

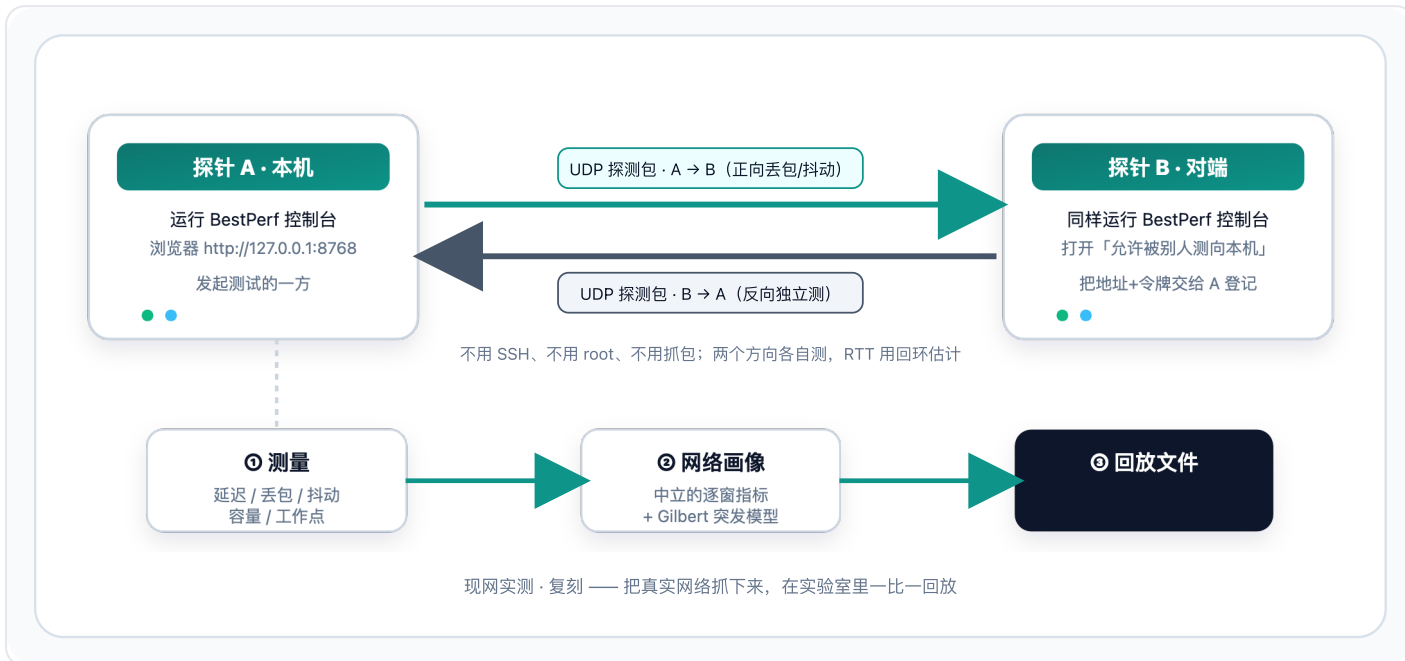
BestPerf 使用手册 · 封面

产品名称：BestPerf（网准通 BestPerf） 软件版本：v0.1.0 日期：2026 年 6 月 标语：现网实测 · 复刻

一句话说清楚 BestPerf 是做什么的

关键理解：在现网（真实的办公室、机房、跨城专线.....）的两台普通电脑上，各开一个 BestPerf 控制台，让它们用 UDP 互相测一阵子网络，得出这条链路的「网络画像」（延迟、丢包、抖动有多大），再导出一个「回放文件」。把这个回放文件导入实验室里的 WAN 仿真器，实验室就能复刻出和现网一模一样的网络坏脾气。这就是标语「现网实测 · 复刻」的全部含义。

它的好处是：不用 SSH、不用 root（管理员）、不用抓包。你只要会双击安装、会用浏览器，就能完成一次现网测量。



上图：两台机器各跑一个 BestPerf 控制台（即「探针」），用 UDP 互相打探针包，攒够样本后生成「网络画像」，再导出「回放文件」交给实验室的 WAN 仿真器复刻现网。

文档信息

下面这张表是本手册和软件的基本参数。装机、排查问题时可以随时回来对照。

项目	内容
产品名称	BestPerf (网准通 BestPerf)
软件版本	v0.1.0
支持系统	Windows 10 及以上 (x64)、macOS 11 及以上 (Intel 与 Apple 芯片通用)、Linux x86_64
浏览器要求	Chrome 90+ / Edge 90+ / Firefox 90+
控制台端口	TCP 8768 (本地浏览器访问 <code>http://127.0.0.1:8768</code>)
探测端口	UDP 39700–39899 (探测); TCP 39800–39899 (TCP 体感) (两台机器之间互测用)

提示：控制台界面是深色主题，在手机或平板的浏览器里也能打开，但本手册以电脑端 (Chrome / Edge / Firefox) 为准。

注意：表里的端口很重要。要让别人测向你这台机器，得放行控制台 TCP 8768、探测 UDP 39700–39899、TCP 体感 TCP 39800–39899；只是你去测别人，则不需要为自己开任何端口。在 Windows 上不用手动逐个开端口，双击桌面上的「2 开放网络权限 BestPerf」即可 (一次 UAC 确认，自动放行 UDP 39700–39899 + TCP 39800–39899 + TCP 8768)。各系统的具体放行方法见后面章节。

三步看懂怎么用 (先有个整体印象)

正式章节里会手把手地教，这里先给你一张「全景图」，让你知道一共就这么三步：

- **第 1 步：自检本机** —— 在控制台「本机信息」页确认本机地址、令牌 (Token)、UDP 探测端口都正常。
- **第 2 步：登记对端** —— 在「探针」页填上对方机器的控制台地址和令牌，把要测的那台机器登记进来。
- **第 3 步：发起测试** —— 在「发起测试」页选好「发起方 / 对端」和测量类型，点亮那颗发光的「开始」按钮，等结果出来，到「结果·样本库」页下载回放文件。

关键理解：测量分四种类型 —— **轻量测量** (量延迟 / 丢包 / 抖动)、**负载画像** (逐档加压找容量上限)、**TCP 体感** (量应用实际能跑多快)、**本机性能自检** (只测本机、无需对端，用来分清瓶颈在网络还是在你这台机器)。后面章节会逐一讲清楚每种该在什么时候用。

文档结构

本手册一共八章，从「这是什么」一路讲到「出问题怎么办」。建议第一次使用的读者按顺序读第一章到第六章，之后按需查阅。

- **第一章 产品概述** —— BestPerf 是什么、解决什么问题、两探针测量与「现网实测·复刻」的原理、四种测量类型一览。
- **第二章 安装与启动 · Windows** —— Windows 系统包的下载、解压、安装与启动，启动器体检 (Doctor)，以及双击「2 开放网络权限 BestPerf」放行被测网络权限。
- **第三章 安装与启动 · macOS** —— macOS (Intel 与 Apple 芯片通用) 系统包的下载、安装与启动，首次运行的安全提示处理与放行被测网络权限。
- **第四章 安装与启动 · Linux** —— Linux x86_64 系统包的下载、解压、启动与体检，以及放行被测网络权限。
- **第五章 打开测试界面与控制台总览** —— 用浏览器打开 `http://127.0.0.1:8768`，认识控制台各个页面 (本机信息、探针、发起测试、结果·样本库) 的整体布局与导航。
- **第六章 手把手：第一次测试** —— 从自检本机、登记对端，到发起一次真实测量、看懂结果、导出回放文件，跟着图一步步走完整流程。
- **第七章 功能详解 (每个功能做什么·注意事项·测试效果)** —— 逐个功能讲清楚它做什么、使用时要注意什么、测出来是什么效果，覆盖四种测量类型与各项参数。
- **第八章 常见问题 (FAQ)** —— 装不上、连不上对端、被测端口不通、结果看不懂等高频问题的排查办法。

提示：手册里所有截图、配图都放在各章节旁边，图片下方都有一句中文说明，照着图一步步操作即可。

本手册描述的所有界面按钮、字段、开关均以 **BestPerf v0.1.0** 实际控制台为准。若你的软件版本不同，个别名称可能略有差异，请以软件实际显示为准。

第一章 产品概述

1.1 产品简介

BestPerf (网准通 BestPerf) 是一款跨平台的网络质量探测工具。它的标语是「现网实测·复刻」——先把真实网的质量实测下来，再在实验室里复刻出来。

它要解决的问题是：你的应用（视频、语音、文件传输、设备）在客户现场、在跨国专线、在弱网环境下时好时坏，但你在实验室里很难重现那种「现场感」。等真出了问题，又没法把现场的网络搬回实验室。BestPerf 就是来填这个缺口的——它让你**不动现网、不装重型工具**，就能把现网的延迟、丢包、抖动这些「脾气」量出来，存成一份文件，带回实验室一键复刻。

BestPerf 采用**两探针模型**：在现网两端的两台机器上，各跑一个 BestPerf 控制台（也就是一个「探针」）。两个探针用 UDP 报文互相对测，量出这条链路真实的质量。测完之后，BestPerf 把样本汇成一份中立的**网络画像 (network profile)**，再导出一份**回放文件 (replay scenario)**——把这份文件导入 WAN 仿真器，实验室里的链路就会表现得和现网一模一样。

关键理解：BestPerf 本身**不制造网络损伤**，它只**测量并记录现网真实的质量**；制造损伤、复刻现网的是 WAN 仿真器。BestPerf 是「现网这一端的耳朵和录音机」，仿真器是「实验室这一端的复读机」。

提示：整个过程**不用 SSH、不用 root 管理员、不用抓包 (pcap)、不用 tc/netem**。每台机器只需在本地跑一个小小的网页控制台，浏览器打开就能用。

1.2 核心能力

四种测量类型

BestPerf 在「发起测试」页提供四种测量类型，一句话各是什么：

测量类型	一句话说明	是否需要 对端
轻量测量 (light)	秒级小流量探针，量 RTT 延迟、双向丢包、抖动、乱序、重复、微中断	需要对端
负载画像 (load)	逐档加压找拐点 / 容量上限，在拐点以内的工作点测延迟和丢包，拟合 Gilbert-Elliott 突发丢包模型	需要对端
TCP 体感 (tcp)	按 RFC 6349，分别测上行 / 下行 goodput (应用真正拿到的吞吐) 加上应用级 RTT	需要对端
本机性能自检 (selfcheck)	本机环回 (loopback) 加压，得出这台机器自己的 UDP 处理上限和负载下的延迟	不需要对端

提示：本机性能自检 是用来「分清瓶颈在网络还是在本机」的。如果一条链路慢，先做一次本机自检，确认不是自己这台机器的处理能力到了顶，再去怀疑网络。

关键特性

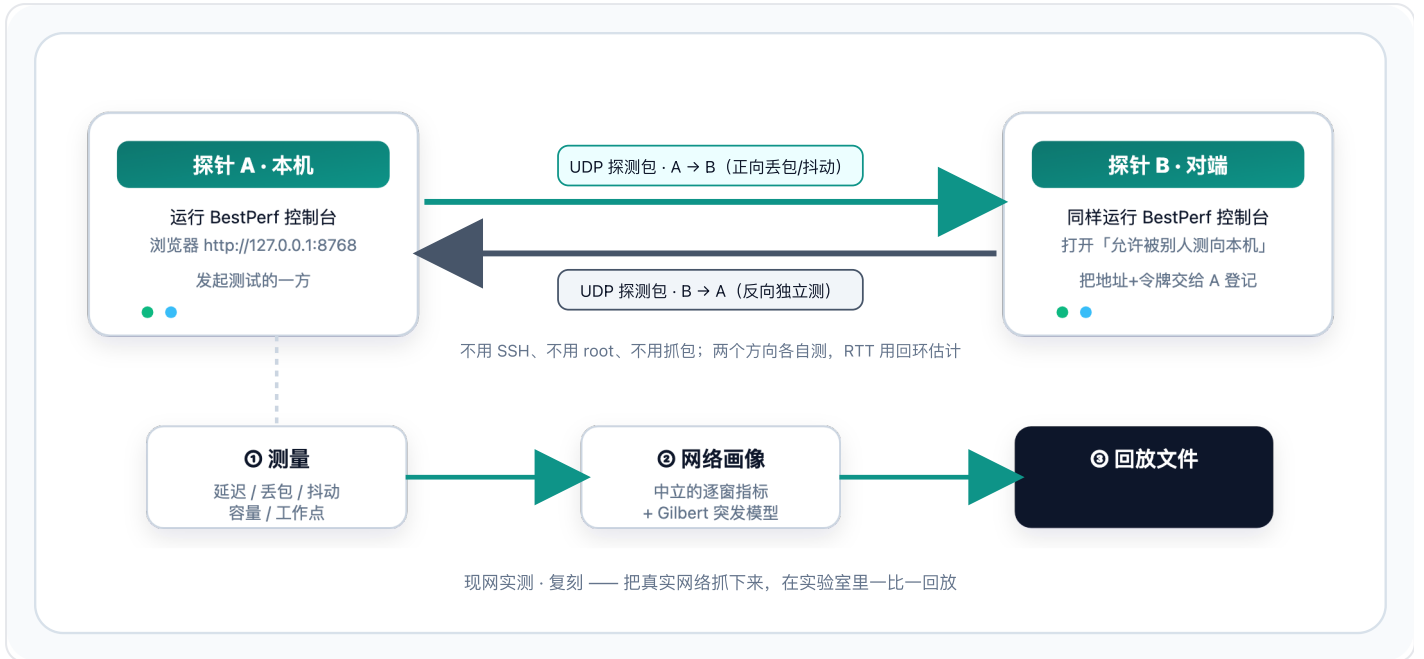
- **双向独立测** — A→B 和 B→A 两个方向各自独立测量、独立出结果。现网很多时候是不对称的（上行好、下行差），BestPerf 不会把两个方向混成一个数字。
- **不用 SSH / root / 抓包** — 纯 UDP 探针 + 一个本地网页控制台。不碰对方系统、不需要管理员权限、不抓任何业务报文。
- **网络画像 + 回放文件导出** — 每次测量都产出一份中立的网络画像，并能一键下载回放文件，直接喂给 WAN 仿真器复刻。
- **诚实测量** — 丢包结果带 **95% 置信上界**（一个都没丢时用「三倍法则」给出上界），并带样本量可靠性标记。也就是说，一次「0% 丢包」的读数不会被当成「绝对不丢包」的结论，而是诚实地告诉你「在这么多样本下，丢包率最多不超过某个值」。
- **区分本机抖动和网络抖动** — 画像构建时会用 `scheduler_lag`（本机调度滞后）把「本机自己卡了一下」和「网络真的抖了」分开，不让本机的卡顿冤枉网络。

关键理解：BestPerf 只用单调时钟，**不声称能测单向延迟**（两台机器的钟没法精确对齐）。它把半程延迟作为「估计值」导出，并标注置信度——这也是「诚实测量」的一部分：测不准的东西就不假装测准。

1.3 工作原理 / 架构

两探针 UDP 互测 → 网络画像 → 回放文件 → WAN 仿真器

BestPerf 的核心链路是一条「从现网到实验室」的流水线：



上图：现网两端的探针 A 和探针 B 用 UDP 互相对测（两个方向都测），各自的样本汇成一份中立的网络画像，再导出回放文件，导入实验室的 WAN 仿真器即可复刻现网。

按步骤拆解：

- **步骤 1 — 两探针 UDP 互测：** 现网两端各跑一个 BestPerf 控制台（探针）。任意一端都可以发起测试。探针之间用 UDP 报文互发：回波探针量 RTT，单向有序流探针量方向性的丢包 / 乱序 / 重复 / 抖动，突发探针量微中断。两个方向同时都测。
- **步骤 2 — 汇成网络画像：** 测量产生的原始样本，被汇成一份中立的网络画像。这份画像只描述「这条网络长什么样」，不绑定任何具体仿真器。
- **步骤 3 — 导出回放文件：** 从画像导出回放文件。导出时会按目标仿真器实际支持的能力来映射——支持不了的会优雅降级并记录警告，绝不悄悄丢掉。
- **步骤 4 — 导入仿真器复刻：** 把回放文件导入实验室的 WAN 仿真器，实验室链路就会按现网的延迟和丢包表现出来，现网被「复刻」到了实验室。

关键理解： 探针之间的协调走 HTTP（控制台对控制台），不走 SSH。发起方只要在「探针」页登记了对端的控制台地址和令牌，就能驱动整次双机测量。前提是对端必须打开「允许被别人测向本机」开关，否则它无法被测向，两探针模型就跑不起来（具体操作见后续章节）。

在「结果·样本库」详情页，回放文件有两个下载按钮，区别在于格式是「专用可直接导入」还是「通用中立」：

下载按钮	适用对象	说明
↓ 损伤仪场景 (可直接导入)	支持原生格式的 损伤仪 / 仿真器	WAN 仿真器原生格式，导进去就能复刻这次测到的网络，无需转换 (文件名 <code>device_scenario.json</code>)
↓ 回放场景 (通用格式)	不支持上面专用 格式的仿真器	通用 / 中立格式，给载入不了原生格式的仿真器用 (文件名 <code>wan_emulator_scenario.json</code>)

四种测量，怎么挑



上图：四种测量类型的定位——轻量测量看「这条网平时怎么样」，负载画像看「压上去能扛多少、到顶时怎么样」，TCP 体感看「应用真正能跑多快」，本机性能自检看「瓶颈是不是我自己这台机器」。

提示：日常先做 **轻量测量** 拿到网络画像和回放文件就够了。需要知道「能扛多大带宽、到拐点时什么表现」时再加做 **负载画像**；想知道「应用真正能跑多快」时做 **TCP 体感**；怀疑是本机性能不够时做 **本机性能自检**。其中 **负载画像** 还会比轻量测量多一个「方向」选择，**本机性能自检** 不需要对端、会隐藏对端和方向。

1.4 典型应用场景

场景一：现网质量验证

应用在客户现场时好时坏，想拿到客观证据。在现场两端各跑一个探针，做一次轻量测量，得到带 95% 置信上界的 RTT / 丢包 / 抖动报告——把「感觉网络不好」变成「这条链路 p95 RTT 是多少、丢包上界是多少」的硬数据。

场景二：跨国 / 弱网复刻

跨国专线、卫星、4G/5G 弱网这类难复现的链路，先在现网实测出网络画像，导出回放文件带回实验室，用 WAN 仿真器复刻——之后所有调试、对比、回归都在实验室进行，不必反复占用现网或飞到现场。

场景三：SD-WAN / 设备测试

要验证 SD-WAN、加速器、网关等设备在真实劣化链路下的行为。用 BestPerf 把现网的脾气录下来，在实验室用仿真器复刻成稳定可重复的链路，反复测设备的路径切换、重传、恢复策略，每次条件都一致、可对比。

场景四：上线前回归

新版本上线前，想确认它在「已知的几条现网链路」下没有退化。把历次现网测出的回放文件存成一组基准场景，每次发版都在这些场景下跑一遍——网络条件固定，差异就只来自你的代码，回归一目了然。

1.5 术语表

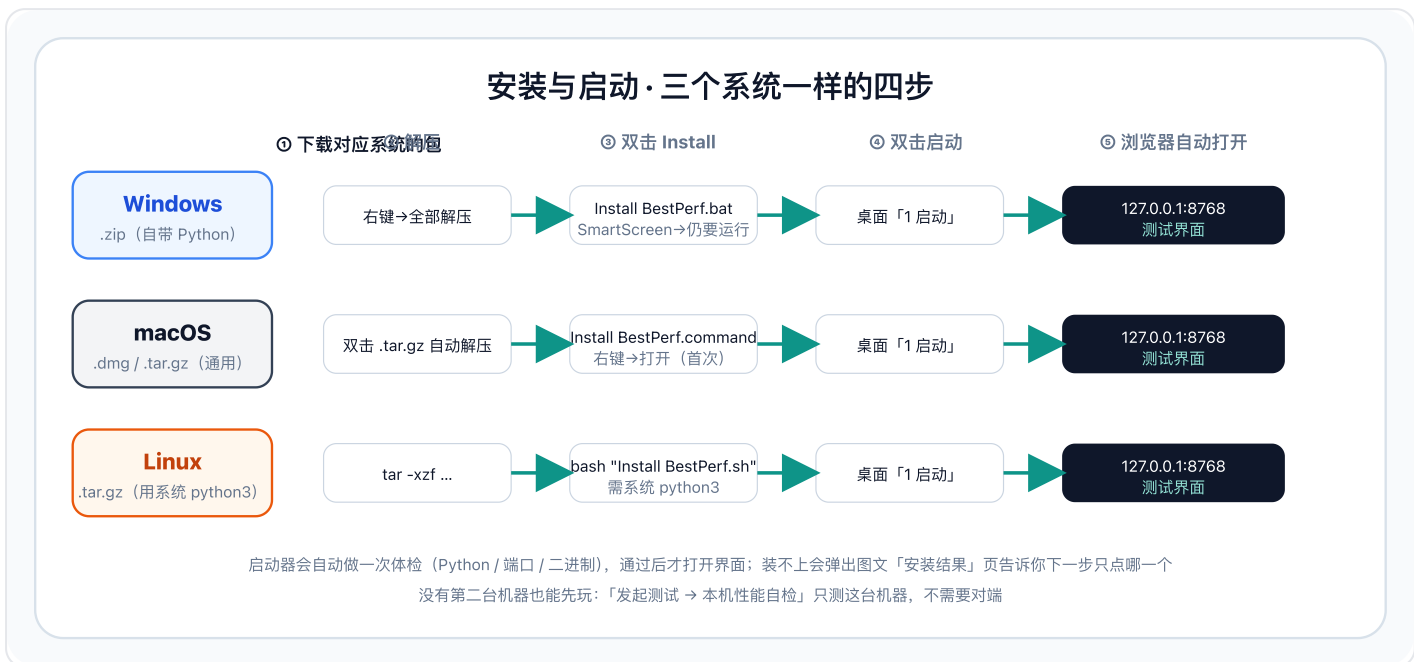
术语	说明
探针 (Probe)	一台跑着 BestPerf 控制台的机器。两个探针用 UDP 互测，构成一次测量
本机	你正在操作的这台机器、它本地跑的这个控制台。「本机信息」页展示的就是它自己的连接信息
对端 (Peer)	测量的另一端那台探针。在「探针」页登记它的控制台地址和令牌后，才能对它发起测试
网络画像 (network profile)	由测量样本汇成的、中立的网络质量描述，不绑定任何具体仿真器，是导出回放文件的中间产物
回放文件 (replay scenario)	从网络画像导出的文件，导入 WAN 仿真器即可在实验室复刻现网。详情页提供两个下载按钮：「损伤仪场景(可直接导入)」是原生格式、可直接导入（文件名 <code>device_scenario.json</code> ），「回放场景(通用格式)」是通用 / 中立格式、给不支持原生格式的仿真器用（文件名 <code>wan_emulator_scenario.json</code> ）
工作点 (working point)	负载画像里，在拐点以内选定的一个稳定加压档位，在这个点上测延迟和丢包，代表链路「正常受载时」的表现
拐点 (knee)	负载画像逐档加压时，链路开始饱和的转折点——先是延迟抬头，接着交付速率到平台（即容量上限），再往上才是丢包上升
Gilbert-Elliott	一种双状态突发丢包模型。负载画像会把测到的丢包拟合成这个模型，用来刻画「成串地丢」而非「均匀地丢」的突发丢包特性
令牌 (Token)	控制台的访问令牌。要测向某个对端，必须拿到对端的令牌并填进「探针」页；这是探针之间互信的凭据

下一章预告：第二章将手把手带你在 Windows / macOS / Linux 上安装并启动 BestPerf 控制台，确认浏览器能打开 <http://127.0.0.1:8768>。

第二章 安装与启动 · Windows

本章手把手带您在 Windows 电脑上装好 BestPerf，并第一次打开它的操作界面。全程只需「下载 → 解压 → 双击安装 → 双击启动」四步，不用安装 Python、不用管理员账号、不用敲命令。照着做，大约 5 分钟就能看到界面。

关键理解： BestPerf 的每台机器都在本地跑一个「控制台」（也就是操作界面）。您在本机装好后，浏览器会自动打开 `http://127.0.0.1:8768`，这就是您这台机器的控制台。要做两台机器互测，对方也要照本章装一遍。



上图：Windows 安装四步——下载、解压、双击安装、双击启动，界面自动打开。

2.1 开始之前：准备好这些

系统要求

项目	要求	说明
操作系统	Windows 10 或 Windows 11, 64 位	绝大多数现代电脑都满足
内存	至少 4 GB	用于跑测量
磁盘空间	预留约 200 MB 以上的空闲空间	用来解压和安装
Python	无需另装	Windows 包自带 Python, 这是最省事的一个版本
账号权限	普通用户即可	安装与发起测试都不需要管理员; 只有「要被别人测向本机」时才需要点一次 UAC (见 2.3)

提示：Windows 包是三个系统里最省心的——它把 Python 一起打包了，您什么都不用预装。只要电脑能开机上网、能解压 zip，就能装。

我该下哪个包？

BestPerf 为每个系统准备了对应的安装包。请按下表对号入座：

您的电脑	下载这个包
Windows 10 / 11 (本章)	<code>BestPerf-windows-x64.zip</code>
macOS (Intel 或 Apple 芯片)	<code>BestPerf-macos-universal.tar.gz</code> (见第三章)
Linux	<code>BestPerf-linux-x64.tar.gz</code> (见第四章)

本章只讲 Windows，请下载 `BestPerf-windows-x64.zip`。如果您用的是 Mac 或 Linux，请跳到第三章或第四章。

2.2 一步步装好并启动

整个过程就四步：**解压** → **双击 Install** → **处理一次蓝屏提示** → **双击桌面「1 启动」**。请严格按顺序做。

步骤 1：解压压缩包（这一步最关键）

下载到的是一个 `BestPerf-windows-x64.zip` 压缩包。

1. 在文件资源管理器里找到这个 `.zip` 文件
2. 右键点它 → 选择「全部解压」(Extract All)
3. 解压完成后，会得到一个文件夹；**进入这个文件夹再继续操作**

注意（最常见的错误）：一定要先「全部解压」，再进解压出来的文件夹操作。千万不要在压缩包预览窗口里直接双击运行里面的程序——那样会因为找不到配套文件而失败。看到的是「压缩包窗口」还是「真正的文件夹」，是这一步成败的关键。

怎么判断这步成功了：你已经身处一个真正的文件夹（地址栏里不再有 `.zip`），里面能看到一个名为 `Install BestPerf` 的安装文件。

提示：解压出来的文件夹里还有一个「0 先看我 START HERE」文件，双击它能看到同样的三步图文指引，可以随时回看。

步骤 2：双击「Install BestPerf.bat」开始安装

在解压出来的文件夹里：

1. 双击 `Install BestPerf.bat`
2. 它会自动把程序装好，并在桌面生成启动图标
3. 这一步**不需要管理员权限**，整个过程自动完成

安装位置：BestPerf 会装到 `%LOCALAPPDATA%\BestPerf`（也就是当前用户目录下，不影响系统其他用户）。

步骤 3：遇到蓝色 SmartScreen 提示时这样点

第一次运行时，Windows 可能会弹出一个蓝色全屏提示，写着「Windows 已保护你的电脑 / Windows protected your PC」。这是 SmartScreen 在拦截没有数字签名的内部程序，属于正常现象。

1. 点蓝屏左下角的「更多信息 / More info」
2. 再点出现的「仍要运行 / Run anyway」

注意：BestPerf 是内部工具、没有购买数字签名，所以才被 SmartScreen 拦一下，可以放心运行。如果蓝屏里看不到「仍要运行」按钮，是因为你还没点「更多信息」——先点它，按钮才会出现。

安装这一步只确认文件是否就位（装到哪、桌面入口建好、引导脚本与界面资源齐全），完成后会弹出一张图文「安装结果」页——绿色的「安装成功」就代表装好了（详见 2.4）。真正的「体检 (Doctor)」——检查 Python、端口、二进制是否都就绪——是在步骤 4 双击「1 启动 BestPerf」时才进行的，通过后才会自动打开浏览器。

