Kea DHCP

(A powerful DHCP Server)

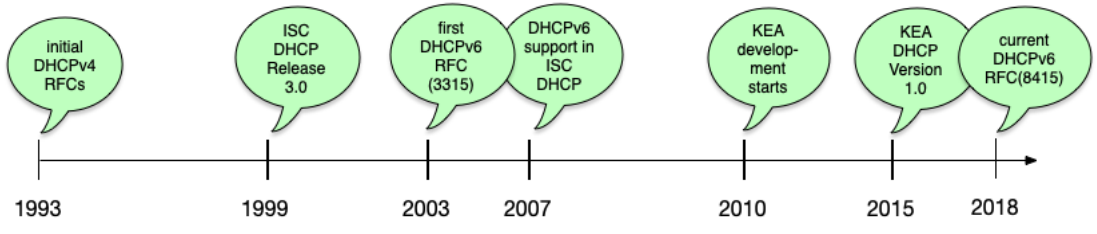
Carsten Strotmann

CREATED: 2023-02-02 THU 16:56

About DHCP

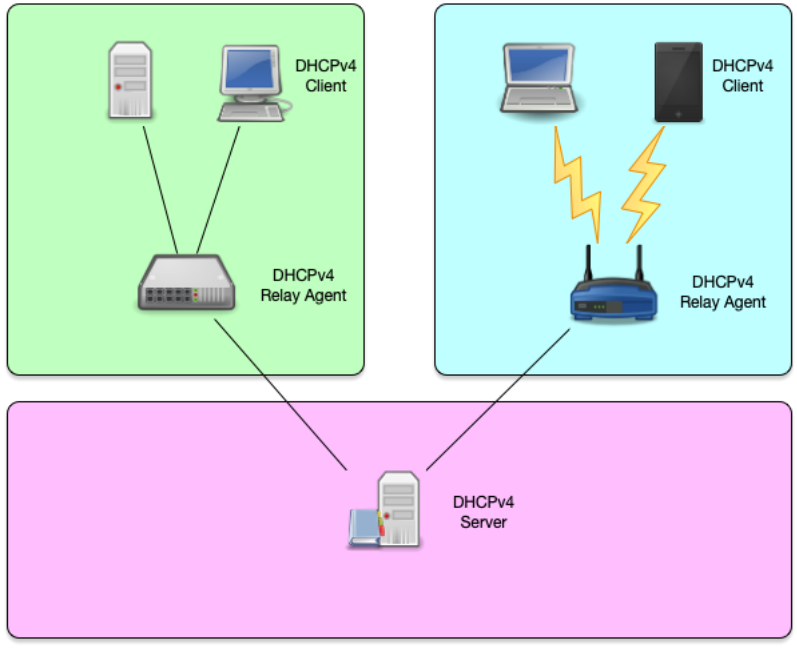
- DHCP is short for Dynamic Host Configuration Protocol
  - the Internet standard protocol to assign IP addresses and network related configuration to TCP/IP connected machines
  - for IPv4: RFC 2131
  - for IPv6: RFC 8415

Short history of DHCP (including ISC-DHCP and Kea)

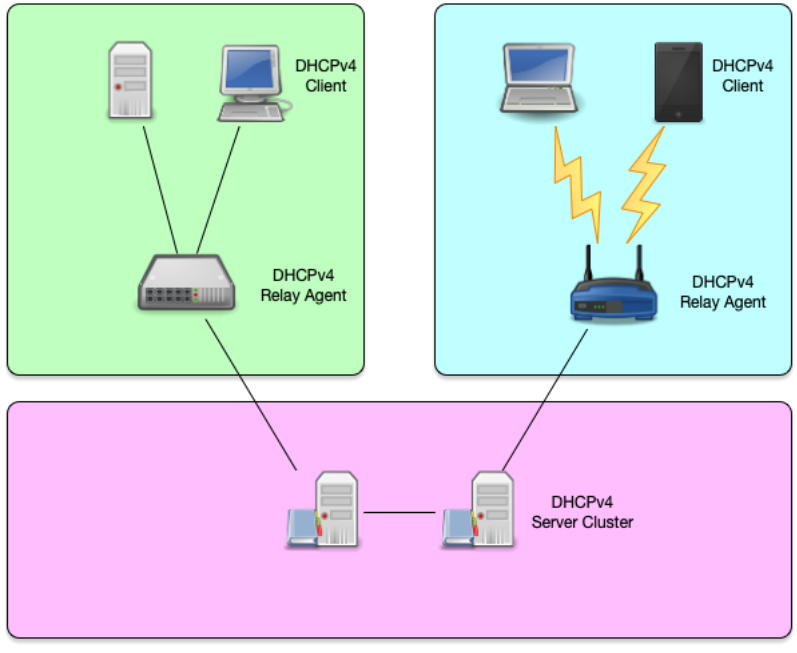


DHCPv4 overview

DHCPv4 overview (1/2)



DHCPv4 overview (2/2)



DHCPv4 protocol

- the DHCPv4 protocol uses UDP broadcast and (in some situations) unicast
    - the DHCPv4 server or relay agents listen on port 67
    - a DHCPv4 client listens on port 68 for messages from a server or relay agent
    - the initial request from a client requires layer 2 (Ethernet) communication

DHCPv4 Lease concept

DHCPv4 Lease (1/3)

- when using the DHCP protocol, a client can never keep an IP Address forever
    - each IP Address given out by a DHCP server has a "lease" time
    - this is the time in seconds that the client is allowed to use the IP Address

DHCPv4 Lease (2/3)

