

Установка ПО «Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa»

1. Общая информация

Программа предназначена для осуществления суммирования пропускной способности каналов передачи данных (подвижные, проводные, спутниковые).

Программа применяется в устройствах и системах передачи данных для увеличения надежности и пропускной способности каналов передачи данных.

Программа при своей работе использует специальные алгоритмы распределения и последующего суммирования пакетов на оконечном устройстве связи (маршрутизаторе универсальном – сумматоре трафика) и на выделенном сервере (сервере терминирования).

Программа динамически перераспределяет трафик от оконечных устройств до серверов терминирования и обеспечивает автоматическое переключение при наступлении аварийной ситуации.

Функциональные возможности ПО:

- Суммирование каналов передачи данных L2 и L3 VPN
- Управление приоритезацией трафика (QoS модели по типу трафика)
- Управление статической и динамической маршрутизацией
- Управление сетевыми интерфейсами и VLAN
- Управление пакетным сетевым фильтром
- Управление работой модемов и SIM-карт
- Инструменты тестирования сети
- Регистрации ключевых метрик работы устройств с инструментарием ретроспективного анализа состояния устройств в любой момент времени
- Управление внешними подключениями
- Управление и мониторинг состояний устройств и интерфейсов
- Удаленное обновление экземпляров ПО на оконечных устройствах

Программное обеспечение «Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa» является встроенным в ПАК «Avotel Summa» и разделяется на:

- ПО для конечных устройств CPE
- Подсистему серверного ПО

Все части ПО функционируют в среде Linux-подобной операционной системы Debian не ниже версии 10.9.

ПО для конечных устройств CPE представлено программной частью “AVOTEL-SUM-CLIENT”.

Подсистема серверного ПО представлена несколькими основными программными частями (модулями, программными модулями):

- “AVOTEL-SUM-SERVER”
- “AVOTEL-NMS-SERVER”
- “AVOTEL-LIC-SERVER”

Программные модули устанавливаются на отдельных физических или виртуальных серверах. Допускается устанавливать программные модули “AVOTEL-NMS-SERVER” и “AVOTEL-LIC-SERVER” на одном физическом сервере. Для обеспечения отказоустойчивости и резервирования рекомендуется устанавливать два экземпляра программных модулей “AVOTEL-SUM-SERVER” на двух разнесенных серверах.

Программные модули подсистемы серверного ПО во время установки объединяются в логическую сущность DC Instance.

2. Установка серверной подсистемы ПО “Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa”

2.1. Среда установки ПО “Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa”

Для установки серверной подсистемы ПО “Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa” должна быть развернута среда установки на отдельном ПК (машине), либо на виртуальной машине. Пользователь заранее должен согласовать параметры доступа (имя DC Instance, количество лицензий, идентификатор клиента, имя клиента и пароль, в соответствии с которыми будет обеспечен доступ к ПО на сайте Правообладателя).

2.1.1. Общие технические требования для платформы среды установки серверной подсистемы.

Параметр	Минимальная конфигурация
CPU (vCPU)	2
RAM (GB)	8
Storage (GB)	64

Операционная система	Версия системы
Debian	10.9

- Должна быть обеспечена полная сетевая доступность между машиной, на которой развернута среда установки, и остальными серверами;
- У машины на время сеанса установки должен быть сетевой доступ до реестра Правообладателя ПО
- У всех серверов должно быть корректное время;
- В инфраструктуре должен быть dns сервер.

2.1.2. Подготовка среды установки:

1) Установить параметр для изменения профайлов:

```
allow_world_readable_tmpfiles=1
```

2) Установить коллекцию Ansible:

```
ansible-galaxy collection install git+https://a.b.c.d/avotel/ansible.git#summa/dc -f
```

3) Установить пакет rsync:

```
apt install rsync
```

4) Создать inventory файл. см. inventory/hosts.yaml:

#В inventory/hosts.yaml, заменить <dc_instance> на идентификатор DC Instance all:

```
children:
  #by environment
  stage:
    vars:
      env: test
    children:
      <dc_instance>:
external:
  children:
    <dc_instance>:
dc:
  children:
    dc_terminal:
      hosts:
        <dc_instance>:
    dc_license:
      hosts:
        <dc_instance>:
    dc_monitoring:
      hosts:
        <dc_instance>:
dc-instances:
  children:
    <dc_instance>:
      hosts:
        <dc_instance>:
          ansible_host: <адрес сервера>
```

