Профили каналов

Профили измерений канала

Один канал = один сервис на пути. Один профиль = 1–2 пробы, которые не мешают друг другу.

Scope: профили и каталог ниже — только для **wisla-wiprobe-agent** (софт на хосте). Классический SLAMon/wiProbe ≥ 1.14.10 в списке агентов — инвентарь, не jobs. Зонды в разрыве (WPE / Berkut, SLAMon на железе) — отдельная волна. Матрица: SLAMon ≥ 1.14.10.

Так делают ThousandEyes, Cisco IP SLA, TWAMP/SLAMon: **не вешают все тесты на один объект**. Нагрузка (ёмкость) и непрерывный SLA — разные тесты, часто разные объекты между теми же концами.

Изоляция метрик (hub / star)

На агенте-хабе может быть много channel jobs. Ключи пишутся как:

`channel.{slug}.{c2r|r2c}.s{8hex}.{metric}`

где `s{8hex}` — короткий id сервиса канала. Так last-value и графики не смешивают peer'ов. Backend при чтении отдаёт канонический ключ без токена; планировщики availability/порогов читают **только ключи этого сервиса** (без полного dump `channel.*` по агенту).

Агент: `limits.channel_max_concurrent` по умолчанию **1** — за тик берётся не больше N самых «просроченных» jobs; первый запуск размазан по интервалу (stagger), чтобы хаб не бил ICMP по всем peer разом.

Роль **channel**: с пакета **0.2.17** стоит в дефолтном `agent.yaml` вместе с `host`. Без jobs агент тихий. Старые установки без роли — alpha-channel-enable-roles.py.

Проблемы и доступность канала

Две **разные** проблемы на карточке **канала** (узел/host тут ни при чём — зонд канала может работать без привязки к узлу):

Ключ	Когда	Смысл
`channel.probe.offline`	central и/или remote зонд без heartbeat > **~180 с**	Заголовок: «Один из зондов измерения недоступен»; **какой** зонд — в opdata (`central/remote probe "..."`)
`icmp.unavailable`	**лучший** loss по обеим сторонам ≈ 100% / `probe.error`, **или** метрики старше `wisla.channel.availability.stale-sec`	Путь / качество по метрикам

Ключ	Когда	Смысл
	(по умолчанию **900 с**) — **и** все назначенные зонды online	

Решение о доступности пути принимает только `ChannelAvailabilityScheduler` (сводка last-value с central **и** remote). Push метрик от одного зонда **не** открывает/закрывает `icmp.unavailable`: иначе при асимметричном traceroute (одна сторона доходит, другая — `* * *`) лента флапает каждые ~2 мин. Ingest только штампует `last_check_at`.

Backend при чтении last-value / series подставляет ключи профиля канала (не только loss/rtt). Для **TRACE_ROUTE** в KPI должны появляться **Хопы** и **Трасса OK** (не прочерк).

В **списке объектов** бейдж **Trace** (не ICMP): зелёный, если `traceroute.ok=1` хотя бы в одном направлении; красный, если оба направления FAIL; серый — нет свежих метрик. Бейдж **Путь — жёлтый (деградация)**, если одно направление c2r/r2c недоступно, а другое OK (раньше был зелёный по «лучшему» направлению). Бейдж **Агент — оба** зонда (remote + central): зелёный только когда оба online; offline любого → красный.

Доступность / SLA для канала считается по **лучшему** результату с обеих сторон (min loss): если хотя бы одно направление доходит — канал «доступен», даже когда обратный traceroute с α на ПК даёт `* * *`. Это не ICMP-ping узла, а измерение пути между зондами. Переключатель **«Центр → удалённый»** / **«Удалённый → центр»** показывает метрики **выбранного** направления: на одном может быть OK (18 ms, 2 hop), на другом — FAIL.

Асимметрия c2r / r2c (частая причина «пустой» стороны)

Каждый зонд шлёт пробы **только в свою** сторону: central → `c2r`, remote → `r2c`. Если на карточке одна вкладка пустая, а другая с цифрами — это **не** «нет сбора вообще», а провал **конкретного** направления.