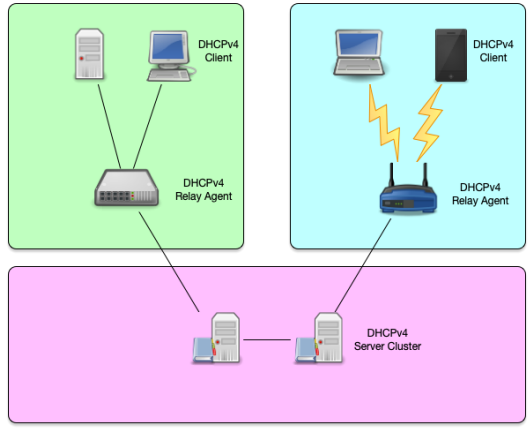
- the "lease" time is delivered in an DHCP option
  - it is a 32bit value
  - the maximum lease time is $FFFFFFFE (= 4294967294 seconds or ~136 years)
  - a lease time of $FFFFFFFF indicates an infinite lease

DHCPv4 Lease (3/3)

- according to the RFC, a DHCP server **must** store the lease information to permanent storage before confirming the IP address to a client
  - this can be a performance bottleneck on a DHCP server
  - we will discuss the various options in one of the following chapters

DHCP Clients, Relays and Server

- because a client machine without IP address can only communicate on the local link, the base DHCPv4 protocol is "link-local" only
- DHCPv4 relay-agents can be used to forward DHCPv4 requests to centralised DHCPv4 server
    - DHCP relay-agents are often found in network equipment (e.g. router)
    - dedicated "software based" relay-agents are available



What is Kea DHCP

Kea DHCP (1/2)

- A modern DHCPv4 and DHCPv6 Server
- Open Source (MPL License)
- JSON/REST API
- Modular design
- High performance (> 1000 leases/seconds with SQL database backend)
- Failover via SQL DB or High-Availability-Hook
- Host reservation support

Kea DHCP (2/2)

- Support for DHCPv6 prefix delegation
- Dynamic reconfiguration
- Dynamic DNS updates
- SQL database backend (MySQL / PostgreSQL ...)
- Statistics module
- PXE Boot support

Platforms and Requirements

Kea Platforms (1/2)

- Kea is officially supported on
  - CentOS Linux
  - Fedora Linux
  - Ubuntu Linux
  - Debian Linux
  - FreeBSD Unix

Kea Platforms (2/2)

- Kea is also know to work on
  - MacOS X
  - Alpine Linux
  - OpenBSD Unix
- Kea currently does not work on Windows operating systems

Kea DHCP requirements

- Kea requires to run
  - A cryptographic library: Botan or OpenSSL
  - log4cplus C++ logging library
  - The Boost C++ system library

- Optional components
  - A database such as MySQL or PostgreSQL
    - Support for Cassandra (CQL) has been removed from Kea DHCP

  - FreeRadius-client library for Radius support
  - Sysrepo for NETCONF support

- See the Kea DHCP documentation for detailed information on software dependencies

Installation

Kea DHCP installation via operating system packages

- Kea DHCP is available in the package repositories of all major Linux and Unix systems
    - If you have support from the operating system vendor (Red Hat, Canonical, Suse), installing from the OS repositories is the best choice
- Kea DHCP can also be installed from source, if you need a special build configuration or the latest features not available in the binary packages

Kea DHCP installation via ISC packages

- ISC offers binary packages of Kea DHCP for our users and customers, hosted on Cloudsmith.
    - If you need the latest Kea version, these packages are an alternative to building Kea from source
- The packages provide fast access to the latest bug fixes
    - ISC provides the binary packages along with sources at the time of release

Kea binary packages from ISC

- The open source packages contain the base Kea software and the following hooks libraries:
  - Flexible Option
  - Lease Commands
  - High Availability
  - Statistics Commands
  - BOOTP
  - Run Script
  - User Check

Packages for support customers

- Users of Kea that purchase professional Kea DHCP support from ISC are entitled to special software features that are not available in the open source version:
    - Class Commands
    - Configuration Backend Commands
    - Flexible Identifier
    - Forensic Logging
    - Host Cache
    - Host Commands
    - RADIUS support
    - Subnet Commands
    - GSS-TSIG (Microsoft AD dynamic DNS Updates)

Kea hooks

The Kea hooks

- The base Kea software implements the basic DHCPv4 and DHCPv6 functions
- These basic functions can be extended via ***hooks***.
    - The ***hooks*** are libraries that contain extra functions that will be called when a DHCP request is processed
    - Hooks allow the core Kea system to stay lean
    - Installations only load the functions used and needed
    - This reduces the complexity and the attack surface of an installation

Types of hooks available

- Hooks that are part of the Kea open source code (source and binary packages)
- Premium hooks that can be purchased online from the ISC website
- Hooks that are available for ISC support subscription customers
- Third party hooks (source code)

Premium/Subscription hooks

- The premium/subscription hooks are available in source and binary (package) form
    - Customers can download the hooks for a period of 12 month
    - As the API between Kea and the hooks might change between Kea versions, care must be taken to install hooks that match the Kea version number

Kea configuration

JSON

- Configuration files for the **DHCPv4**, **DHCPv6**, **DDNS**, Control Agent, and **NETCONF** modules are defined in an extended JSON format.
- Basic JSON is defined in RFC 7159 and ECMA 404.

```
{
"Dhcp4": {
    "interfaces-config": {
        "interfaces": [ "eth0" ]
    },
    "control-socket": {
        "socket-type": "unix",
        "socket-name": "/tmp/kea-dhcp4.socket"
    },
[...]
```

Extended JSON

- Kea components use an extended JSON with additional features:
  - Shell comments: any text after the hash (#) character is ignored.
  - C comments: any text after double slashes (`//`) is ignored.
  - Multiline comments: any text between `/*` and `*/` is ignored. This commenting can span multiple lines.
  - File inclusion: JSON files can include other JSON files by using a statement of the form `<?include "file.json"?>`.