5) Создать файл group_vars/<dc_instance>.yml:

```
# group_vars/<dc_instance>
#required in license
#dc_license
```

```
instance_name: <dc_instance>
```

```
# ПО (будет скачано из репозитория по rsync)
# версия сборки ПО
summa_assembly_version: 2022-06-14.11-52.tailer-rc2
```

```
DC_LICENSE_ADMIN_PASSWORD: "<supersecret>"
```

```
DC_LICENSE_LICENSE_PASSWORD: "<supersecret>"

#for post-install TODO
CLIENT_NAME: "Avotel"
CLIENT_LOGIN: avotel
DC_MONITORING_CLIENT_PASSWORD: <secret>
DC_MONITORING_CLIENT_UUID: "<вставить UUID пользователя>"

##### required in terminal
# see host_vars files
# fall back to default ip by default
DC_TERMINAL_SERVER_HOST: "{{ ansible_default_ipv4.address }}"

# required in monitoring
DC_LICENSE_SERVER_HOST: localhost
DC_MONITORING_ADMIN_PASSWORD: "<supersecret>"
```

6) Задать параметры для настройки в group_vars:

Для установки DC (общее для всех сервисов)

Необходимо задать следующие переменные:

- `instance_name`

Для установки сервера лицензий (dc-license-server)

Необходимо задать:

- `DC_TERMINAL_SERVERS_LIST` - список серверов терминирования. Использовать адреса, доступные клиентам (default: массив из DC_TERMINAL_SERVER_HOST всех серверов терминирования)
- `DC_LICENSE_ADMIN_PASSWORD` - пароль для доступа к API сервера лицензий
- `DC_LICENSE_SERVER_PORT` - TCP-порт
- `VPN_NETWORK_BASE_NET` - сеть в VPN
- `DC_LICENSE_LICENSE_PASSWORD` - используется в ссылке активации для CPE
- `DC_LICENSE_LICENSE` - лицензия, полученная от Avotel. Для установки лицензии запустить плейбук site.yml с `--tags=license`
- `IPC_LS` - endpoint для nsq, используется в конфиге license сервера # практически не используется на практике.

Параметры для подключения к БД:

- `SQL_DB_HOST`, (default: localhost)
- `SQL_DB_PORT`: XXXX
- `SQL_MAIN_PASSWORD`: PPPP
- `SQL_MAIN_USER`: UUUU
- `SQL_DB_USER`: UUUM
- `SQL_DB_PASSWORD`: PPPU
- `SQL_DB_NAME_TS_MS`: DBNAME1
- `SQL_DB_NAME_LS`: DBNAME2
- `SQL_DB_SCHEME`: SCHEME
- `SQL_DB_ALLOW_FROM` - список адресов с маской (/32, /24 и т.д.), с которых допустимо подключение к БД

Для установки сервера терминирования (dc-terminal-server)

Необходимо задать:

- `DC_LICENSE_SERVER_HOST` - адрес сервера лицензий (можно использовать адреса private ip или localhost, если сервисы расположены вместе) (default: {{ ansible_default_ipv4.address }})
- `DC_TERMINAL_SERVER_HOST` - хост/id терминального сервера, используется в id, должен присутствовать в DC_TERMINAL_SERVERS_LIST (default: ansible default ip address)
- параметры БД (см выше)
- `DC_TERMINAL_LICENSE` - лицензия, полученная от Avotel. Для установки лицензии запустить плейбук с `--tags=license -e "DC_TERMINAL_LICENSE=<license key>"`
- `DC_LICENSE_SERVER_PORT` - TCP-порт сервера лицензий
- `DC_TERMINAL_SERVER_PORT` - порт терминального сервера
- `IPC_TS_MS` - endpoint для nsq (default: "localhost:XXXX")
- `DC_TERMINAL_SERVER_LISTEN_ADDR` - адрес на котором слушает dc-terminal-server. (default: "", что соответствует `DC_LICENSE_SERVER_HOST`, можно заменить на 0.0.0.0)
- `DC_TERMINAL_WEB_PROXY_BIND_ADDRESS` - адрес на котором будет запускаться веб-прокси (default: отсутствует)
- `DC_TERMINAL_PROXY_EXTENDED_REDIR` - нужно выставить как `true` для интеграционных тестов (default: false)

