УДК 004.7

Галяутдинов Д.А.,
студент 4 курса,
факультет Комплексной Безопасности ТЭК,
Российский государственный университет нефти и газа
(Национальный исследовательский университет)
имени И.М. Губкина
(г. Москва, Россия)

ВНЕДРЕНИЕ И ТЕСТИРОВАНИЕ СЕТЕВОЙ ТЕЛЕМЕТРИИ ВМЕСТО SNMP НА ОСНОВЕ ECOROUTEROS, RFC 9232

Аннотация: *в статье рассматриваются принципы построения систем сетевого мониторинга и направление их развития от модели периодического опроса к модели потоковой телеметрии. Проводится сравнительный анализ протоколов SNMP и IPFIX по архитектурным различиям, нагрузке на управляющую плоскость маршрутизатора и степени детализации собираемых данных. Описываются особенности реализации экспорта потоков в отечественной операционной системе EcoRouterOS. Приводится методика развертывания коллектора на базе агента Telegraf. Обосновывается применимость рассмотренного подхода в корпоративных и операторских сетях в условиях импортозамещения.*

Ключевые слова: *сетевой мониторинг, сетевые протоколы, потоковая телеметрия, SNMP, IPFIX, EcoRouterOS, нагрузка на процессор.*

Galyautdinov D.A.,
4th-year student,
Faculty of Integrated Fuel and Energy Complex Safety,
Gubkin Russian State University of Oil and Gas

(National Research University)

(Moscow, Russia)

IMPLEMENTATION AND TESTING OF NETWORK TELEMETRY AS AN SNMP REPLACEMENT BASED ON ECOROUTEROS AND RFC 9232

Abstract: *the article examines the principles of network monitoring system design and the transition from periodic polling to streaming telemetry. A comparative analysis of SNMP and IPFIX protocols is carried out in terms of architectural differences, control plane load, and data granularity. The specifics of flow export implementation in the domestic EcoRouterOS operating system are described. A methodology for deploying a Telegraf-based collector is presented. The applicability of the approach in corporate and carrier-grade networks under import substitution conditions is substantiated.*

Keywords: *network monitoring, network protocols, streaming telemetry, SNMP, IPFIX, EcoRouterOS, CPU load.*

ВВЕДЕНИЕ

Сетевые инфраструктуры непрерывно усложняются, а объемы генерируемого трафика растут высокими темпами. Классический подход к управлению сетью через протокол SNMP работает по модели периодического опроса оборудования. У этой схемы два структурных недостатка. Первый состоит в генерации избыточной вычислительной нагрузки на центральный процессор маршрутизатора. Второй выражается в появлении слепых зон: кратковременные аномалии, возникающие между интервалами опроса, система не фиксирует.

Альтернативой стала потоковая сетевая телеметрия, базовые принципы которой описаны в RFC 9232. В этой модели маршрутизатор не ожидает внешних запросов, а самостоятельно и непрерывно выгружает

данные на коллектор в режиме реального времени. Применение данного подхода на базе отечественной платформы EcoRouterOS является актуальной инженерной задачей в условиях курса на импортозамещение в телекоммуникационной отрасли. Вопросы внедрения отечественного сетевого оборудования и подготовки кадров для работы с ним рассматриваются в ряде современных публикаций.

Объект исследования: инфраструктура мониторинга сетевых потоков на базе маршрутизатора с операционной системой EcoRouterOS. Предмет исследования: переход от мониторинга по SNMP к потоковой телеметрии на основе архитектуры RFC 9232. Цель работы: проектирование и тестирование системы сетевой телеметрии для замены традиционных методов мониторинга и повышения детализации данных о трафике.

Гипотеза: переход от SNMP к протоколу IPFIX снижает пиковую нагрузку на центральный процессор маршрутизатора EcoRouter при аналогичной интенсивности транзитного трафика. Методы исследования: сравнительный анализ архитектур мониторинга, инструментальное тестирование утилитами iperf3 и snmpwalk, измерение загрузки ЦПУ, анализ журналов и данных в формате JSON.