JSON Editor

- When working with KEA, it helps to have an editor that understands the JSON format, can check the syntax and can highlight and reformat JSON data
  - Emacs
  - VIM
  - Visual Studio Code
  - TextMate / BBEdit
  - ...

EMACS JSON Mode

- EMACS JSON Mode: https://www.emacswiki.org/emacs/JSON
- Enable *JSON-Mode* in Emacs with **`ESC-X json-mode<enter>`**
- Re-format a JSON file with **`CTRL+c-CTRL+f`**

VIM JSON Syntax Highlighting

- Using **`vim`**, syntax highlighting for JSON can be enabled in the command mode with

```
: set syntax=json
```

Location of the KEA configuration files

- On most Linux/Unix systems, the Kea configuration files can be found under **`/etc/kea/`**
    - Some have their own locations such as **`/usr/local/etc/kea`** on FreeBSD

Kea configuration files

- The main Kea configuration file are
  - **`kea-ctrl-agent.conf`** - Kea control agent
  - **`kea-dhcp-ddns.conf`** - Kea dynamic DNS updater
  - **`kea-dhcp4.conf`** - Kea DHCPv4 server
  - **`kea-dhcp6.conf`** - Kea DHCPv6 server
  - **`keactrl.conf`** - configuration file for **`keactrl`** script (not in JSON format)

Documentation

- The example configuration files provided by the Kea project contain extensive comments
- The full documentation can be found online at https://kea.readthedocs.io

A basic Kea DHCPv4 configuration

Network Interface and control socket

- The Kea DHCP server needs to know on which network interfaces the DHCP service should listen on
- The control socket defines the communication interface between the DHCP server process and the administration tools

```
{
"Dhcp4": {
    "interfaces-config": {
        "interfaces": [ "eth0" ]
    },
    "control-socket": {
        "socket-type": "unix",
        "socket-name": "/tmp/kea-dhcp4.socket"
    },
[...]
```

Lease database definition

- Kea DHCP needs to know where to store the lease information. The configuration snippet below defines a *in-memory* database

```
[...]
    "lease-database": {
        "type": "memfile",
        "lfc-interval": 3600
    },
[...]
```

Global configuration and options

- Some DHCP configurations are ***global*** and apply to all the subnets and pools managed by the DHCP server

```
[...]
    "renew-timer": 900,
    "rebind-timer": 1800,
    "valid-lifetime": 3600,
[...]
```

IPv4-Subnet and Pool definition

- The example of a subnet below with DHCP pool definition includes subnet specific options (default router option: `routers`)

```
[...]
    "subnet4": [
        {
            "subnet": "192.0.2.0/24",
            "pools": [ { "pool": "192.0.2.100 - 192.0.2.200" } ],
            "option-data": [
                {
                    "name": "routers",
                    "data": "192.0.2.1"
                }
            ]
        }
    ]
},
[...]
```

Logging

- Kea DHCP has comes with a flexible and powerful logging framework
- The configuration snippet below configures a log-file for the DHCPv4 service

```
[...]
  "loggers": [
    {
        "name": "kea-dhcp4",
        "output_options": [
            {
                "output": "/var/log/kea-dhcp4.log"
            }
        ],
        "severity": "INFO",
        "debuglevel": 0
    }
  ]
}
```

Checking the configuration for syntax errors

Kea configuration syntax check

- After changes to a configuration file, and before reloading the new configuration into the Kea server, the configuration file should be checked for errors
    - Syntax checks can be done with the `-t` (test) parameter

Kea configuration check examples (1/2)

- KEA-DHCP4 Syntax Check with error (line 33, char 9)

```
# kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
Syntax check failed with: /etc/kea/kea-dhcp4.conf:33.9: syntax error, unexpected }
```

Kea configuration check examples (2/2)

- Successful Syntax check (output wrapped for readability)

```
# kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
INFO  [kea-dhcp4.dhcpsrv/51] DHCPSRV_CFGMGR_ADD_IFACE listening on interface serve
INFO  [kea-dhcp4.dhcpsrv/51] DHCPSRV_CFGMGR_SOCKET_TYPE_DEFAULT "dhcp-socket-type"
                             using default socket type raw
INFO  [kea-dhcp4.dhcpsrv/51] DHCPSRV_CFGMGR_NEW_SUBNET4 a new subnet has been adde
                             192.0.2.0/24 with params: t1=900, t2=1800, valid-life
```

Starting, restarting and stopping Kea

keactrl

- **`keactrl`** is a shell script that can be used to control the Kea services
- Care must be taken not to conflict with process supervision services such as **`systemd`**, **`runit`** or **`s6`**

Start KEA DHCPv4 module via systemd

- On Linux systems, Kea comes with a set of systemd unit files that control the Kea services

```
# systemctl start kea-dhcp4
```

Check the status of the Kea DHCPv4 service (Linux systemd)

```
# systemctl status kea-dhcp4
● kea-dhcp4.service - Kea DHCPv4 Server
   Loaded: loaded (/usr/lib/systemd/system/kea-dhcp4.service; enabled; vendor pres
   Active: active (running) since Thu 2018-12-06 10:13:26 UTC; 4s ago
     Docs: man:kea-dhcp4(8)
 Main PID: 63 (kea-dhcp4)
    Tasks: 1 (limit: 1144)
   Memory: 1.9M
   CGroup: /machine.slice/libpod-2e3e4a67333cf94630baa9c268ae84f8e77353abf14b074ed
           └─63 /usr/sbin/kea-dhcp4 -c /etc/kea/kea-dhcp4.conf

Dec 06 10:13:26 2e3e4a67333c systemd[1]: Started Kea DHCPv4 Server.
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.674 INFO  [kea-dhc
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.676 INFO  [kea-dhc
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.676 INFO  [kea-dhc
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.677 INFO  [kea-dhc
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.677 INFO  [kea-dhc
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.677 INFO  [kea-dhc
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.679 INFO  [kea-dhc
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.680 INFO  [kea-dhc
```