Для установки сервера управления (dc-monitoring-server)

Необходимо задать:

- `DC_LICENSE_SERVER_HOST` - адрес сервера лицензий (можно использовать адреса private ip или localhost, если сервисы расположены вместе)
- параметры БД (см выше)

- `DC_LICENSE_SERVER_PORT` - TCP-порт сервера лицензий
- `DC_MONITORING_ADMIN_PASSWORD` - админский пароль для доступа на сервер мониторинга (default: генерится случайным образом)
- `DC_MONITORING_RSYNC_PASSWORD` - shared secret для rsync (default: генерится случайным образом)
- `IPC_TS_MS` - endpoint для nsq (default: "localhost:XXXX")
- `IPC_PEER` - адрес IPC пира (что при LS) (default: `DC_LICENSE_SERVER_HOST:8140`)
- `DC_MONITORING_TOKEN_USER` - (default: UUUU)
- `DC_MONITORING_SERVER_PORT` - (default: XXXX)
- `DC_MONITORING_TOKEN_PASSWORD` - ? (default: генерится в папке host_secrets)

7) Создать файл плейбука Ansible:

плейбук site.yml

- name: Prepare avt hosts (external)
 - hosts: all
 - become: yes
 - tags: prepare
 - collections:
 - summa.dc
 - roles:
 - name: summa_pkg
 - name: docker
 - when: install_docker is defined and install_docker | bool
 - vars:
 - disable_docker_networking: yes
 - name: common
 - name: shell
 - name: avt_rc_local
 - when: install_avt_rc_local is defined and install_avt_rc_local | bool
- name: Setup Summa DC license servers
 - gather_facts: no
 - hosts: dc_license
 - become: yes
 - roles:
 - name: summa.dc.dc_license_server
- name: Setup Summa DC monitoring servers
 - gather_facts: no
 - hosts: dc_monitoring

become: yes

roles:

- name: summa.dc.dc_monitoring_server

- name: Setup Summa DC terminal servers

gather_facts: no

hosts: dc_terminal

become: yes

collections:

- summa.dc

roles:

- name: summa.dc.dc_terminal_server

- name: iperf_server

- name: summa_iperf3_exporter

when: install_iperf3_exporter is defined and install_iperf3_exporter | bool

- name: Harden site (iptables)

hosts: all:!nomad_remote

gather_facts: yes

#remote_user: roots

tags: iptables

become: yes

collections:

- summa.dc

roles:

- name: summa_iptables

when: install_iptables is defined and install_iptables | bool

- name: Setup SQL exporter

hosts: dc_monitoring

become: yes

collections:

- summa.dc

roles:

- name: summa_sql_exporter

when: >

install_sql_exporter is defined and install_sql_exporter | bool and
(k8s_setup is not defined or k8s_setup is false)

- name: Harden site

hosts: all:!nomad_remote

gather_facts: no

#remote_user: roots

tags: harden

become: yes

collections:

- summa.dc

roles:

- name: harden

2.2. Установка программных модулей серверной подсистемы ПО «Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa»

Будет выполнена установка и настройка всех программных модулей серверной подсистемы ПО «Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa» на лицензионный сервер, сервер терминирования, сервер управления, а также NSQ и postgresql посредством исполнения сценария Ansible.

Ввести следующую команду (исполнение сценария):

```
ansible-playbook site.yml -l <dc_instance>
```

2.3. Добавление пользователей и лицензий для серверной подсистемы ПО «Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa»

1) В файл group_vars/<dc_instance>.yml добавьте переменную client_list, где укажите необходимое количество лицензий:

```
clients_list:
- name: "client2"
  login: "client2"
  # vault: no - задать вручную, ограниченная поддержка режима без vault
  vault: no
  CLIENT_NAME: client2
  CLIENT_LOGIN: client2
  DC_MONITORING_CLIENT_PASSWORD: mypass
  #DC_MONITORING_CLIENT_UUID: xxxx
  #DC_MONITORING_CLIENT_ID:
  licenses:
  - {"interfaces": 5, "count": 250}
  - {"interfaces": 2, "count": 2}
```