步骤 4：双击桌面「1 启动 BestPerf.cmd」打开界面

安装成功后，桌面上会出现一个图标「1 启动 BestPerf.cmd」。

1. 双击桌面「1 启动 BestPerf」
2. 它会再做一次快速体检，通过后自动打开浏览器
3. 浏览器地址栏显示 `http://127.0.0.1:8768`，看到 BestPerf 深色主题的操作界面

关键理解：127.0.0.1 是「本机自己」的意思。这个地址只有你这台电脑能打开，是你本机的控制台。别人要测向你时用的是另一个地址（你的局域网 IP，见 2.3 和后续章节）。

怎么判断这步成功了：浏览器出现 BestPerf 界面，默认停在顶部第一个标签页「本机信息」，能看到「本机连接信息」卡片和「自检本机 → 登记对端 → 发起测试」三步引导。到这里，安装就完成了。

提示：以后再用，只需双击桌面「1 启动 BestPerf」即可——不必再解压、不必再装。

2.3 让别人测向本机（可选）

这一步是可选的，先判断你要不要做：

- 只发起测试（由你这台机器去测别人）：不需要做这一步，跳过即可。
- 要被别人测向本机（你这台机器当「对端 / 应答方」，让另一台 BestPerf 来测你）：才需要做下面三件事。

关键理解：两台机器互测时，发起测试的那台不用开任何权限；被测的那台才需要放行入站端口、并打开界面里的开关。如果你既要发起、又要被测，那就把下面三步做一遍。

步骤 1：放行防火墙入站端口

双击桌面「2 开放网络权限 BestPerf」。会弹出一 Windows 的 UAC（用户账户控制）窗口，点「是 / Yes」即可。它会自动以管理员身份放行以下入站端口：

端口	协议	用途
39700–39899	UDP	网络探测（测量报文走这里）
39800–39899	TCP	TCP 体感测量
8768	TCP	控制台（让对端能协调这台机器）

提示：这是整个安装流程中唯一需要点 UAC 的地方，且只需点一次「是」。你不用懂防火墙规则——脚本会自动加好。

步骤 2：在界面里打开开关

回到 BestPerf 界面的「本机信息」页，在「本机连接信息」卡片里，打开「允许被别人测向本机」开关（默认是关的）。

步骤 3：把地址和令牌发给对方

同一张「本机连接信息」卡片里，会显示控制台地址（`http://本机IP:8768`）和访问令牌（Token）。把这两样发给要测你的人——对方在他的「探针」页登记你时要用到。

注意：「允许被别人测向本机」默认关闭。如果你只发起测试、不打算被别人测，请保持它关着，更安全。

2.4 怎么判断装成功了

看「安装结果」页

安装完成后弹出的图文「安装结果」页，是判断成败最直接的依据。



上图：绿色的「安装成功」页。它会告诉你装到哪里了，以及下一步——只需点桌面「1 启动 BestPerf」。

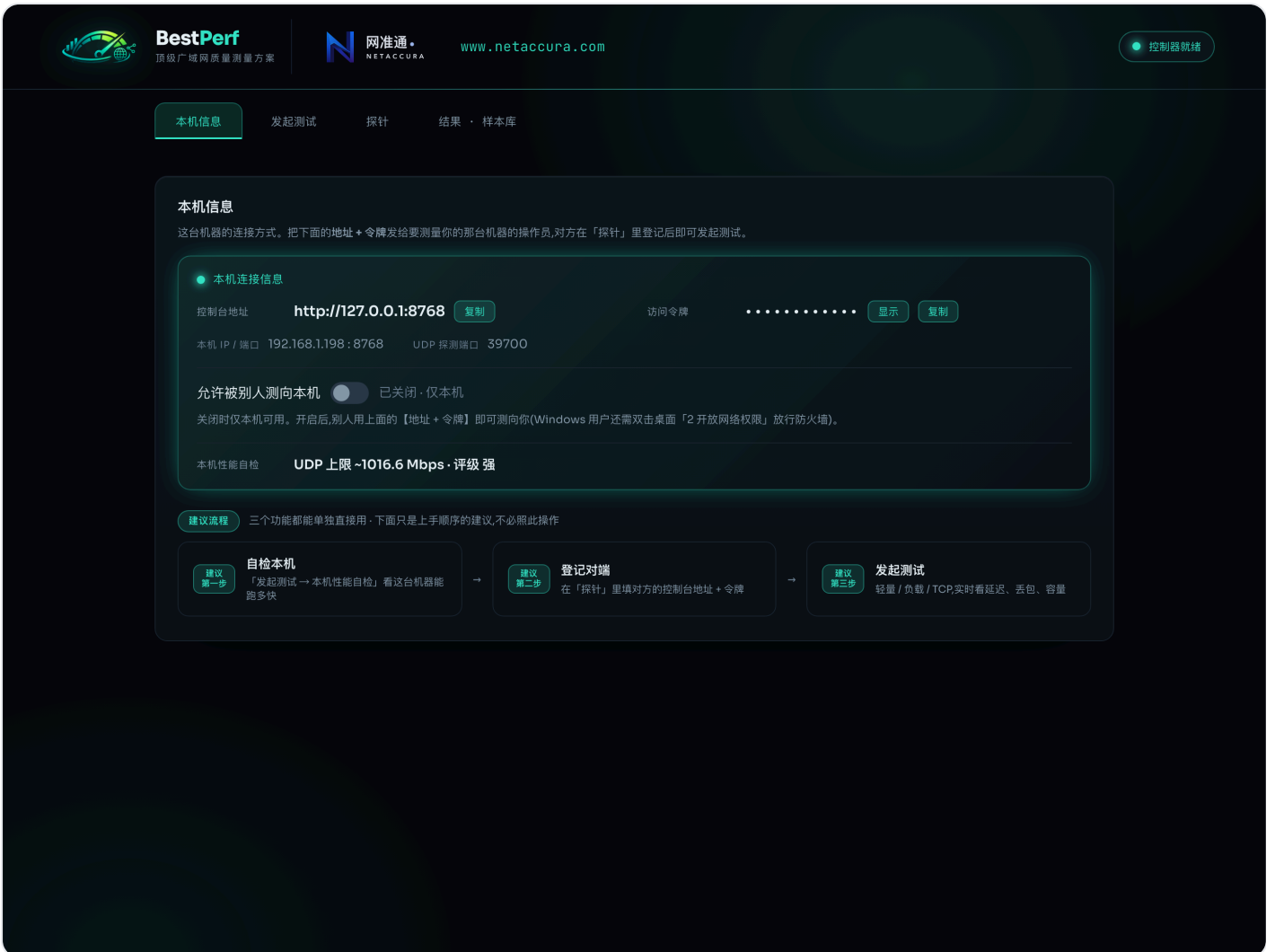
- 绿色「安装成功」= 装好了，照页面提示双击桌面「1 启动」即可。
- 如果这一页变成红色，说明没装成功——红色页面用一句话给出唯一的下一步动作，照着做即可（也可看 2.5）。

看桌面图标

安装成功后，桌面会出现启动入口「1 启动 BestPerf.cmd」；如果你做了 2.3，还会多一个「2 开放网络权限 BestPerf」。

看浏览器界面（最终确认）

双击「1 启动」后，浏览器自动打开并显示 BestPerf 的「本机信息」页，就代表一切正常、可以开始用了。



上图: BestPerf 主界面的「本机信息」页。看到顶部四个标签 (本机信息 / 发起测试 / 探针 / 结果·样本库) 和「本机连接信息」卡片, 就说明装好并启动成功了。

关键理解: 能看到这张「本机信息」页 = 安装 + 启动都成功。接下来怎么做第一次测量, 请看后续章节。

2.5 装不上 / 打不开怎么办

如果中途卡住, 先按下面快速自查:

现象	先试这样做
双击 Install 没反应或很快闪退	确认你是在解压出来的文件夹里双击的，而不是在压缩包窗口里（回到 2.2 步骤 1）
蓝色 SmartScreen 拦住、找不到「仍要运行」	先点蓝屏里的「更多信息 / More info」，按钮才会出现（见 2.2 步骤 3）
「安装结果」页是红色	按红色页上写的那一句下一步动作做
桌面没出现「1 启动」图标	重新双击一次 <code>Install BestPerf.bat</code> ，看「安装结果」页
双击「1 启动」浏览器没自动打开	手动打开浏览器，地址栏输入 <code>http://127.0.0.1:8768</code> 试试
别人测不到你这台机器	确认 2.3 三步都做了：放行端口、打开「允许被别人测向本机」开关、把地址+令牌给了对方

如果还是不行，需要找支持人员协助时：把解压文件夹里的 `install_summary.latest.txt` 文件（这是安装过程的体检记录）发给支持人员，能帮他们快速定位问题。

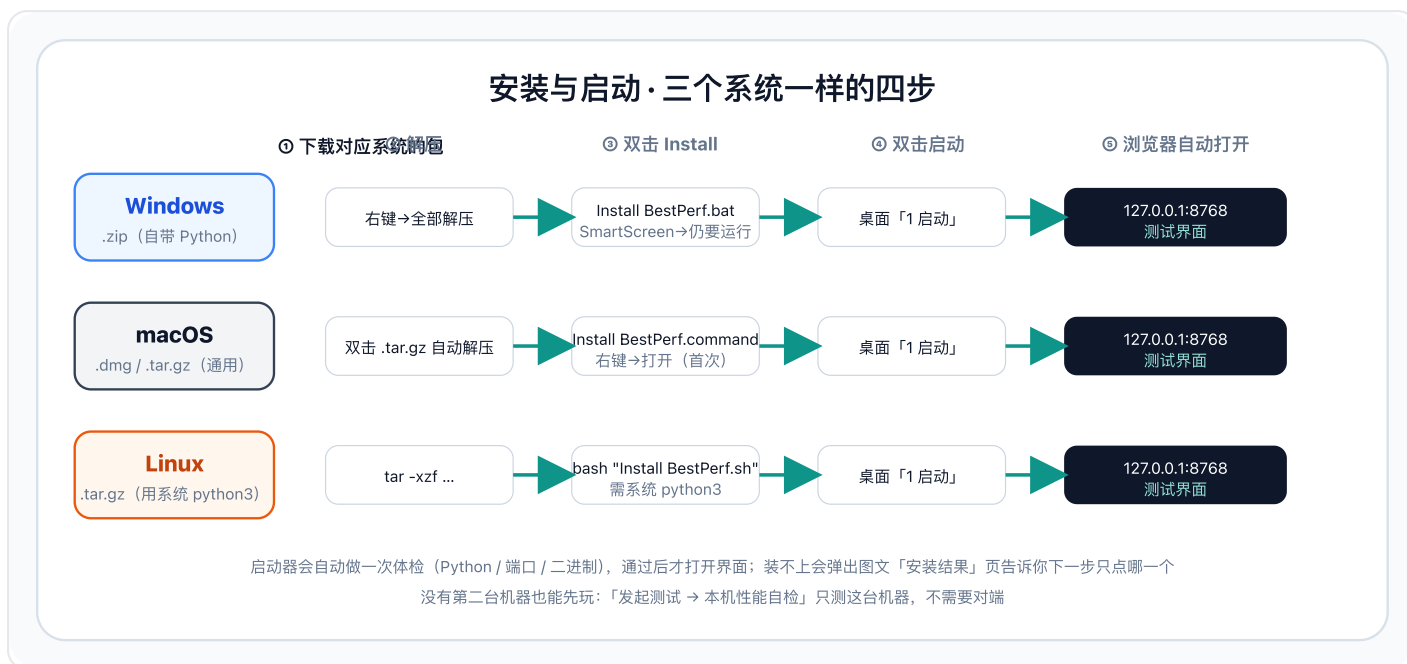
提示：更详细的排查（端口被占用、浏览器打不开、对端连不上等）请看本手册的「常见问题（FAQ）」一章。

第三章 安装与启动 · macOS

本章手把手带您在 Mac 上装好 BestPerf 控制台，并第一次打开界面。全程不需要 SSH、不需要管理员 (root)、也不需要懂命令行——绝大多数操作都是「双击」。跟着步骤走即可。

关键理解： BestPerf 的工作方式是——每台机器本地各跑一个「控制台」（也叫探针），用浏览器打开 `http://127.0.0.1:8768` 来操作。本章就是把您这台 Mac 上的控制台装好、跑起来。装好后，两台机器互相登记，就能用 UDP 互测网络、得出「网络画像」。

提示： 控制台默认端口是 8768；若 8768 被占用，会自动改用其它空闲端口，以界面 / 终端实际显示为准。



上图：所有系统通用的安装四步——下载 → 解压 → 安装 → 启动。macOS 的具体操作见下文。

3.1 系统要求

开始之前，先确认您的 Mac 满足下面的条件。

项目	要求	说明
操作系统	macOS 11 (Big Sur) 及以上	较新的 macOS 都满足
芯片	Apple 芯片 (M 系列) 或 Intel 均可	同一个通用包, 不用区分
Python	系统自带 <code>python3</code> , 版本 3.9 或更高	Mac 一般已自带; 没有时一行命令补上
安装位置	<code>~/Applications/BestPerf</code>	装在当前用户目录下, 无需管理员密码

关键理解: BestPerf 的 macOS 包是通用 (**universal**) 包。无论您的 Mac 是 Apple 芯片 (M1/M2/M3 等) 还是老的 Intel 芯片, 下载的是同一个文件, 不用纠结选哪个版本。

怎么判断自己装没装 python3

BestPerf 在 Mac 上使用系统自带的 `python3`。多数 Mac 已经带了, 但保险起见可以自己确认一下:

步骤 1: 打开「终端」。

- 在「访达 / Finder」里依次进入「应用程序 → 实用工具 → 终端」;
- 或按 `Command + 空格` 打开聚焦搜索, 输入「终端」回车。

步骤 2: 在终端里输入下面这行, 按回车:

```
python3 --version
```

步骤 3: 看结果。

- 如果显示类似 `Python 3.11.6 (3.9 或更高)`, 说明已就绪, 可以直接进入 3.2 安装。
- 如果提示找不到命令, 或弹出「要安装命令行开发者工具吗」之类的窗口, 说明还没装。运行下面这行, 按提示点「安装」, 装完再回来:

```
xcode-select --install
```

提示: `xcode-select --install` 是苹果官方的开发者命令行工具, 里面就带了 `python3`, 装一次以后都不用再管。整个过程点几下「同意 / 安装」即可, 不影响您电脑里的其它东西。

注意：就算您不确定有没有 python3，也可以**直接往下装**——3.2 的启动器自带「体检 (Doctor)」，会自动检查 python3 在不在。缺 python3 时，安装结果页 / 启动器会提示「安装 Python 3.9 或更新版本」；在 Mac 上推荐用 `xcode-select --install` 来装 python3（这是我们的建议，不是界面原话），不会让您卡在猜。

3.2 一步步安装与启动

下面是 macOS 上从下载到打开界面的完整流程。每一步都标了「怎么判断这步成功了」。

步骤 1：下载安装包

下载 macOS 版安装包，文件名是其中之一：

- `BestPerf-macos-universal.tar.gz`（压缩包，最常见）
- 或 `BestPerf-macos-universal.dmg`（磁盘映像，用法见 3.3）

提示：两种格式装出来的东西完全一样，按手头拿到的那个用即可。本节用 `.tar.gz` 演示；`.dmg` 见 3.3 节。

步骤 2：双击解压

在「访达 / Finder」里找到刚下载的 `BestPerf-macos-universal.tar.gz`，**双击它**。macOS 会自动解压，在同一个文件夹里得到一个名为 `BestPerf-macos-universal` 的文件夹。

提示：如果您习惯用终端，也可以在文件所在目录运行：`tar -xzf BestPerf-macos-universal.tar.gz`，效果一样。

怎么判断成功：旁边出现了一个 `BestPerf-macos-universal` 文件夹，双击能打开，里面能看到「Install BestPerf.command」。

步骤 3：双击「Install BestPerf.command」安装

打开 `BestPerf-macos-universal` 文件夹，找到「Install BestPerf.command」，双击它。

安装程序会：

- 把 BestPerf 安装到 `~/Applications/BestPerf`；
- 在桌面生成启动入口「1 启动 BestPerf.command」。

步骤 4：第一次被 Gatekeeper 拦住——右键打开（只首次）

第一次双击「Install BestPerf.command」时，macOS 多半会弹一个窗口，提示**「无法打开，因为来自身份不明的开发者」**。

这是正常现象，不是出错。macOS 有个叫 **Gatekeeper** 的安全机制：凡是没有经过苹果付费签名/公证的程序，第一次运行都会被拦一下，让您确认「这是我自己要运行的」。BestPerf 是您主动下载安装的，放行即可。

放行方法（只需做这一次）：

步骤 4.1：在「Install BestPerf.command」上点右键（或按住 `Control` 单击），在菜单里选「打开」。

步骤 4.2：这次弹窗里会多出一个「打开」按钮，点它。

提示：关键区别在于——直接双击时系统只给「取消」，不让运行；而**右键 → 「打开」**时系统会多给一个「打开」按钮，这就是放行的入口。

提示：另一种放行方式——直接双击被拦后，去「系统设置 → 隐私与安全性」，往下翻能看到被拦的那一项，点它右侧的「仍要打开」，效果一样。

注意：这个「右键 → 打开」只有第一次需要。一旦放行过，以后双击桌面的「1 启动 BestPerf.command」就会直接运行，不会再被拦。

步骤 5：看到绿色「安装结果」页 = 装成功了

安装顺利完成后，会弹出一张图文「安装结果」页。

- 显示绿色「安装成功」：说明全部就绪，您要做的下一步只有一件——双击桌面的「1 启动 BestPerf.command」。
- 万一显示红色：页面会用一句话给出唯一的下一步动作，照着做即可。最常见是缺 python3——这时页面会提示「安装 Python 3.9 或更新版本」；在 Mac 上推荐用 `xcode-select --install` 来装 python3（这是我们的建议，不是界面原话），装完再重装。



上图：图文「安装结果」页。绿色「安装成功」表示一切就绪，下一步只点桌面「1 启动」。

步骤 6：双击桌面「1 启动 BestPerf.command」

回到桌面，双击「1 启动 BestPerf.command」。

启动器会先做一次体检（Doctor）——自动检查 Python、端口、二进制（程序文件）是否都就绪；全部通过后，才会自动打开界面。

提示：如果您更喜欢用终端，也可以运行 `bash ~/Applications/BestPerf/BestPerf.command`，效果与双击桌面入口一样。

怎么判断成功：体检通过后，浏览器会自动打开 `http://127.0.0.1:8768`，显示 BestPerf 的深色主题界面。

步骤 7：浏览器打开界面

浏览器自动打开后，您会看到 BestPerf 控制台。顶部有四个标签页：

标签页	作用
本机信息	看本机连接信息、令牌、开关「允许被别人测向本机」，还有三步引导
发起测试	选发起方/对端、测量类型，按「开始」跑测试
探针	登记要测的对端
结果·样本库	看历史测试结果、下载回放文件

第一次进来，建议按界面「本机信息」页的引导走：**自检本机 → 登记对端 → 发起测试**。如果浏览器没有自动弹出，手动打开浏览器、地址栏输入 `http://127.0.0.1:8768` 回车即可。

关键理解： `http://127.0.0.1:8768` 是「本机自己」的地址，只有这台 Mac 上的浏览器能打开它，别人访问不到——这是控制台，不是被测端口。真正用来互测的是 UDP（默认探测端口 39700），那是另一回事，见 3.4 节。

3.3 用 .dmg 安装包

如果您拿到的是 `BestPerf-macos-universal.dmg`（磁盘映像），安装的内核步骤和 3.2 一样，只是打开方式略有不同。

步骤 1： 双击 `.dmg` 文件，macOS 会把它挂载成一个磁盘，并自动弹出一个窗口。

步骤 2： 在弹出的窗口里找到「**Install BestPerf.command**」，双击它来安装。

提示： 有些 `.dmg` 会提示把图标拖到「应用程序」文件夹或拖到桌面。如果是这样，先拖出来，再去拖出的那份上双击「**Install BestPerf.command**」运行——不要在挂载的只读磁盘里直接运行。

步骤 3： 和 3.2 步骤 4 完全一样——第一次被 Gatekeeper 拦住时，在「**Install BestPerf.command**」上右键 → 「打开」 → 再点「打开」（只首次）。

步骤 4: 装好后，记得在「访达」侧边栏点一下那个磁盘旁边的**「推出（弹出）」**按钮，把挂载的 `.dmg` 卸下来——安装已经写进了 `~/Applications/BestPerf`，磁盘可以推出，不影响以后启动。

注意：无论用 `.tar.gz` 还是 `.dmg`，装好后真正用来启动的入口都是桌面的「1 启动 BestPerf.command」，而不是 `.dmg` 磁盘里的那个文件。`.dmg` 只在安装时用一次。

3.4 让别人「测向」本机（被测端）

默认情况下，您这台 Mac 只能主动去测别人，别人测不进来。如果您希望对方机器能「测向」您这台 Mac（也就是让本机当被测端 / 应答方），需要两步。

步骤 1: 打开「允许被别人测向本机」开关

在控制台「本机信息」页的「本机连接信息」卡片里，找到「允许被别人测向本机」开关，把它打开。

注意：这个开关默认是关的。只有当您确实要被别人测时才打开它；测完可以关回去。

步骤 2: 把【地址 + 令牌】发给对方

同一张卡片里能看到本机的控制台地址（`http://本机IP:8768`）、访问令牌（Token）、本机 IP/端口，以及 UDP 探测端口 39700。把地址和令牌发给要测您的人——对方在他自己的「探针」页登记您时要用到。

步骤 3: 第一次被测时，点「允许」

当对方第一次测向您这台 Mac 时，macOS 会弹出一个系统提示「是否允许 接受传入网络连接」。点「允许」。

提示：这一步是 macOS 自带的应用防火墙在问您「要不要放行进来的连接」。点「允许」之后，对方就能正常测向本机了，以后一般不会再问。

关键理解：「主动去测别人」不需要打开任何开关、也不需要点「允许」——开开关、放行入站连接，只在您要当被测端（让别人测向您）时才需要。

3.5 怎么判断装成功

判断「装好了、能用了」，看两个信号就够：

信号一：绿色「安装结果」页

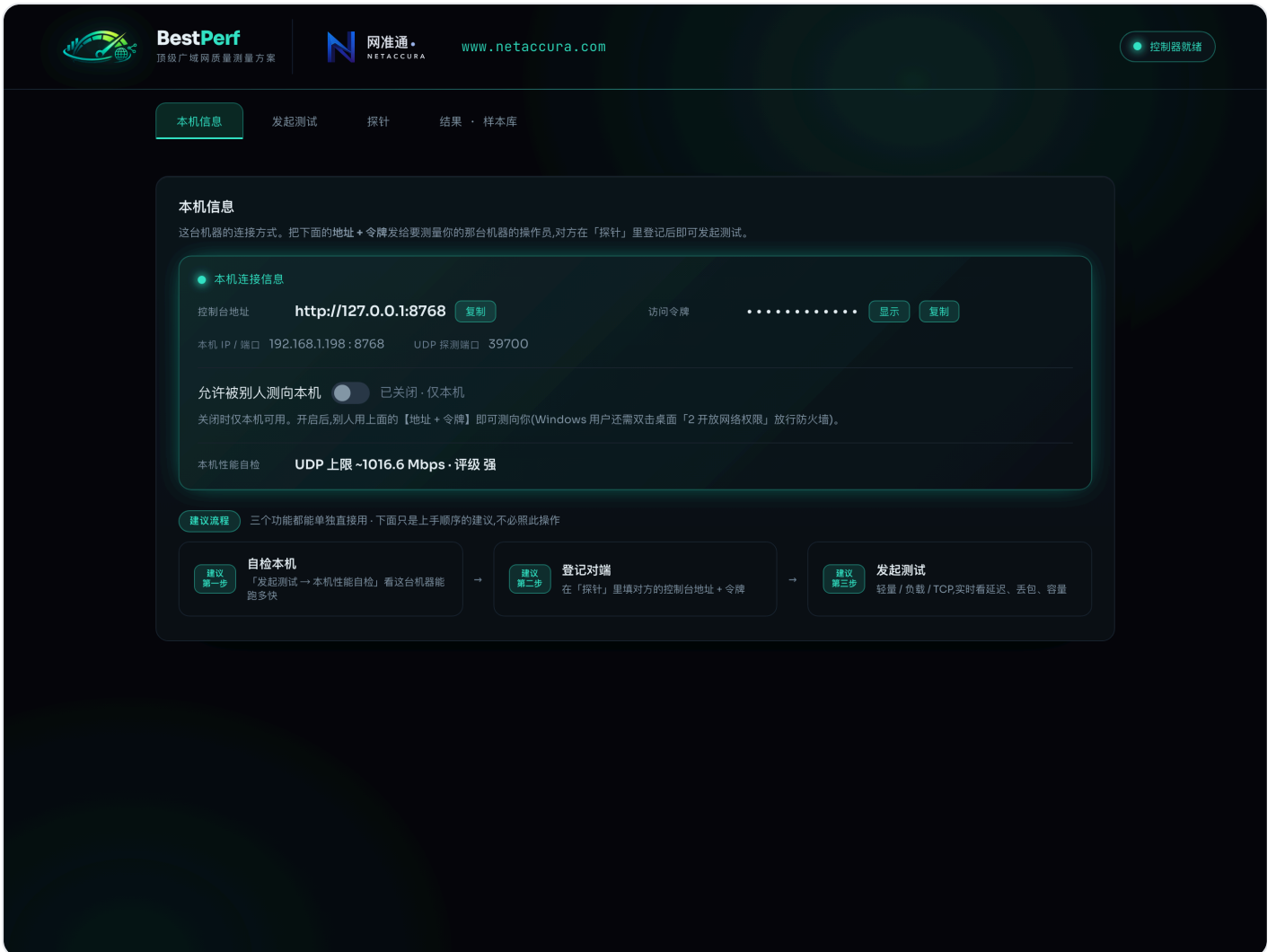
安装结束时弹出的图文页显示绿色「安装成功」，并提示下一步只点桌面「1 启动」。这说明 Python、端口、程序文件都就绪了。



上图：绿色「安装成功」——装好了，下一步双击桌面「1 启动 BestPerf.command」。

信号二：浏览器打开后看到完整的「本机信息」页

双击桌面「1 启动」后，浏览器自动打开 `http://127.0.0.1:8768`，能看到深色主题界面、顶部四个标签页，并且「本机信息」页里的「本机连接信息」卡片正常显示（有地址、令牌、UDP 端口 39700）。看到这个画面，就说明控制台真正跑起来了。



上图：「本机信息」页——能看到这个完整画面（连接信息卡片 + 三步引导），就说明 BestPerf 已经装好并正常运行。

提示：万一装不上或打不开，把 `~/Applications/BestPerf/install_summary.latest.txt`（或安装时终端窗口里的输出）发给技术支持人员，里面有诊断信息，能帮您快速定位问题。

3.6 装不上 / 打不开怎么办

遇到问题先别急，对照下表找到您的症状，照「该怎么办」做一遍，多半就解决了。

症状	该怎么办
双击「Install BestPerf.command」没反应	确认您是在解压后的 <code>BestPerf-macos-universal</code> 文件夹里双击，而不是在压缩包或只读磁盘里
被 Gatekeeper 拦（「来自身份不明的开发者」）	在该文件上右键 → 「打开」 → 再点「打开」（只首次，见 3.2 步骤 4）
提示缺 Python	装好 <code>python3</code> （推荐 <code>xcode-select --install</code> ）后，再重新跑一次安装
安装结果页显示红色	页面会给出唯一的下一步动作，照它说的做即可，不用自己猜
浏览器没自动打开界面	手动打开浏览器，地址栏输入 <code>http://127.0.0.1:8768</code> 回车（若 8768 被占用，以界面 / 终端显示的端口为准）
别人测不到本机	在「本机信息」页打开**「允许被别人测向本机」开关，并在系统弹窗里点「允许」**放行传入连接（见 3.4）

提示：如果上面都试过还没好，翻一下后面的 **常见问题（FAQ）** 章节，或把 `~/Applications/BestPerf/install_summary.latest.txt`（在解压出来的安装包文件夹里也有一份）发给技术支持，里面的诊断信息能帮您快速定位。

