通配符（*.b.com，仅匹配任意层级子域、不含 b.com 自身）
domainRouting[].upstreamAddress	string	—	[scheme://][userinfo@]host[:port][/path]（scheme=udp/tcp/dot/doh，缺省 udp:53；与上游共用 UpstreamAddressParser）；禁止指回本网关自身监听地址（保存时校验拒绝）
domainRouting[].timeoutMs	int	2000	专属上游独立超时（0=默认 2000ms），超时直接返回 SERVFAIL

filter（过滤规则）

键	类型	默认值	说明
filter.blocklists[]	list	data/filters/ad-hosts.txt	全局黑名单文件（Adblock 双斜线拦截、hosts IP 行、纯域名、*. 通配、/re/ 正则）
filter.allowlists[]	list	data/filters/allow.txt	全局白名单（@@ 例外优先于黑名单）
filter.rewrites[]	list	data/filters/rewrites.txt	重写规则（源域名 + 目标 IP/IPv6/CNAME，CNAME 自动检环）
filter.clients[].client	string	—	客户端 IP/CIDR/逗号组合或 *
filter.clients[].blocklists[] / allowlists[]	list	—	该客户端分组的专属名单（用户专属覆盖全局）

规则语法速记：||domain^ Adblock 拦截；@@domain^ 例外；0.0.0.0 host hosts 拦截；host 目标 重写行；$important 高优先；$client=1.2.3.0/24 按客户端；$dnstype=AAAA 按类型；/regex/ 正则（带防爆炸保护）；|http://...| URL 锚定。

cache（缓存）

键	类型	默认值	说明
cache.enabled	bool	true	缓存开关（false 时全量回源）
cache.maxEntries	int	100000	缓存容量上限（默认频率感知引擎；Caffeine 回退时为 W-TinyLFU 容量）
cache.defaultTtlSeconds	int	300	正向应答默认 TTL
cache.negativeTtlSeconds	int	60	NXDOMAIN 负缓存 TTL
cache.minTtlSeconds	int	0	最小缓存 TTL（低于提升；0=不限制）
cache.maxTtlSeconds	int	0	最大缓存 TTL（高于截断；0=不限制）

缓存引擎：**默认「频率感知动态 TTL」**（按域名访问热度浮动 TTL，热点更久、冷域名更快过期）；JVM 参数 `-Ddns.cache.engine=caffeine` 显式回退到 Caffeine W-TinyLFU，未知值告警后仍按频率感知处理；引擎仅在启动时读取一次，切换须重启；配置字段与 REST 契约不变。SERVFAIL/REFUSED/UPSTREAM 系瞬态错误一律不缓存。

storage（数据存储）

键	类型	默认值	说明
storage.dataDir	string	data	查询日志等数据落盘基准；相对路径以 jar 所在目录解析，可填绝对路径

logging（日志）

键	类型	默认值	说明
logging.queryLogEnabled	bool	true	查询日志开关
logging.queryLogPath	string	querylog	相对 storage.dataDir 的子目录（绝对路径直接使用）
logging.retentionHours	int	168	查询日志保留时长（小时）：24/168/720/2160 或任意自定义
logging.statsRetentionHours	int	24	统计窗口（小时）：24=最近 1 天 / 168=最近 7 天
logging.maxEntries	int	100000	内存环形缓冲最大条目
logging.bufferSize	int	8192	异步写入队列容量
logging.storageType	string	JSONL	JSONL（按天轮转）/ MEMORY（仅内存）
logging.level	string	INFO	TRACE/DEBUG/INFO/WARN/ERROR

web（管理面）

键	类型	默认值	说明
web.enabled	bool	true	管理面开关
web.bindAddress	string	127.0.0.1	管理面监听地址（与 DNS 数据面分离，建议仅内网/本机）
web.port	int	8080	管理面端口
web.adminUsername	string	admin	管理员账号
web.adminPassword	string	admin	管理员口令（⚠ 生产务必修改）

plugins（插件私有配置）：任意键值对，插件经 DnsConfigProvider.subset("plugins.<name>") 点分读取；示例：

```
plugins:
  rate-limiter:
    qps: 5000
```