2) Создайте плейбук post-install.yml:

```
#post-install.yml
```

```
- name: Update customer secrets
  hosts: "!keepalived_replicas"
  run_once: yes
  tasks:
    - debug:
        msg: "{{ clients_list }}"

    - name: Prepare customer ids
      set_fact:
        clients_list_merged: []
      tags: update_license
    - name: Prepare customer ids
      set_fact:
        clients_list_merged: "{{ clients_list_merged + [ c |
combine(lookup('community.general.hashi_vault', ('secret=dc-
instance/data/'+instance_name+''+c.name)) if c.vault else {} ) ] }}"
      loop: "{{ clients_list }}"
      loop_control:
        loop_var: c
      tags: update_license
    - debug:
        msg: "{{ clients_list_merged }}"
    - name: Create license role
      include_role:
        name: summa.dc.dc_client
      vars:
        CLIENT_NAME: "{{ c.CLIENT_NAME }}"
        CLIENT_LOGIN: "{{ c.CLIENT_LOGIN }}"
        # DC_MONITORING_CLIENT_PASSWORD: "{{
c.DC_MONITORING_CLIENT_PASSWORD }}"
        dc_client_licenses: "{{ c.licenses | default(_licenses) }}"
        DC_MONITORING_CLIENT_UUID: "{{ c.DC_MONITORING_CLIENT_UUID |
default('') }}"
        use_vault: "{{ c.vault }}"
      loop: "{{ clients_list_merged }}"
      loop_control:
        loop_var: c
      tags: update_license
    - name: Prepare customer ids (one more time, update UUID)
      set_fact:
        clients_list_merged: []
    - name: Prepare customer ids (one more time, update UUID)
```

```
set_fact:
  clients_list_merged: "{{ clients_list_merged + [ c |
combine(lookup('community.general.hashi_vault', ('secret=dc-
instance/data/'+instance_name+'+c.name)) if c.vault else {} ) ] }}"
  loop: "{{ clients_list }}"
  loop_control:
    loop_var: c
```

- name: Set vars

```
set_fact:
  clients_list_merged: "{{ clients_list_merged }}"
  cacheable: yes
  delegate_to: "{{ item }}"
  delegate_facts: True
  # only for hosts in same dc-instance
  with_items: "{{ (groups['dc_monitoring'] + groups['dc_license']) |
intersect(groups[instance_name]) | unique }}"
```

- name: Create CPE setup instruction(check dir)

```
delegate_to: localhost
delegate_facts: True
file:
  path: "host_extra/{{ instance_name }}"
  state: directory
  mode: '0755'
vars:
  CLIENT_NAME: "{{ client.name }}"
  CLIENT_LOGIN: "{{ client.login }}"
  vault_user: "{{ lookup('community.general.hashi_vault', 'secret=dc-instance/data/{{
instance_name }}/{{ CLIENT_NAME | urlencode }}})"
  DC_MONITORING_CLIENT_PASSWORD: "{{
vault_user.DC_MONITORING_CLIENT_PASSWORD }}"
  DC_MONITORING_CLIENT_UUID: "{{
vault_user.DC_MONITORING_CLIENT_UUID }}"
  loop: "{{ clients_list_merged }}"
  loop_control:
    loop_var: c
```

- name: Create CPE setup instruction

```
delegate_to: localhost
delegate_facts: True
template:
  src: for_cpe.j2
  dest: "host_extra/{{ instance_name }}/for_cpe_{{ instance_name }}_{{
CLIENT_NAME }}"
vars:
```

```

CLIENT_NAME: "{{ c.CLIENT_NAME }}"
CLIENT_LOGIN: "{{ c.CLIENT_LOGIN }}"
DC_MONITORING_CLIENT_PASSWORD: "{{
c.DC_MONITORING_CLIENT_PASSWORD }}"
DC_MONITORING_CLIENT_UUID: "{{ c.DC_MONITORING_CLIENT_UUID }}"
DC_MONITORING_CLIENT_ID: "{{ c.DC_MONITORING_CLIENT_ID }}"
loop: "{{ clients_list_merged }}"
loop_control:
  loop_var: c