装好之后，下一步就是登记对端、发起第一次测试。继续看后面的章节，开始您的第一次「现网实测·复刻」。

第四章 安装与启动 · Linux

本章带您在 Linux 电脑上安装并启动 BestPerf 控制台。BestPerf 不需要 root、不需要改内核、不用抓包，装好后只是在本机后台跑一个小程序，您用浏览器打开它就行。整个过程跟着「步骤 1 / 2 / 3」走即可。

关键理解：BestPerf 是「两台机器各装一个控制台、用 UDP 互相测网络」。本章只负责把这一台 Linux 机器上的控制台装好、跑起来。要做一次真正的双机测试，对端那台机器也要照它自己的系统装一份（Windows 看第二章、macOS 看第三章）。

安装的总体流程如下图，先有个整体印象，再往下逐步操作。

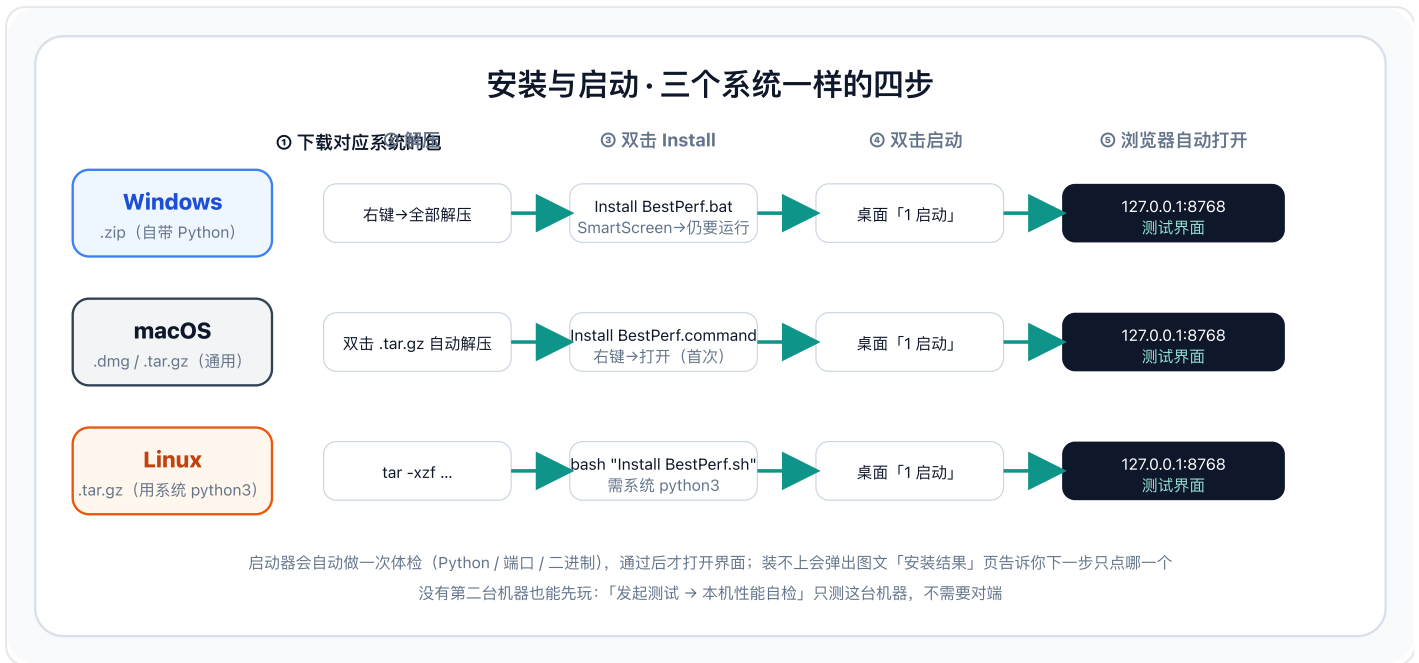


图 4-0 Linux 四步走：解压安装包 → 运行安装脚本 → 双击「1 启动」 → 浏览器自动打开

<http://127.0.0.1:8768>。

4.1 系统要求

开始前，先确认这台 Linux 机器满足下面几条。都满足才往下走，省得装到一半卡住。

项目	要求	怎么确认
CPU 架构	x86_64 (64 位 Intel/AMD)	终端执行 <code>uname -m</code> ，输出 <code>x86_64</code> 即可
Python	系统已装 python3 3.9 或更高	见下方「步骤」
桌面 / 命令行	有图形桌面（双击）或纯命令行（敲命令）都行	两种用法本章都给了
网络	与对端机器三层可达（能互相 ping 通即可）	真正测试时才用得到

提示：BestPerf 的 Linux 包用的是系统自带的 **python3**，包里不再单独塞一份 Python，所以这一步必须先确认 python3 装好。

确认 Python 是否就绪

打开终端，输入：

```
python3 --version
```

- 如果显示类似 `Python 3.10.12` (**3.9 及以上**)，说明 Python 没问题，跳到 4.2。
- 如果显示 `Python 3.8.x` 或更低，版本太老，需要升级或装个新版（见下方）。
- 如果提示 `python3: command not found`，说明这台机器还没装 Python，按下方对应发行版装一下。

没有 / 版本太低时，怎么装 python3

按您的发行版选一条命令执行（需要管理员，会让您输密码）：

```
# Ubuntu / Debian 系
sudo apt update && sudo apt install -y python3

# CentOS / RHEL / Rocky / Alma 系
sudo dnf install -y python3      # 老系统用 sudo yum install -y python3

# Fedora
sudo dnf install -y python3
```

装完再跑一次 `python3 --version` 确认到了 3.9+，就可以进入下一节。

注意： BestPerf 只用标准库，不会去网上装一堆 Python 依赖包，所以您**不需要** pip、**不需要**联网装库。只要 `python3 --version` 过关就行。

4.2 一步步安装与启动

下面假设您已经把安装包 `BestPerf-linux-x64.tar.gz` 下载到了机器上（比如放在「下载」目录 `~/Downloads`）。全程在终端里完成，命令可以直接复制。

步骤 1：解压安装包

进入安装包所在目录，把它解开：

```
cd ~/Downloads
tar -xzf BestPerf-linux-x64.tar.gz
cd BestPerf-linux-x64
```

判断成功： 执行 `ls` 后能看到 `Install BestPerf.sh` 这个文件，并且当前目录已经是 `BestPerf-linux-x64`，就对了。

提示： 如果您是在图形桌面里下载的，也可以右键安装包 → 「提取到此处 / Extract Here」，再用文件管理器进到解压出来的文件夹。但接下来运行安装脚本还是建议在终端里做，看得到提示信息。

步骤 2：运行安装脚本

在刚才那个解压出来的目录里，执行安装脚本：

```
bash "Install BestPerf.sh"
```

注意： 文件名里有空格，所以命令里的引号 `"..."` 不能去掉，否则会找不到文件。

安装脚本会做这些事：

- 把程序安装到 `~/.local/share/BestPerf`（您自己的用户目录下，不动系统、不需要 root）。
- 在桌面生成一个启动入口「1 启动 BestPerf.sh」。
- 做一次环境体检（Doctor）：检查 Python、端口、程序文件是否都齐了。

装完后会弹出 / 打开一张图文「安装结果」页：

- **绿色「安装成功」**：表示一切就绪，下一步只要双击桌面的「1 启动 BestPerf.sh」即可。
- **红色**：表示有问题，页面会用红字给出唯一的下一步动作（通常是「请先装 python3」之类），照着做完再重跑这一步。



图 4-1 安装结束后的「安装结果」页。看到绿色「安装成功」就说明这台机器装好了，记住下一步只点桌面「1 启动」。

判断成功：看到绿色「安装成功」，并且桌面（或 `~/Desktop`）出现了「1 启动 BestPerf.sh」。

步骤 3：启动控制台

有两种方式启动，任选其一：

方式 A · 双击（有图形桌面）

双击桌面上的「1 启动 BestPerf.sh」。第一次双击如果系统问您「是否运行 / 在终端中运行」，选运行即可。

方式 B · 命令行（纯终端也能用）

```
bash ~/.local/share/BestPerf/BestPerf.sh
```

不论哪种方式，启动器都会先做一次体检（Doctor）——确认 Python、端口、程序文件都就绪——然后才在本机起一个小服务并自动打开浏览器，地址是：

```
http://127.0.0.1:8768
```

提示： 127.0.0.1 是「本机自己」的意思，这个地址只有本机能打开，别人访问不到。控制台默认端口是 **8768**。如果浏览器没自动弹出来，手动打开浏览器、把上面这行地址敲进地址栏即可。

判断成功： 浏览器打开后看到 BestPerf 的深色界面，顶部有四个标签页——「本机信息」「发起测试」「探针」「结果·样本库」。看到这个界面，就说明这台 Linux 机器的控制台已经装好并跑起来了。

关键理解： 这个窗口（终端里运行的 BestPerf.sh）就是控制台本体，测试期间别关它。关掉终端 / 停掉脚本，控制台就停了，界面也会打不开。下次要用，再双击桌面「1 启动」或重跑一遍命令即可。

4.3 防火墙：要被别人测向本机时才需要

BestPerf 的角色分两种：

- 你去测别人（发起方）：本机只往外发流量，不需要动防火墙，装好就能用。
- 让别人测向你这台机器（当对端 / 应答方）：别人的流量要能进到你这台机器，这时如果本机开了防火墙，就得放行几个入站端口。

关键理解： 只有当这台 Linux 机器要被别人测（充当对端）时，才需要做本节的放行。纯发起方可以跳过整节。

放行的同时，别忘了去界面里把开关打开：在「本机信息」页的「本机连接信息」卡片里，打开「允许被别人测向本机」开关（默认是关的），再把卡片上显示的**控制台地址 + 访问令牌（Token）**发给要测你的人。

用 ufw (Ubuntu / Debian 常见)

依次执行下面三条（需要管理员，会让您输密码）：

```
sudo ufw allow 8768/tcp           # 控制台端口 (HTTP)
sudo ufw allow 39700:39899/udp   # UDP 探测端口段
sudo ufw allow 39800:39899/tcp   # TCP 体感测试端口段
```

放行的端口含义对照：

端口	协议	用途
8768	TCP	控制台本身 (HTTP)，两台控制台之间也走它协调
39700-39899	UDP	UDP 探测端口段（默认探测端口 39700），轻量测量 / 负载画像都走这里
39800-39899	TCP	TCP 体感 (goodput) 测试用

用 firewalld (CentOS / RHEL / Fedora 常见)

如果您的机器用的是 firewalld，可一句带过——放行同样的三段：

```
sudo firewall-cmd --permanent --add-port=8768/tcp \
  --add-port=39700-39899/udp --add-port=39800-39899/tcp
sudo firewall-cmd --reload
```

提示：很多 Linux 桌面/服务器默认压根没开防火墙。可以先不动；如果对端测过来连不上，再回来做本节的放行。

4.4 怎么判断装成功

两道关，过了就说明这台机器彻底就绪了。

第一道关：安装结果页是绿色

步骤 2 装完后那张「安装结果」页显示绿色「安装成功」（图 4-1）。这一关确认「程序文件、Python、端口」都齐了。

第二道关：界面正常打开

启动后浏览器打开 `http://127.0.0.1:8768`，看到下面这样的「本机信息」页：

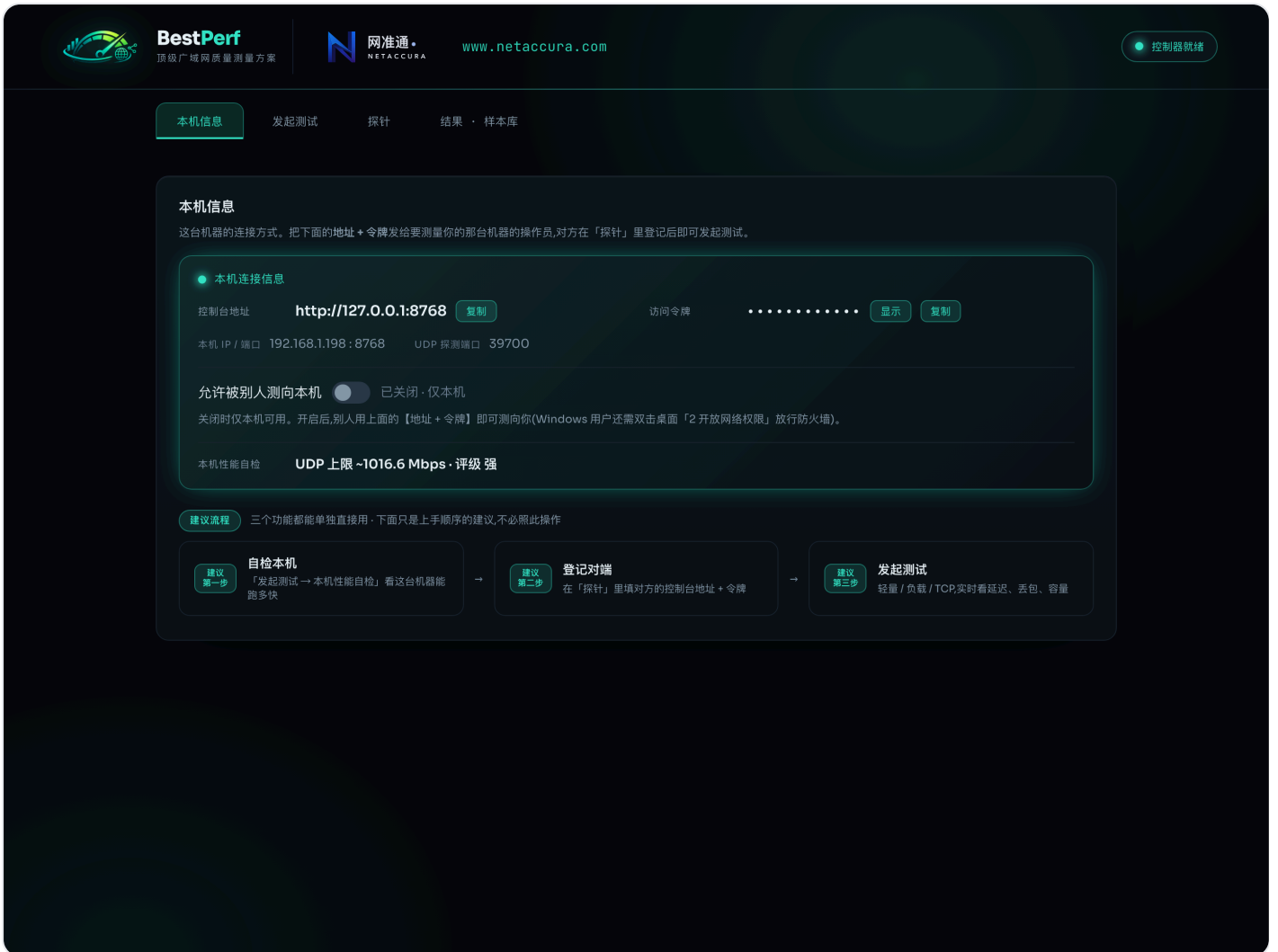


图 4-2 控制台「本机信息」页。看到这个深色界面，并且「本机连接信息」卡片里有控制台地址、访问令牌 (Token)、本机 IP/端口、UDP 探测端口 39700，就说明装好了。

对照检查这几样，全有就算通过：

- 顶部四个标签页：本机信息、发起测试、探针、结果 · 样本库。
- 「本机连接信息」卡片里能看到：控制台地址 `http://本机IP:8768`、访问令牌 (Token)、本机 IP/端口、UDP 探测端口 39700，以及「允许被别人测向本机」开关 (默认关)。
- 页面上有「自检本机 → 登记对端 → 发起测试」三步引导。

两关都过，这台 Linux 机器的控制台就装好了。

下一步：如果只想先确认这台机器自己没问题，可以在「发起测试」页选「**本机性能自检**」（不需要对端）跑一下。要做真正的双机测试，请先让对端那台机器也装好控制台、打开「允许被别人测向本机」，把它的**地址 + 令牌**给你，然后到「探针」页把它登记进来——具体操作见后续章节。

装不上 / 打不开怎么办：把解压出来的安装包目录里的 `install_summary.latest.txt`（也就是你运行 `Install BestPerf.sh` 时所在的那个目录；安装时终端也会打印它的完整路径）发给支持人员，里面记录了体检结果，方便快速定位问题。

第五章 打开测试界面与控制台总览

装好 BestPerf 之后，真正干活的地方是一个网页界面，我们把它叫做**控制台**。它跑在你自己这台机器上，用浏览器打开就行——不用装别的软件，不用记复杂的网址。

本章带你把控制台打开，认全四个标签页，看懂「本机信息」页上那张连接信息卡片，并讲清两个最容易让新手犯迷糊的概念：**本机和对端、令牌是干嘛的**。把这一章看完，你就知道这台机器是谁、它的「门牌号」是什么、什么时候需要把门打开让别人来测你。

关键理解：BestPerf 测的是「两台机器之间的网络」。所以每台参与测试的机器，本地都要各自跑一个控制台。你在自己这台的控制台上操作，对端那台也开着它自己的控制台。两台控制台通过网络互相对话，一起把这段网络的延迟、丢包测出来。

5.1 怎么打开测试界面

方法一：双击「1 启动」入口（推荐）

第三章安装完成后，桌面上会出现一个启动入口，三个系统的名字略有不同：

系统	桌面入口名称
Windows	1 启动 BestPerf.cmd
macOS	1 启动 BestPerf.command
Linux	1 启动 BestPerf.sh

双击它就行。启动器会先做一次「体检」（Doctor）——检查 Python、端口、程序文件是不是都就绪——一切正常后，会自动帮你打开浏览器，停在控制台界面上。

提示：第一次启动可能要等几秒，启动器在后台拉起控制台并等它准备好。看到浏览器弹出深色的 BestPerf 界面，就说明成功了。

方法二：手动在浏览器里打开

如果浏览器没有自动弹出，或者你不小心关掉了页面，**不用重新启动**，只要控制台还在后台运行，直接打开浏览器，在地址栏输入：

```
http://127.0.0.1:8768
```

按回车就回到控制台了。

地址组成	含义
127.0.0.1	「本机」的固定地址，意思就是「我自己这台电脑」
8768	控制台的端口号，BestPerf 默认用这个

注意： `http://127.0.0.1:8768` 只能在装了 **BestPerf** 的这台机器上打开。别人的电脑上输入这个地址，打开的是他们自己的控制台，不是你的。要让别人访问到你，得用下面 5.1 第三段讲的「本机 IP + 令牌」。

方法三（多机时）：别人怎么访问你

当你要和另一台机器互相测网络时，对方需要能联系上你这台控制台。对方不是在浏览器里直接打开你的页面，而是在他的控制台「探针」页里，把你的控制台地址 + 你的令牌登记进去（详见第六章）。你要发给对方的两样东西是：

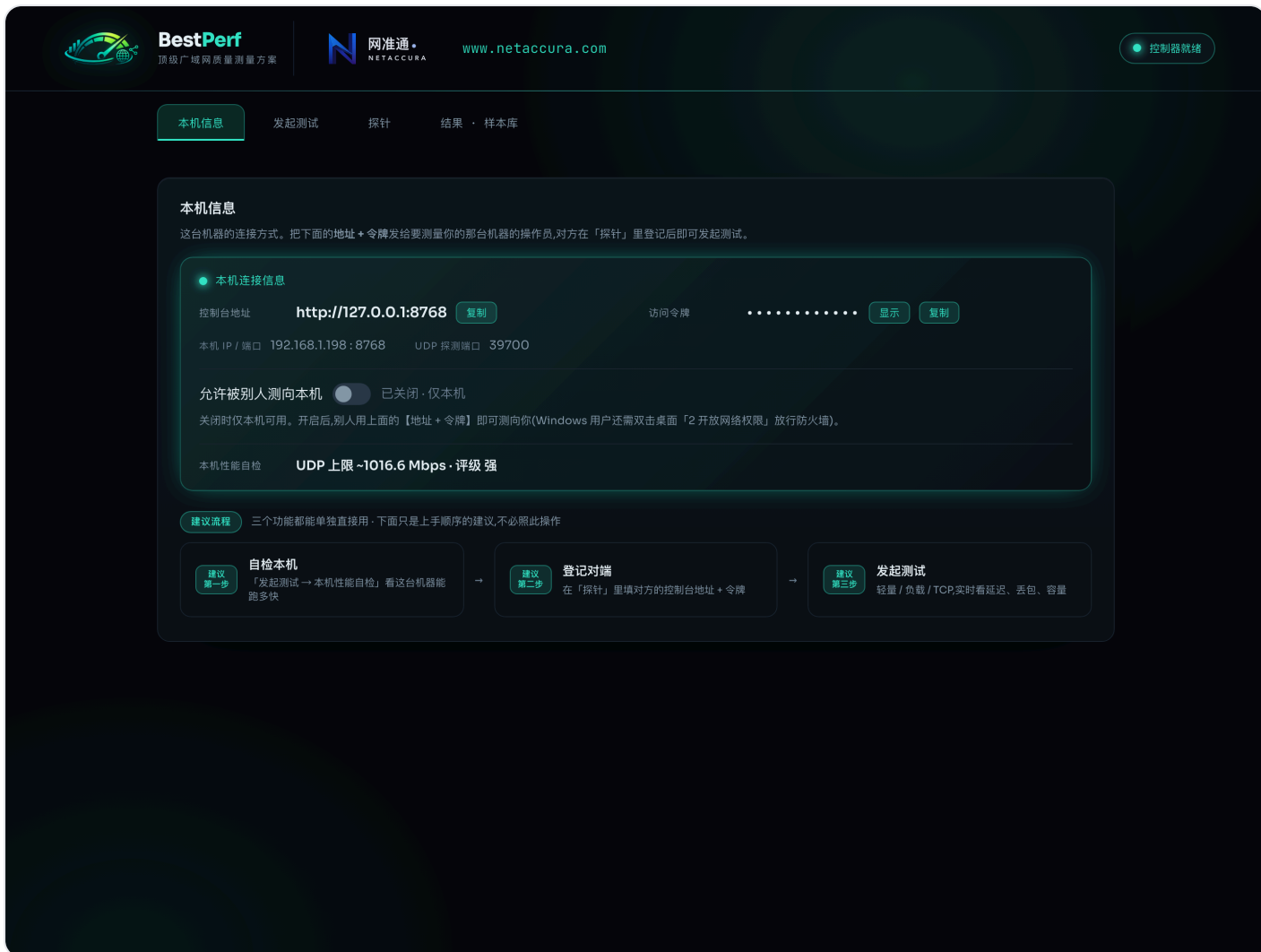
1. 你的控制台地址：形如 `http://你的本机IP:8768`（不是 `127.0.0.1`，要用你在局域网里的真实 IP）。
2. 你的访问令牌（Token）：一串随机字符，相当于「来访密码」。

这两样东西都能在「本机信息」页一眼看到（见 5.3）。

关键理解： 控制台监听的是 8768 端口，但 BestPerf 真正打网络测量包用的是另一组端口——**探测 UDP 39700–39899、TCP 体感 TCP 39800–39899**。它们各司其职：控制台 TCP 8768 是两台控制台「商量怎么测」的通道，探测 / 体感端口是真正灌测量流量的通道。这些都要能通，测试才跑得起来（防火墙放行办法见第三章和第六章）。

5.2 四个标签页总览

控制台顶部有一条标签栏，四个标签页就是你全部的操作入口。点哪个标签，下面就切到对应的页面。



上图：控制台打开后默认停在「本机信息」页，顶部从左到右是四个标签页。

四个标签页各管一摊事：

标签页	一句话作用	什么时候用
本机信息 (home)	显示这台机器自己的「身份证」：地址、令牌、端口，以及上手引导	打开控制台第一眼看的页；要把地址/令牌发给别人时来这拿
发起测试 (test)	选发起方、对端、测量类型和参数，按「开始」跑测试	真正动手测网络的地方
探针 (probes)	登记你要测的对端机器（填对方的地址和令牌）	测两台机器之间的网络前，先在这里把对端加进来
结果·样本库 (results)	列出每一次测过的记录，点开查看详情、下载回放文件	测完看结果、导出给 WAN 仿真器复刻现网

提示：打开控制台默认就停在「本机信息」页。先在这里把自己看明白，再去「探针」登记对端，最后到「发起测试」开跑——这也是页面上「建议流程」给的顺序。

这四个标签页里：

- **发起测试**页有四种测量类型可选：**轻量测量 / 负载画像 / TCP 体感 / 本机性能自检**（第七章详讲）。其中「本机性能自检」是测这台机器自己的处理能力，不需要对端。
- **探针**页是配置对端的地方（第六章详讲）。
- **结果 · 样本库**页里能下载**回放文件**，导入 WAN 仿真器就能在实验室复刻现网（后续章节详讲）。

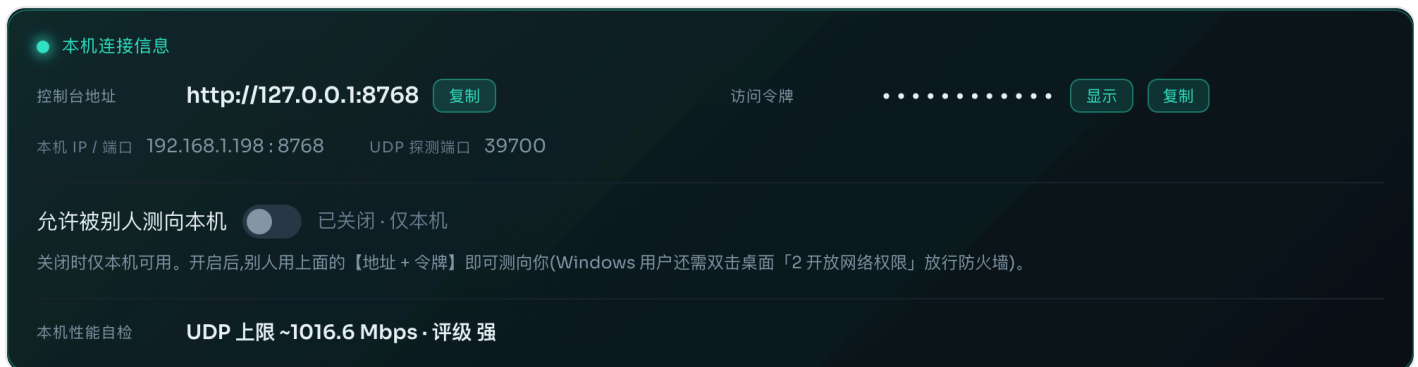
本章先把第一个标签页——「本机信息」——讲透，因为它是你和别人协作的起点。

5.3 「本机信息」页详解

「本机信息」页是控制台的门面。它告诉你**两件事**：这台机器的连接方式（上半部的卡片），以及建议的上手顺序（下半部的三步引导）。

5.3.1 「本机连接信息」卡片

页面上半部分是一张卡片，列出这台机器的全部「身份信息」。



上图：「本机连接信息」卡片。把这里的「地址 + 令牌」发给要测你的那台机器的操作员，对方登记后就能发起测试。

卡片上的字段逐个看：

字段	内容样子	它是干嘛的
控制台地址	<code>http://本机IP:8768</code>	别人要访问你这台控制台时填的地址。发给对端的就是这一串
访问令牌 (Token)	一串随机字符	相当于「来访密码」。别人登记你为对端时要一起填上，否则连不进来
本机 IP / 端口	如 <code>192.168.1.10 / 8768</code>	这台机器在网络里的地址和控制台端口
UDP 探测端口	默认 <code>39700</code> (若被占用会自动顺延到下一个空闲端口，卡片显示实际使用的端口)	真正灌测量流量的端口。两台机器之间要能通这个 UDP 端口，测试才跑得动
允许被别人测向本机 (开关)	默认关	控制「别的机器能不能测向我」的总闸。默认关着，只有要被别人测时才打开

注意：默认（未打开「允许被别人测向本机」开关）时，控制台地址显示 `http://127.0.0.1:8768`（只有本机能开）；打开开关后，才会变成你这台机器的局域网地址 `http://192.168.x.x:8768` ——要发给对方登记的，是开开关之后的这个局域网地址。直接复制卡片上显示的那一串发出去就对了。