8.3 典型配置示例

场景一：家庭广告拦截（最小化）

```
server: { bindAddress: 0.0.0.0, udpPort: 53, tcpPort: 53 }
upstreams:
  - { address: 223.5.5.5, protocol: UDP }
  - { address: 119.29.29.29, protocol: UDP }
filter:
  blocklists: [data/filters/ad-hosts.txt]
  clients:
    - client: 192.168.2.0/24          # 儿童设备分组
      blocklists: [data/filters/kid-block.txt]
cache: { enabled: true, maxEntries: 100000 }
web: { enabled: true, bindAddress: 127.0.0.1, port: 8080 }
```

场景二：对外加密 DNS（DoT + DoH + DoQ）

```
server:
  bindAddress: 0.0.0.0
  udpPort: 53
  tcpPort: 53
  hostname: dns.example.com
  tls:
    certFile: TLS/fullchain.pem
    keyFile: TLS/privkey.pem
  dot: { enabled: true, port: 853 }
  doh: { enabled: true, port: 443, path: /dns-query }
  doq: { enabled: true, port: 853 }
web:
  enabled: true
  bindAddress: 10.0.0.5          # 仅内网可访问管理面
  port: 8080
```

场景三：YiANLink 专用通道（供自研客户端）

```
yianlink:
  enabled: true
  port: 6833
  psk: "3q2+7wEYxkBeK0vF0uYn2O2o0S6mJ0yQ1aB9cD8eF0g="   # 示意：Base64 32 字节
server: { bindAddress: 0.0.0.0, udpPort: 53, tcpPort: 0 }
```

场景四：多上游（含 URL 写法）+ 趋势预测 + 域名分流

```
upstreams:
  - { address: 223.5.5.5, protocol: UDP }
  - { address: "https://doh.pub/dns-query", protocol: DOH }
```

```
    - { address: "tls://dot.pub", protocol: DOT }
  upstream:
    strategy: TREND_PREDICTION
  domainRouting:
    - { domainPattern: "*.corp.local", upstreamAddress: udp://10.0.0.53:53 }
    - { domainPattern: "stream-vip.example", upstreamAddress: doh://dns.alidns.com, timeoutMs:
  0 }
---
```

9. 运维与监控

9.1 日志体系与排查方法

查询日志（logging 段）：

- 存储：JSONL 按天轮转 `querylog-yyyyMMdd.jsonl`（storage.dataDir 下 querylog/），异步虚拟线程写线程批量刷盘；启动自动清理超过 retentionHours 的文件；目录不可写自动降级内存模式（WARN）；
- 内存侧：环形缓冲（maxEntries）+ 有界写队列（bufferSize，满则丢弃计数 droppedByBuffer）+ 24 小时统计桶（供 /api/stats）；
- 每条记录包含：时间戳、域名、查询类型、rcode、客户端 IP、上游、耗时、来源协议（UDP/TCP/DOT/DOH/DOQ/YIANLINK）、是否拦截、命中规则（matchedRule）、是否缓存命中（fromCache）。

排查方法：

```
# 查看今天日志中某域名的记录
grep "example.com" data/querylog/querylog-$(date +%Y%m%d).jsonl
# 控制台快速验证
curl -H "Authorization: Bearer <token>" "http://127.0.0.1:8080/api/queries?limit=20"
```

运行日志：stdout 输出（级别 logging.level）；TLS 层可加 `-Ddns.tls.debug=true` 输出带 SNI 的握手诊断。

9.2 监控指标说明

上游节点指标（仪表盘 / 上游服务器页）：

- `queries`：尝试次数——每次向上游发起尝试前 +1（含重试）；
- `failures`：失败次数（传输异常/超时/可重试 rcode）；
- `avgLatencyMs`：仅成功应答后记录（失败**刻意不写**，避免污染调度器 EMA）；无成功样本返回 **-1**（前端显示"无成功应答"）；**0** 表示有样本但耗时不足 1ms（前端显示 "<1 ms"）；
- 判读口径："有次数无延迟" = 该上游全部尝试被 failover 掩盖（从未成功），应优先排查连通性。