Testing DHCPv4 with the ISC dhcp client

ISC dhclient as a debugging tool

- Most Linux distributions provide the ISC DHCP client tool **`dhclient`**
- This tool can be used as an simple DHCP debugging tool

dhclient as a debugging tool (1/2)

- Create a new shell script in **`/usr/local/sbin/dhclient-debug.sh`** with the lines below

```
#!/bin/sh
env
```

    - This script will print all variables in it's execution environment

- Make the script executeable

dhclient as a debugging tool (2/2)

- Execute the **`dhclient`** tool with this script

```
dhclient -sf /usr/local/sbin/dhclient-debug.sh
```

- The script will print out all the information received from the DHCP server (via environment variables)
- It will **not** reconfigure the client machines network stack!

Performance benchmarking: perfdhcp

- Kea comes with a DHCP benchmarking tool: **`perfdhcp`**
- This tool can be used to benchmark Kea, but also other DHCP server systems
- For details, see the perfdhcp documentation

Kea control agent

Kea control agent

- The Kea control agent is a process that provides a HTTP(s) REST interface
- The control agent can be used to dynamically reconfigure the Kea services (without manually changing the configuration files)
- The kea control agent communicates with the running Kea services via unix control sockets

Configuration of the Kea control agent

- By default, the Kea control agent listens on the (first) IPv4 loopback address `127.0.0.1` Port `8000`
  - This can be changed in the configuration file **`kea-ctrl-agent.conf`**

Kea shell

- The Kea Shell is a Python command line tool to interact with the Kea Control Agent REST API

Kea shell example

- The Kea shell returns the JSON data from the Kea-Modules REST API
  - Tools such as **`jq`** can be used to pretty print the output

```
# kea-shell --service dhcp4 --host 127.0.0.1 --port 8000 version-get | jq
[
  {
    "arguments": {
      "extended": "1.7.10-git\ngit 9bade6ae294f570976e7614e84a76a34ac4915b1\nlinke
nMemfile backend 2.1"
    },
    "result": 0,
    "text": "1.7.10-git"
  }
]
```

Reading configuration data

- The REST interface has been designed to be used from a Kea configuration application (such as Kea Stork or Kea Shell)
- However, API calls can be send to the Kea control agent from the command line via the **`curl`** tool
- Here we send the **`config-get`** command to the DHCPv4 server

```
[kea-server]# curl -X POST -H "Content-Type: application/json" \
  -d '{ "command": "config-get", "service": [ "dhcp4" ] }' \
  http://127.0.0.1:8000/
```

Pretty printing the JSON output

- The output is unformatted JSON. The tool **`jq`** can be used to pretty-print the output

```
[kea-server]# curl -X POST -H "Content-Type: application/json" \
  -d '{ "command": "config-get", "service": [ "dhcp4" ] }' \
  http://127.0.0.1:8000/ | jq
```

JSON queries with jq

- `jq` can be used to filter specific parts of the configuration. The `jq` filter `".[0].arguments"` can be used to produce a valid KEA configuration file.
    - The example below prints the logging config of the DHCPv4 server:

```
[kea-server]# curl -X POST -H "Content-Type: application/json" \
  -d '{ "command": "config-get", "service": [ "dhcp4" ] }' \
  http://127.0.0.1:8000/ | jq ".[0].arguments.Dhcp4.loggers"
```

Result:

```
[
  {
    "debuglevel": 0,
    "name": "kea-dhcp4",
    "output_options": [
      {
        "output": "/opt/kea/var/log/kea-dhcp4.log"
      }
    ],
    "severity": "INFO"
  }
]
```

List API commands

- The **`list-commands`** command returns the API commands available for a specific KEA module

```
[kea-server]# curl -X POST -H "Content-Type: application/json" \
  -d '{ "command": "list-commands", "service": [ "dhcp4" ] }' \
  http://127.0.0.1:8000/ | jq
```

Dynamic changes to the Kea configuration file (1/6)

- With the REST API, it is possible to
  - Remotely fetch the current running config of a Kea server
  - Change the config
  - And write the config back to the server

Dynamic changes to the Kea configuration file (2/6)

- Dump the current configuration into a file

```
curl -s -X POST -H "Content-Type: application/json" \
     -d '{ "command": "config-get",  "service": [ "dhcp4" ] }' \
     http://127.0.0.1:8000/ | jq ".[0]" > kea-dhcp4.tmp
```

Dynamic changes to the Kea configuration file (3/6)

- Edit the file
  - Add the **`command`** and **`service`** information
  - Make changes to the configuration
  - Remove the **`result`** from the JSON file

```
{
    "command": "config-set",
    "service": [ "dhcp4" ],
    "arguments": {
        "Logging": {
            "loggers": [
                {
                    "severity": "INFO",
                    "output_options": [
[...]
```

Dynamic changes to the Kea configuration file (4/6)

- Send the new configuration to the server

```
[kea-server]# curl -s -X POST -H "Content-Type: application/json" \
   -d @kea-dhcp4.tmp http://127.0.0.1:8000/ | jq
[
  {
    "result": 0,
    "text": "Configuration successful."
  }
]
```

Dynamic changes to the Kea configuration file (5/6)