1. ОГРАНИЧЕНИЯ SNMP И КОНЦЕПЦИЯ ПОТОКОВОЙ ТЕЛЕМЕТРИИ

Протокол SNMP появился в конце 1980-х годов и предложил единый стандарт получения метрик с оборудования разных производителей. Архитектура построена на модели менеджер-агент: сервер мониторинга по заданному таймеру запрашивает значения из базы управляющей информации MIB сетевого устройства. На протяжении нескольких десятилетий этот подход оставался отраслевым стандартом.

Ограничения SNMP проявились с массовым внедрением дата-центров, виртуализации и SDN-архитектур. Скорости маршрутизации

вышли за отметки в десятки и сотни гигабит в секунду, и структурные проблемы протокола стали практическими. Первая проблема: каждый запрос GetRequest требует полного цикла обработки на центральном процессоре маршрутизатора, а в сети из сотен устройств, где с каждого снимаются тысячи метрик, суммарный объем служебного трафика достигает значительных значений при минимальной полезной нагрузке. Вторая проблема: один и тот же процессор одновременно обслуживает протоколы динамической маршрутизации и отвечает на SNMP-запросы. При агрессивном поллинге процесс snmpd начинает вытеснять задачи управляющей плоскости, что приводит к задержкам конвергенции таблиц маршрутизации и ложным разрывам соседских отношений. Третья проблема: протокол оперирует только количественными счетчиками интерфейсов и не содержит данных о структуре проходящих потоков, IP-адресах, портах или типах трафика. Наконец, базы MIB представляют собой иерархическое дерево с вендорно-специфичными OID, несовместимое с современными форматами хранения данных, что вынуждает строить дополнительные слои конвертации.

Потоковая телеметрия по RFC 9232 меняет логику взаимодействия устройства с системой мониторинга. Инициатором обмена выступает сам маршрутизатор, который непрерывно выгружает данные на коллектор без внешних триггеров. Принципиальное отличие от SNMP состоит в уровне обработки: сбор метрик интегрирован непосредственно в плоскость коммутации данных. Записи о потоках формируются в момент прохождения пакета через интерфейс с использованием аппаратных буферов, центральный процессор управляющей плоскости в этот процесс не вовлечен. Мониторинг и маршрутизация перестают конкурировать за один ресурс.

2. ПРОТОКОЛ IPFIX И ЕГО РЕАЛИЗАЦИЯ В ECOROUTEROS

Протокол IPFIX описан в RFC 7011 и является открытым вендор-независимым стандартом, разработанным на основе NetFlow v9 компании Cisco. Архитектура системы включает три компонента. Экспортер встроен в сетевое устройство: он анализирует проходящие пакеты, группирует их в потоки по ключевым признакам L3/L4 и по истечении тайм-аутов отправляет сформированные сообщения. Коллектор принимает данные от экспортеров, хранит шаблоны структур и конвертирует бинарный поток в структурированный формат. Анализатор сохраняет данные в базу временных рядов и предоставляет интерфейс визуализации.

Формат передачи данных в IPFIX основан на шаблонах. Устройство один раз отправляет описание структуры данных, которое коллектор кэширует, а затем передает только сами значения без повторяющихся заголовков. За счет этого объем служебного трафика по сравнению с SNMP снижается кратно. Каждое поле данных идентифицируется числовым идентификатором из открытого реестра IANA, что обеспечивает совместимость между различными производителями.

Маршрутизаторы EcoRouterOS используют технологию DPDK для программного ускорения обработки пакетов на процессорах x86. Эта архитектура определяет особенности реализации IPFIX-сенсора. Сбор статистики встроен непосредственно в процесс быстрой коммутации: при поступлении пакета система извлекает идентификаторы потока и обновляет счетчики в высокоскоростной таблице активных потоков в оперативной памяти. Жизненный цикл потоков управляется двумя тайм-аутами. Active Timeout определяет максимальный интервал, по истечении которого статистика активного потока принудительно экспортируется независимо от того, продолжается ли передача данных. Idle Timeout определяет период ожидания: если за это время новых пакетов потока не поступило, запись считается завершенной и экспортируется. Процесс экспорта работает асинхронно в отдельном фоновом потоке, что позволяет

вынести все операции формирования IPFIX-сообщений за пределы критического пути маршрутизации.