提示：令牌是 BestPerf 自动生成并记住的，你不用自己设，也不用记。每次要给别人时，照着卡片复制就行。它只会显示给在这台机器本地打开控制台的你；别人从局域网访问你的控制台时，是看不到令牌的——所以必须由你主动把令牌发给对方。

5.3.2 「允许被别人测向本机」开关

这个开关是新手最容易忽略、却最常导致「对方测不通」的地方，单独说清楚：

- **默认关闭。**关着的时候，别的机器联系你这台控制台会被直接挡在门外（返回「对端无法连接」）。这是为了安全——没人能在你不知情时测你。
- **只有当你这台机器要作为「对端」被别人测时，才需要把它打开。**
- **打开之后，你自己主动去测别人不受影响**——发起测试任何时候都能做，这个开关只管「别人能不能测进来」。

关键理解：这个开关只管「入站」。

- 我去测别人：**不用**开这个开关。
- 别人来测我：**必须**开这个开关，并且把我的地址 + 令牌给他。

一段网络要做双向测量，两头都得能互相联系，所以**两台机器一般都要把这个开关打开**，并互相交换地址和令牌。

注意：打开开关只是软件层面放行。系统防火墙那一层也要放行测量端口（探测 UDP 39700–39899、TCP 体感 TCP 39800–39899，以及控制台 TCP 8768），对端才真正连得通。各系统的防火墙放行办法见对应的安装章节（第二/三/四章）：Windows 双击「2 开放网络权限 BestPerf」、macOS 弹窗点「允许」、Linux 用 `ufw` 放行。

5.3.3 三步上手引导

卡片下面是一条「建议流程」，把第一次使用的顺序画成了三步：

顺序	动作	在哪做
建议第一步	自检本机	「发起测试 → 本机性能自检」，先看这台机器自己能跑多快
建议第二步	登记对端	「探针」页填对方的控制台地址 + 令牌
建议第三步	发起测试	选轻量 / 负载 / TCP，实时看延迟、丢包、容量

提示：页面上明确写着——这只是建议的上手顺序，不必照此操作。三个功能都能单独直接用。比如你只想测一段网络，完全可以跳过自检，直接登记对端、发起测试。

先做一次「本机性能自检」的好处是：它测出这台机器自己的处理上限。万一后面测网络时数字不好看，你能分清是网络差还是本机扛不住——这个判断对后续排查很有用。

5.4 关键理解：本机、对端、令牌

这一节把全章最核心的几个概念集中讲一遍，看懂了后面的章节都顺。

本机 vs 对端

叫法	指谁	在界面上
本机	运行你正在操作的这个控制台的机器，就是「我自己」	「探针」列表里带 本机 标签的那一条
对端	网络另一头、你想和它互测的那台机器	你在「探针」页登记进来的机器

关键理解：BestPerf 测的从来不是「一台机器」，而是「本机 ↔ 对端之间的那段网络」。所以两台机器缺一不可，而且两台都要各自跑着 **BestPerf 控制台**。你这台是本机，对方那台是对端；如果换到对方的电脑前操作，那台就成了本机，你这台反过来成了它的对端。**本机和对端只是相对的角色，不是固定身份。**

例外：**本机性能自检**这种测量不需要对端——它让本机自己跟自己测（环回 loopback），单台机器就能跑，用来看这台机器自身的处理能力。其余三种（轻量测量、负载画像、TCP 体感）都需要一个对端。

令牌是干嘛的

令牌（Token）就是访问你控制台的「来访密码」。

- 每台控制台都有一个**自动生成、长期固定**的令牌，显示在「本机信息」页的卡片上。
- 别人要把你登记成对端、向你发起测试，**必须同时拿到你的地址和令牌**。地址告诉他「往哪连」，令牌证明「这是你允许的人」。
- 没有正确令牌的连接会被拒绝。这样即使你开了「允许被别人测向本机」，也只有拿到令牌的人能真正测你，陌生人测不进来。
- 令牌只对本机操作者显示。别人从网络访问你的控制台是看不到令牌的，所以**必须由你亲手把令牌发给对方**（微信、邮件、当面念都行）。

注意：令牌相当于密码，**别贴到公开的地方**。只发给你确实想让他测你的那位操作员即可。

什么时候要开「允许被别人测向本机」

把整章的判断浓缩成一张表，照着对就不会错：

你想做的事	需要开「允许被别人测向本机」吗	需要把地址+令牌发给对方吗
自己测自己（本机性能自检）	不需要	不需要
我主动去测对端	不需要（这是我发起）	不需要（我去拿对方的）
让对端来测我	需要	需要（把我的发给他）
两台机器双向互测	两台都需要	互相交换

关键理解：绝大多数现网测量都是双向的——既要测「本机→对端」，也要测「对端→本机」。所以实际操作里，**两台机器通常都要打开这个开关、互相交换好地址和令牌**，然后由其中任意一台发起测试即可。

下一章（第六章）我们就动手做这件事：在「探针」页把对端登记进来，让两台控制台真正「认识」彼此，为发起第一次测试做好准备。

第六章 手把手：第一次测试

前面几章把 BestPerf 装好、把控制台打开了。本章带您从头到尾完整跑一次测试，把界面上的每个按钮都点一遍，再把结果和回放文件看明白。

关键理解： BestPerf 不区分操作系统的使用方式。无论您用的是 Windows、macOS 还是 Linux，控制台界面和操作步骤**完全一样**——都是浏览器打开 `http://127.0.0.1:8768`，对着同一套深色界面操作。本章所有截图与步骤对三个系统通用。

本章分两条路走：

- **只有一台机器：**先看 6.1，用「本机性能自检」热个身，不需要第二台机器。
- **手上有两台机器：**从 6.2 开始，登记对端、互相测一次真实网络。

6.1 没有第二台机器也能先玩：本机性能自检

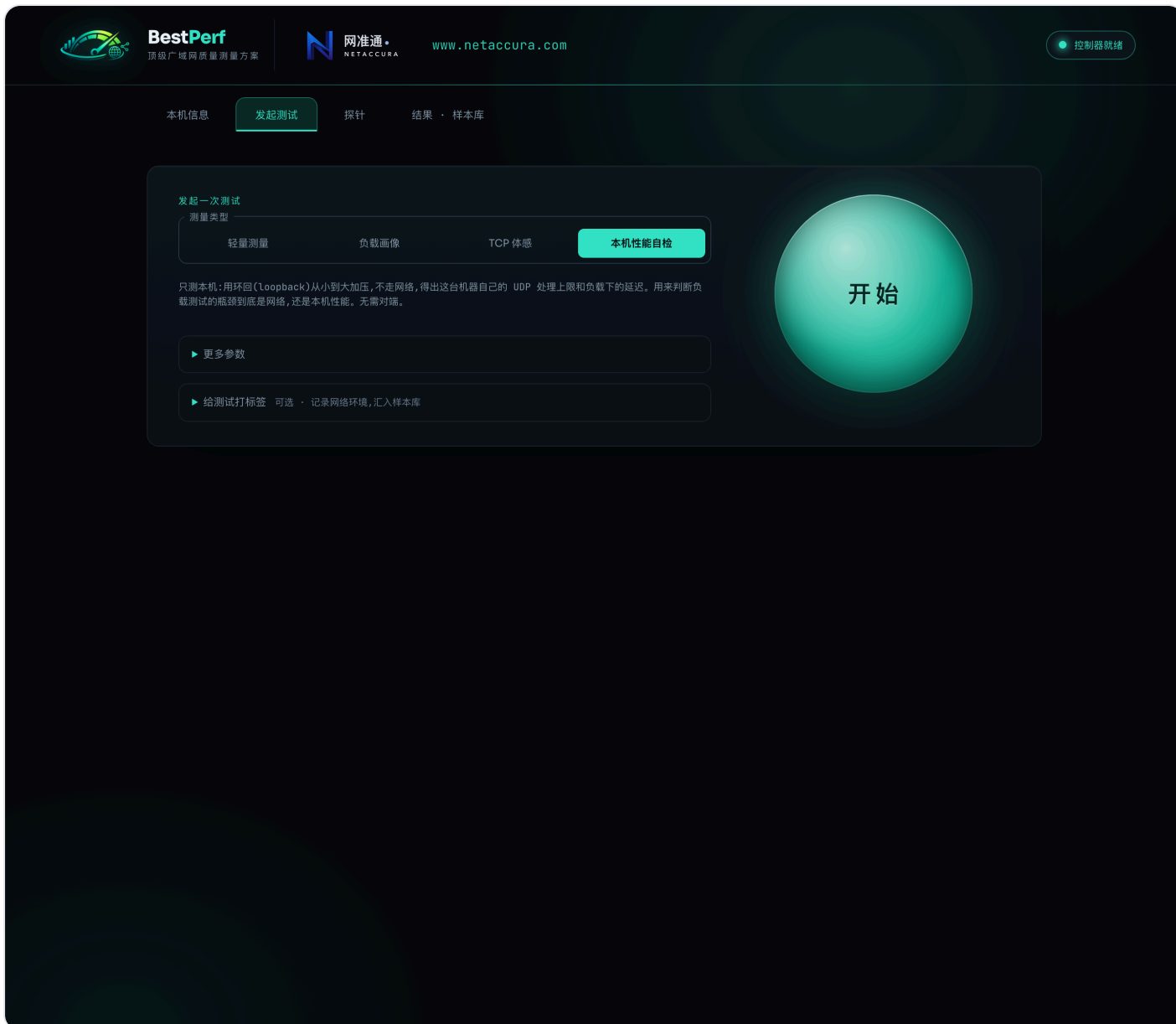
测网络要两台机器对测。但在找到第二台机器之前，您可以先用「本机性能自检」摸清这台机器自己能跑多快——它在本机内部（环回 loopback）从小到大加压，不走网卡、不打扰网络，得出这台机器的 UDP 处理上限和负载下的延迟。

它有什么用？ 以后做真实网络测试时，如果发现速度上不去，您就能分清：到底是**网络的瓶颈**，还是**本机性能**本身就只有这么快。自检给的就是这台机器的「天花板」。

步骤 1：点顶部标签页「发起测试」。

步骤 2：在「测量类型」分段开关里点「本机性能自检」（最右边那个）。

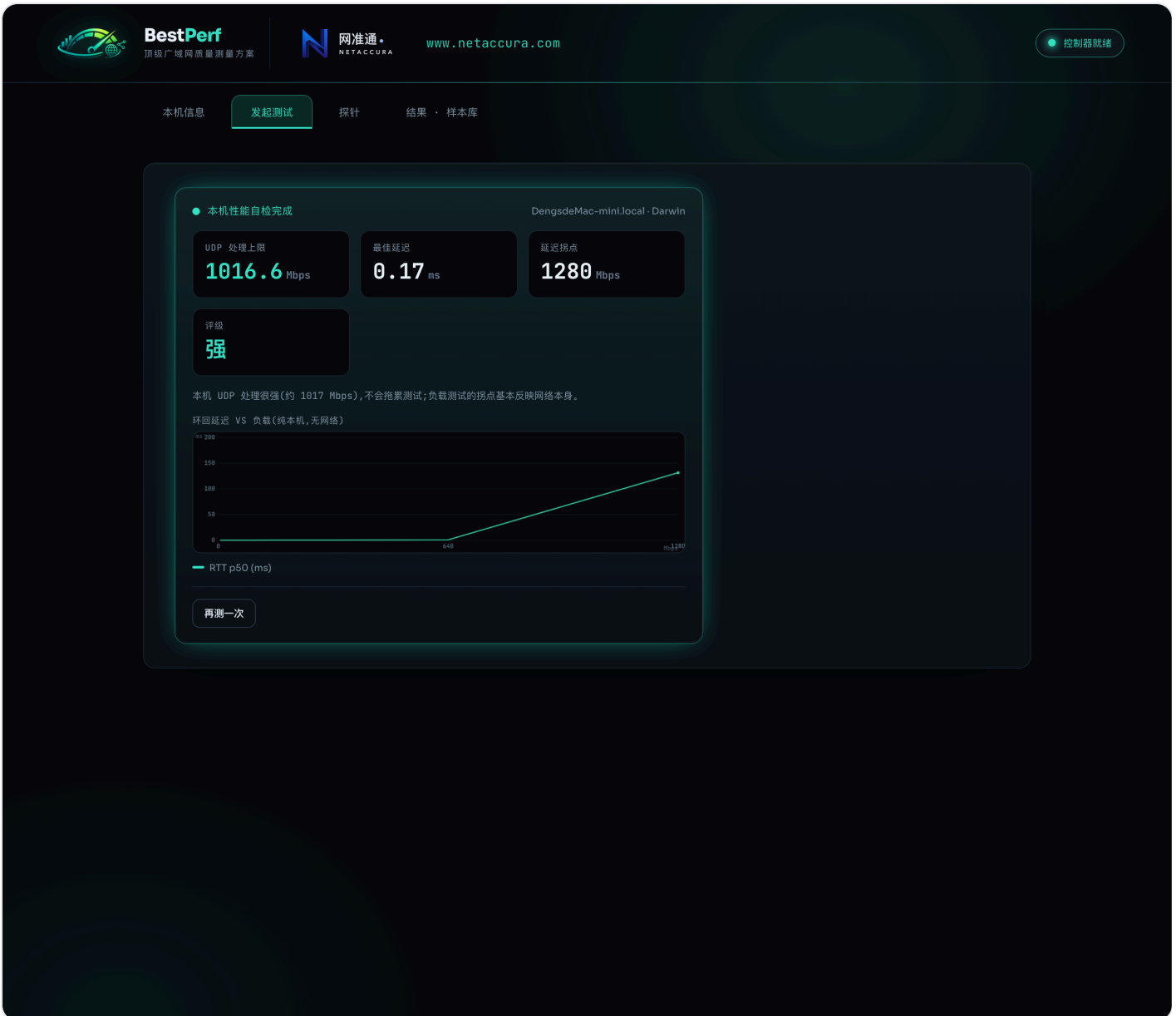
提示：选了「本机性能自检」之后，界面会自动**隐藏**上方的「对端」下拉和「方向」选择——因为自检只测本机，不需要对端。这是正常的，不是出 bug 了。



选「本机性能自检」后，界面只剩本机一项，发光的「开始」按钮在右侧。

步骤 3：点右侧那颗发光的「开始」按钮。

测试只跑几秒。界面会从发起表单切换成实时进度，自检完成后直接在这块区域里给出结果，无需翻页。



自检完成卡片：上方四块数字，下方是「环回延迟 vs 负载」曲线。

怎么看自检结果？完成卡片上有四块数字：

指标	含义
UDP 处理上限	这台机器在本机环回下能稳定处理的 UDP 速率 (Mbps)。这就是它的天花板。
最佳延迟	轻负载下能达到的最低往返延迟 (ms)，代表这台机器的延迟下限。
延迟拐点	加压到多大时延迟开始明显抬头 (Mbps)。没出现就显示「—」。
评级	一句话结论，分三档：强 / 中等 / 弱。

下方曲线是「环回延迟 vs 负载」——横轴是施加的负载 (Mbps)，纵轴是 RTT 往返延迟 (ms)，看延迟随压力怎么往上走。注意标题里写着「纯本机，无网络」，提醒您这条曲线测的是机器本身，不是网络。

关键理解：自检评级是「弱」不代表机器坏了，只是说它的 UDP 处理能力一般。如果以后真实网络测试跑出来的速度，正好卡在自检给出的「UDP 处理上限」附近，那瓶颈多半是**本机**，不是网络——换台更强的机器再测就能验证。

6.2 两台机器互测的准备：开放对端 + 登记探针

要测一段真实网络，需要**两台机器**，每台都装了 BestPerf 控制台。一台当「发起方」（您操作的这台），一台当「对端」（被测的那台）。下面分两步把对端准备好。

第一步：在「对端」机器上打开「允许被别人测向本机」

去到要被测的那台机器（对端），在它的控制台上：

1. 点顶部「本机信息」标签页。
2. 找到「本机连接信息」卡片，把「允许被别人测向本机」开关打开（默认是关的）。
3. 记下卡片上显示的两样东西：
 - **控制台地址：**形如 `http://对端IP:8768`
 - **访问令牌 (Token)：**一串字符

注意：「允许被别人测向本机」默认关闭，目的是安全——平时别人测不到您。只有当这台机器要被别人测时才打开它。关闭时这台机器只能自己用（比如做 6.1 的自检）。

macOS / Windows 用户特别注意：

- macOS 第一次被别人测时，系统会弹「允许接受传入网络连接」，点「允许」。
- Windows 用户除了打开开关，还要双击桌面「2 开放网络权限 BestPerf」放行防火墙（UAC 点一次「是」），否则对端探测包会被挡在门外。详见第二章安装说明。

把对端的控制台地址和访问令牌抄下来，回到您操作的这台机器（发起方）。

第二步：在「本机」的「探针」页登记对端

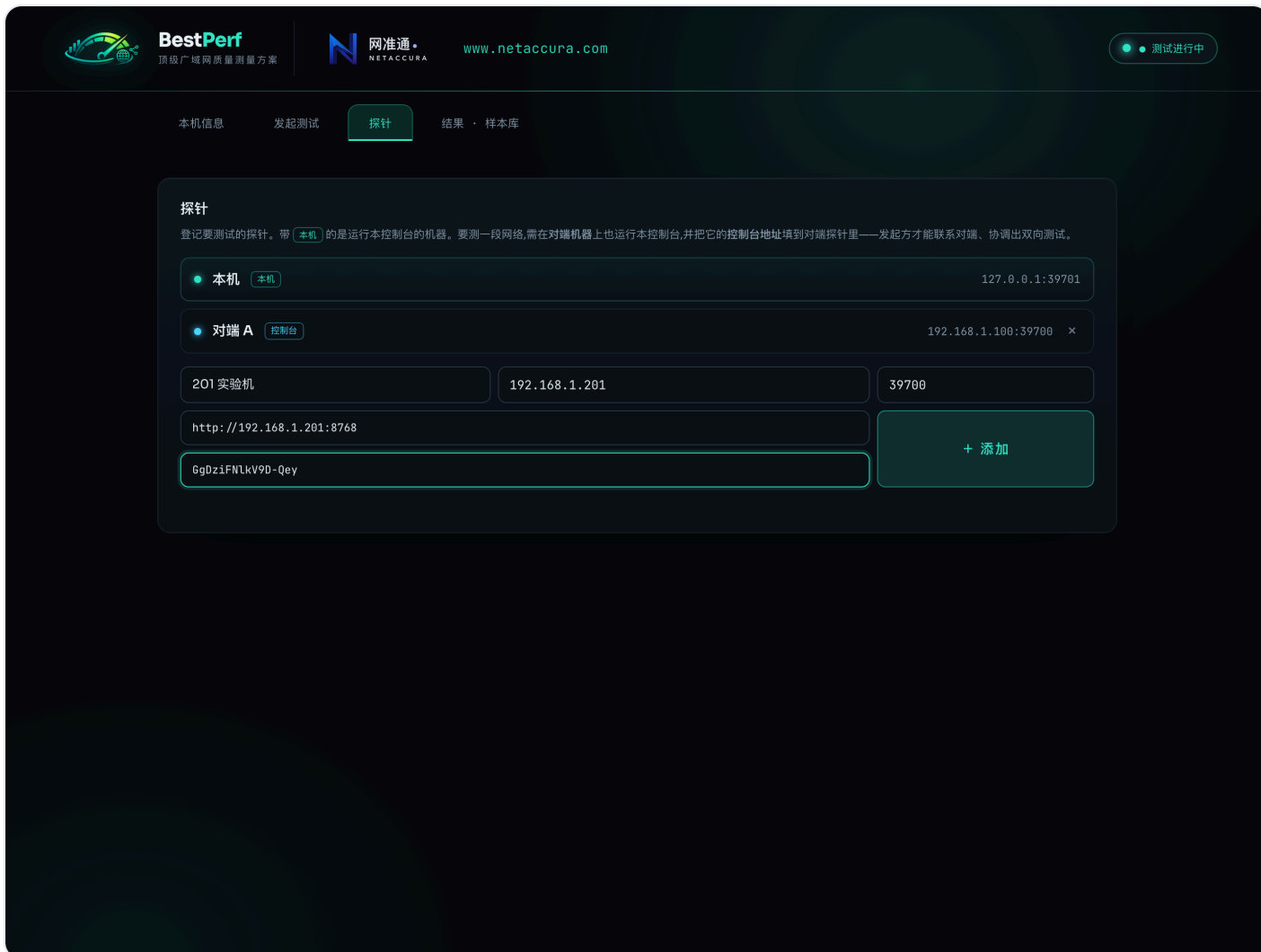
回到发起方机器的控制台，点顶部「探针」标签页。这一页用来登记您要测的对端。

前提：对端也装了控制台、已打开「允许被别人测向本机」开关、并且把它的**地址 + 令牌**给了您（就是上一步抄下来的）。三者缺一，登记后也测不通。

在页面下方的表单里填写：

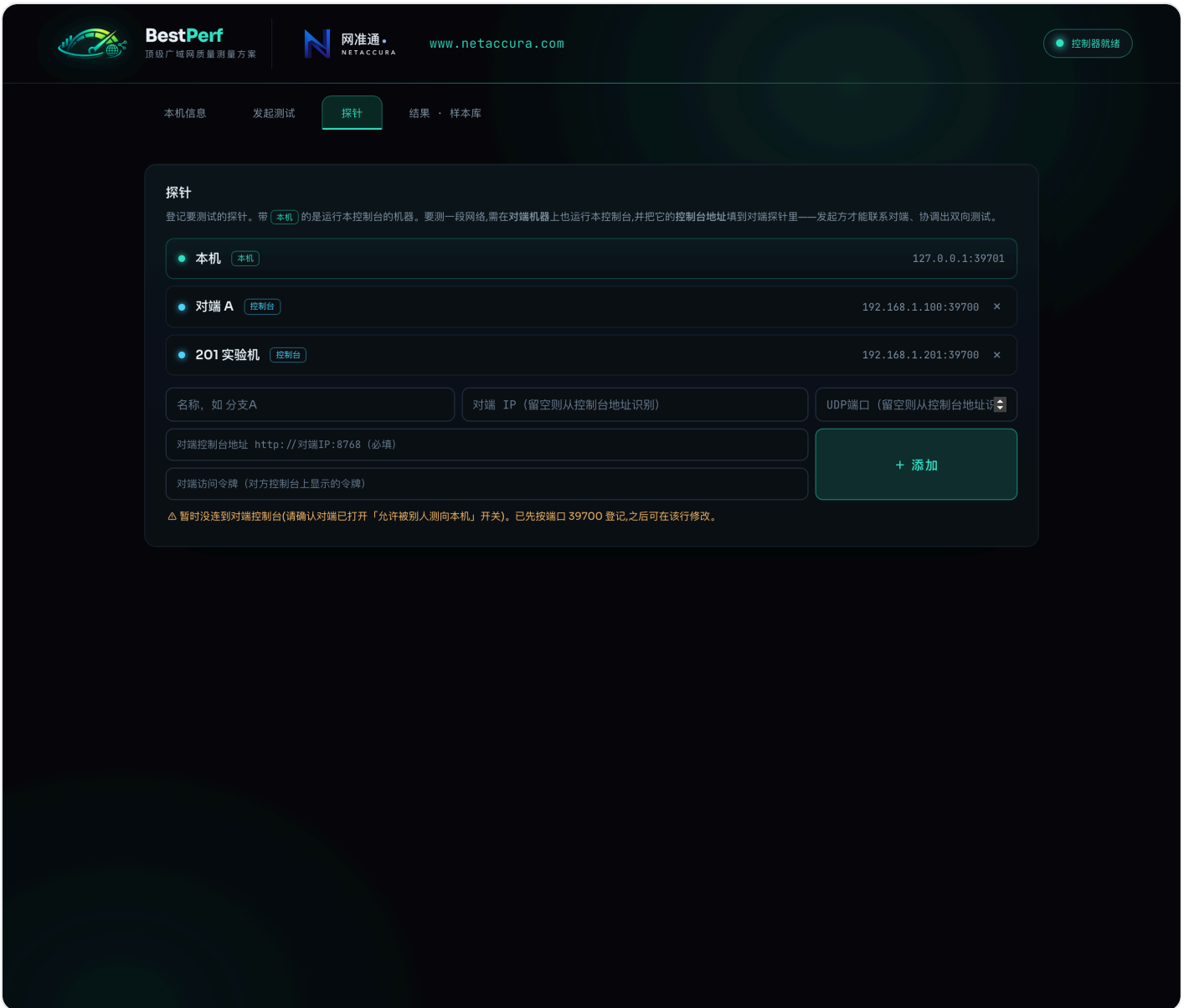
字段	填什么	示例
名称	给对端起个好记的名字	分支A
对端 IP	对端的 IP（留空则从下面的控制台地址自动识别）	192.168.1.201
UDP 端口	对端的 UDP 探测端口；端口可留空（会从对端控制台地址自动识别），识别不到时才用默认 39700	39700
对端控制台地址（必填）	对端「本机信息」上的控制台地址	http://192.168.1.201:8768
对端访问令牌	对端「本机信息」上显示的那串令牌	（粘贴对方令牌）

提示：「对端 IP」和「UDP 端口」可以留空——只要填对了「对端控制台地址」，控制台会自动从里面识别出 IP。最省事的填法是：只填**名称、控制台地址、令牌**三项。



表单填好后的样子：名称、对端控制台地址、令牌都已就位。

填好后点表单右侧「+ 添加」按钮。登记成功后，对端会出现在上方的探针列表里。



登记成功：对端已在探针列表里，运行本控制台的机器自带「本机」标记。

注意：如果点添加后提示「暂时没连到对端控制台」，多半是对端那台机器没打开「允许被别人测向本机」开关。BestPerf 会先按您填的信息把它登记下来（之后可在该行修改），但要等对端开了开关才真正测得通。