- All dynamic changes are stored in memory
  - To make the changes persistent, write the in-memory configuration back to a file with the **`config-write`** command (be careful, any comments in the file will be gone and the formatting will be different)

Dynamic changes to the Kea configuration file (6/6)

```
[kea-server]# curl -s -X POST -H "Content-Type: application/json" \
                  -d '{ "command": "config-write", "arguments": { "filename": "/e
                  http://127.0.0.1:8000/ | jq
[
  {
    "arguments": {
      "filename": "/etc/kea/kea-dhcp4-new.json",
      "size": 3248
    },
    "result": 0,
    "text": "Configuration written to /etc/kea/kea-dhcp4-new.json successful"
  }
]
```

Kea DHCPv6

DHCPv6

- From a birds eye view, DHCPv6 works the same way as DHCP4
- In the details, there are many differences
    - Pure layer 3 protocol
    - Clean message design (not based ob BOOTP or DHCPv4)
    - Prefix Delegation
    - Multiple IPv6-Addresses per lease (possible)
    - Rapid Commit

Kea DHCPv6 configuration

- The Kea DHCPv6 server is independent from the Kea DHCPv4 server
- Both can be started together on the same machine, or on separate machines
- The configuration file for the Kea DHCPv6 server is **`kea-dhcp6.conf`**
- The Kea DHCPv6 server can be controlled from the **`keactrl`** script or through systemd (on Linux)
- The DHCPv6 configuration can be managed through the Kea Control Agent and Kea Shell

Software Migrations

About migrations

- Migrations from old to new software products are required from time to time
- Not all software can be upgraded to new functionality
    - Sometimes a clean start is needed to update the underlying implementation
    - Such new implementations often cannot provide 100% compatibility
    - Examples: BIND 8 -> BIND 9, Python2 -> Python3, ISC DHCP -> Kea DHCP

Migration strategies

- Possible migration strategies
  - Upgrade of the existing infrastructure
  - Re-design of the existing infrastructure

Re-Design of an existing infrastructure

- A re-design can be used to remove shortcomings of the old design
  - A migration is a good time to review the current design
  - A new design can make use of modern features that were not available at the time of the original design
  - Maybe it is time to introduce IPv6?
  - Implement new features
    - Automatic configuration management
    - Monitoring
    - Better documentation

Configuration migration

- A DHCP configuration can be quite complex
  - Migrating such a configuration can be challenging
  - There could be unknown pitfalls
  - Testing and Monitoring are important (the Kea DHCP team has created tools to help with this)

About automated configuration migration

- Automated migration of an existing DHCP configuration might be possible
- But be aware:
  - Automated migrations don't utilize the power of the new system
  - Automated migrations create a non-optimal configuration that might be harder to maintain

- Recommendation: if possible, **take time and do a manual migration**

Kea Migration Assistant (KeaMA)

Kea Migration Assistant (KeaMA)

- The Kea Migration Assistant (KeaMA) is a tool provided by ISC to help migrating an existing ISC DHCP configuration to Kea DHCP
  - The output of KeaMA can be used as a starting point for creating a Kea DHCP configuration
  - Don't expect miracles from KeaMA: in most cases, it will not be able to convert the complete ISC DHCP configuration

What KeaMA can do

- KeaMA is good at converting simple configuration structures
  - Reservations
  - Subnet and Shared-Network definitions
  - DHCP Option definitions

What KeaMA cannot do

- There are some ISC DHCP configuration statements that cannot be translated into a Kea DHCP configuration
    - configurations where ISC DHCP and Kea DHCP differ
        - Failover vs. High-Availability
        - Support for hardware types other than Ethernet
        - Shared-Network pools
        - ISC DHCP spawning classes
        - ...

Building Kea Migration Assistant (KeaMA)

- KeaMA is part of the ISC DHCP Server
  - It reuses the ISC DHCP configuration file parser
  - ISC provides experimental RPM/DEB packages on Cloudsmith: https://cloudsmith.io/~isc/repos/keama/packages/
  - It is recommended to install and use KeaMA on a dedicated ***migration*** machine
    - Copy the ISC DHCP configuration to that machine and migrate the configuration there

KeaMA: getting the source

- Download the latest ISC DHCP source code and extract the tar file

```
wget http://ftp.isc.org/isc/dhcp/4.4.2/dhcp-4.4.2.tar.gz
tar xvfz dhcp-4.4.2.tar.gz
```

KeaMA: preparing the source

- The KeaMA source must be adjusted to be compiled on your Linux/Unix system and hardware architecture
    - the **`configure`** script will check the running operating system and will create the build files required to compile KeaMA

```
cd dhcp-4.2.2/
./configure
```

KeaMA: Compile from source

- Once the **`configure`** script returns without error, the KeaMA tool can be build

```
cd keama
make
```

KeaMA: using KeaMA

- It is not required to ***install*** KeaMA, it can be used directly from the build directory
- KeaMA syntax

```
keama [ -4 | -6] [ -D ] [ -N ] [ -r {perform|fatal|pass} ] \
  [ -l hook-library-path ] [ -i input-file ] [ -o output-file ]
```

KeaMA: Example usage

- This is an example of using KeaMA on an ISC DHCPv4 configuration file
    - the option **`-N`** will place host reservations in the appropriate subnet
    - The option **`-r pass`** will pass host names into the Kea DHCP configuration. Often these host names need to be replaced with their IP addresses manually.

```
./keama -4 -N -r pass -i dhcpd.conf -o kea-dhcp4.conf
```