```

- name: Post-install dc_monitoring
 - hosts: dc_monitoring
 - become: yes
 - vars:
 - DC_MONITORING_CLIENT_UUID_LIST: []
 - tasks:
 - name: Prepare DC_MONITORING_CLIENT_UUID_LIST
 - set_fact:
 - # CLIENT_NAME: "{{ c.CLIENT_NAME }}"
 - # CLIENT_LOGIN: "{{ c.CLIENT_LOGIN }}"
 - # DC_MONITORING_CLIENT_PASSWORD: "{{
 c.DC_MONITORING_CLIENT_PASSWORD }}"
 - # DC_MONITORING_CLIENT_UUID: "{{ c.DC_MONITORING_CLIENT_UUID }}"
 - DC_MONITORING_CLIENT_UUID_LIST: "{{
 DC_MONITORING_CLIENT_UUID_LIST + [c.DC_MONITORING_CLIENT_UUID] }}"
 - loop: "{{ clients_list_merged }}"
 - loop_control:
 - loop_var: c
 - name: Prepare DC_MONITORING_CLIENT_UUID_LIST
 - set_fact:
 - DC_MONITORING_TOKEN_USER: "{{ clients_list_merged[0].CLIENT_LOGIN }}"
 - name: Print DC_MONITORING_CLIENT_UUID_LIST
 - debug:
 - var: DC_MONITORING_CLIENT_UUID_LIST
 - name: Update config
 - include_role:
 - name: summa.dc.dc_monitoring_server
 - tasks_from: add-clients-to-config

DC_MONITORING_CLIENT_UUID is set to config

- meta: flush_handlers

- name: Post-install dc_monitoring (create user)
 - hosts: dc_monitoring:!keepalived_replicas
 - become: yes

```

run_once: yes
tasks:
- name: Create user in monitoring
  include_role:
    name: summa.dc.dc_monitoring_server
    tasks_from: create-user
  vars:
    CLIENT_NAME: "{{ c.CLIENT_NAME }}"
    CLIENT_LOGIN: "{{ c.CLIENT_LOGIN }}"
    DC_MONITORING_CLIENT_PASSWORD: "{{
c.DC_MONITORING_CLIENT_PASSWORD }}"
    DC_MONITORING_CLIENT_UUID: "{{ c.DC_MONITORING_CLIENT_UUID }}"
    DC_MONITORING_CLIENT_ID: "{{ c.DC_MONITORING_CLIENT_ID }}"
  loop: "{{ clients_list_merged }}"
  loop_control:
    loop_var: c

- name: Restart replicas (to init standby)
  hosts: keepalived_replicas
  become: yes
  tasks:
  - name: Restart keepalived to init standby.sh
    systemd:
      name: keepalived
      state: restarted
  - name: Wait for postgresql replica to become ready
    wait_for:
      host: localhost
      port: 5432
      delay: 1
      timeout: 300
  - name: Wait for dc-license-server replica to become ready
    wait_for:
      host: localhost
      port: 18090
      delay: 1
      timeout: 300

```

3) Создание пользователей и активации лицензий:

```
ansible-playbook post-install.yml -l <dc_instance>
```

Серверная подсистема ПО “Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa” установлена и готова к работе.

3. Установка ПО “Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa” на оконечное клиентское устройство CPE

На устройстве CPE будет установлен и настроен программный модуль “AVOTEL-SUM-CLIENT”.

3.1. Общие технические требования для выполнения процесса установки ПО на CPE:

- Устройство CPE должно иметь доступ в интернет;
- Должна обеспечена возможность доступа на устройство CPE по протоколу SSH;
- На устройстве CPE должен быть установлен пакет rsync, если его нет то его необходимо установить, выполнив команду: `apt install rsync`
- Должен быть известен актуальный пароль для доступа к файловому хранилищу по протоколу RSYNC, в командах его будет необходимо указать вместо ACTUAL_RSYNC_PASSWORD. Актуальный пароль необходимо заранее согласовать с Правообладателем ПО.
- У Правообладателя ПО должен быть запрошен и получен по электронной почте файл с параметрами подключения активации и ПО на CPE.