6.3 发起一次轻量测量

对端登记好了，就可以发起第一次真实网络测试了。第一次推荐用「轻量测量」——它流量小、不打扰网络，几秒钟就量出延迟、丢包、抖动、乱序、重复和微中断，最适合摸底。

步骤 1：点顶部「发起测试」标签页。

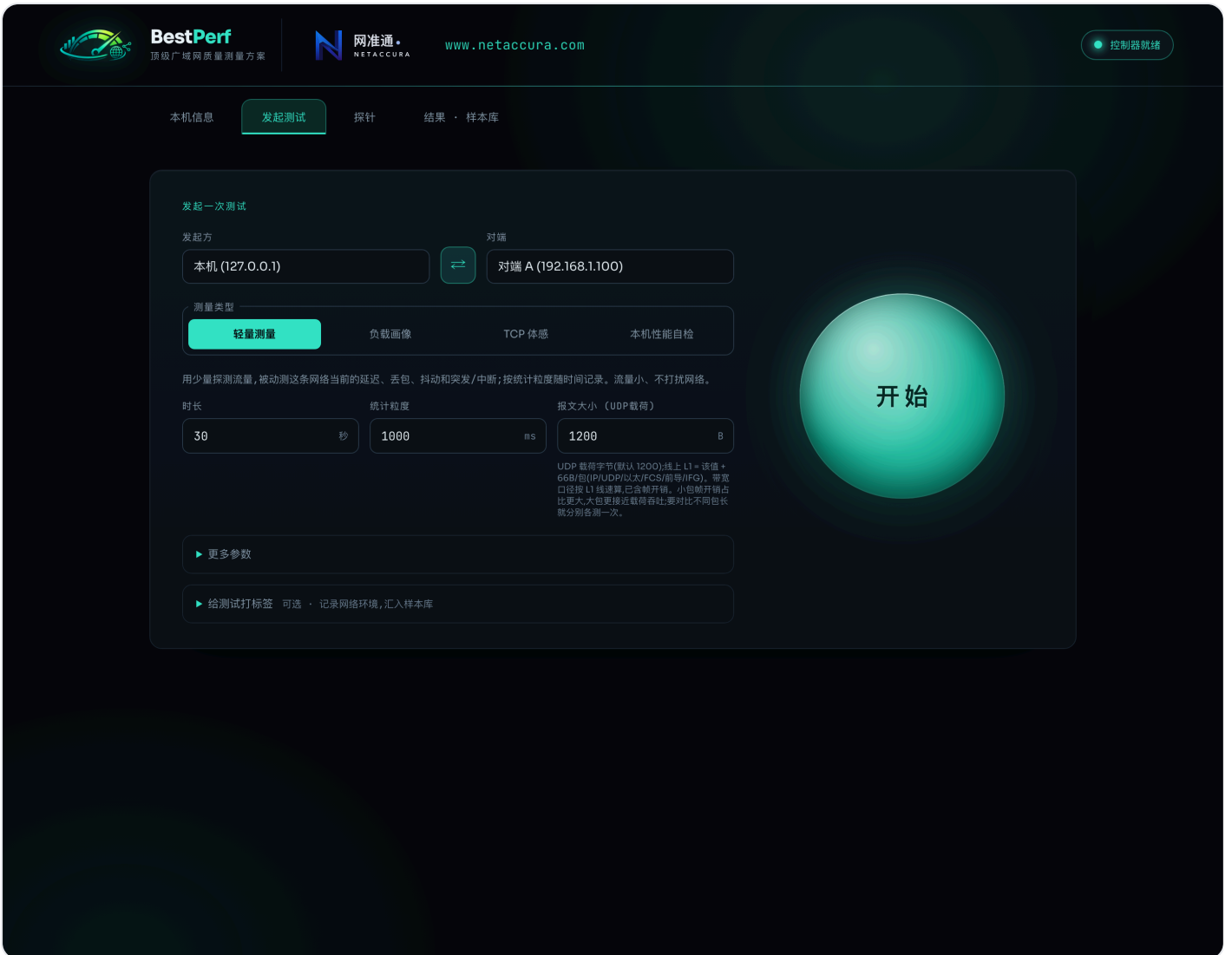
步骤 2：设置两端：

- 「发起方」下拉选「本机」。
- 「对端」下拉选您刚登记的那台（如「分支A」）。
- 中间那颗「⇄」交换按钮可以一键对调发起方和对端，想反过来测就点它。

步骤 3：「测量类型」里点「轻量测量」（默认就是它，通常不用改）。**步骤 4：设置参数：**

参数	含义	默认值
时长	测多久（秒）	30
统计粒度	多久取一段统计（ms）	1000
报文大小（UDP 载荷）	每个探测包的载荷字节数	1200

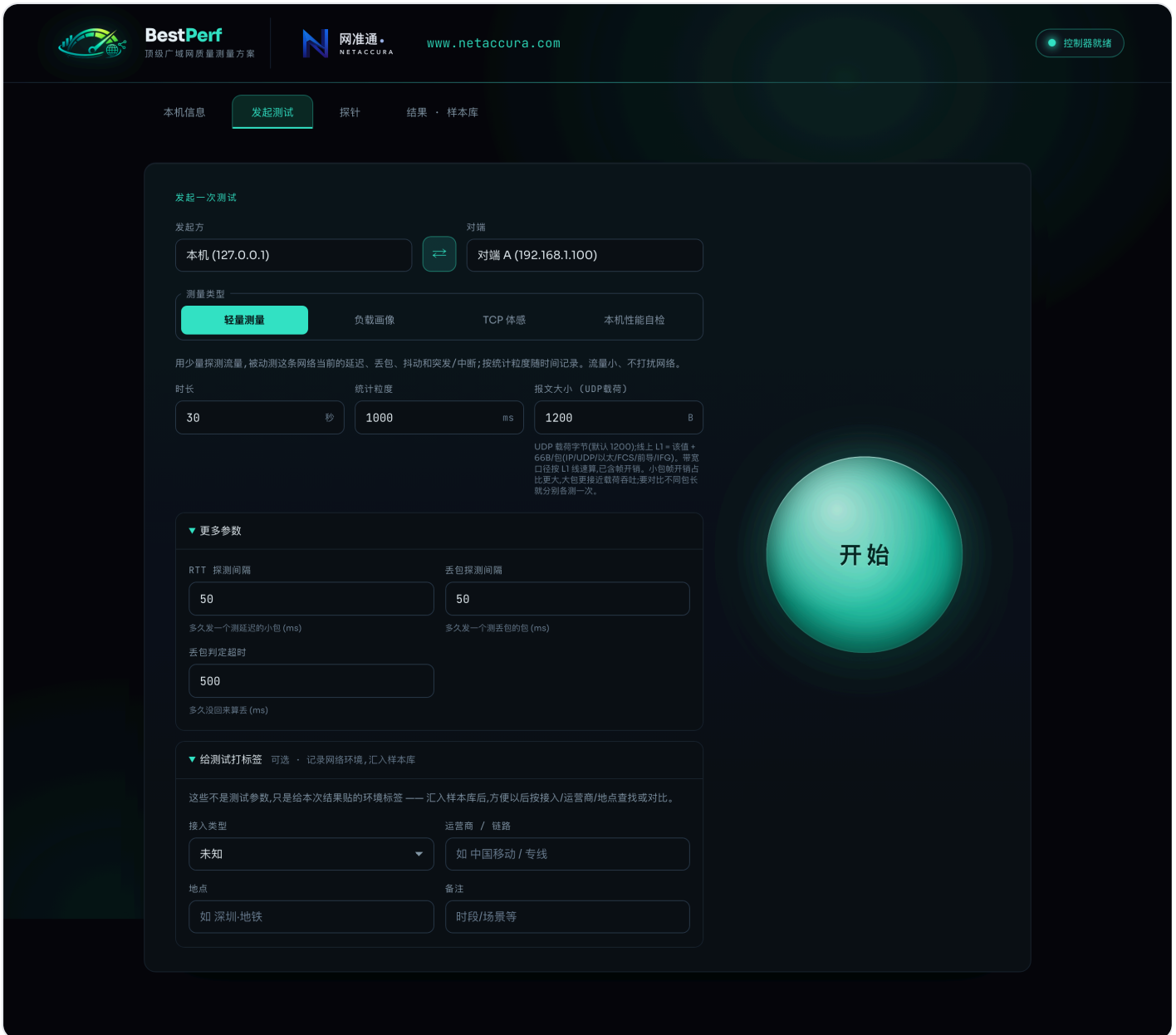
提示：第一次测，参数全部用默认值即可。时长 30 秒、报文 1200 字节是日常摸底的稳妥设置。



轻量测量就绪：对端、时长、报文大小都设好了，等着点「开始」。

步骤 5（可选）：想更细地控制或留个记录，可以展开两个折叠区：

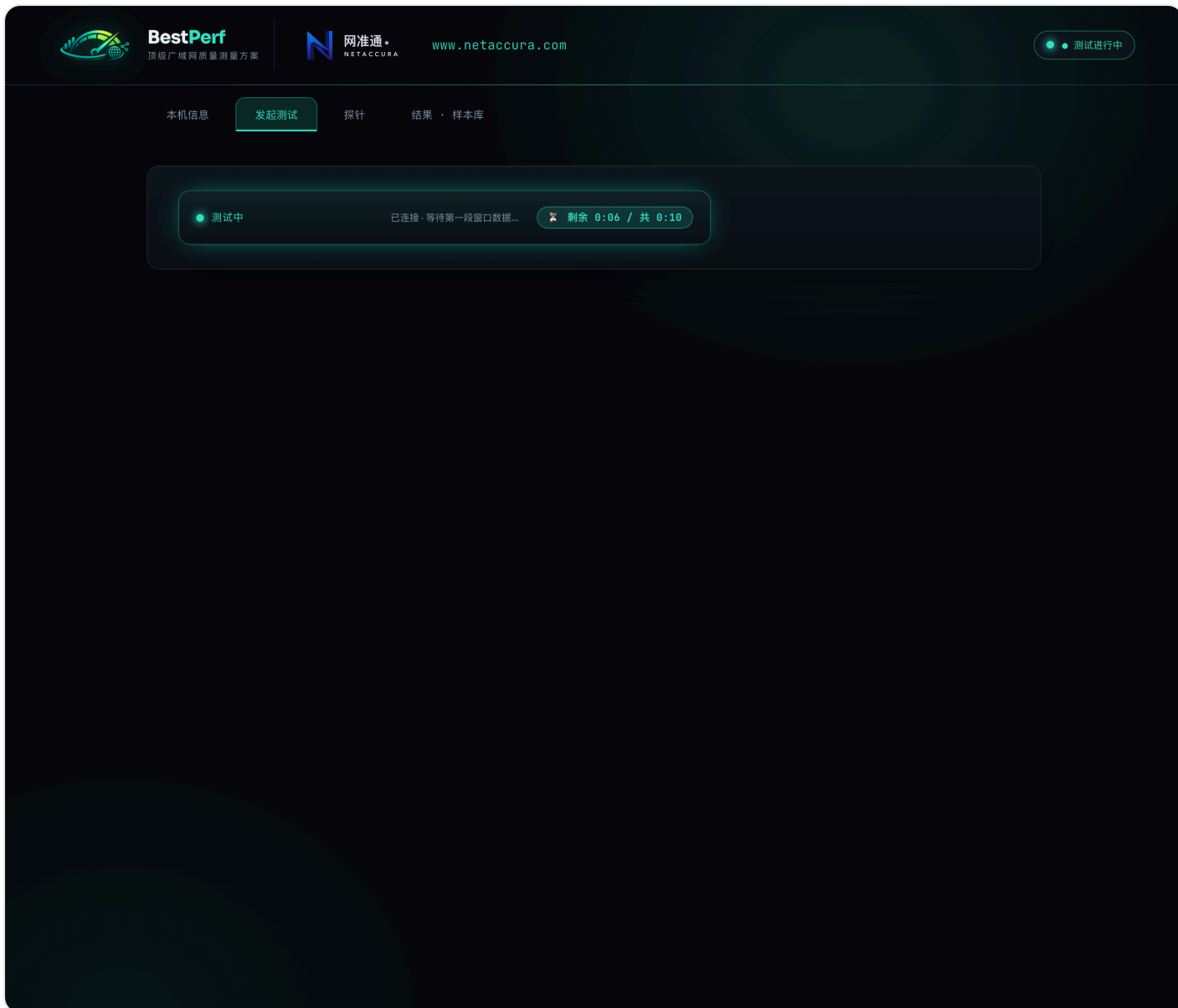
- 「更多参数」：调 RTT 探测间隔、丢包探测间隔、丢包判定超时等高级项。
- 「给测试打标签」：填接入类型（WiFi / 4G / 5G / 卫星 / 企业 WAN / 家庭宽带 / 有线局域网...）、运营商、地点、备注。这些不是测试参数，只是给本次结果贴的环境标签，汇入样本库后方便以后按接入方式、运营商、地点查找和对比。



展开后的高级参数与环境标签，按需填写，不填也能测。

步骤 6：点右侧发光的「开始」按钮。

界面会从发起表单切换成实时进度，显示剩余时间和实时的延迟、丢包等数字。测试期间耐心等它跑完即可。



测试进行中：发起表单换成实时进度块，顶部健康标记也变成「● 测试进行中」。

6.4 看结果

第一步：完成卡片 → 查看报告

测试跑完，实时进度块会变成「测试完成」卡片，上面有两个按钮：

- 「查看报告」：跳到「结果·样本库」页并自动打开本次详情。
- 「再发起一次」：换回发起表单，再测一遍。

点「查看报告」。

第二步：在「结果·样本库」里逐块看详情

「结果·样本库」页左边是历次 run 列表（每条带 **轻量 / 负载 / 自检** 标签、**完成 / 失败** 状态、以及您打的环境标签如「有线LAN」），右边是选中那条的详情。下面对着轻量测量的详情逐块讲。



本机信息 发起测试 探针 **结果 · 样本库**

结果 · 样本库

样本库 7 条

全部接入

搜索 run / 运营商 / 地点 / 备注

全选 已选 0 项

轻量 light---201-2026-06-14T10-13-57 完成

↓ 回放场景(通用格式) ↓ 网络模型 画像报告(PDF) ↓ 网络画像 删除样本

轻量 light---201-2026-06-1...
完成 06/14 18:13
有线LAN

负载 load-2026-06-14T10-09-...
完成 06/14 18:09
有线LAN

负载 load-2026-06-14T07-14-...
完成 06/14 15:14
有线LAN

轻量 light---201-2026-06-1...
完成 06/14 15:13
有线LAN

轻量 light---201-2026-06-1...
完成 06/14 15:10
有线LAN

轻量 light---201-2026-06-1...
完成 06/14 15:10
有线LAN

原始数据 session_config

环境标签: 有线LAN(自动)

综合 RTT

p50	60.73 ms
p95	60.87 ms
p99	60.95 ms
最大	61.11 ms
样本数	524
超时	-

发起方 → 对端 (A-B)

丢包率	2.86 %
到达间隔抖动 p95	0.27 ms
乱序率	0 %
重复率	0 %
中断次数	0

发起方 → 对端 (A-B) · 质量随时间

时长 14s · 14 个窗口



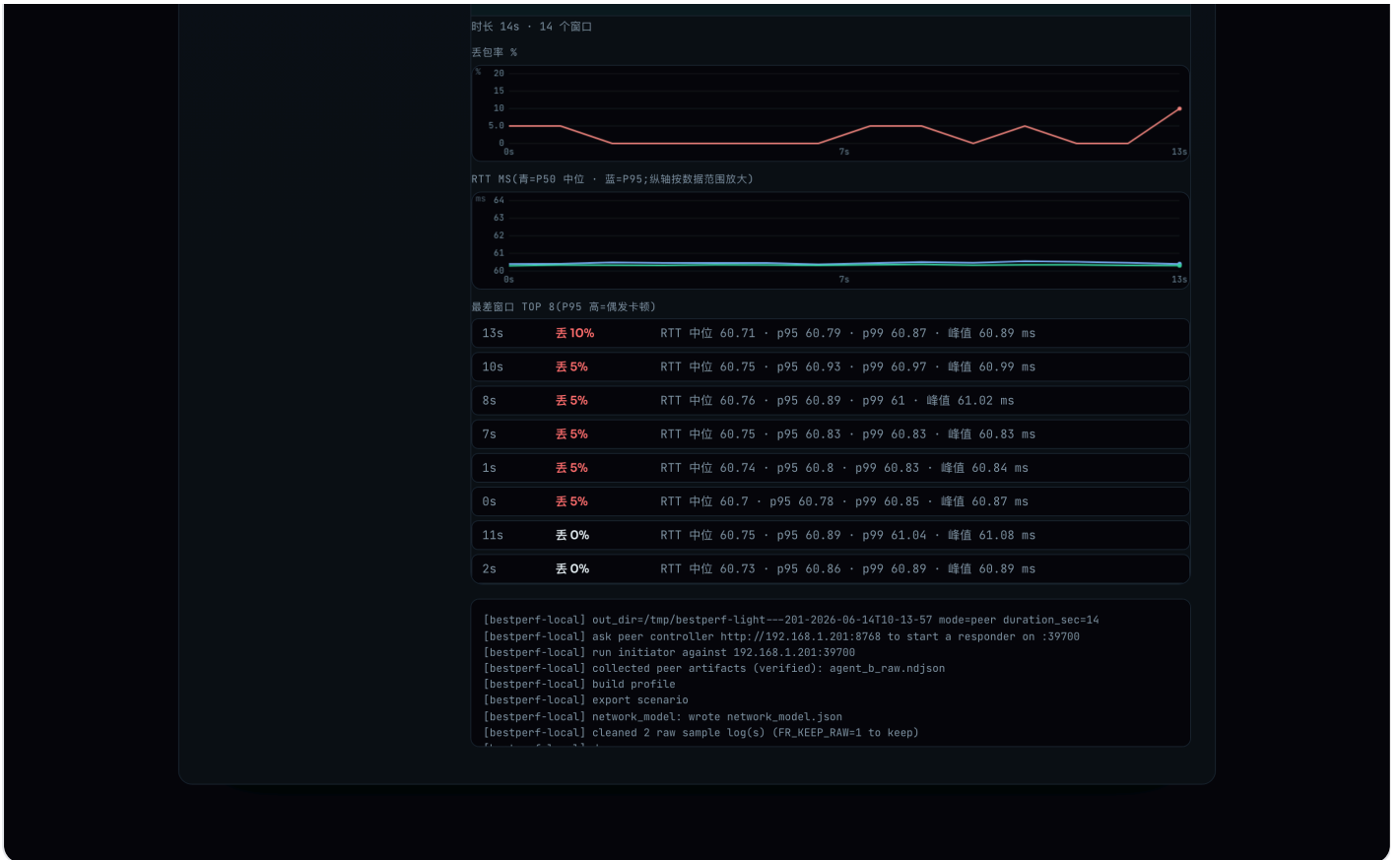
最差窗口 TOP 8 (P95 高=偶发卡顿)

11s	丢 10%	RTT 中位 60.77 · p95 60.86 · p99 60.86 · 峰值 60.86 ms
5s	丢 10%	RTT 中位 60.73 · p95 60.85 · p99 60.96 · 峰值 60.99 ms
3s	丢 5%	RTT 中位 60.75 · p95 60.93 · p99 60.93 · 峰值 60.93 ms
4s	丢 5%	RTT 中位 60.75 · p95 60.89 · p99 60.92 · 峰值 60.93 ms
8s	丢 5%	RTT 中位 60.78 · p95 60.87 · p99 60.91 · 峰值 60.91 ms
2s	丢 5%	RTT 中位 60.73 · p95 60.81 · p99 60.85 · 峰值 60.86 ms
1s	丢 0%	RTT 中位 60.75 · p95 60.92 · p99 61.07 · 峰值 61.11 ms
7s	丢 0%	RTT 中位 60.75 · p95 60.91 · p99 60.99 · 峰值 61.01 ms

对端 → 发起方 (B-A)

丢包率	2.5 %
到达间隔抖动 p95	0.11 ms
乱序率	0 %
重复率	0 %
中断次数	0

对端 → 发起方 (B-A) · 质量随时间



轻量测量详情全貌：上方综合 RTT，中间按方向分卡，下方质量随时间曲线。

① 综合 RTT（最上面一块）

字段	含义
p50	典型往返延迟（中位数，ms）
p95	尾部往返延迟（95% 的包都比它快，ms）
p99	更极端的尾部（99 分位，ms）
最大	本次见过的最大往返延迟（ms）
样本数	一共测了多少个 RTT 探测
超时	多少个探测包没等回来

关键理解：RTT 是往返时延，是准的。两端没有对时，所以 BestPerf 只给往返 RTT，不给单向延迟——别去找「单向多少毫秒」，界面上不会有，那是诚实，不是缺功能。

② 按方向的卡片（A→B 和 B→A 各一块）

每个方向单独一块，因为去和回的网络质量可能不一样：

字段	含义
丢包率	这个方向丢了百分之几的包
到达间隔抖动 p95	包到达节奏忽快忽慢的程度 (p95, ms)
乱序率	包到达顺序乱掉的比例
重复率	收到重复包的比例
中断次数	出现「一阵子完全没包」的次数

如果这个方向出过突发丢包、微中断或延迟尖峰，卡片下方还会有一行「事件」，列出每类事件出现几次、最长持续多久。

③ 质量随时间（每个方向卡片下方的曲线）

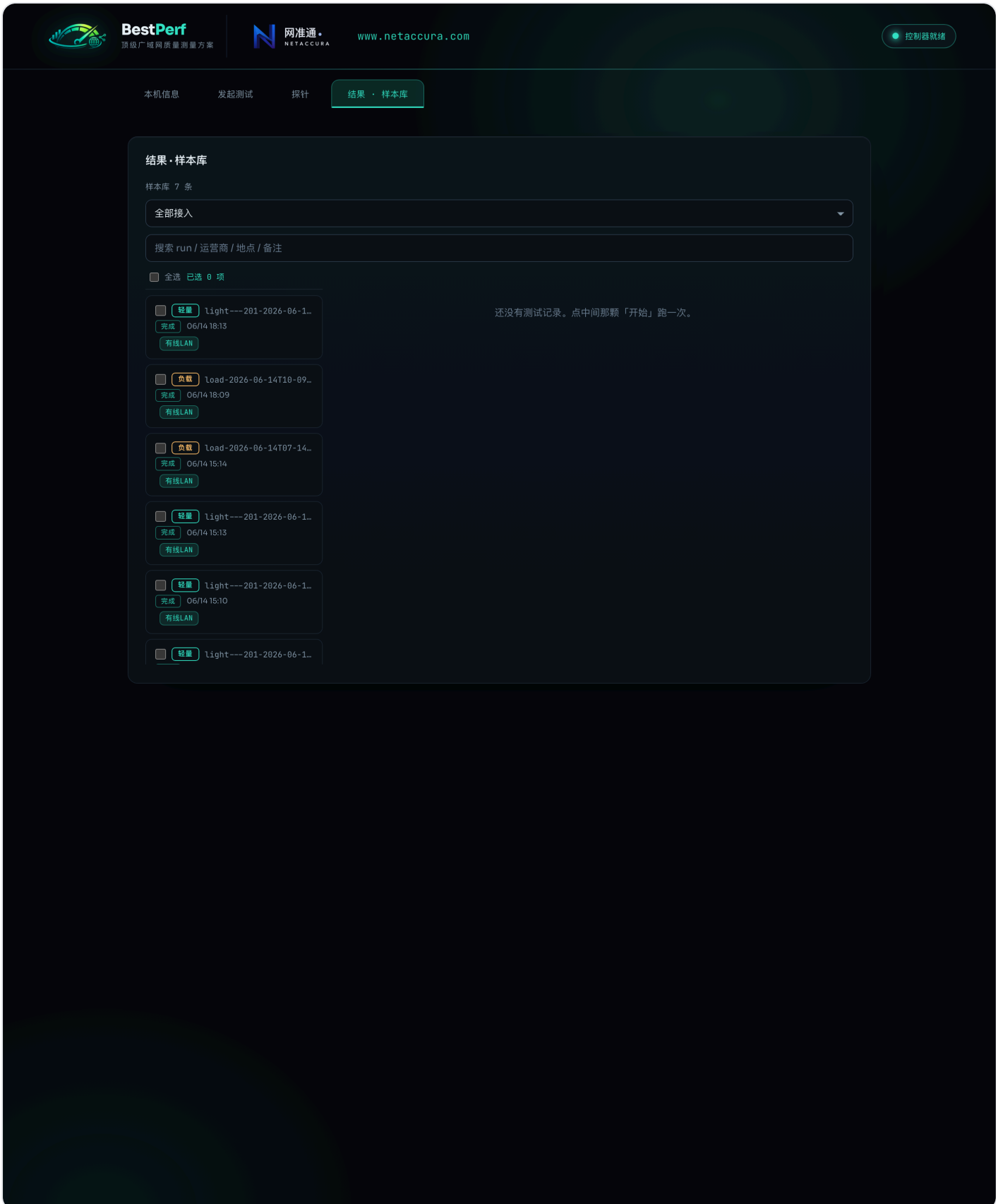
- 一条「丢包率 %」曲线：每段统计窗口丢了多少包，随时间画出来。全程没丢就标「全程 0% 丢包」。
- 一条「RTT ms」曲线：青线是 p50 中位、蓝线是 p95 尾部，随时间画出来。
- 下面还列出最差的几个窗口（逐窗行）：哪一秒丢得最多、那一秒的 RTT 中位 / p95 是多少，让您一眼定位到最糟糕的时刻。

6.5 下载回放文件

测出了网络画像，最有价值的一步是把它导出成回放文件，拿到实验室的 WAN 仿真器里复刻这段真实网络——这正是 BestPerf 「现网实测·复刻」的核心。

在详情页顶部有一排下载按钮。轻量测量通常能下载：


按钮	是什么	什么时候用
↓ 损伤仪场景 (可直接导入)	WE 原生格式 (<code>device_scenario</code>), 带延迟 + 丢包	导进 WAN 仿真器就能复刻这次测到的网络, 首选这个
↓ 回放场景 (通用格式)	通用 / 中立格式 (<code>wan_emulator_scenario</code>), 主要是延迟	给不支持上面专用格式的仿真器用
↓ 网络模型	本次提炼出的标准网络画像, 供长期积累 / 建模	做样本库沉淀用
📄 画像报告 (PDF)	排版好的报告, 浏览器打印→另存为 PDF	给人看、留档
↓ 网络画像	本次画像的原始数据 (延迟 / 抖动 / 丢包 / 乱序...)	给工具 / 脚本分析



详情顶部的下载按钮区：回放场景、网络模型、画像报告、网络画像一应俱全。

两种回放格式怎么选？

- 您的仿真器就是 WAN 仿真器 / 损伤仪本身 → 用「**损伤仪场景（可直接导入）**」，延迟和丢包都带上，复刻最完整。
- 用的是别家的、不认这个专用格式的仿真器 → 用「**回放场景（通用格式）**」，用中立格式，主要复刻延迟。

提示：详情顶部最右边还有「 删除样本」按钮，删掉这次记录。删除不可恢复，删之前确认一下。

6.6 关键理解与易错点

关键理解：BestPerf 测的是一段网络的「双向画像」，再导出成「回放文件」复刻现网。两台机器各跑一个控制台，用 UDP 互相打探测包，量出 RTT、双向丢包、抖动、乱序、重复、中断，提炼成网络画像；把回放文件导进 WAN 仿真器，就能在实验室里复刻这段现网网络。整个过程不用 SSH、不用 root、不用抓包。

最常见的易错点：结果显示 0ms / 0 丢包，并在详情顶部弹出橙色提示。

如果测完发现 RTT、丢包全是 0，而且详情上方出现一条橙色的△提示横幅，写着类似「本次测试没有收到任何回包（所有探测超时）.....通常是对端 responder 没启动 / 被残留进程占了 UDP 端口，或防火墙挡了 UDP.....本次统计无意义」——这不是网络完美，而是探测包根本没通。

注意：0ms / 0 丢包配橙色提示，几乎一定是下面三种情况之一：

1. 对端没打开「允许被别人测向本机」开关 —— 回 6.2 第一步，去对端把它打开。
2. 对端防火墙没放行 UDP —— Windows 双击「2 开放网络权限 BestPerf」；macOS 在弹窗里点「允许」；Linux 用 `ufw` 放行（见第四章）。
3. 对端有残留进程占了 UDP 端口 —— 清理对端残留进程后重测。

关键理解：橙色提示出现时，不要相信那份全 0 的结果——它没有任何意义。先按上面排查、把探测打通，再重测一次。更详细的排查步骤见 FAQ（常见问题）章节。

恭喜！ 您已经完整跑通了 BestPerf 的第一次测试——从自检热身，到登记对端、轻量测量，再到看懂结果、导出回放文件。接下来的章节会逐一展开负载画像、TCP 体感等更深入的测量类型。

第七章 功能详解（每个功能做什么·注意事项·测试效果）

本章把控制台里能用的每一个功能拆开讲清楚。每一节的结构都一样，方便你对照着看：

- **【做什么】** —— 这个功能到底量的是什么、解决什么问题。
- **【怎么操作 / 配什么】** —— 在界面的哪个位置、点哪个按钮、填哪些参数。
- **【注意事项 / 易错点】** —— 容易踩的坑、配置前提、常见误会。
- **【能得到什么测试效果】** —— 测完后在「结果·样本库」里能看到哪些数字、能导出哪些文件。

关键理解： BestPerf 的全部测量都靠两台机器各跑一个控制台、用 UDP 互相打探针完成。不用 SSH、不用 root、不用抓包。一台是发起方（本机），另一台是对端。除了「本机性能自检」只用本机、不需要对端之外，其余三种测量都需要对端先装好控制台、打开「允许被别人测向本机」开关，并把它的控制台地址 + 令牌给你（详见 7.5 探针管理）。

四种测量类型集中在顶部「发起测试」标签页的「测量类型」分段开关里：

测量类型	需要对端？	一句话
轻量测量 (light)	需要	小流量探针，量这条网络当前的延迟、丢包、抖动、乱序、重复、微中断
负载画像 (load)	需要	逐档加压找拐点/容量，再在拐点以内的工作点量延迟/丢包
TCP 体感 (tcp)	需要	按 RFC 6349 测应用真正拿到的上行/下行速度 + 应用层 RTT
本机性能自检 (selfcheck)	不需要	本机环回加压，测这台机器自己的 UDP 处理上限

7.1 轻量测量 (light)