Для приема данных были рассмотрены четыре варианта программных коллекторов: Logstash, Vector, Telegraf и KUMA. Logstash работает на базе JVM и требует значительных ресурсов. Vector написан на Rust и отличается высокой производительностью, но ориентирован прежде всего на обработку текстовых событий. KUMA является коммерческим отечественным решением, ориентированным на задачи информационной безопасности. Для стенда выбран Telegraf: агент написан на Go, потребляет минимум оперативной памяти и содержит нативный плагин inputs.ipfix для приема и разбора IPFIX-датаграмм без дополнительной настройки.

3. ПРАКТИЧЕСКАЯ ЧАСТЬ

3.1. Состав и топология испытательного стенда

Стенд развернут на сервере виртуализации. В его составе три виртуальные машины. Маршрутизатор под управлением EcoRouterOS: 4 ядра, 4 ГБ ОЗУ, 6 ГБ диска, 3 сетевых интерфейса. Сервер-коллектор под управлением ALT Linux: 1 ядро, 1 ГБ ОЗУ, 30 ГБ диска, установлены Telegraf и iperf3. Генератор нагрузки под управлением ALT Linux: 1 ядро, 1 ГБ ОЗУ, 30 ГБ диска, установлен iperf3. Логическая топология: генератор нагрузки подключен к одному интерфейсу маршрутизатора, коллектор к другому, трафик проходит через устройство транзитом.

Маршрутизатору выделено 4 ядра, однако в исходной конфигурации EcoRouterOS технология DPDK резервирует часть процессорных ядер под опрос сетевых очередей методом активного ожидания. По этой причине даже в отсутствие транзитного трафика наблюдается фоновая загрузка ЦПУ на уровне 78-78,3%, отражаемая операционной системой. Эта

особенность учитывалась при анализе результатов: оценивался не абсолютный уровень загрузки, а прирост нагрузки относительно базового значения при включении подсистемы мониторинга.

Эксперимент проводился в три этапа: измерение нагрузки при SNMP-мониторинге, измерение нагрузки при IPFIX-телеметрии, сравнительный анализ результатов.

3.2. Методика измерения нагрузки при SNMP

На маршрутизаторе EcoRouter был настроен SNMP-агент версии SNMPv2c с community string public. Доступность данных проверена утилитой snmpwalk. Параметры эксперимента приведены ниже для обеспечения воспроизводимости. Со стороны сервера-коллектора запускался непрерывный цикл опроса набора из пяти OID с интервалом 5 секунд, что при наблюдаемой средней длительности ответа давало порядка 1 запроса в секунду на каждый OID. Опрашиваемый набор: ifInOctets, ifOutOctets, ifInErrors, ifOutErrors, sysUpTime. Транзитная нагрузка генерировалась утилитой iperf3 по протоколу TCP в течение 60 секунд без ограничения полосы пропускания.

Загрузка ЦПУ фиксировалась в два момента. На первом шаге маршрутизатор находился в состоянии простоя без транзитного трафика и без активного опроса; фоновая загрузка составляла 78-78,3% по причине, описанной в разделе 3.1. На втором шаге iperf3 генерировал транзитный трафик, а сервер-коллектор одновременно выполнял SNMP-запросы. В активной фазе загрузка ЦПУ на интервале 10 секунд достигала 100%, что видно на рисунке 1. Процессор управляющей плоскости при каждом GET-запросе прерывался для обхода дерева OID, формирования ответного пакета и его отправки. При увеличении числа опрашиваемых параметров или сокращении интервала опроса нагрузка возрастала пропорционально.