查询指标（GET /api/metrics）：

指标	含义
dns.query.total	查询总数
dns.query.dropped	请求链丢弃数
dns.query.filter.respond	过滤器短路应答数
dns.query.filter.allow / dns.query.blocked	放行 / 拦截数
dns.query.rewritten	重写数（非 NXDOMAIN 型拦截）
dns.query.cache.hit	缓存命中数
dns.query.latency	查询平均延迟
dns.protocol{UDP/TCP/DOT/DOH/DOQ/YIANLINK}	按来源协议的计数（新增协议必须同步白名单/配

	色/回归断言）

连接指标（GET /api/connections / /api/status）： 活跃连接 totalActiveConnections = TCP + DoT + DoH + yianLinkConnections；YiANLink 依赖 60s 读超时回收半开连接保证计数归零。

系统状态（GET /api/system/info，纯 JDK/JMX 采集，零第三方库）：

字段	口径
uptime	JVM 运行时长；<1 天按小时、≥1 天按"天+小时"展示
load1/5/15	load1 取 getSystemLoadAverage（Windows 不可用返回 -1 → 前端 N/A）；load5/15 为 3 秒采样点的 EMA（窗口 300s/900s）
线程数	当前 JVM 线程数
堆/栈/直接内存	堆实测；栈=线程数×单线程栈大小（读 -Xss，缺省 1MB，估算值）；直接内存取 direct BufferPool
GC	YoungGC/OldGC 次数与累计耗时；recentGcPauseMs 为相邻采样间累计 GC 耗时增量（最近刷新周期的停顿）
系统/Java	os.name/version/arch 与 java.version/vendor/runtime/vm（等宽小字三行）

不可用值统一以 -1 表示、前端渲染 N/A。

9.3 常见问题排查（FAQ / 故障处理手册）

现象	排查与结论
nslookup 全部超时	先排查本机 DNS 劫持/接管工具（YogaDNS、广告拦截、安全软件）与 53 端口占用；换非 53 端口 + 原生 UDP 探针二分；服务端数据面正常时勿当服务端 Bug
启动即退出，exit=3	端口被占用：netstat/资源监视器定位占用者，改端口或停冲突进程
启动即退出，exit=2	配置校验失败：控制台逐条打印字段路径与原因，按提示修正
启动失败提示"TLS 证书未覆盖对外服务域名"	server.hostname 与证书 SAN/CN 不匹配；更换证书或修正 hostname（错误信息列出证书实际覆盖域名）
DoT 客户端失败而 DoQ 正常	证书未被客户端信任或地址不在 SAN（DoQ 工具常跳过校验）；另注意 DoT 固定 TLS1.2 且禁用 ALPN 属预期行为
ALPN 告警日志	客户端 ALPN 不在白名单被明确拒绝（可观测设计）；确认客户端协议（h2/HTTP1.1/doq）
Web 保存配置未生效	保存是整体原子校验：查看返回的校验错误；写盘失败也会整体不生效

删除再重加的上游直接熔断	上游状态按身份复用（authority+序号）；确需清零用"重置上游状态"按钮或 resetStateOnUpdate=true
YiANLink 连接计数不清零	半开连接不发 FIN 属正常；60s 读超时自动回收归零（回归用例保障）
Web 忘记口令	改配置文件 web.adminPassword → 重启；旧会话全部失效
缓存命中率低	检查 min/max TTL 配置、负缓存 TTL；瞬态错误（SERVFAIL 等）按设计不缓存；热点域名多时保持默认频率感知引擎即可，如需回退加 -Ddns.cache.engine=caffeine
Windows 负载显示 N/A	getSystemLoadAverage 平台限制（返回 -1），属预期
上游"有次数无延迟"	该上游从未成功应答（全被 failover 掩盖）：检查地址/协议/连通性；注意 DoH 上游地址须写完整 URL（如 https://doh.pub/dns-query），scheme 与 protocol 必须一致