KeaMA: example result

- Comments point to parts of the configuration that most likely need manual adjustment
- Known issues are referenced with their Gitlab issue number
    - Example: Issue #245 below can be found at https://gitlab.isc.org/isc-projects/kea/-/issues/245

```
{
  # dhcpd.conf
  /// This configuration declares some subnets but has no interfaces-config
  /// Reference Kea #245
  "Dhcp4": {
//  "statement": {
//    "config": {
//      "value": "allow",
//      "name": "allow-booting",
//      "code": 9
//    }
//  },
    "dhcp-ddns": {
      "qualifying-suffix": "home.example.com",
      "enable-updates": true
    },
[...]
```

Differences between Kea and ISC DHCP

Shared Networks

- ISC DHCP permits pools at shared-network level
- In Kea DHCP, a pool must belong to a subnet
- In Kea DHCP, selecting a lease from a shared-network has a performance penalty compared to selecting a lease from a plain subnet

Client classification

- Kea DHCP does not have a concept similar to **`permit`** or **`deny`** in ISC DHCP
  - In ISC DHCP, this is used to permit or deny certain client classes in subnets
  - This can be done in Kea DHCP as well, but the logic is different
- Kea DHCP does not support ***spawning*** classes (which are used for dynamic lease limit configurations)
  - There is (currently) no per client / per class lease limit in Kea DHCP

DHCP Options

- ISC DHCP can resolve DNS names to IP addresses for options that require an IP address. Kea DHCP does not resolve DNS names
    - The Kea Migration Assistant can resolve the DNS names into IP addresses while converting an ISC DHCP configuration
- Option inheritance scoping is different between ISC DHCP and Kea DHCP

High-Availability

- ISC DHCP supports the IETF DHCPv4 Failover Protocol draft
  https://datatracker.ietf.org/doc/html/draft-ietf-dhc-failover
- Kea DHCP supports its own DHCP high availability implementation for DHCPv6 and DHCPv4
- Both create a highly available DHCP service, but the implementation and configuration details are different
- Document comparing failover in ISC DHCP to HA in Kea DHCP
  https://kb.isc.org/docs/aa-01617

Kea High Availability vs ISC DHCP Failover (1)

- Number of servers in an HA cluster
  - ISC DHCP: 2
  - Kea DHCP: 2 active + unlimited backup servers
- Failover relationships
  - ISC DHCP: one per subnet
  - Kea DHCP: one per instance
- Load balancing
  - ISC DHCP: Flexible split (RFC3074)
  - Kea DHCP: fixed 50/50 split (RFC3074)

Kea High Availability vs ISC DHCP Failover (2)

- Lazy lease updates (MCLT)
  - ISC DHCP: yes (server responds to the client immediately)
  - Kea DHCP: no (server waits for lease update completion before responding to client)
- Send lease updates to external entity
  - ISC DHCP: no
  - Kea DHCP: yes (via backup server or custom hook library)
- Rebalancing pools
  - ISC DHCP: yes
  - Kea DHCP: no

Kea High Availability vs ISC DHCP Failover (3)

- Database replication for sharing lease info
  - ISC DHCP: no
  - Kea DHCP: yes (optional)
- API
  - ISC DHCP: omapi
  - Kea DHCP: RESTful API

Host reservations

- In ISC DHCP all host declarations are global
- Kea DHCP supports global and per-subnet/shared-network reservations
- ISC DHCP can have reservations that are not viable on the subnet where the clients are attached
    - Kea does not start if configured with an address or prefix that is not viable on its subnet; it displays an error

Scripts and Tools

- Most 3rd party tools and scripts that work with ISC DHCP do not work with Kea DHCP
  - parse lease file
  - SNMP monitoring agents
  - Configuration orchestration
- Many use cases are covered by the Kea API

Manual Migration

Client Classes

- The Kea Migration Assistant will translate the client classification rules from ISC DHCP to Kea DHCP
    - the Kea DHCP configuration will have the original client classification as a comment

```
[...]
    "client-classes": [
      {
        "name": "virtualbox",
        /// from: match if (substring(hardware, 1, 3)) = 0x080027
        "test": "substring(pkt4.mac,0,3) == 0x080027"
      },
[...]
```

- Understanding Client Classification
  https://kb.isc.org/docs/understanding-client-classification

Client Classes

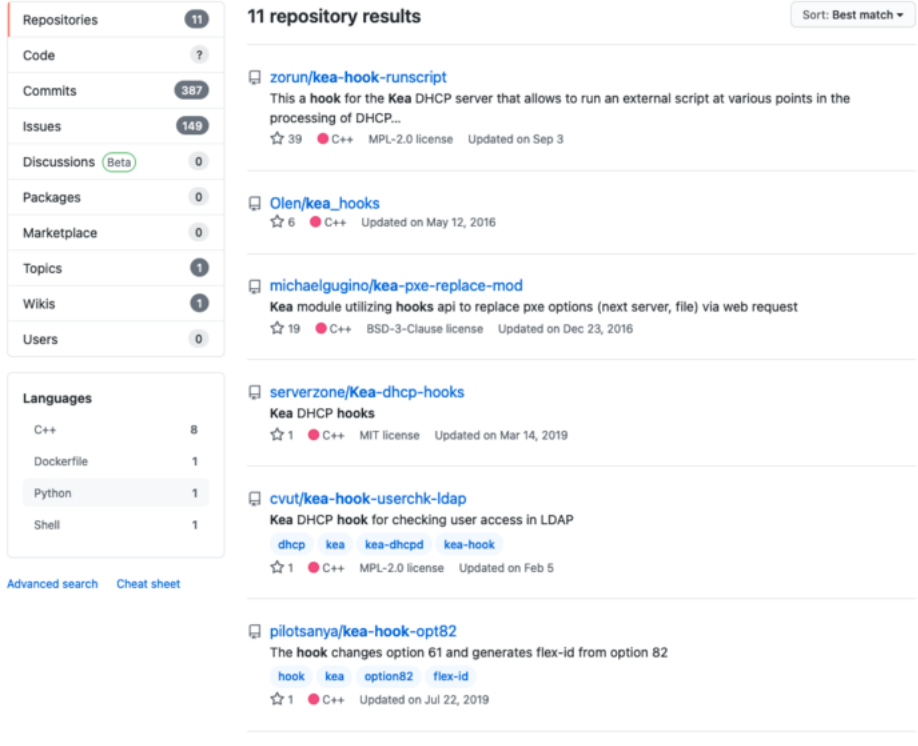
- Review and, if possible, simplify the client classification tests
  - Kea DHCP might have expressions that better define the class
  - Using Expressions in Classification: https://kea.readthedocs.io/en/kea-1.8.1/arm/classify.html#using-expressions-in-classification

```
[...]
      {
        "name": "gen#virtualbox#!KNOWN#_AND_#!microsoft-client#",
        "test": "(member('virtualbox') or not member('KNOWN')) and not member('mic
      },
      {
        "name": "gen#!KNOWN#_AND_#!virtualbox#!microsoft-client#",
        "test": "(not member('KNOWN')) and not member('virtualbox') and not member
      }
[...]
```