【做什么】

轻量测量用秒级的小流量探针，被动地量出这条网络当前的质量画像，包含 6 项指标：

指标	含义
RTT 延迟	往返时延（一个包打过去、回过来的总时间）
双向丢包	A→B 方向和 B→A 方向各自独立的丢包率
抖动 (Jitter)	包到达节奏相对发送间隔的偏离（到达忽快忽慢）
乱序 (Reorder)	后发的包反而先到的比例
重复 (Duplicate)	同一个包被收到多次的比例
微中断 (Micro-outage)	一小段时间内一个包都没到（短暂掐断）

它流量很小、不打扰网络，适合先摸一遍底：这条线现在好不好用、卡在哪。

关键理解 —— 为什么丢包是「双向独立」的？ 网络两个方向的质量往往不一样（比如上行挤、下行空）。BestPerf 用带序号的单向流分别测 A→B 和 B→A，所以看到的是两个方向各自的丢包率，而不是把两边混成一个数。这正是后面「回放文件」能分方向复刻的基础。

关键理解 —— 为什么 RTT 用回环估计、不直接报「单向延迟」？ 探针只用各自机器的单调时钟，两台机器之间没有对时。在没对时的前提下硬报单向延迟会骗人（不对称网络里误差很大）。所以界面上逐窗显示的是真实测到的往返 RTT（同一时钟，准）；导出回放文件时才把 RTT 做一次有标注的对称对半拆分（`rtt_half_estimate`，置信度标为中等），仅供仿真器建模用，并不声称它是真单向延迟。

关键理解 —— 为什么丢包给「95% 上界」？ 短测量里如果一个包都没丢，并不等于「这条线 100% 不丢包」，只是这次没采到丢包样本。BestPerf 会在丢包率旁给出一个 **95% 置信上界**（样本全到时用三倍率法则估上界），并标注样本是否充足。这样「0% 丢包」永远不会被误读成「绝对不丢」——它的真实含义是「丢包率低于这个上界」。

【怎么操作 / 配什么】

步骤 1 打开「发起测试」标签页，「测量类型」选「轻量测量」（默认就是它）。

步骤 2 顶部两个下拉：左边「发起方」选 `本机`，右边「对端」选你登记好的那台机器。中间的交换按钮可以对调两端。

注意：发起方目前只能是「本机」。如果你想从对端那台机器发起，请去那台机器上打开它的控制台操作。在本机这边想看对端方向，直接选 本机 → 对端 就行，两个方向 BestPerf 都会量到。

步骤 3 设置参数（不改也能跑，用默认值即可）：

参数	默认值	说明
时长（秒）	30	测多久。想让「丢包上界」更可信就测长一点
统计粒度	1000 ms	多久汇总成一段（曲线/逐窗丢包的颗粒度）
报文大小（UDP 载荷）	1200 B	每个探针包的载荷字节数

展开「更多参数」还能调回环/单向探针的发送间隔、超时等（一般不用动）。展开「给测试打标签」可以记录接入方式、运营商/链路、地点、备注——这些会作为环境标签汇入样本库，方便日后检索（见 7.6）。

步骤 4 点右侧那颗发光的「开始」按钮。页面会原地切换成**实时进度**：按方向显示收发速率、丢包率、RTT p50/p95、到达间隔抖动 p95，还有随时间走的丢包曲线和 RTT 曲线。测完自动回到结果。

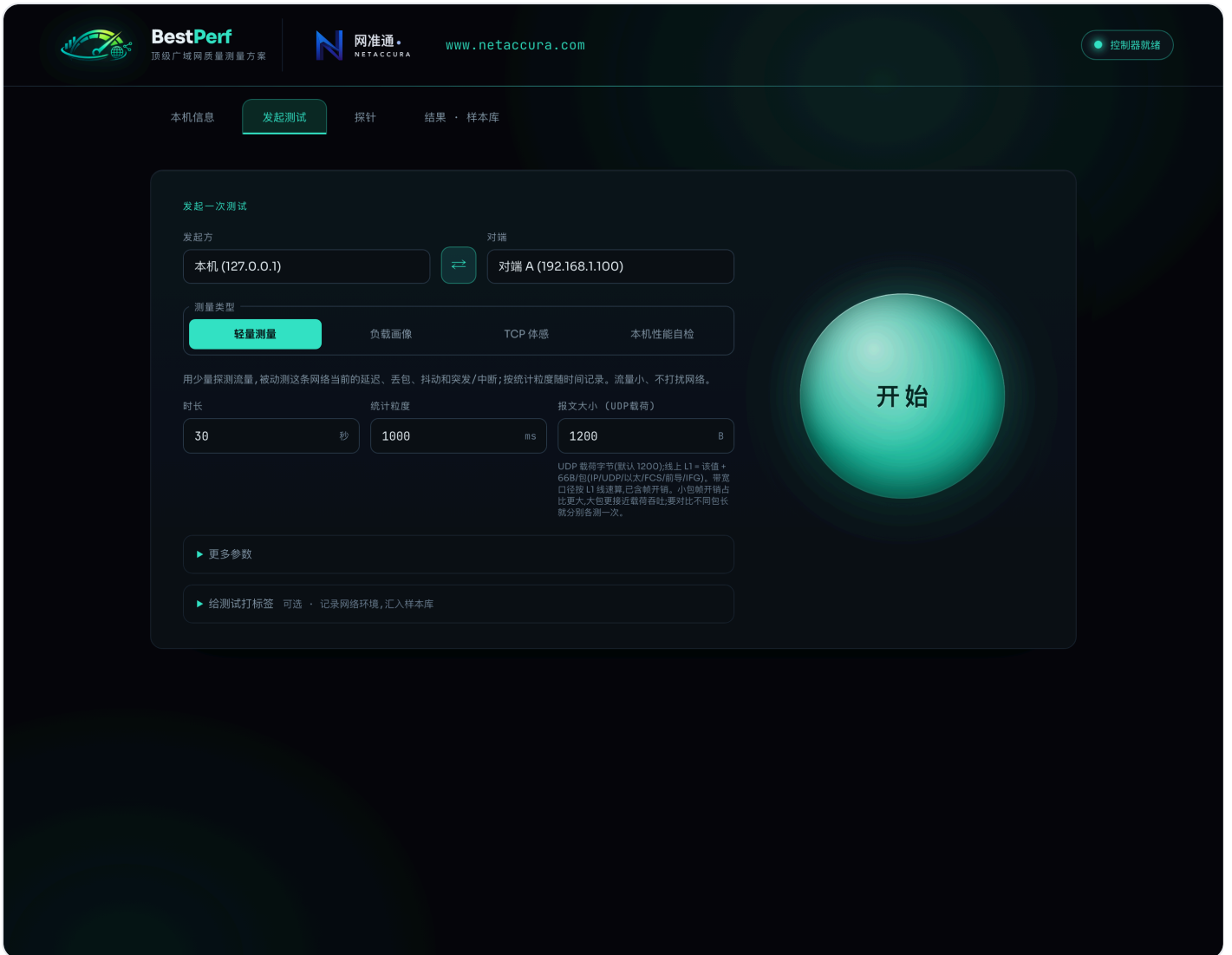


图 7-1 「发起测试」页：测量类型选「轻量测量」，发起方=本机、对端=已登记的探针，右侧「开始」即可。

【注意事项 / 易错点】

注意：对端必须也在跑控制台、并打开了「允许被别人测向本机」开关，否则发起方联系不上它，测出来会是「所有探测超时、统计全 0」。如果详情页顶部出现红色横幅提示「没有收到任何回包」，通常是对端 responder 没启动、UDP 端口被残留进程占了，或防火墙挡了 UDP——清理后重测即可。

注意：完整的双向画像需要对端那一侧的日志。本机发起端跑完后，若双向画像还没合成，详情页会出现「对端数据收集」卡片，点「开始收集」从对端控制台拉取（支持断点续传，网络差可暂停稍后续传，数据不丢、不会退化成往返估计）。

【能得到什么测试效果】

在「结果·样本库」点开这条 **轻量** 记录，详情里能看到：

- **综合 RTT**：p50 / p95 / p99 / 最大 / 样本数 / 超时数。
- **按方向 (A→B、B→A) 的卡片**：丢包率、到达间隔抖动 p95、乱序率、重复率、中断次数；下面还有「事件」一行，把突发丢包、微中断、RTT 尖峰按类型聚合（例如「微中断 ×3（最长 120ms 无包到达）」）。
- **每个方向的「质量随时间」**：丢包率曲线 + RTT (p50/p95) 曲线，外加「最差窗口 Top 8」列表，专门把偶发卡顿挑出来。



本机信息 发起测试 探针 **结果 · 样本库**

结果 · 样本库

样本库 7 条

全部接入

搜索 run / 运营商 / 地点 / 备注

全选 已选 0 项

轻量 light---201-2026-06-14T10-13-57 完成

↓ 回放场景(通用格式) ↓ 网络模型 📄 画像报告(PDF) ↓ 网络画像 🗑️ 删除样本

轻量 light---201-2026-06-1...
完成 06/14 18:13
[有线LAN](#)

负载 load-2026-06-14T10-09-...
完成 06/14 18:09
[有线LAN](#)

负载 load-2026-06-14T07-14-...
完成 06/14 15:14
[有线LAN](#)

轻量 light---201-2026-06-1...
完成 06/14 15:13
[有线LAN](#)

轻量 light---201-2026-06-1...
完成 06/14 15:10
[有线LAN](#)

轻量 light---201-2026-06-1...
[有线LAN](#)

原始数据 session_config

环境标签: 有线LAN(自动)

综合 RTT

p50	60.73 ms
p95	60.87 ms
p99	60.95 ms
最大	61.11 ms
样本数	524
超时	-

发起方 → 对端 (A→B)

丢包率	2.86 %
到达间隔抖动 p95	0.27 ms
乱序率	0 %
重复率	0 %
中断次数	0

发起方 → 对端 (A→B) · 质量随时间

时长 14s · 14 个窗口



最差窗口 TOP 8 (P95 高=偶发卡顿)

11s	丢 10%	RTT 中位 60.77 · p95 60.86 · p99 60.86 · 峰值 60.86 ms
5s	丢 10%	RTT 中位 60.73 · p95 60.85 · p99 60.96 · 峰值 60.99 ms
3s	丢 5%	RTT 中位 60.75 · p95 60.93 · p99 60.93 · 峰值 60.93 ms
4s	丢 5%	RTT 中位 60.75 · p95 60.89 · p99 60.92 · 峰值 60.93 ms
8s	丢 5%	RTT 中位 60.78 · p95 60.87 · p99 60.91 · 峰值 60.91 ms
2s	丢 5%	RTT 中位 60.73 · p95 60.81 · p99 60.85 · 峰值 60.86 ms
1s	丢 0%	RTT 中位 60.75 · p95 60.92 · p99 61.07 · 峰值 61.11 ms
7s	丢 0%	RTT 中位 60.75 · p95 60.91 · p99 60.99 · 峰值 61.01 ms

对端 → 发起方 (B→A)

丢包率	2.5 %
到达间隔抖动 p95	0.11 ms
乱序率	0 %
重复率	0 %
中断次数	0

对端 → 发起方 (B→A) · 质量随时间

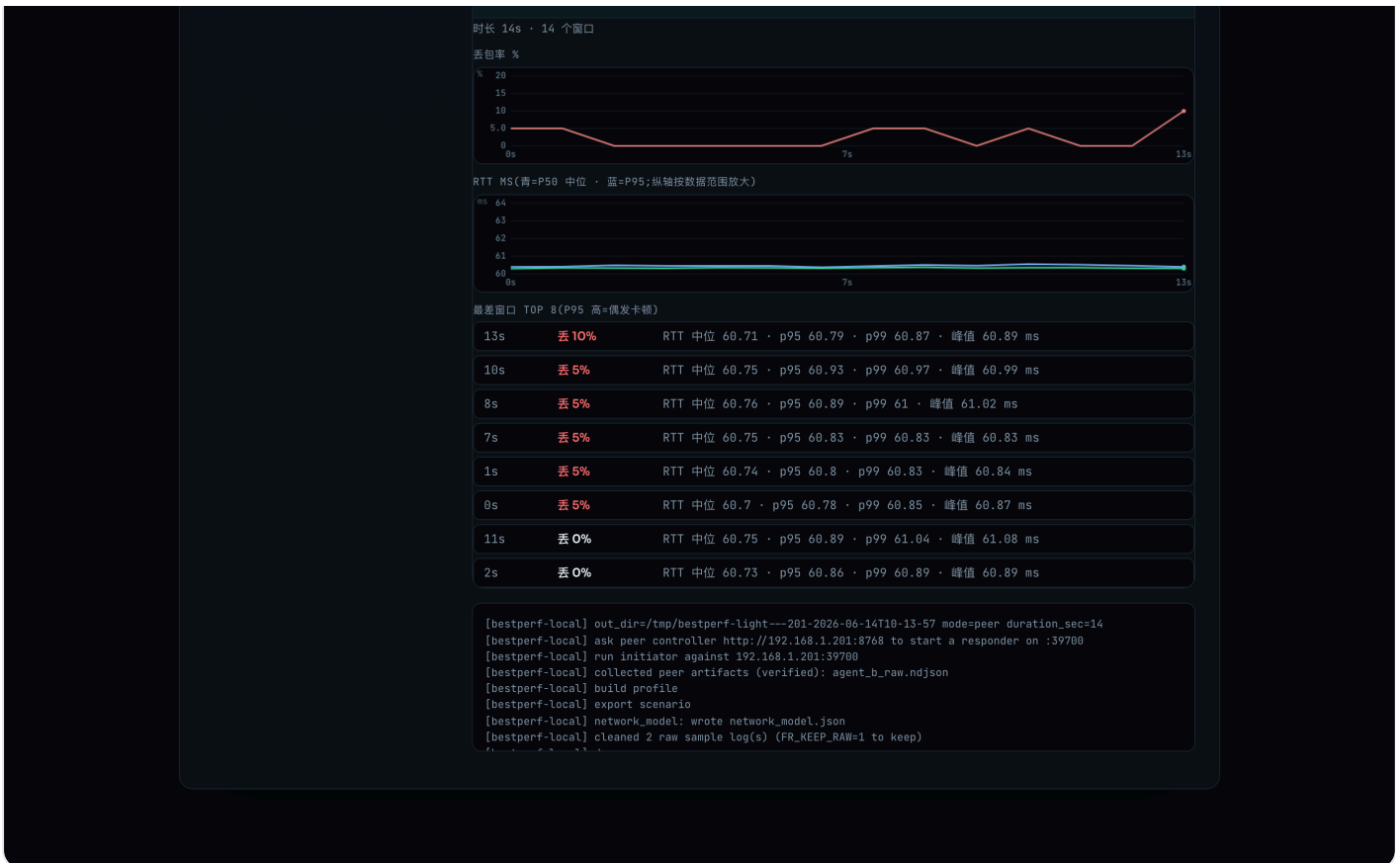


图 7-2 轻量测量详情页：综合 RTT 概况 + 分方向指标 + 随时间的丢包/延迟曲线。

详情顶部的下载按钮可导出：**网络画像 (JSON)**、**画像报告 (PDF)**、以及给仿真器用的回放场景（专用 / 通用两种格式，怎么选见 7.6）。

7.2 负载画像 (load)

【做什么】

负载画像主动从小到大逐档加压，回答三个问题：

1. 这条网络能扛多大？ —— 一边加大发送速率，一边盯着延迟和到达速率，找到拐点（饱和点）：延迟开始上翘 / 到达速率封顶 / 丢包起点，三者哪个先到就以哪个为准。
2. 拐点对应的容量是多少？ —— 给出可用容量（Mbps, L1 线速率，含帧开销），并附一个「应用层」口径（UDP 载荷）方便和 TCP 体感对照。
3. 在能稳定承载的负载下，质量怎么样？ —— 在拐点以内的工作点量延迟/丢包，并拟合一个 **Gilbert-Elliott 双态突发丢包模型**（描述网络在「好/坏」两态间切换、坏态成串丢包的特性）。

重点澄清 —— 会不会把对端打死？ 不会。 负载画像不是「往死里灌」的压测。它先用逐档爬坡找出拐点，然后退回到拐点以内的工作点去做质量刻画——也就是说，长时间的测量是发生在网络能稳定承载的负载上的，不是顶着饱和点猛打。爬坡阶段每档只停留很短的时间（约 1 秒一档）用来探边界，找到边界就收手。所以它量的是「满负荷附近还稳的那个点」的表现，而不是把链路或对端压垮。

【怎么操作 / 配什么】

步骤 1 「发起测试」页，「测量类型」选「**负载画像**」。

步骤 2 选好「发起方（本机）」和「对端」。

步骤 3 负载画像会多出一个「方向」选择：

方向	含义
双向	同时给两个方向加压（控制台默认，最贴近真实争抢）
发起方→对端	只压 A→B 这一个方向
对端→发起方	只压 B→A 这一个方向

步骤 4 设置参数：

参数	默认值	说明
刻画时长（秒）	30	在工作点上刻画质量的时长
带宽上限（L1线速）	50 Mbps	爬坡时尝试加到的最高速率上限
报文大小（UDP 载荷）	1200 B	每个负载包的载荷字节数

注意： 时间线粒度（负载质量每隔多久取一段，默认 1000 ms）不在上面这一行主参数里，而是收在「**更多参数**」（高级）折叠区内——主参数行只有时长 / 带宽上限 / 报文大小。

步骤 5 点「开始」。实时页会画出「**发出 (offered) vs 到达 (delivered)**」两条吞吐曲线、当前发送/到达速率、发包数和丢包率——你能亲眼看到加压到某一档时到达速率封顶、丢包抬头，那就是拐点。

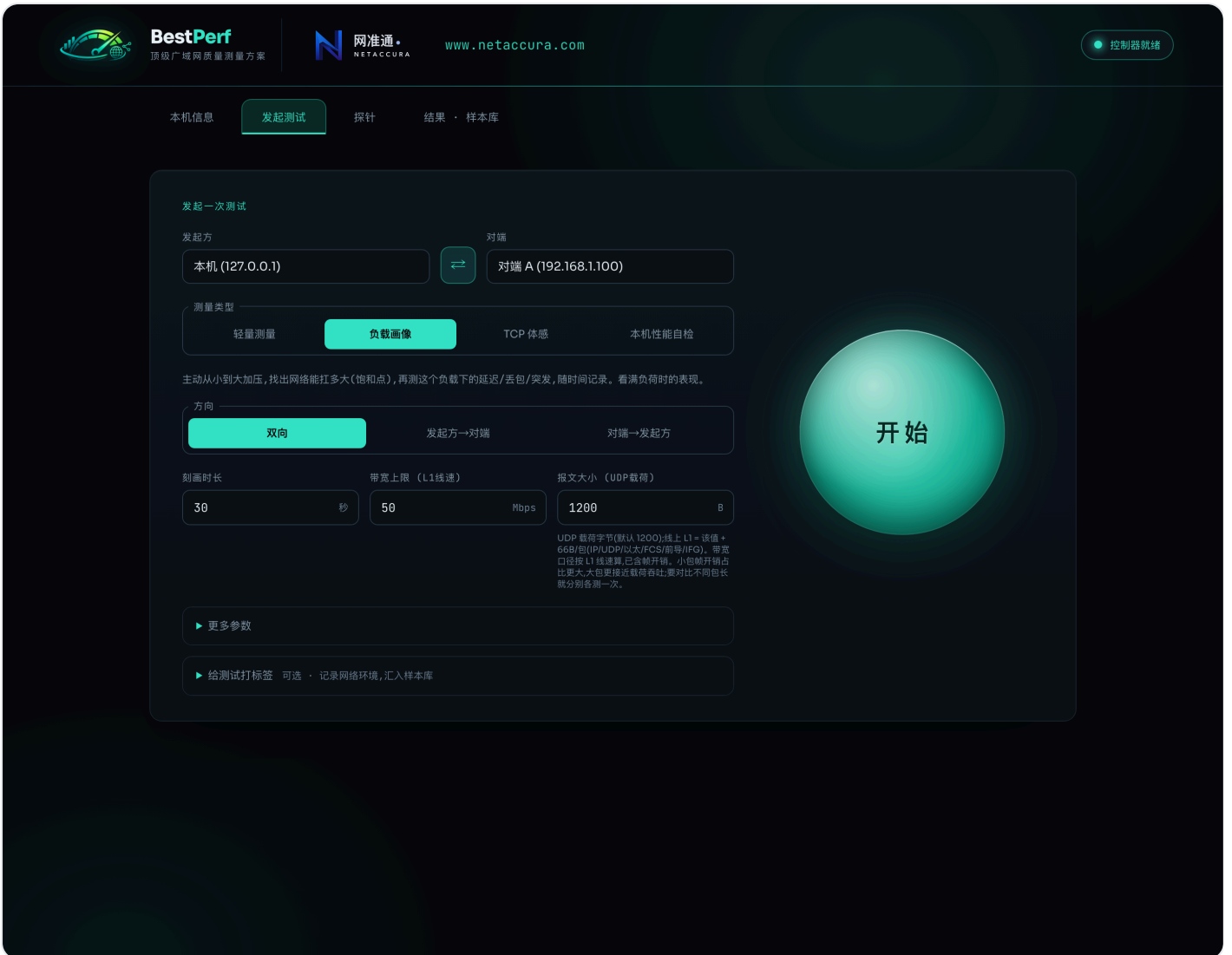


图 7-3 「负载画像」表单：比轻量测量多出「方向」分段（双向/发起方→对端/对端→发起方）。

【注意事项 / 易错点】

注意：如果详情里出现「 Δ 受设备限制」的提示，说明这次没压到网络的真实上限，而是本机自己的 **UDP 处理能力先到顶了**。这时容量数字反映的是机器瓶颈而非网络瓶颈——请先跑一次「本机性能自检」（见 7.4）确认本机能跑多快，再回头看这条负载结果。

提示：丢包数字的可信度，BestPerf 会如实标注：**实测丢包（对端收包记录） = forward truth** 比 **丢包估计（往返近似）** 更可信。控制台默认会用对端的收包记录把每个方向的丢包修正到「前向真值」。详情里还会给出 95% 上界和样本是否充足，**样本偏少时会提示「延长时长更可信」**。

【能得到什么测试效果】

详情页（[负载](#) [记录](#)）按方向展示：

- **容量 / 工作点**：可用容量（Mbps L1 线速）、应用层容量、工作负载、饱和判据（时延上翘 / 投递封顶 / 丢包起点）、工作点丢包、工作点 RTT p50。
- **突发丢包模型 (Gilbert-Elliott)**：丢包率、平均突发长度、突发阵数、 p （进坏态概率）、 r （出坏态概率）。
- **工作点 · 质量随时间**：工作点上的丢包与 RTT 曲线 + 最差窗口列表。
- **负载 vs 质量 (爬坡)**：每一档「发出→到达」对应的丢包和 RTT，把加压曲线逐档列出来。



本机信息 发起测试 探针 **结果 · 样本周**

结果 · 样本周

样本周 7 条

全部接入

搜索 run / 运营商 / 地点 / 备注

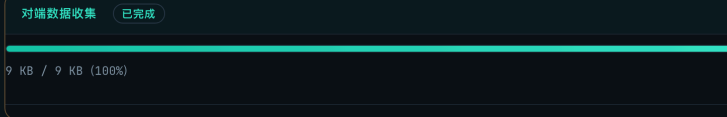
全选 已选 0 项

负载 Load-2026-06-14T10-09-55

轻量 light---201-2026-06-1...
完成 06/14 18:13

负载 Load-2026-06-14T10-09-...
完成 06/14 18:09

原始数据 session_config



环境标签: 有线LAN (自动)

发起方 → 对端 (A→B)

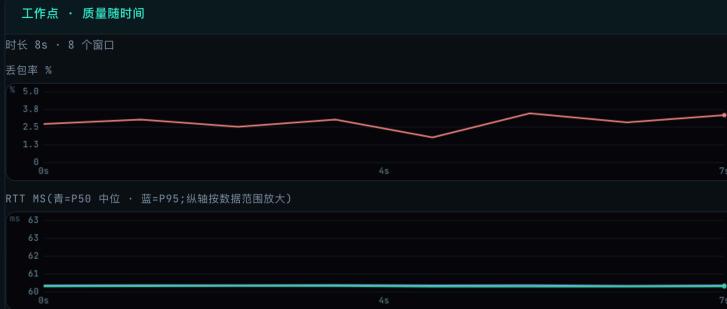
容量 / 工作点	
可用容量	28.31 Mbps L1线速
可用容量(应用层)	26.84 Mbps payload
工作负载	16 Mbps L1线速
饱和判据	时延上翘
工作点丢包	2.85%
工作点 RTT p50	60.3 ms

容量/吞吐为 L1 线速率(含 IP/UDP/以太/FCS/前导/IFG 帧开销,即链路真实占用);“应用层”为 UDP payload,可与下方 TCP 体感吞吐直接对照。

突发丢包模型 (Gilbert-Elliott)	
丢包率	2.85%
平均突发长度	1.03 包
突发阵数	350
p (进坏态)	0.03
r (出坏态)	0.97

测量方式: 实测丢包(对端收包记录)

置信: 丢包 2.85% · 95%上界 < 3.1% · 样本 12640



最差窗口 TOP 8(P95 高=偶发卡顿)

5s	丢 3.48%	RTT 中位 60.29 · p95 60.36 · p99 60.39 · 峰值 60.93 ms
7s	丢 3.35%	RTT 中位 60.3 · p95 60.35 · p99 60.36 · 峰值 60.38 ms
3s	丢 3.04%	RTT 中位 60.32 · p95 60.36 · p99 60.38 · 峰值 60.45 ms
1s	丢 3.04%	RTT 中位 60.31 · p95 60.36 · p99 60.38 · 峰值 60.4 ms
6s	丢 2.85%	RTT 中位 60.29 · p95 60.33 · p99 60.35 · 峰值 60.41 ms
0s	丢 2.72%	RTT 中位 60.3 · p95 60.35 · p99 60.37 · 峰值 60.54 ms
2s	丢 2.53%	RTT 中位 60.32 · p95 60.35 · p99 60.38 · 峰值 60.41 ms

4s 丢 1.77% RTT 中位 60.29 · p95 60.35 · p99 60.39 · 峰值 60.4 ms

负载 vs 质量 (爬坡)

1→0.96 Mbps	丢 4.38% · RTT 60.31ms
2→1.92 Mbps	丢 3.64% · RTT 60.3ms
4→3.89 Mbps	丢 2.78% · RTT 60.28ms
8→7.75 Mbps	丢 3.11% · RTT 60.28ms
16→15.52 Mbps	丢 3.01% · RTT 60.29ms
32→29.16 Mbps	丢 8.87% · RTT 218.63ms

对端 → 发起方 (B→A)

容量 / 工作点

可用容量	28.32 Mbps L1线速
可用容量(应用层)	26.84 Mbps payload
工作负载	16 Mbps L1线速
饱和判据	时延上翘
工作点丢包	3.09%
工作点 RTT p50	60.31 ms

容量/吞吐为 L1 线速率(含 IP/UDP/以太/FCS/前导/IFG 帧开销,即链路真实占用);“应用层”为 UDP payload,可与下方 TCP 体感吞吐直接对照。

突发丢包模型 (Gilbert-Elliott)

丢包率	3.09%
平均突发长度	1.03 包
突发阵数	378
p (进环态)	0.03
r (出环态)	0.97

测量方式:实测丢包(对端收包记录)

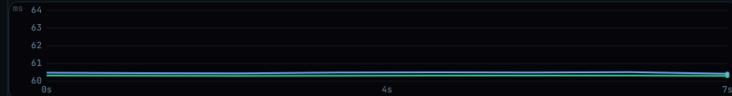
置信:丢包 3.09% · 95%上界 < 3.35% · 样本 12640

工作点 · 质量随时间

时长 8s · 8 个窗口



RTT MS (青=P50 中位 · 蓝=P95;纵轴按数据范围放大)



最差窗口 TOP 8(P95 高=偶发卡顿)