9.4 升级与回滚流程

升级：

1. 备份：配置文件（config.yaml）、data/（查询日志与规则）；

2. 替换 jar 为新版本（配置向后兼容：旧策略值自动归一、已废弃字段如上游 weight 被忽略且保存后随列表替换消失）；

3. 启动并核对横幅版本号（含发行代号）；

4. 观察仪表盘与 /api/system/info 确认服务健康。

回滚：换回旧 jar + 旧配置文件重启即可；web 界面保存会写回配置文件，必要时用备份恢复。

版本策略：pom/manifest/jar 文件名使用纯语义版本（如 1.0.3）；展示层（README、启动横幅）在 1.0.0 → 1.2.20 区间附加发行版代号 "Marhaba"。

---

10. 开发者文档

10.1 代码仓库结构说明

```
DNS-Program Files/
├── pom.xml                    # 父 POM（聚合 + dependencyManagement）
├── build.ps1 / smoke.ps1      # Windows 构建/冒烟脚本（内置 Maven 环境修复）
├── maven-settings.xml         # 项目级 Maven 设置（镜像 + 本地仓库）
├── dns-server-api/            # 接口层
├── dns-core/                  # dnsjava 编解码与应答工厂
├── plugin-api/                # 插件扩展点（7 接口 + sealed 决策）
├── config/                    # 配置模型/加载/校验/热更新 + 示例配置
├── logging/                   # 查询日志（JSONL/内存、统计桶）
├── metrics/                   # 内存指标注册表
├── filter-engine/             # 过滤引擎（规则解析/索引/客户端档案）
├── upstream-resolver/         # 上游传输/选择器/熔断/趋势预测调度/域名分流/地址解析
├── cache/                     # 频率感知缓存（默认引擎）+ Caffeine 回退
├── dns-proxy/                 # 传输服务端（UDP/TCP/DoT/DoH/DoQ/YiANLink）、流水线、预测闸门、
启动入口
├── web-admin/                 # 管理 REST API（Java）+ frontend/（Vue3 控制台）
├── tools/                     # 冒烟等工具
└── config/                    # config.example.yaml / config.example.json
```

关键入口：`com.example.dns.proxy.DnsGatewayApplication`（main + 12 步装配 + 优雅关闭）；前端源码 `web-admin/frontend/`，构建产物输出至 `web-admin/src/main/resources/static`（打进 jar）。

10.2 本地开发环境搭建

步骤	说明
1. JDK 21	安装并配置 JAVA_HOME（OpenJDK/Semeru/毕昇均可）
2. Maven 3.6+	全局 mvn；本项目构建始终走 `maven-settings.xml`（勿依赖全局 settings）
3. Node.js（改前端才需要）	v20+；进入 web-admin/frontend 执行 `npm install`
4. IDE	任意；语言级别 21

受限环境说明：build.ps1 会注入 `MAVEN_OPTS=-Daether.syncContext.named.factory=noop`（规避文件锁拦截，普通环境无副作用）；本地仓库路径由 maven-settings.xml 指定。

10.3 构建、测试与发布流程

操作	命令
全量构建（含全部单测 + fat jar）	`powershell -File build.ps1 clean package`
单模块测试（必带 -am）	`powershell -File build.ps1 -pl <module> -am clean test`
端到端冒烟（先 package）	`powershell -File smoke.ps1`（9 项断言）
前端构建	`cd web-admin/frontend && npm run build`（产物进 resources/static）
直接 Maven（跨平台）	`mvn -s maven-settings.xml -B clean package`

fat jar 机制（dns-proxy 模块 maven-shade-plugin）：合并全部内部模块与第三方依赖；ServicesResourceTransformer 合并 SPI；剔除签名文件与 module-info；manifest 写入 `Main-Class: com.example.dns.proxy.DnsGatewayApplication`、`Implementation-Version` 与 `Multi-Release: true`（dnsjava 多版本 JAR 须排除其 InetAddressResolverProvider SPI，避免接管 JDK 全局解析）。打包后清理 shade 中间产物 `original-*.jar`，保证产物唯一。