Expressions

- ISC DHCP allows complex expressions in the configuration file
  - The ISC DHCP configuration file is almost a programming language
  - Most effects of ISC DHCP expressions can be created with the Kea DHCP functions, or existing hook libraries (`flex_id` or `flex_option` hooks)
  - In other cases, a custom hook can be used to implement almost any logic required
    - 3rd party hook examples: https://github.com/search?q=kea+hook

Custom Kea hooks



Repositories 11
Code ?
Commits 387
Issues 149
Discussions Beta 0
Packages 0
Marketplace 0
Topics 1
Wikis 1
Users 0

Languages
C++ 8
Dockerfile 1
Python 1
Shell 1

Advanced search Cheat sheet

11 repository results

Sort: Best match

zorun/kea-hook-runscript
This a hook for the Kea DHCP server that allows to run an external script at various points in the processing of DHCP...
☆ 39 C++ MPL-2.0 license Updated on Sep 3

Olen/kea_hooks
☆ 6 C++ Updated on May 12, 2016

michaelgugino/kea-pxe-replace-mod
Kea module utilizing hooks api to replace pxe options (next server, file) via web request
☆ 19 C++ BSD-3-Clause license Updated on Dec 23, 2016

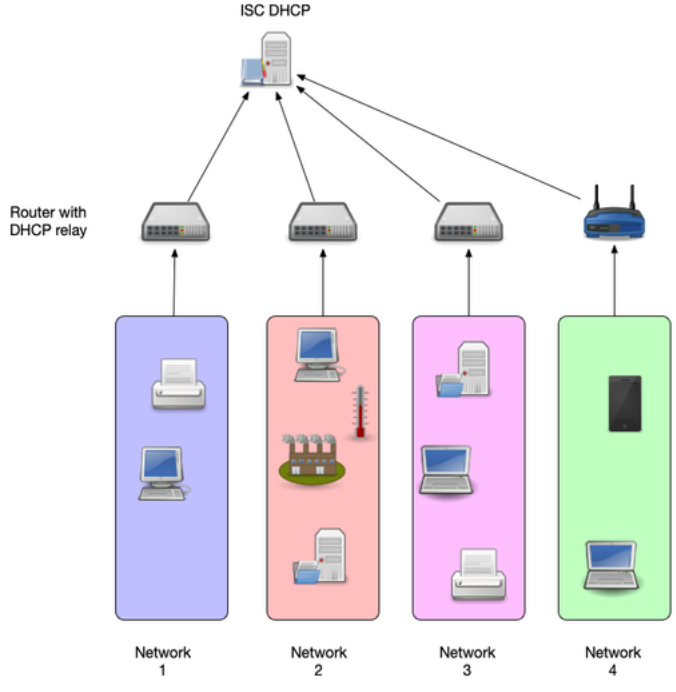
serverzone/Kea-dhcp-hooks
Kea DHCP hooks
☆ 1 C++ MIT license Updated on Mar 14, 2019

cvut/kea-hook-userchk-ldap
Kea DHCP hook for checking user access in LDAP
dhcp kea kea-dhcpd kea-hook
☆ 1 C++ MPL-2.0 license Updated on Feb 5

pilotsanya/kea-hook-opt82
The hook changes option 61 and generates flex-id from option 82
hook kea option82 flex-id
☆ 1 C++ Updated on Jul 22, 2019