4s	丢 3.48%	RTT 中位 60.31 · p95 60.47 · p99 60.48 · 峰值 60.54 ms
3s	丢 3.42%	RTT 中位 60.31 · p95 60.46 · p99 60.51 · 峰值 60.52 ms
1s	丢 3.35%	RTT 中位 60.3 · p95 60.43 · p99 60.48 · 峰值 60.49 ms
6s	丢 3.23%	RTT 中位 60.32 · p95 60.48 · p99 60.51 · 峰值 60.52 ms
7s	丢 3.1%	RTT 中位 60.29 · p95 60.4 · p99 60.49 · 峰值 60.55 ms
5s	丢 2.97%	RTT 中位 60.32 · p95 60.46 · p99 60.5 · 峰值 60.53 ms
2s	丢 2.66%	RTT 中位 60.3 · p95 60.42 · p99 60.5 · 峰值 60.53 ms
0s	丢 2.47%	RTT 中位 60.31 · p95 60.45 · p99 60.49 · 峰值 60.5 ms

负载 vs 质量 (爬坡)

1→0.98 Mbps	丢 1.94% · RTT 60.31ms
2→1.95 Mbps	丢 2.33% · RTT 60.32ms
4→3.87 Mbps	丢 3.19% · RTT 60.3ms
8→7.78 Mbps	丢 2.75% · RTT 60.3ms
16→15.55 Mbps	丢 2.84% · RTT 60.31ms
32→29.17 Mbps	丢 8.87% · RTT 218.62ms

```
[bestperf-local] out_dir=/tmp/bestperf-load-2026-06-14T10-09-55 mode=load_bidir duration_sec=6
[bestperf-local] ask peer controller http://192.168.1.201:8768 to start a bidir-peer endpoint on :39700
[bestperf-local] run bidir initiator against 192.168.1.201:39700
[bestperf-local] collected peer artifacts (verified): load_report_peer.json, forward_record_peer.json
```

```
[bestperf-local] recompute A→B (forward truth from peer): bperf-load: corrected /tmp/bestperf-load-2026-06-14T10-09-55/load_report.json to forward truth (12280 peer-received working packets)
[bestperf-local] recompute B→A (forward truth from us): bperf-load: corrected /tmp/bestperf-load-2026-06-14T10-09-55/load_report_peer.json to forward truth (12250 peer-received working packets)
[bestperf-local] recompute bperf-load: peer-received working packets /tmp/bestperf-load-2026-06-14T10-09-55
```

图 7-4 负载画像详情页：容量与工作点、Gilbert-Elliott 突发模型、随负载变化的质量曲线。

可导出：负载报告（PDF）、负载报告（JSON，按方向打包完整原始数据），以及回放场景（专用 / 通用格式，带延迟 + 丢包）。

7.3 TCP 体感 (tcp)

【做什么】

TCP 体感按 RFC 6349 测应用真正能拿到的吞吐：分别测上行和下行的 goodput（有效吞吐），再加一个应用层 RTT。

关键理解 —— 物理容量 vs 应用实得，差在哪？ 7.2 负载画像 用 UDP 量的是网络的物理容量（链路最多能塞多少）。但 TCP 不会顶着上限猛冲——它一遇到丢包/排队就主动退让（拥塞控制）。所以同一条线上，TCP 实际跑到的速度通常低于 UDP 测出的物理容量，这个差值正是「网络容量」和「应用体感」的鸿沟。两个数放在一起看，才知道「线有多宽」和「应用实际享受到多少」。

【怎么操作 / 配什么】

步骤 1 「发起测试」页，「测量类型」选「TCP 体感」。

步骤 2 选好发起方（本机）和对端。TCP 体感没有「方向」选择——它总是把上行、下行分开各测一遍（这样两个方向不会互相抢瓶颈带宽，数字才干净）。

步骤 3 参数主要是「每向时长」（每个方向各测多久）。

步骤 4 点「开始」。实时页会显示上行、下行的吞吐曲线和当前 RTT。

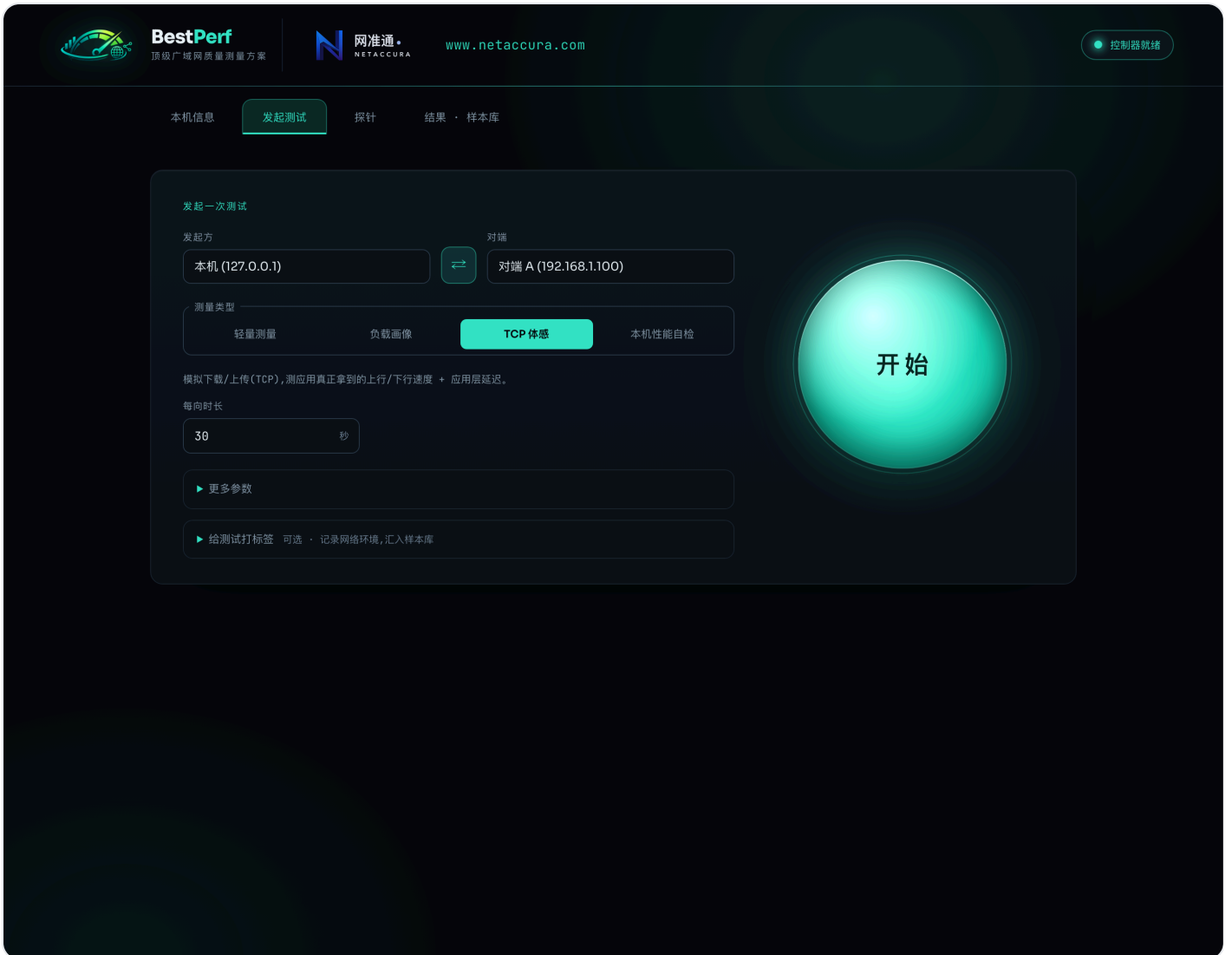


图 7-5 「TCP 体感」表单：无方向选择，按 RFC 6349 分别测上行与下行。

【注意事项 / 易错点】

注意： TCP 体感走的是 TCP 端口（控制台默认 39800 段），和轻量/负载用的 UDP 探测端口不是一回事。被别人测向本机时，对端开放网络权限会把这些端口一并放行（Windows 双击「2 开放网络权限 BestPerf」、macOS/Linux 见安装章节）。

提示： TCP 吞吐前一两秒是慢启动爬坡，不代表稳态。详情里会给出「去慢启动后的稳态平均」和「整段平均」两个口径，看稳态更接近真实体感。

【能得到什么测试效果】

详情页（TCP 记录）展示：

- 上行 / 下行 稳态吞吐 (去慢启动) 与 整段平均 (Mbps)。
- 应用层 RTT、测试时长。
- 稳定度 (每秒): 上行、下行的「最低 / 波动」, 看跑得稳不稳。
- TCP 吞吐随时间 曲线 (上行青、下行琥珀)。

可导出 **TCP 报告 (JSON)**。把它和负载画像的容量并排, 就能直观看出「物理容量 vs 应用实得」的差距。

7.4 本机性能自检 (selfcheck)

【做什么】

本机性能自检只测这台机器自己: 用环回 (loopback) 从小到大加压、完全不走网络, 得出两个数:

- 这台机器自己的 **UDP 处理上限** (能扛多大)。
- 负载下的延迟 (压上去之后延迟从哪个点开始上翘)。

它的用途只有一个: 分清瓶颈到底在网络, 还是在本机。

关键理解 —— 什么时候先跑它? 当你做负载画像、发现容量上不去或被标了「 Δ 受设备限制」时, 先跑一次**本机性能自检**。如果环回 (不走网络) 就已经只能跑到、比如 80 Mbps, 那网络测出来的 80 Mbps 就是被你这台机器卡住的, 不是网络的真实上限。反过来, 如果自检能轻松跑到几百 Mbps, 而网络测量只有几十, 那瓶颈才真的在网络上。先把本机底摸清, 负载画像的数字才读得准。

【怎么操作 / 配什么】

步骤 1 「发起测试」页, 「测量类型」选「**本机性能自检**」。

步骤 2 选完这个类型后, 界面会自动隐藏「发起方 / 对端」两端选择和「方向」选择 (自检只测本机, 不需要选两端) —— 因为它根本不需要对端。你不用登记任何探针就能跑。

步骤 3 直接点「开始」。它在本机环回上跑一段短爬坡即可。

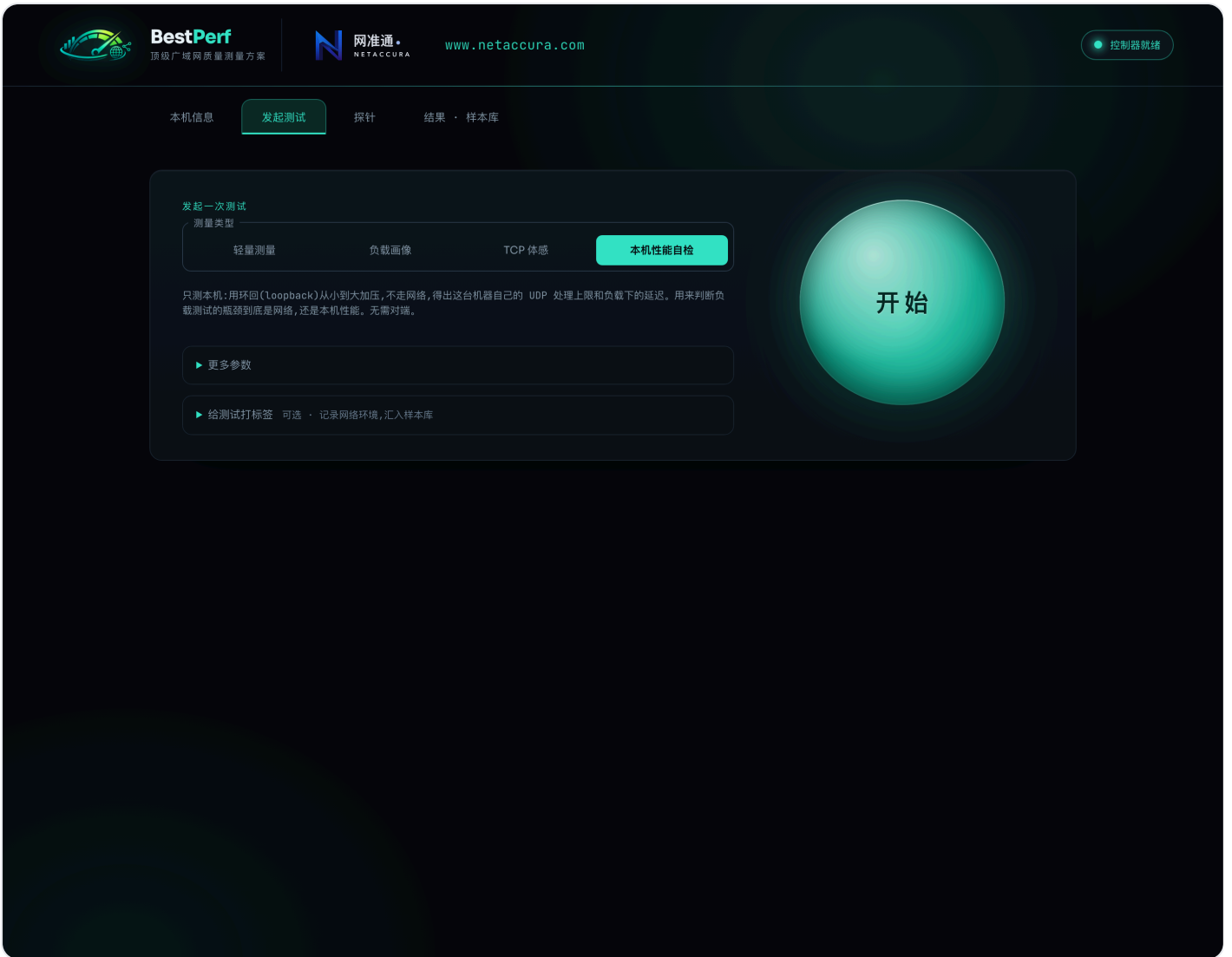


图 7-6 「本机性能自检」表单：自动隐藏对端与方向选择，无需任何对端即可开跑。

【注意事项 / 易错点】

注意：自检测的是纯本机环回，结果里的延迟是这台机器自己处理 UDP 的延迟，和网络无关。别拿它当网络延迟看。

提示：换了机器、或机器上跑了别的吃 CPU 的程序后，建议重跑一次自检——本机的处理上限会随机器负载变化。

【能得到什么测试效果】

自检跑完会当场给出结论卡片（也会进样本库的 `自检` 记录）：

- UDP 处理上限（Mbps）。

- 最佳延迟（环回最低 RTT, ms）。
- 延迟拐点（加压到哪个 Mbps 延迟开始上翘；没出现就显示「未出现」）。
- 评级：强 / 中等 / 弱，外加一句文字结论。
- 环回延迟 vs 负载 曲线（纯本机、无网络）。

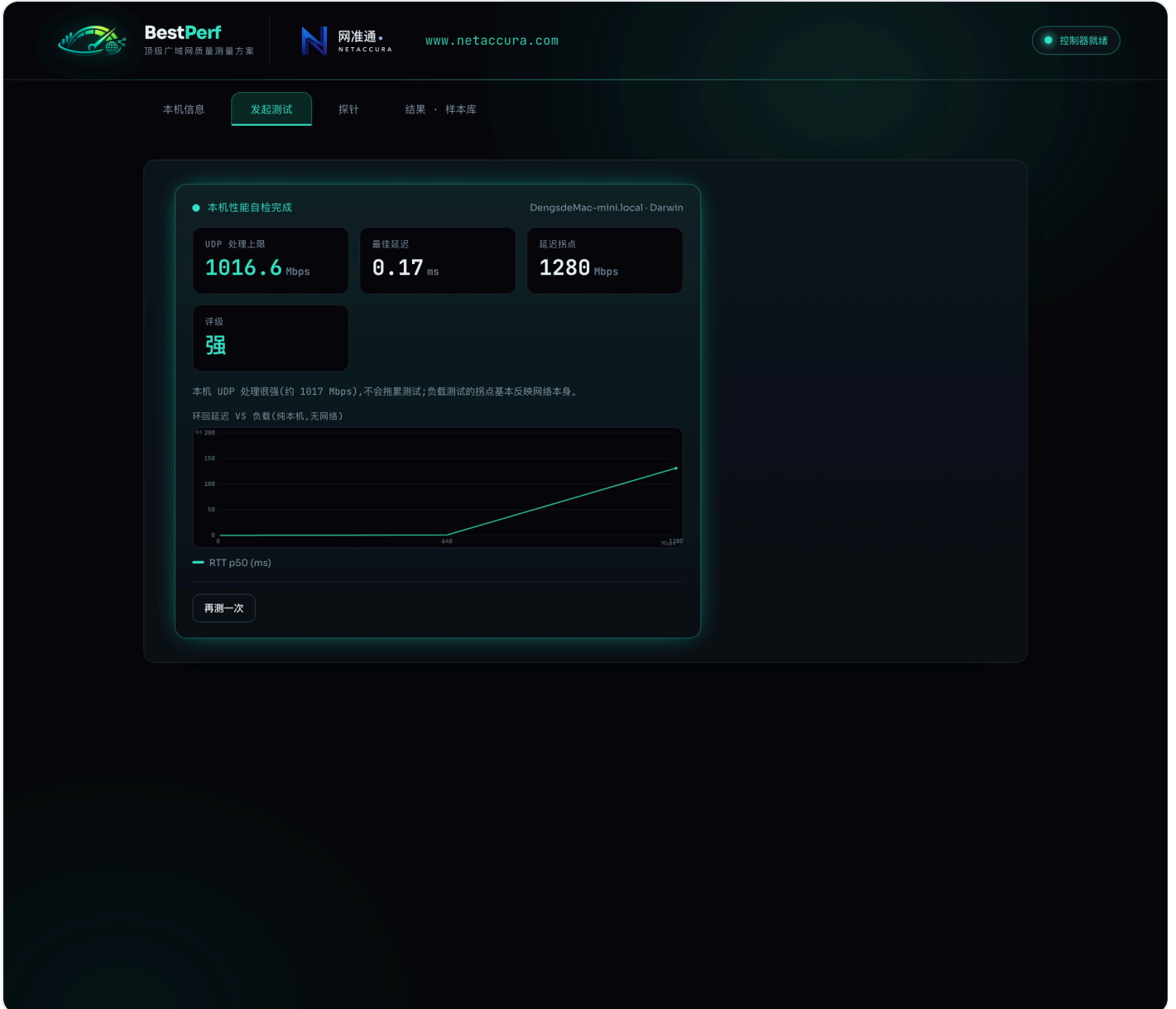


图 7-7 本机性能自检结果：这台机器的 UDP 处理上限、延迟地板与拐点、整体评级。

提示：自检结果还会刷新「本机信息」页上的一行标定信息，作为这台机器的「体检底数」。

7.5 探针管理（登记对端·令牌与地址）

【做什么】

「探针」标签页用来**登记你要测的对端机器**。带「本机」标签的那一条是运行本控制台的机器（不能删）；其余都是你登记的对端。要测一段网络，必须在**对端机器**上也运行本控制台，并把它的**控制台地址**填进来——发起方才能联系对端、协调出双向测试。

【怎么操作 / 配什么】

前提：对端那台机器要满足三件事——①也装了控制台；②打开了「允许被别人测向本机」开关；③把它的**控制台地址 + 令牌**给了你。

步骤 1 打开「探针」标签页。还没登记过对端时，列表里只有「本机」一条。

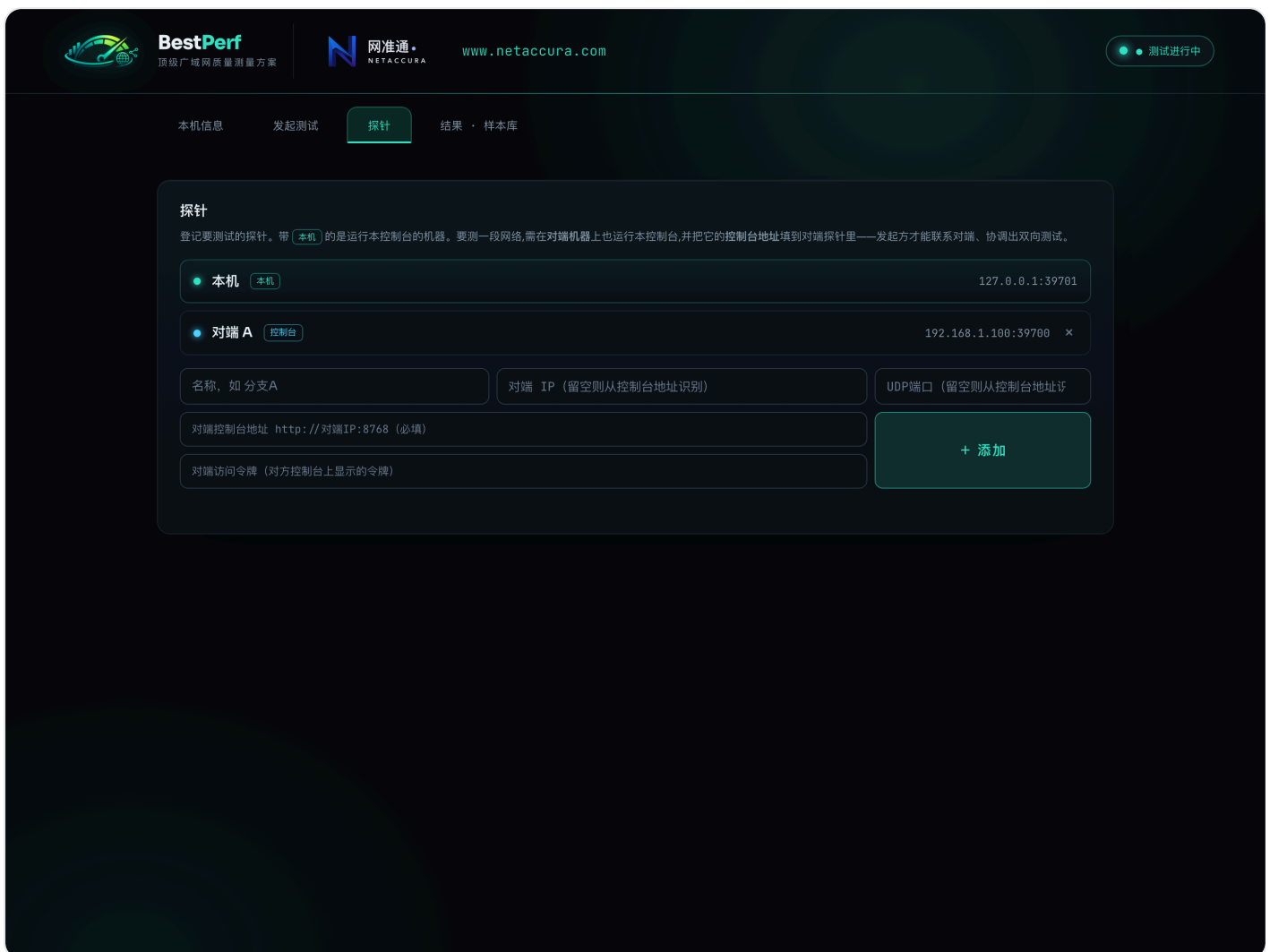


图 7-8 「探针」页：列表里带「本机」标签的是运行本控制台的机器；下方表单用来登记对端。

步骤 2 在下方表单里填写（字段说明如下）：

字段	必填	说明
名称	是	给这台对端起个名，如「分支A」
对端 IP	否	对端的主机 / IP；留空则从控制台地址自动识别
UDP 端口	否	对端的 UDP 探测端口；留空则从控制台地址自动识别（对端默认 39700，被占用会自动避让）
对端控制台地址	是	http://对端IP:8768 （对端控制台的地址，端口默认 8768）
对端访问令牌	否	对端控制台上显示的那串令牌（Token）

步骤 3 提交后，BestPerf 会尝试连一下对端控制台：连通会提示「√ 已连到对端...，UDP 探测端口 XXX」并自动读到真实端口；暂时没连上会提示「△ 暂时没连到对端控制台（请确认对端已打开『允许被别人测向本机』开关）」，并先按缺省端口登记，之后可在该行修改。

步骤 4 删除对端：在列表里对应行点右侧的 × 即可（本机 那条没有删除按钮）。

【注意事项 / 易错点】

关键理解 —— 地址 = ID，令牌 = 密码。 「本机信息」页的「本机连接信息」卡片里有这台机器的控制台地址、访问令牌、本机 IP/端口、以及 **UDP 探测端口 39700**。要让别人测你，就把这里的地址 + 令牌发给对方，并打开「允许被别人测向本机」开关（默认是关的，只有要被别人测时才打开）。这套机制类似远程协助软件的「ID + 密码」。

注意：对端的 UDP 探测端口被占用时会**自动避让**（换一个端口），所以**不要把端口写死**——留空让控制台从对端 identity 自动读取最准。

注意：只登记探针**不会自动开测**。登记好之后，到「发起测试」页选它当对端、点「开始」才真正跑测量。

【能得到什么测试效果】

探针登记本身不产生测试数据，但它是其余三种测量（轻量 / 负载 / TCP）的**前置条件**——登记好的对端会出现在「发起测试」页的「对端」下拉里供你选择。

7.6 结果与样本库

【做什么】

「结果·样本库」标签页是所有测试的归档处。每跑一次测量就多一条记录 (run)，左侧是列表、右侧是详情。它解决两件事：① 把历次测量按环境归类、检索；② 下载报告和给仿真器用的回放文件。

【怎么操作 / 配什么】

列表怎么读：每条记录有

- **类型徽标**：轻量 / 负载 / TCP / 自检，一眼看出是哪种测量；
- **状态**：完成 / 失败 等；
- **环境标签**：如 有线LAN、WiFi、4G、5G、卫星、企业WAN、家宽，后面还能带地点（这些来自发起测试时「给测试打标签」填的内容）。

怎么检索（列表上方）：

工具	作用
接入类型下拉	按 WiFi / 4G / 5G / 卫星 / 企业WAN / 家宽 / 有线LAN / 其它 等筛选
搜索框	按 run 名 / 运营商 / 地点 / 备注 模糊搜索
行首勾选框 + 全选	多选后可批量删除（进行中的测试会被跳过、删不掉）