测试规模：全工程累计 350+ 项单测；关键模块参考量级——dns-proxy 80+、config 60+、upstream-resolver 50+、filter-engine 60、web-admin 45、cache 30+。

版本升级流程：批量替换版本号（12 个 pom + frontend/package.json + package-lock.json + README，共 15 处口径）→ grep 确认无残留 → `clean package` 全绿 → 删除 original-*.jar → manifest 与实跑横幅双验证。发行代号规则：区间 1.0.0 → 1.2.20 使用 "Marhaba"；超出区间须确认新代号。

10.4 贡献指南

- **代码基线**：Java 21（record/sealed/虚拟线程惯用）；除非要求，不写注释；命名与变量以英文为主，对外策略/功能命名使用中文语义名并映射英文常量（如 趋势预测 → TREND_PREDICTION）；
- **架构红线**（违反视为回归）：

1. 查询主链路保持无锁（无锁读发布数组/volatile 快照；不引入 synchronized 热路径）；

2. 数学公式语义不可改（工程加固不得改公式；新取整点必须考虑防抖死区吸收）；

3. 并发闸门必须"先 tryAcquire 成功再创建线程，finally 归还"；Semaphore 缩容只能走 reducePermits 子类路径；

4. YiANLink 必须传递真实对端 clientIp（IP 黑白名单/限流/日志依赖）；

5. 新增协议必须同步：协议计数白名单/前端配色/活跃连接统计/回归断言；

6. "可缓存 rcode"判断两处（DnsHandlerPipeline 与 CachingDnsCacheProvider）必须保持一致；

- **测试要求**：新行为/修复必须带单测；轮询公平性用严格数值断言（如 3 节点 1 熔断 600 次首选严格 300/300）；临时验证代码用完即删；
- **PR 流程**：fork → 分支 → 全量 `clean package` 全绿 + 冒烟通过 → 提交（附构建/测试证据）→ 关联 Issue 说明动机与影响面。

---

11. 安全说明

11.1 安全模型与威胁假设

信任边界：局域网客户端（半可信）、管理面（仅管理员）、上游 DNS（不可信链路/可信内容）、公网暴露面（如启用加密协议对外开放）。

威胁假设与对应控制：

威胁	控制措施
明文 DNS 窃听/劫持	DoT/DoH/DoQ 加密传输（TLS1.2+，现代 ECDHE/AEAD 套件白名单；SSLv3/TLSv1/1.1 永久封杀）；自研 YiANLink AES-256-GCM + 60s 时间戳窗口防重放
非法客户端接入加密端口	DoT/DoH TLS 证书认证；YiANLink PSK 隐式认证（Tag 失败即断连）
ALPN/协议混淆攻击	ALPN 白名单，不匹配明确拒绝并留首条告警日志（可诊断）
管理面未授权访问	PBKDF2（12 万迭代）口令哈希 + 常数时间比较；Bearer 会话 8h；默认仅绑 127.0.0.1；敏感配置脱敏返回
口令破解侧信道	MessageDigest.isEqual 常数时间比较
资源耗尽/洪峰	预测性容量闸门：许可耗尽快速拒绝（先拒绝后建线程，拒绝日志限流防雪崩）；上游并发闸门与总超时
恶意/滥用域名	过滤引擎（important→白→重写→黑 评估序）+ IP 黑白名单/限流（真实 clientIp 贯穿）
DNS 放大滥用	不对外递归的代理语义（按需关闭 UDP、限制监听网段）
配置半写/篡改	整体原子校验 + 原子写回；校验失败整体不应用；域名分流禁止指回自身（防回环自杀）
上游返回污染	缓存只收 NOERROR/NXDOMAIN；SERVFAIL/REFUSED 等瞬态错误不缓存；可选 DNSSEC 上游
证书域名冒用	server.hostname 启动校验 SAN/CN 覆盖；服务端 SNI 不校验（任意 SNI 均下发当前证书——单证书部署的兼容取舍）