3.2. Установка ПО на CPE

1) Для установки необходимо зайти на устройство по SSH и запустить подряд три команды:

- `env RSYNC_PASSWORD="ACTUAL_RSYNC_PASSWORD" rsync -azv --no-o --no-g --progress --partial rsync://user@server/store/files/part1/install.sh install.sh && chmod +x install.sh && ./install.sh`

- `env RSYNC_PASSWORD="ACTUAL_RSYNC_PASSWORD" rsync -azv --no-o --no-g --progress --partial rsync:// user@server /store/files/part2/install.sh install.sh && chmod +x install.sh && ./install.sh store/files/part2`
- `SRV="server"; PASS="ACTUAL_RSYNC_PASSWORD"; RELE="2022-06-22.13-57.tailer-rc3"; env RSYNC_PASSWORD="$PASS" rsync -cazv --no-o --no-g rsync://user@$SRV/store/cpe/$RELE/install.sh ./install.sh && chmod +x install.sh && ./install.sh $RELE $PASS $SRV`

2) Изменить следующие параметры путем редактирования файлов. Доступ производится через протокол SSH:

- через редактирование файла `/home/roots/.ssh/authorized_keys` актуализировать список ключей ssh;
- через редактирование файла `/etc/systemd/timesyncd.conf` изменить адрес сервера NTP на текущий адрес;
- через редактирование файла `/opt/ap/etc/openvpn-client.conf` изменить адрес сервера OpenVPN на текущий адрес;

3) Изменить настройки CPE. Доступ через WEB интерфейс.

- Добавить необходимые интерфейсы в суммирование:
меню Настройки -> Сеть, в шапке таблицы справа от заголовка столбца "Summa" иконка "гаечный ключ";
- Добавить параметры подключения к серверу:
меню Настройки -> Файл конфигурации, вставить параметры подключения в начало окна. Параметры представляют набор строк, которые берутся из файла берутся из простого текстового файла

3.3. Настройка подсети на устройстве CPE:

- 1) Получить подсеть для конкретного CPE, например 1.2.0.0/28
- 2) Включите CPE
- 3) Зайдите в веб-интерфейс сервера управления (далее СУ)
- 4) Найдите CPE, например поиском по серийному номеру
- 5) Определите адрес SummaVPN CPE, например 3.4.0.5
- 6) Зайдите по протоколу SSH на сервер управления
- 7) В командной строке введите команду
`sudo -u user env PAGER="less -S" psql -d dbname`

- 8) Запустится клиент БД PostgreSQL, приглашение должно стать dbname=#
- 9) Используя подсеть из п.1 и адрес из п.5 введите следующий запрос к БД, заменив значения из примеров на реальные:

```
update devices set network_variant=1,routes='1.2.0.0/28' where ip='3.4.0.5';
```
- 10) В случае успешного выполнения команды появится сообщение UPDATE 1
- 11) Зайдите в веб-интерфейс управления CPE через СУ
- 12) В меню выберите Настройки, Сеть
- 13) В таблице найдите интерфейс с названием LAN и запомните имя интерфейса, указанное в скобках (для моделей A9x это br0)
- 14) В шапке таблицы нажмите иконку справа от заголовка столбца "Интерфейсы"
- 15) Откроется окно с сетевыми настройками, найдите в нем интерфейс по имени из п.13
- 16) Замените адрес в строке address на первый адрес в полученной подсети из п.1, например для подсети 1.2.0.0/28 настройка интерфейса должна выглядеть следующим образом:

```
auto br0
iface br0
inet static
bridge_ports lan1 lan2 lan3 lan4address 1.2.0.1/28
```

- 17) Поставьте галочку "перезапустить сервис networking после сохранения /etc/network/interfaces" и нажмите кнопку "Сохранить"
- 18) В таблице, в строке с интерфейсом LAN, в колонке "**DHCP/DNS Server**", нажмите иконку справа от переключателя "вкл/выкл". Откроется окно с настройками DHCP/DNS сервера для выбранного интерфейса
- 19) Необходимо отредактировать параметр dhcp-range, указав начальный и конечный адреса диапазона IP адресов, из которого будут выдаваться адреса подключаемым к CPE устройствам, например для подсети 10.16.0.0/28 параметр должен выглядеть так:

```
dhcp-range=1.2.0.2,1.2.0.14,12h
```
- 20) Убедитесь что указаны правильные адреса DNS серверов, которые будут получать устройства, в случае необходимости скорректируйте:

```
dhcp-option={{.INTERFACE_NAME}},6,8.8.8,8.8.4.4
```
- 21) Нажмите кнопку "Сохранить" для записи настроек
- 22) Перезапустите CPE, выбрав в меню Настройки, Питание и нажав иконку "Перезагрузить"

Клиентская программная часть ПО “Система управления и мониторинга маршрутизатором универсальным – сумматором трафика Avotel Summa” на клиентском устройстве СРЕ установлено, устройство готово к работе.