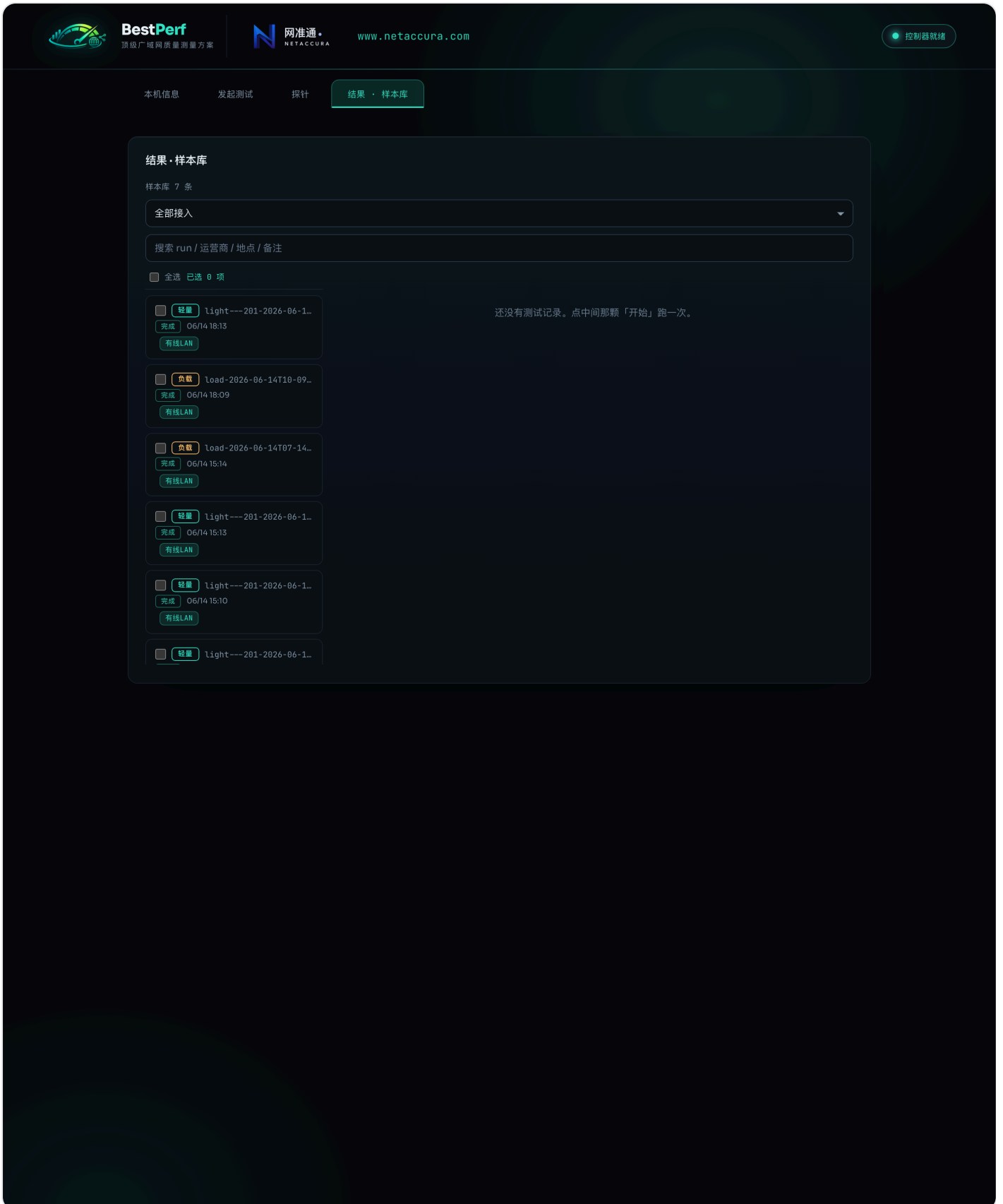


图 7-9 「结果 · 样本库」：按接入类型/运营商/地点检索，每条记录带类型徽标与环境标签。

点开一条记录，右侧详情顶部就是下载按钮区。各按钮含义：

下载项	出现在	是什么
损伤仪场景（可直接导入）	有回放数据时	<code>device_scenario.json</code> ，WE 原生专用格式，带延迟 + 丢包，可直接导入仿真器复刻这次网络
回放场景（通用格式）	有回放数据时	<code>wan_emulator_scenario.json</code> ，通用 / 中立格式，主要是延迟，给不支持专用格式的仿真器用
网络模型	有模型时	提炼出的标准网络画像，汇入样本库做长期建模
画像报告（PDF）	轻量测量	排版好的画像报告，浏览器打印→另存为 PDF
网络画像	轻量测量	<code>network_profile.json</code> ，延迟/抖动/丢包/乱序等原始画像
负载报告（PDF / JSON）	负载画像	排版报告 / 完整原始数据（容量、拐点、丢包、时间线，按方向打包）
TCP 报告	TCP 体感	<code>tcp_throughput.json</code> ，上下行吞吐 + RTT
删除样本	全部	永久删除这条记录的全部数据（进行中不可删）

【注意事项 / 易错点 —— 两种回放格式怎么选】

关键理解 —— `device_scenario` vs `wan_emulator_scenario`：

- 专用格式 `device_scenario`：WE（WAN 仿真器）的原生格式，可直接导入，同时带延迟和丢包，复刻最完整。只要你的仿真器是 WE / 支持这个格式，优先选它。
- 通用格式 `wan_emulator_scenario`：中立 / 通用格式，主要承载延迟，给不支持上面专用格式的仿真器用。

一句话：能用专用就用专用（复刻得全），仿真器不认专用格式时再退而用通用格式。

怎么导入 WAN 仿真器复刻现网：把下载下来的回放文件（`device_scenario.json` 或 `wan_emulator_scenario.json`）拿到实验室，导入你的 WAN 仿真器的回放/场景功能，仿真器就会按这次现网测到的延迟、丢包等参数施加损伤——于是实验室里的链路行为就复刻了现网。这就是 BestPerf 的核心闭环：现网实测·复刻。

注意：负载画像的回放场景带延迟 + 丢包；轻量测量的回放主要是延迟与基础损伤。需要复刻「满负荷下的丢包行为」时，用负载画像导出的场景更贴近。

【能得到什么测试效果】

- 一个可检索的样本库：随时间积累不同接入方式、运营商、地点的网络画像，按标签翻查。
- 每条记录的**报告（PDF/JSON）**用于阅读和留档。
- 每条记录的回放文件用于在实验室仿真器里**复刻现网**——把第一线测到的真实网络，原样搬进实验室。

提示：删除是永久的、不可恢复。批量删除时，正在跑的测试会被自动跳过。

本章小结：先用轻量测量摸底（延迟/双向丢包/抖动/乱序/重复/微中断，丢包给 95% 上界）；要看满负荷表现就上**负载画像**（找拐点→容量→工作点质量→Gilbert-Elliott，且只在拐点以内刻画、不会打死对端）；想知道应用实得多少就跑 **TCP 体感**（对照物理容量）；容量不对劲先用**本机性能自检**分清瓶颈在网络还是本机。所有结果都进**样本库**，按环境检索、导出报告与回放文件，最后导入 WAN 仿真器**复刻现网**。

第八章 常见问题 (FAQ)

本章按「问题 → 原因 → 怎么办」的方式，把电脑小白最容易卡住、最容易疑惑的点逐条讲清楚。**遇到问题先在这里查一遍，多数情况一两步就能解决。**

关键理解： BestPerf 的全部功能都在浏览器里的那个深色控制台界面（ `http://127.0.0.1:8768` ）完成。它不需要 SSH、不需要管理员/root、不需要抓包。如果某一步让你去敲命令行、装一堆东西，多半是走错了方向 —— 回到「双击启动入口」这条主线即可。

本章分为五节：

- 8.1 启动与界面打不开
- 8.2 安装被系统拦住
- 8.3 对端连不上 / 结果全是 0
- 8.4 测量与结果的疑问
- 8.5 安全、令牌与端口速查

8.1 启动与界面打不开

Q：双击「1 启动」后，浏览器没有自动打开？

原因： 启动器会先做一次「体检」（Doctor：检查 Python、端口、二进制是否都就绪），全部通过后才会去打开浏览器。如果体检没过，或者浏览器默认程序有问题，就不会自动跳出页面。

怎么办：

- **步骤 1：** 先别急着重启电脑。启动器没有报错关闭的话，控制台其实已经在后台跑起来了。手动打开任意浏览器（Chrome / Edge / Firefox 都行），在地址栏输入 `http://127.0.0.1:8768` 回车，看是否能打开深色界面。
- **步骤 2：** 如果能打开，说明只是「自动打开浏览器」这一步没成功，控制台本身是好的，照常用即可。
- **步骤 3：** 如果打不开，关掉启动器窗口，再双击一次桌面的「1 启动」入口，留意启动器窗口里的文字提示 —— 它会告诉你卡在了哪一步。

提示： 地址里是 `127.0.0.1`（也可以写 `localhost`），这是「本机」的意思，永远指你正在操作的这台电脑，不是别人的机器。

Q：浏览器打开后提示「无法访问 / 拒绝连接」？

原因：常见两种 —— ① 控制台进程没起来（体检没过）；② 默认端口 8768 被别的程序占用了。

怎么办：

- **步骤 1**：确认你双击的是桌面上的「1 启动」入口（Windows 是 1 启动 BestPerf.cmd ， macOS 是 1 启动 BestPerf.command ， Linux 是 1 启动 BestPerf.sh ），而不是别的文件。
- **步骤 2**：看启动器窗口的体检结果。它会逐项报告 Python、端口、二进制是否就绪；哪一项是红的，就是哪一项的问题。
- **步骤 3**：如果提示端口被占用，最省事的办法是重启一次电脑再启动（占端口的残留进程会被清掉）；高级用户可以给启动器设置一个固定的其他端口（环境变量 FIELDREPLAY_PORT ），但小白一般用不到。

注意：界面顶部有一个状态指示。显示「控制器就绪」表示一切正常；显示「控制器离线」表示浏览器开着、但后台控制台没连上 —— 这时去看启动器窗口，或重新双击「1 启动」。

Q：提示「需要 Python」或体检里 Python 那项是红的？

原因：macOS 和 Linux 的包用的是系统自带的 python3 ；如果系统里没有 3.9 以上版本，体检就过不去。

Windows 包自带 Python，不会出现这个问题，可忽略本条。

怎么办（按你的系统对号入座）：

系统	怎么办
Windows	不用管。Windows 包（ BestPerf-windows-x64.zip ）自带 Python。
macOS	Mac 一般自带 python3 。如果没有，打开「终端」运行一次 xcode-select --install ，按提示装好命令行工具即可，然后重新双击「1 启动」。
Linux	需要系统 python3 3.9 及以上。用你发行版的包管理器装一下（例如 Debian/Ubuntu: sudo apt install python3 ），再重新启动。

关键理解：BestPerf 不需要你会写 Python，也不需要你懂命令行。它只是用 Python 来跑后台控制台 —— 你只要保证系统里「有」Python 就行，剩下都是双击。

8.2 安装被系统拦住

第一次双击安装文件时，系统的安全机制可能会拦一下。这是**所有**未购买昂贵签名证书的小工具都会遇到的正常现象，不是软件有问题。放行一次以后就不会再问。

Q: macOS 双击安装时弹「来自身份不明的开发者」，打不开？

原因：macOS 的 Gatekeeper 对没有付费签名的程序默认拦截。这是系统行为，只在**第一次**出现。

怎么办：

- **步骤 1**：不要双击。改为**右键点击**（或按住 Control 单击） `Install BestPerf.command`。
- **步骤 2**：在弹出的菜单里选「打开」。
- **步骤 3**：这次系统弹窗里会多出一个「打开」按钮，点它。安装就会继续。

之后桌面会出现「1 启动 BestPerf.command」，以后正常双击它即可，不会再被拦。

提示：被别人测向你这台 Mac 时，macOS 还会弹一次「是否允许接受传入网络连接」，点「允许」即可。这是放行测量流量进来，必须允许，否则别人测你时会全是超时。

Q: Windows 双击安装时跳出蓝色的「Windows 已保护你的电脑」(SmartScreen) ？

原因：和 macOS 同理，SmartScreen 对不常见的程序会先拦一下。同样只在**第一次**出现。

怎么办：

- **步骤 1**：在蓝色窗口里点「更多信息」（这几个字默认是收起的，要点一下才展开）。
- **步骤 2**：下方会出现「仍要运行」按钮，点它。安装继续。

注意：Windows 解压时要用「右键 → 全部解压」把 zip **真正解开**，不要直接在压缩包里双击 `Install BestPerf.bat`。在压缩包内部双击会因为找不到配套文件而装不上。

Q: 装完后弹出来的那个「安装结果」页是什么？

原因：这是安装器给你的图文回执，告诉你装得成不成、下一步该点哪。

怎么办：看颜色就行 ——

- **绿色「安装成功」**：装好了。下一步只需要双击桌面的「1 启动」，别的都不用管。
- **红色**：没装上。页面会用红字给出**唯一的下一步动作**，照着那一句做即可（通常就是上面 8.1 / 8.2 里的某一条）。

8.3 对端连不上 / 结果全是 0

这是双机测量最常见的一类问题，单独拎出来讲。**症状**通常是下面之一：

- 在「探针」页登记对端时，提示「△ 暂时没连到对端控制台」。
- 测完打开结果，RTT 全是 0ms、丢包全是 0，详情顶部还有一条**橙色/警告横幅**，写着「本次测试没有收到任何回包（所有探测超时）」。

关键理解：结果里出现 0ms、0 丢包加橙色警告，不是说网络完美，而是一个包都没回来，这次数据没有意义，要排查后重测。真正「零丢包」的好网络不会配这条橙色警告。

请按下面五条逐一排查，多数是第 1、2 条。

Q：对端到底要先做什么，我才能测它？

前提（缺一不可）：对端那台机器必须 ——

1. 也安装并启动了 BestPerf 控制台；
2. 在它自己的「本机信息」页打开了「允许被别人测向本机」开关；
3. 把它的「控制台地址 + 令牌」给了你，由你在「探针」页登记。

关键理解：「允许被别人测向本机」这个开关只能在那台机器自己的控制台上打开，你在自己这边开不了别人的。如果对端没开这个开关，你这边登记时就会看到「请确认对端已打开『允许被别人测向本机』开关」。

Q：登记对端时提示「暂时没连到对端控制台」？

按顺序检查这几项：

- **第 1 项 · 对端开关：**对端是否打开了「允许被别人测向本机」？（最常见的原因。）
- **第 2 项 · 控制台地址：**你填的「控制台地址」是不是对端的 `http://对端IP:8768`？IP 要是对端那台机器在局域网里的真实 IP，端口是 `8768`。
- **第 3 项 · 令牌：**填的「令牌」是不是对端的 Token（在对端「本机信息」页能看到）？令牌填错或填了自己的，都连不上。
- **第 4 项 · 防火墙：**对端的防火墙是否放行了控制台端口 `8768`、以及 UDP 探测端口 `39700`？见下一问。

提示：登记成功时，界面会回一条绿色提示，类似「✓ 已连到对端『主机名』，UDP 探测端口 39700」，并自动把端口填好。看到这条就说明地址和令牌都对。

Q：地址、令牌都对，结果还是全超时、全 0？

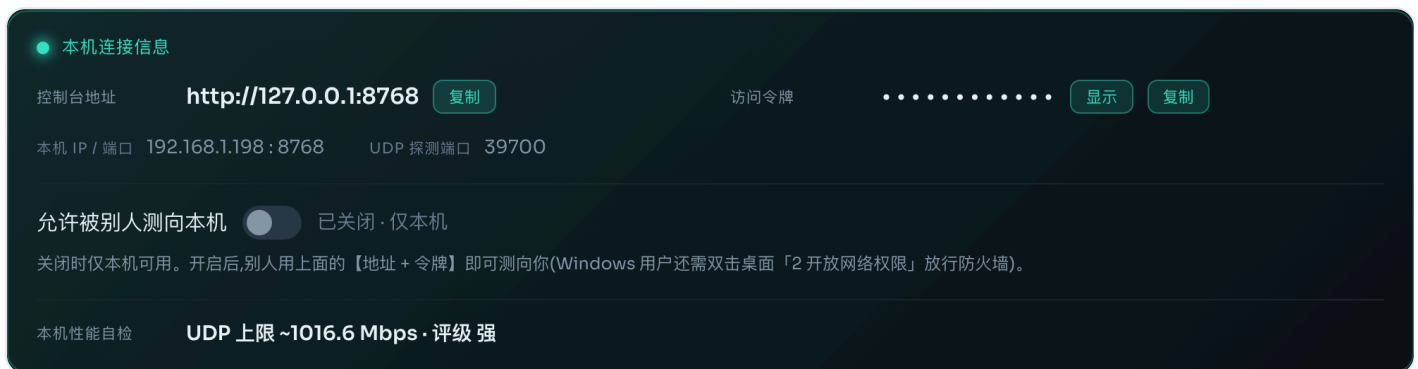
原因：八成是对端的防火墙挡住了 UDP 探测端口，或者对端有残留进程占着 UDP 端口。控制台之间用 8768 握手没问题，但真正的测量流量走的是 UDP 39700，这一段被挡住，包就回不来。

怎么办（让对端在它自己的机器上放行）：

对端系统	放行方法
Windows	双击桌面「2 开放网络权限 BestPerf」，在 UAC 弹窗点一次「是」。它会自动放行入站 UDP 39700-39899、TCP 39800-39899、TCP 8768，不用懂防火墙。
macOS	被测时系统弹「允许接受传入网络连接」，点「允许」。
Linux	用 ufw 放行三条： <code>sudo ufw allow 8768/tcp</code> 、 <code>sudo ufw allow 39700:39899/udp</code> 、 <code>sudo ufw allow 39800:39899/tcp</code> 。

注意：发起测试的一方不需要做这些放行。只有「要被别人测向」的那台机器（也就是对端 / 响应方）才需要开开关 + 放行入站端口。你从自己机器主动发起去测别人，自己这边什么都不用开。

排查端口速查见 8.5 节的表。



「本机信息」页的「本机连接信息」卡片：这里能看到本机的控制台地址、令牌 (Token)、本机 IP/端口、UDP 探测端口 39700，以及「允许被别人测向本机」开关。把别人测你需要的「地址 + 令牌」就从这里拿。

8.4 测量与结果的疑问

Q：负载画像会不会把对端或者网络「打死」？会不会影响别人上网？

不会。负载画像 (load) 是逐档加压：从很小的速率开始，一档一档往上加，在过程中找到网络的「拐点 / 容量上限」(再加就开始大量丢包、延迟飙升的那个点)。

关键在于：找到拐点后，它测延迟/丢包/突发用的「工作点」是落在拐点以内的，不会一直顶着最大压力猛灌。所以它刻画的是「网络满负荷时的真实表现」，而不是把链路打瘫。

提示：如果还是担心，可以把「时长」设短一点、报文大小用默认值先跑一次小的。另外「负载报告」里如果出现橙色的「△ 受设备限制」字样，说明瓶颈其实在**机器本身**而不是网络——这时可以先做一次「本机性能自检」分清楚（见下）。

Q：详情里给出两个回放文件，导入 WAN 仿真器到底用哪个？

结论：优先用「损伤仪场景（可直接导入）」那个。

每条成功的测量，详情顶部的下载区里可能有两个回放文件按钮：

下载按钮	内部格式	特点	什么时候用
↓ 损伤仪场景 (可直接导入)	<code>device_scenario</code>	WE 原生格式，可直接导入仿真器，带延迟 + 丢包	你的仿真器支持这个专用格式时，首选它——复刻得最全
↓ 回放场景 (通用格式)	<code>wan_emulator_scenario</code>	通用/中立格式，主要是延迟	给不支持上面那个专用格式的仿真器用，作为兜底

关键理解：两个文件都是「把这次实测的现网行为搬进实验室」。区别是专用格式 (device_scenario) 信息更全（连丢包也带上），能直接导入；通用格式是为了兼容更多设备而做的中立版本，偏重延迟。能用专用格式就用专用格式。

本机信息 发起测试 探针 结果 · 样本库

结果 · 样本库

样本库 7 条

全部接入

搜索 run / 运营商 / 地点 / 备注

全选 已选 0 项

轻量 light---201-2026-06-14T10-13-57 完成

轻量 light---201-2026-06-1...
完成 06/14 18:13
有线LAN

回放场景(通用格式) 网络模型 画像报告(PDF) 网络画像 删除样本

原始数据 session_config

环境标签: 有线LAN(自动)

负载 load-2026-06-14T10-09...
完成 06/14 18:09
有线LAN

负载 load-2026-06-14T07-14...
完成 06/14 15:14
有线LAN

轻量 light---201-2026-06-1...
完成 06/14 15:13
有线LAN

轻量 light---201-2026-06-1...
完成 06/14 15:10
有线LAN

轻量 light---201-2026-06-1...
完成 06/14 15:10
有线LAN

综合 RTT

p50	60.73 ms
p95	60.87 ms
p99	60.95 ms
最大	61.11 ms
样本数	524
超时	-

发起方 -> 对端 (A-B)

丢包率	2.86 %
到达间隔抖动 p95	0.27 ms
乱序率	0 %
重复率	0 %
中断次数	0



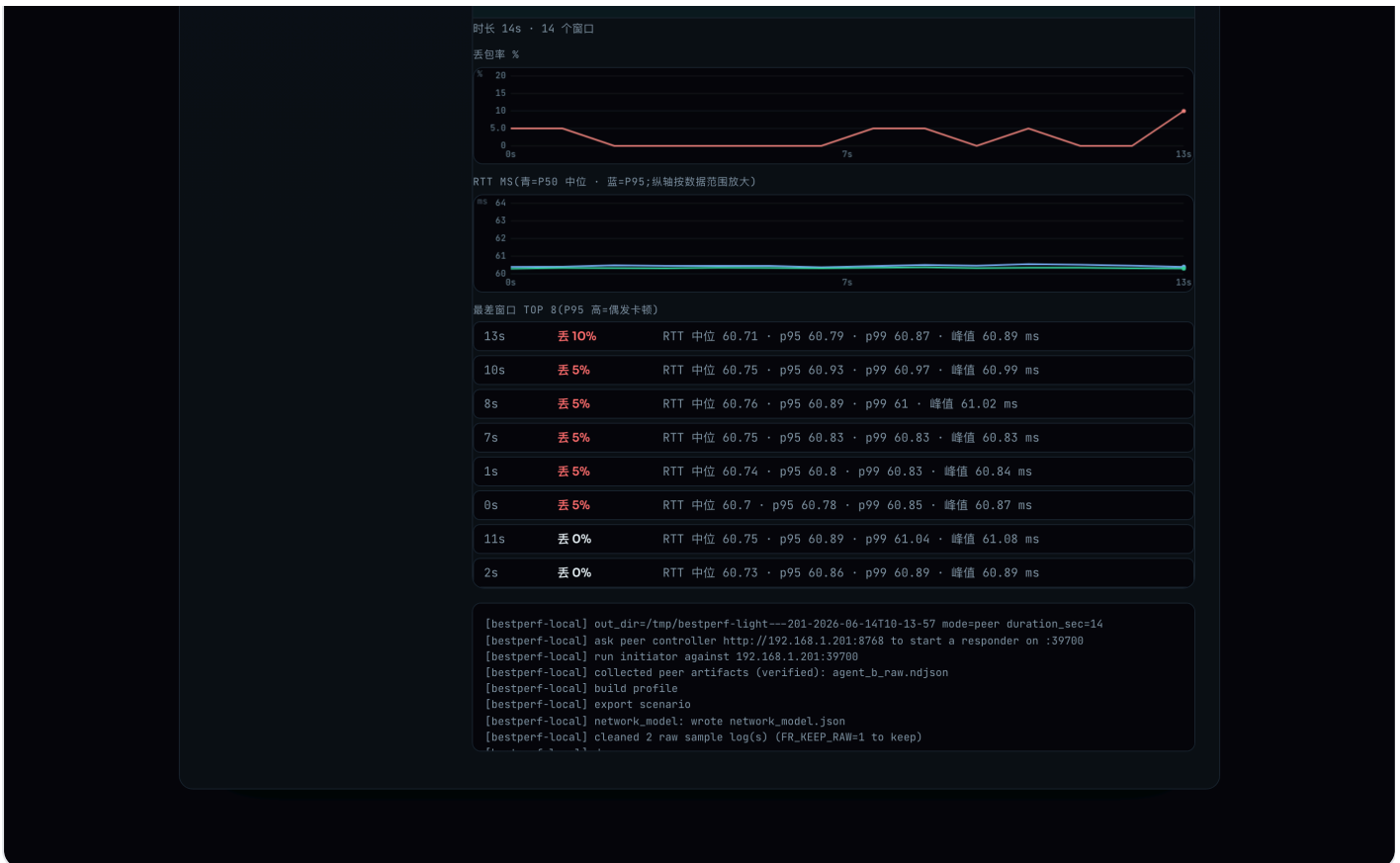
最差窗口 TOP 8 (P95 高=偶发卡顿)

11s	丢 10%	RTT 中位 60.77 · p95 60.86 · p99 60.86 · 峰值 60.86 ms
5s	丢 10%	RTT 中位 60.73 · p95 60.85 · p99 60.96 · 峰值 60.99 ms
3s	丢 5%	RTT 中位 60.75 · p95 60.93 · p99 60.93 · 峰值 60.93 ms
4s	丢 5%	RTT 中位 60.75 · p95 60.89 · p99 60.92 · 峰值 60.93 ms
8s	丢 5%	RTT 中位 60.78 · p95 60.87 · p99 60.91 · 峰值 60.91 ms
2s	丢 5%	RTT 中位 60.73 · p95 60.81 · p99 60.85 · 峰值 60.86 ms
1s	丢 0%	RTT 中位 60.75 · p95 60.92 · p99 61.07 · 峰值 61.11 ms
7s	丢 0%	RTT 中位 60.75 · p95 60.91 · p99 60.99 · 峰值 61.01 ms

对端 -> 发起方 (B-A)

丢包率	2.5 %
到达间隔抖动 p95	0.11 ms
乱序率	0 %
重复率	0 %
中断次数	0

对端 -> 发起方 (B-A) · 质量随时间



结果详情页：上方有概况 RTT (p50/p95/p99/最大)、按方向 (A→B / B→A) 的丢包/抖动/中断，下方是每秒 RTT 曲线和逐窗丢包表。详情顶部的下载区就是你下载「回放场景」「网络模型」「网络画像」「画像报告 (PDF)」的地方。

Q：我只有一台机器，没有第二台，能测吗？

能——用「本机性能自检」。在「发起测试」页，「测量类型」分段开关里选「本机性能自检」。它在本机做环回 (loopback) 加压，不走网络、不需要对端（这时界面会自动隐藏对端和方向选择）。

它的产出是：这台机器自己的 UDP 处理上限和负载下的延迟，给出一个评级（强 / 中等 / 弱）。

用途：当你做负载测试发现速率上不去时，先跑一次自检，就能分清「瓶颈到底在网络，还是在我这台机器自己」。

提示：本机性能自检只衡量本机性能，不会产出「网络画像」或「回放文件」——因为它根本没走网络。要得到画像和回放文件，需要轻量测量或负载画像（两者都需要一个对端）。

Q：测出来 RTT 的 p50 / p95 是什么意思？为什么不给「单向延迟」？

简单说：p50 是「一半的包比这快」的典型值，p95 是「尾部」——被 Wi-Fi 缓冲、抖动拖慢的那批包就落在 p95。两个一起看，能区分「平时挺好、偶尔卡一下」和「整体都慢」。

至于**单向延迟**：BestPerf 只用各自机器的单调时钟，**两端没有对时**，所以诚实地只给往返时延（RTT），不号称单向延迟。（导出回放文件时，会按对称的 RTT/2 拆分，并明确标注为「估算」。）

8.5 安全、令牌与端口速查

Q：「令牌（Token）」是什么？会不会有泄露风险？

令牌就是这台控制台的「访问密码」（类似远程协助软件里的临时密码）。在「本机信息」页的「本机连接信息」卡片里能看到。控制台地址好比「ID」，令牌好比「密码」，两个配齐别人才能测向你。

关于安全，记住三点：

1. **本机操作不用管令牌**。你在自己机器上用 `127.0.0.1:8768` 打开界面时，控制台会自动认证，你看不到也不用输令牌。
2. **令牌只在「要被别人测向你」时才需要交出去** —— 而且只发给你信任的、要测你这台机器的那位操作员。不打算被别人测，就完全不用把令牌给任何人。
3. 控制台默认**不允许对端访问**（「允许被别人测向本机」开关默认关）。只有你主动打开那个开关、并把地址 + 令牌给了别人，别人才测得到你。不需要被测时就保持关闭，最安全。

注意：令牌只是用来「让指定的人能测向你」，不是你电脑的登录密码，泄露它不会让别人登录你的系统；但为稳妥起见，仍然只发给需要测你的人，用完可以关掉「允许被别人测向本机」开关。

Q：端口 / 防火墙速查表（被别人测向本机时才需要放行）

下面这些端口，只有「要被别人测向」的那台机器需要放行入站。主动发起测试的一方不用动。

用途	端口	协议	说明
控制台 / 界面	8768	TCP	浏览器访问界面、两台控制台之间握手协调
UDP 探测（测量主力）	39700 – 39899	UDP	轻量测量、负载画像的实测流量走这里；探测端口默认从 39700 起
TCP 体感吞吐	39800 – 39899	TCP	TCP 体感（RFC 6349）测上行/下行 goodput 时用

各系统放行方法：

系统	放行方法
Windows	双击桌面「2 开放网络权限 BestPerf」→ UAC 点一次「是」，自动放行以上全部端口。
macOS	被测时系统弹「允许接受传入网络连接」→ 点「允许」。
Linux	<pre>sudo ufw allow 8768/tcp ; sudo ufw allow 39700:39899/udp ; sudo ufw allow 39800:39899/tcp 。</pre>

提示：UDP 探测端口默认是 39700，但如果这个端口已被别的程序占用，控制台会自动避让到下一个空闲端口（所以放行的是一个范围，而不是单个端口）。登记对端探针时，界面会自动读取对端的真实端口并填好，你不用手动猜。

本章小结：80% 的卡点集中在两件事 —— ① 启动/安装被系统拦了一下（放行一次即可，见 8.1 / 8.2）；② 测别人时对端没开「允许被别人测向本机」开关或没放行 UDP 39700（见 8.3）。记住「发起方什么都不用开，被测方才要开开关 + 放端口」这一句，多数问题迎刃而解。