11.2 已知限制与最佳实践

已知限制：

1. 默认口令 admin/admin（示例配置明文）——公开知识，必须修改；

2. 管理面为单管理员模型，无多用户/细粒度 RBAC；REST API 鉴权为单一 Bearer token；

3. DoT 固定纯 TLS1.2 并禁用 ALPN（兼容取舍，放弃 TLS1.3）；legacyRsaCompat 开启后放宽至 TLS_RSA_*（仅限无法升级的老设备）；

4. DoQ（Kwik）仅支持 RSA 私钥（PKCS#8），EC/Ed25519 不支持；

5. 自签证书需客户端显式信任；SAN 必须覆盖客户端访问的主机名/IP；

6. YiANLink PSK 即信任根：泄露需全员换钥；协议无版本协商字段（v1）；

7. 查询日志含明文域名（隐私敏感，注意日志目录访问控制与保留时长）；

8. 栈内存为估算值（线程数×栈大小）；Windows 无系统负载（-1 → N/A）。

最佳实践：

- 修改默认口令；web.bindAddress 限内网/本机，公网暴露管理面前加反向代理 + 额外认证；
- 加密协议对外开放时使用正规 CA 证书并配置 server.hostname 校验；
- 生产环境为 YiANLink 生成独立高熵 PSK 并定期轮换（轮换时两端同步）；
- 开启防火墙仅放行所需端口；关注熔断与闸门拒绝计数（指标口径见 9.2）；
- 定期更新远程规则源；按 compliance 要求设置 retentionHours；
- 升级前备份 config 与 data；回滚演练纳入变更流程。

附录

附录 A. YiANLink 报文格式速查表

传输帧（TCP 6833，逐帧）

字段	偏移	长度	说明
Length	0	4B	大端无符号；= 12 + N + 16（其后全部长度）
IV	4	12B	每帧随机 nonce（严禁复用）
Ciphertext	16	N	AES-256-GCM 加密后的明文
Auth Tag	16+N	16B	GCM 认证标签

帧内明文

字段	偏移	长度	说明
Timestamp	0	8B	大端 Unix 毫秒；接收方校验偏差 ≤ 60s，超差静默丢帧
StreamId	8	4B	大端；响应必须以原 StreamId 回包
Payload	12	M	完整 DNS 报文（wire 格式，不含 TCP 长度前缀）

附录 B. 数学公式推导完整过程

B1. 指数项 exp(0.01·μ)：要求代价对 RTT 单调递增且边际递增（高时延惩罚非线性放大）。取指数族 μ → e^(0.01μ)，μ=0 → 1，μ+100ms → ×e。系数 0.01 设定"每 100ms 放大 e 倍"的尺度。

B2. 趋势项 2·max(0, v)：v>0 表示正在变慢，应惩罚；v<0（变快）不奖励——防止节点在低 RTT 基数上故意波动刷奖励。系数 2 使趋势项权重高于同等幅度的加速度项。

B3. 曲率项 1.5·｜a｜：二阶差分 a 为加速度，取绝对值使"恶化加速"与"震荡"（a 正负交替）均被惩罚；系数 1.5 次于趋势项。

B4. 离散项 σ/μ（变异系数）：σ 是绝对波动，除以 μ 得尺度无关的相对波动；μ→0 时定义域退化，工程定义为该项 0。

B5. Softmax 平移不变性证明：令 m=max logit_k，则 exp(logit_i − m)/Σexp(logit_k − m) = [exp(logit_i)·e^(−m)]/[e^(−m)·Σexp(logit_k)] = exp(logit_i)/Σexp(logit_k)，故先减 m 不改变结果，仅防下溢。