Типичные причины на стенде:

Симптом	Частая причина
ICMP/UDP c2r OK, RFC/TCP/DSCP c2r = 100% loss	На **peer** не открыты inbound-порты echo-блока **8787–8797** (Windows Firewall режет TCP 8787 и UDP 8792/8797, а базовый UDP 8787 — пропускает)
r2c OK, c2r FAIL (или наоборот)	Асимметрия ACL/firewall/NAT между зондами
Обе стороны «Ожидание»	Нет jobs (роль `channel`), зонд offline, или первый цикл ещё не завершился

На агенте echo-сервисы слушают блок портов от `channel.echo_port` (по умолчанию **8787**) + смещения: UDP/TCP **+0**, RFC **+5** (=8792), DSCP **+10** (=8797). Если базовый блок занят, агент выбирает следующий свободный (`8788`, …) и сообщает **`channel_echo_port`** менеджеру; jobs на peer получают порты от **его** base, не от константы 8787.

Проверка на peer: `netstat -an | findstr 8787` (Windows) / `ss -ulnp | grep 8787` (Linux) — после первого channel job должны быть слушатели.

Path MTU на Linux использует raw ICMP с флагом DF (не зависит от flavor `ping`; GNU inetutils без `-M do` раньше давал 100% loss в направлении r2c). Сервис агента по умолчанию — `User=root` в unit; для не-root нужен `CAP_NET_RAW`.

Это не ICMP SLA и **не** Protocol MTU Wisla 5: зонд делает **двоичный поиск** размера с DF (~10 проб за цикл), а не N пакетов фиксированного размера. В параметрах канала задаются **интервал** и **потолок поиска** (ICMP payload или L3 MTU ≥ 1500). Поле «Пакетов» к Path MTU

не относится. На карточке KPI — `channel.mtu.bytes` и `channel.mtu.ok`; 11 отправленных — шаги поиска, не «пакеты SLA».

MTU (UDP / Protocol) (`mtu_rfc`) — паритет с Wisla 5 `MTULoadExecutor` / `PROTOCOL:MTU`: UDP load к peer echo, шаги размера, `packet_count` пакетов на шаг, критерий потерь. Метрики: `channel.mtu_rfc.bytes` / `channel.mtu_rfc.ok`. Не путать с Path MTU.

Offline зонда **не** открывает ICMP и **гасит** уже открытый `icmp.unavailable`: пока висит `channel.probe.offline`, loss≈100% / stale обычно значат «нечем мерить», а не подтверждённый обрыв пути.

Почему узлы могут «кричать», а канал молчит дольше по ICMP: при полном сбое агенты перестают пушить пробы, а last-value ещё держит loss=0 — пока не сработает stale. События **«Изменение конфигурации хоста»** относятся к паспорту узла и на карточку канала больше не вешаются.

История статусов (полоса над графиком) красится по порогам шаблона: жёлтый = деградация, красный = отказ. Для RTT берётся худшее из **avg** и **max** (пик max не теряется, если avg ещё «зелёный»). В тултипе графика значения подсвечиваются тем же цветом, что и сегмент истории в этой точке (как на обзоре узла).

Как выбрать

В форме канала, шаг **Измерения** — плитки всех шаблонов + быстрый фильтр; профили сверху подставляют набор. Интервал/пакеты/размер — шаг **Параметры**.

Профиль	Что внутри	Когда
Качество IP	P-Test (ICMP)	Базовый SLA: жива ли линия, потери, задержка. Ставьте почти всегда.
Голос	U-Test (UDP) + MOS	Клиенту важны звонки. Не вместе с ICMP.
Приложения	TCP RTT	Быстро ли поднимается TCP к far-end. HTTP/HTTPS — в доступности услуги.
Маршрут	Traceroute	Разбор инцидента: где обрыв. Не с плотным UDP.
Ёмкость	RFC burst	Приёмка полосы. **Только этот** шаблон на канале.

Каталог слева — рабочие тесты пути: ICMP, UDP, TCP, traceroute, RFC, RFC sustained, DSCP, Path MTU, **MTU UDP (Protocol)**.
DNS / HTTP / HTTPS / SIP — вкладка **Шаблоны → Доступность услуги**.

`TWAMP` (soft subset), `Netflow`, `Y.1731`, `L2`, `OUTER` и похожие hardware / stub типы в операторском каталоге скрыты. Если нужен такой режим, это отдельная волна под железные зонды / специализированные коннекторы, а не обычный канал между `wisla-wiprobe-agent`.

Если нужны и качество, и ёмкость

Создайте **два канала** между теми же зондами:

1. «… качество» → профиль **Качество IP**

2. «… ёмкость» → профиль **Ёмкость** (реже, не круглосуточно плотным burst)

Так метрики не портят друг друга.

Чего не делать

- Все шаблоны каталога на один канал
- RFC вместе с ICMP/UDP на одном канале
- Несколько профилей сразу (клик по карточке заменяет набор)

См. также: Сервисы, Шаблоны каналов.