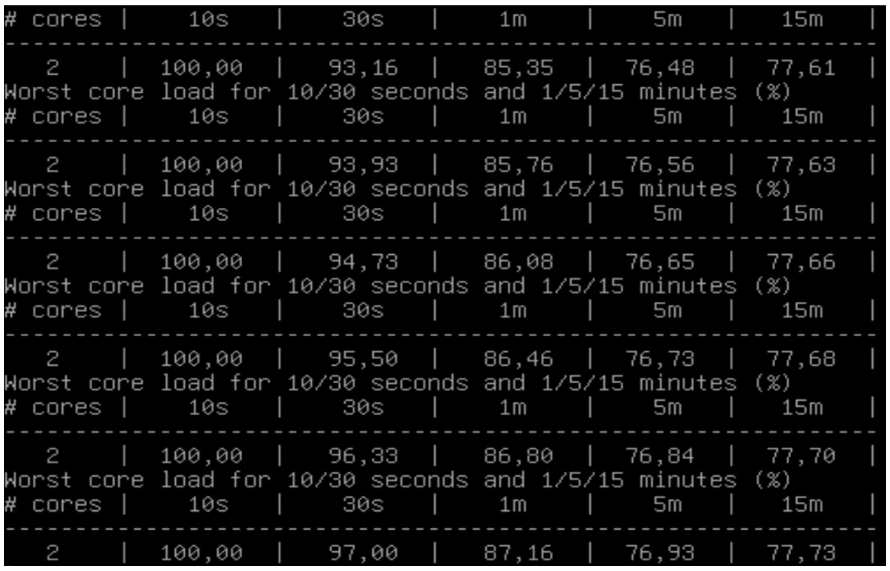

Рисунок 1 – Пиковая загрузка ЦПУ маршрутизатора при активном SNMP-опросе под нагрузкой

3.3. Методика измерения нагрузки при IPFIX

На маршрутизаторе EcoRouter был создан профиль экспорта потоков с указанием IP-адреса и UDP-порта 4739 сервера-коллектора. Значения тайм-аутов жизненного цикла потоков задавались следующими: Active Timeout 60 секунд, Idle Timeout 15 секунд. На коллекторе в конфигурационном файле Telegraf активирован плагин inputs.ipfix, прослушивающий порт 4739. Для верификации настроен плагин вывода, записывающий разобранные метрики в JSON-файл.

Нагрузка генерировалась теми же параметрами iperf3, что и при тестировании SNMP. В процессе прохождения трафика маршрутизатор начал формировать и отправлять IPFIX-сообщения. Telegraf корректно принял шаблоны, разобрал бинарные данные и записал структурированную информацию о сетевых потоках. Фрагмент журнала коллектора с разобранными записями потоков приведен на рисунке 2.

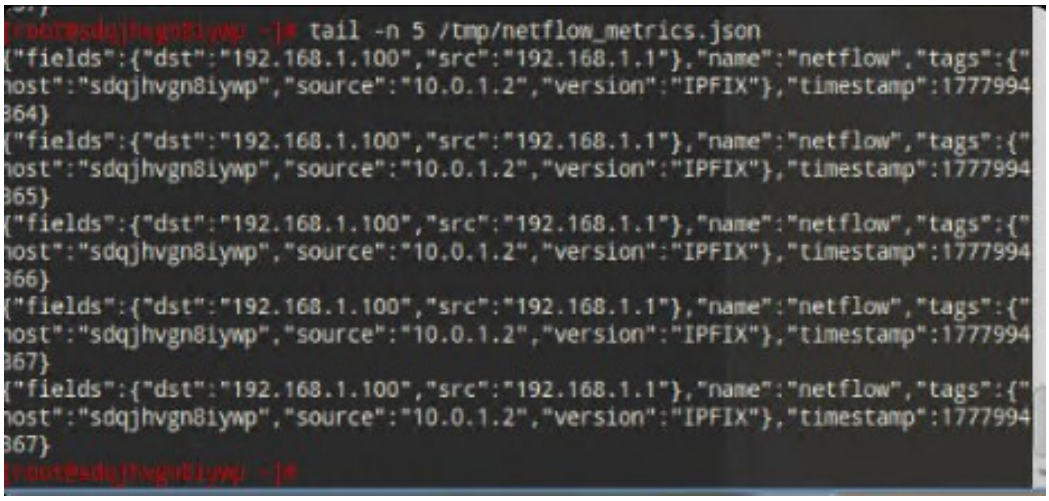

Рисунок 2 – Разобранные IPFIX-записи потоков в JSON-файле на коллекторе

Загрузка ЦПУ в активной фазе зафиксирована на уровне 83,5-84,4%, что отражено на рисунке 3.