B6. SWRR 守恒证明：设一轮调用前 current 总和为 S。所有节点 +weight 后总和 S+Σw；选中 i* 后 −Σw，总和回到 S。故任意调用序列下总和恒等于初始 Σweight，长期选中频率严格收敛于 weight 比例。

B7. 泰勒预测与余项：设负载曲线 f 三阶可微，$f(t+h)=f(t)+f'(t)h+f''(t)/2\cdot h^2+R$。h=5：一阶项 5v，二阶项 12.5a。皮亚诺余项 $R=o(h^3)$——噪声主导（σ>3.0）时高阶项被噪声淹没，弃用二阶项（降阶）。拉格朗日余项 $R=f^{(3)}(\xi)/3!\cdot h^3$，取 $|f^{(3)}|\le M$ 得 $|R|\le M\cdot 125/6\approx 20.83\cdot M$；由 jerk 窗口最大值估计 M（再钳制上限 16.0），故安全阈值预留 R_bound=(M/6)·125，且不低于 MIN_SAFETY_THRESHOLD=1000。

B8. EMA 防抖：稳态时浮点误差可使 EMA=6.000000000000011，ceil 后目标 7 → 每秒 ±1 振荡。死区（|目标−当前|≤1 不调整）吸收取整尾差；回归要求"稳态采样下许可调整次数为 0"。

B9. Michaelis-Menten 饱和 mult(f)：需求：冷域名缩 TTL（mult(0)=2/3）、热点放大有上限（mult(∞)=1.75）、半饱和点 k 处为中值。形如 mult(f)=A+(B−A)·f/(f+k)（标准饱和动力学），f=0→A=2/3，f→∞→B=1.75，f=k→(A+B)/2≈1.2083。

B10. 频率衰减 f←f·e^(−λΔt)+1：连续化为一阶衰减 + 脉冲注入；半衰期 T½ 满足 e^(−λT½)=1/2 → λ=ln2/T½，T½=300s。性质：稳态 f* = 1/(1−e^(−λ))（每秒 1 次访问），热度自动收敛而非无界增长。

附录 C. CLI 命令 / API 速查表

CLI

```
# 启动（配置探测：参数 > ./config.yaml/yml/json > 内置默认）
java -jar dns-proxy-1.0.3.jar [配置文件路径]

# 构建 / 单模块测试 / 冒烟
powershell -File build.ps1 clean package
powershell -File build.ps1 -pl dns-proxy -am clean test
powershell -File smoke.ps1

# 验证 DNS
nslookup baidu.com 127.0.0.1
kdig +tls @dns.example.com -p 853 example.com          # DoT
kdig +https @dns.example.com example.com               # DoH
kdig +quic @dns.example.com -p 853 example.com         # DoQ

# 可选 JVM 参数
# （默认即频率感知缓存引擎，无需设置；以下仅用于回退/诊断）
-Ddns.cache.engine=caffeine    # 回退标准 Caffeine W-TinyLFU 引擎
-Ddns.tls.debug=true           # TLS 握手诊断（带 SNI）
```

REST API（鉴权：Authorization: Bearer <token>；login 与静态页除外）

```
POST /api/auth/login | POST /api/auth/logout | GET /api/auth/me
GET  /api/status | GET /api/connections | GET /api/metrics
GET  /api/queries?limit=N | GET /api/queries/detail | GET /api/stats?topN=N
GET/POST/PUT /api/rules?group= | GET/POST/PUT/DELETE /api/rules/sources | POST
/api/rules/sources/refresh
GET/PUT /api/config | GET /api/system/info
POST /api/system/restart | POST /api/system/shutdown
POST /api/maintenance/clear-query-log | POST /api/maintenance/clear-stats
POST /api/upstream/reset | GET /api/plugins
GET  /index.html（匿名管理页）
```

退出码：0 正常 / 1 启动失败 / 2 配置错误 / 3 端口占用。

附录 D. 版本更新日志（Changelog）

版本	日期	主要变更
0.1.0-SNAPSHOT	2026-10	12 reactor 骨架、插件扩展点（plugin-api 7 接口）
YiAn1.0.0-RC1	2026-10-25 ~ 10-26	阶段 3 核心层（dnsjava 编解码/流水线）、阶段 4 上游转发（四协议传输+熔断）、阶段 5 缓存（Caffeine）、阶段 6 过滤引擎（AdGuard 风格）、阶段 7 分层配置、阶段 8 查询日志（JSONL+统计）、阶段 9 Web 管理 REST、阶段 10 启动入口（12 步装配/退出码）、fat jar 打包
1.0.1_RC1	2026-10-27	阶段 11 安全传输：DoT/DoH/DoQ 服务端（Netty/Kwik/零 native）、server.hostname 校验
1.0.1_RC2	2026-10-28 ~ 10-30	Vue3 管理控制台前端、登录鉴权拆分、移除 Mock、DoH HTTP/1.1 支持、ALPN 不匹配明确拒绝、TLS 加固（完整链/会话缓存/握手超时）、DoT 固定 TLS1.2、TLS_RSA 兼容开关、上游查询策略（负载均衡/并行请求）与热更新
1.0.2-RC1	2026-10-31 ~ 11-05	系统状态监控板块（/api/system/info）、系统服务重启/关闭、管控规则页四列表+远程规则源、SNI/SslHandler 加固、YiANLink 接入闭环（传输+前端+监控契约）、频率感知动态 TTL 缓存
1.0.3-RC2	2026-11-06 ~ 11-16	预测性容量控制（泰勒余项闸门）接入数据面、趋势预测动态权重调度（J+Softmax+SWRR，三策略定名）、手动权重废弃、UpstreamStateRegistry 热更新状态复用、发行版代号 Marhaba、Split DNS 域名分流：上游地址兼容完整 URL（https://doh.pub/dns-query、tls://dot.pub）与裸域名简写（UpstreamAddressParser，网关

		与域名分流共用）；CNAME 重写联动解析（应答附带目标 A/AAAA，客户端无需二次查询）；上游指标语义修正（queries 尝试计数、failures、latency 仅成功记录、无样本显示"无成功应答"/"<1 ms"）；频率感知缓存引擎默认化切换；管控规则页重写表格多行串行修复与分页焦点保持；规则加载链路流式重构（50 万条 ≤150MB）
1.0.3	2026-11-16 ~ 11-17	正式版：全量回归（全模块单测 + smoke 9 项断言）通过；文档与 README 双语口径同步（默认缓存引擎、上游 URL 语法、指标语义）

各版本详细行为以 README 与源码为准；构建层版本号恒为纯语义版本。

附录 E. 参考资料

类别	资料
DNS 基础	RFC 1034 / RFC 1035（DNS 概念与实现）、RFC 7766（DNS over TCP）
加密 DNS	RFC 7858（DoT）、RFC 8484（DoH）、RFC 9250（DoQ）
TLS/QUIC	RFC 8446（TLS 1.3）、RFC 5246（TLS 1.2）、RFC 9000/9001（QUIC 传输/TLS 集成）、RFC 6125（证书身份校验）
算法	Nginx Smooth Weighted Round-Robin；Caffeine W-TinyLFU（Einbuch et al., 2017）；Michaelis-Menten 饱和动力学；泰勒定理（皮亚诺/拉格朗日余项）
工程实现	dnsjava 3.6.x 文档；Netty 4.1 用户指南；Kwik（纯 Java QUIC）项目文档；BouncyCastle PEM 处理；Java 21 虚拟线程（JEP 444）
许可证	Mulan PSL v2（木兰宽松许可证）