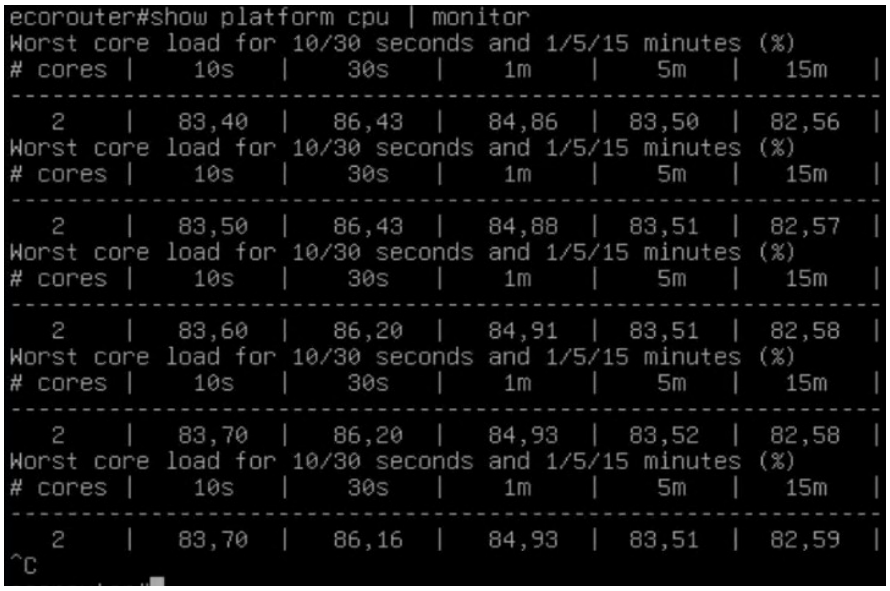

Рисунок 3 – Загрузка ЦПУ маршрутизатора при IPFIX-телеметрии под нагрузкой

Сбор метрик выполнялся в плоскости коммутации данных параллельно с маршрутизацией, процесс экспорта работал асинхронно и не создавал прерываний для управляющей плоскости.

3.4. Анализ результатов

Графики загрузки ЦПУ в двух режимах, приведенные на рисунках 1 и 3, демонстрируют принципиальное различие протоколов при одинаковой транзитной нагрузке. Для модели SNMP характерно достижение 100% на коротком интервале наблюдения, синхронное с моментами опроса. Для модели IPFIX загрузка остается стабильной в диапазоне 83,5-84,4% без выраженных скачков. Сравнительные данные двух режимов сведены в таблицу 1.

Таблица 1. Сравнение результатов тестирования

Показатель	**SNMP**	**IPFIX**
Фоновая загрузка ЦПУ (без трафика)	78–78,3%	78–78,3%
Пиковая загрузка ЦПУ под нагрузкой	100%	83,5–84,4%
Прирост нагрузки от мониторинга	до 22 п.п.	≈6 п.п.
Влияние на управляющую плоскость	Высокое	Минимальное
Объем служебного трафика	Высокий	Низкий
Детализация по потокам (L3/L4)	Нет	Есть

В отличие от предварительной оценки, строка детализации в таблице отражает принципиальное различие протоколов. SNMP предоставляет только агрегированные счетчики интерфейсов и не содержит сведений об IP-адресах, портах и протоколах отдельных потоков. IPFIX по экспортируемому шаблону передает полный контекст уровней L3 и L4, что позволяет идентифицировать источники и приемники трафика. Поэтому по критерию детализации IPFIX однозначно превосходит SNMP, что согласуется с тезисом введения.

Для иллюстрации различия в нагрузке применяется показатель относительного снижения пиковой утилизации ЦПУ:

$$\Delta CPU = (CPU\_SNMP - CPU\_IPFIX) / CPU\_SNMP \times 100\%, \quad (1)$$

где CPU_SNMP - пиковая утилизация при активном SNMP-опросе, CPU_IPFIX - пиковая утилизация при потоковом экспорте IPFIX. При

подстановке зафиксированных значений: ΔCPU = (100 - 84,4) / 100 x 100% = 15,6%. Полученная величина носит иллюстративный характер и отражает результат одной серии измерений. Каждый режим в рамках стенда тестировался однократно, поэтому статистическая достоверность разницы не оценивалась. Для получения верифицированного результата требуется провести не менее пяти серий измерений с расчетом среднего значения и стандартного отклонения, а также исследовать зависимость нагрузки от интервала опроса SNMP и значений тайм-аутов IPFIX. Полученные цифры следует рассматривать как качественную демонстрацию архитектурного преимущества push-модели, а не как точную количественную метрику.

Различие в поведении ЦПУ объясняется архитектурой протоколов. При SNMP каждый входящий GET-запрос прерывает работу процессора управляющей плоскости: требуется обход дерева OID, формирование ответа и его отправка. При IPFIX формирование записей о потоках происходит в плоскости коммутации данных одновременно с маршрутизацией пакетов. Процесс экспорта выполняется асинхронно в фоновом потоке и не создает прерываний для управляющей плоскости. Шаблоны отправляются один раз и кэшируются на коллекторе, поэтому последующие сообщения содержат только значения метрик без повторяющихся структурных заголовков.

ЗАКЛЮЧЕНИЕ

Проведенное исследование подтверждает гипотезу о снижении нагрузки на ЦПУ маршрутизатора при переходе с модели SNMP-опроса на потоковую телеметрию IPFIX, при этом полученные количественные оценки носят предварительный характер и требуют статистической верификации. Протокол SNMP создает структурный конфликт на управляющей плоскости и не предоставляет детализации по сетевым

потокам. Протокол IPFIX устраняет оба недостатка за счет интеграции сбора статистики в плоскость коммутации данных и применения асинхронного экспорта.

Операционная система EcoRouterOS успешно реализует описанную архитектуру: сенсор встроен в Fast Path, экспорт выполняется независимым процессом и не влияет на качество обслуживания транзитного трафика. Агент Telegraf с плагином inputs.ipfix обеспечивает надежный прием и разбор бинарных IPFIX-потоков при минимальном потреблении ресурсов коллектора. Практическая значимость результатов определяется возможностью применения разработанной методики при проектировании систем мониторинга корпоративных и операторских сетей на базе отечественного оборудования.

Использованные источники

1. A Framework for Network and VPN Service Performance Monitoring : RFC 9232 / A. Clemm, L. Ciavaglia, L. Granville, J. Tantsura. – IETF, 2022. – URL: https://www.rfc-editor.org/rfc/rfc9232 (дата обращения: 15.04.2026).
2. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information : RFC 7011 / B. Claise, B. Trammell, P. Aitken. – IETF, 2013. – URL: https://www.rfc-editor.org/rfc/rfc7011 (дата обращения: 15.04.2026).
3. Claise, B. Cisco Systems NetFlow Services Export Version 9 : RFC 3954 / B. Claise. – IETF, 2004. – URL: https://www.rfc-editor.org/rfc/rfc3954 (дата обращения: 15.04.2026).
4. Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX / R. Hofstede, P. Celeda, B. Trammell [et al.] // IEEE Communications Surveys & Tutorials. – 2014. – Vol. 16, № 4. – P. 2037–2064. – DOI: 10.1109/COMST.2014.2321898.

5. Trammell, B. An Introduction to IP Flow Information Export (IPFIX) / B. Trammell, E. Boschi // IEEE Communications Magazine. – 2011. – Vol. 49, № 4. – P. 89–95. – DOI: 10.1109/MCOM.2011.5741152.

6. Уймин, А. Г. Применение отечественного сетевого оборудования Eltex и EcoRouter в рамках специальности 09.02.06 «Сетевое и системное администрирование». Вопросы импортозамещения и подготовки квалифицированных кадров в сетевом оборудовании / А. Г. Уймин, И. М. Толмачев // Автоматизация и информатизация ТЭК. – 2025. – № 11 (628). – С. 58–62. – EDN DMHQJU.

7. Голованов, А. А. Сравнительный анализ систем обнаружения аномалий с использованием потока сетевого трафика и протокола NetFlow / А. А. Голованов, О. И. Мельникова // Международный научно-исследовательский журнал. – 2023. – № 6 (132).

8. Документация по EcoRouterOS. Настройка экспорта потоков (Flow Export) : официальный сайт. – URL: https://docs.ecorouter.ru/ (дата обращения: 20.04.2026).

9. Telegraf Documentation. Plugin inputs.ipfix : официальный сайт InfluxData. – URL: https://docs.influxdata.com/telegraf/v1/ (дата обращения: 07.05.2026).

10. iPerf3: a TCP, UDP, and SCTP network bandwidth measurement tool : официальный сайт. – URL: https://iperf.fr/ (дата обращения: 09.05.2026).