ISC DHCP to Kea DHCP Migration Plan

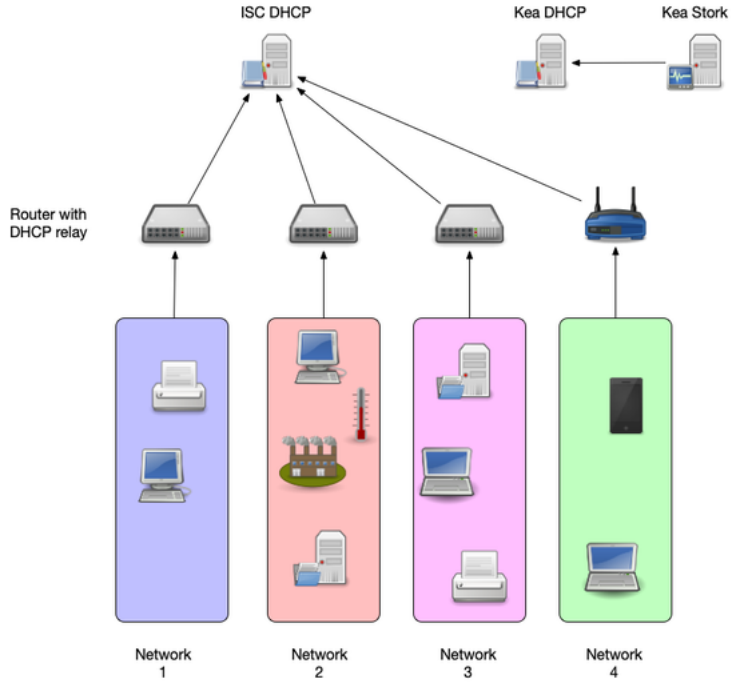
Proposed migration steps



Proposed migration steps

- Get familiar with Kea DHCP
  - Run a production Kea DHCP for some time (some weeks) in a small and low risk network
  - Test features that will be used in the larger production networks

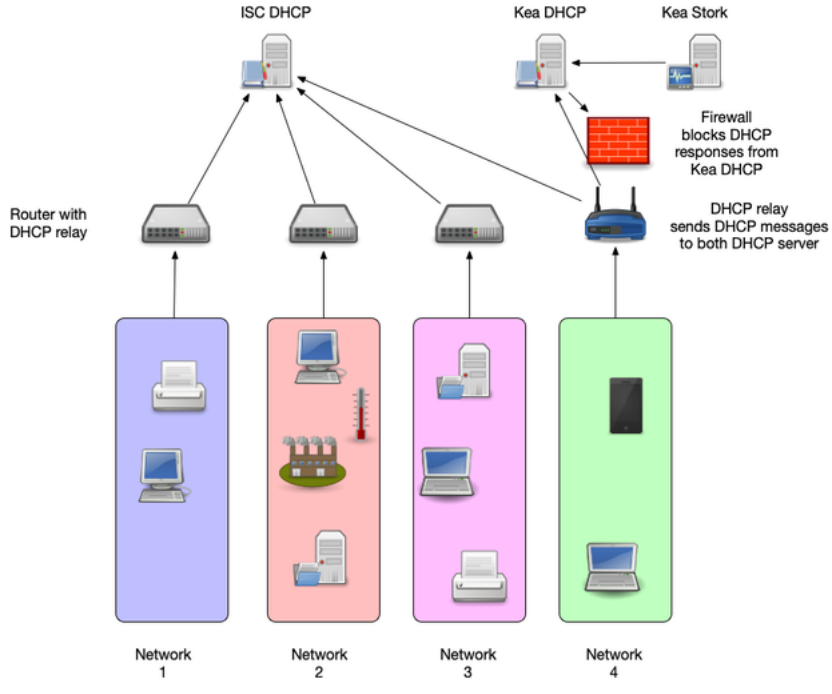
Proposed migration steps



Proposed migration steps

- Install Kea DHCP alongside the existing ISC DHCP
- Write and test the Kea DHCP configuration
    - Configure the DHCP relays to forward DHCP messages to ISC DHCP (production) and Kea DHCP (test)
        - Block the responses from Kea DHCP in the host firewall of the Kea DHCP OS (for example Linux nftables)
        - Inspect the responses from Kea DHCP and compare with the responses from ISC DHCP
- Implement Logging and Monitoring

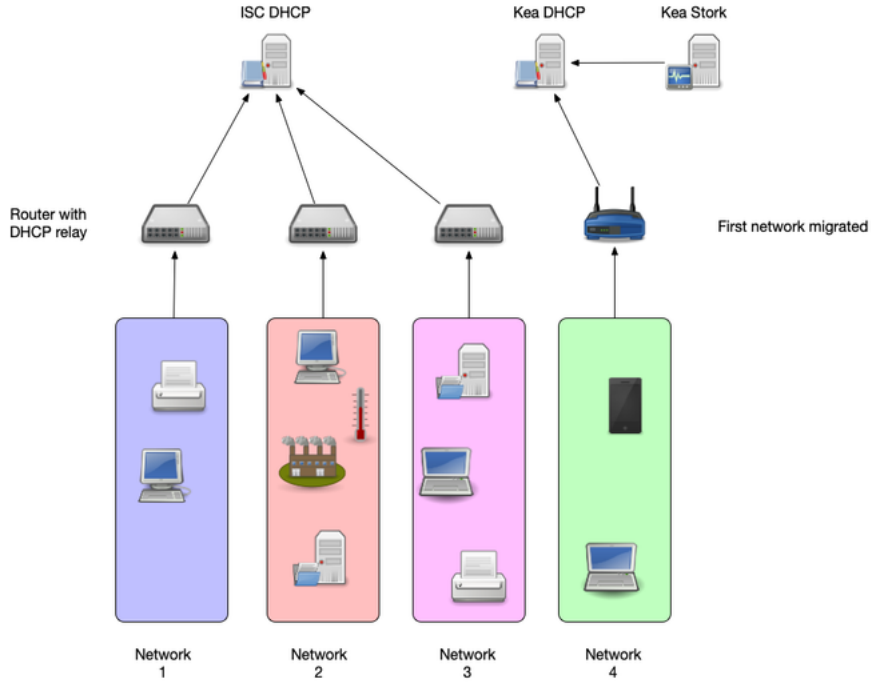
Proposed migration steps



Proposed migration steps

- Define the time line for the migration
- Lower the lease/refresh times on the ISC DHCP before the migration
    - One hour DHCP refresh is safe for most devices
    - Modern operating systems (Windows, Linux, macOS etc) can work with low DHCP refresh times, such as 5 minutes
    - Embedded or older DHCP clients (MS-DOS, Windows 9x, QNX etc) that can be found in industrial control units need safe refresh values

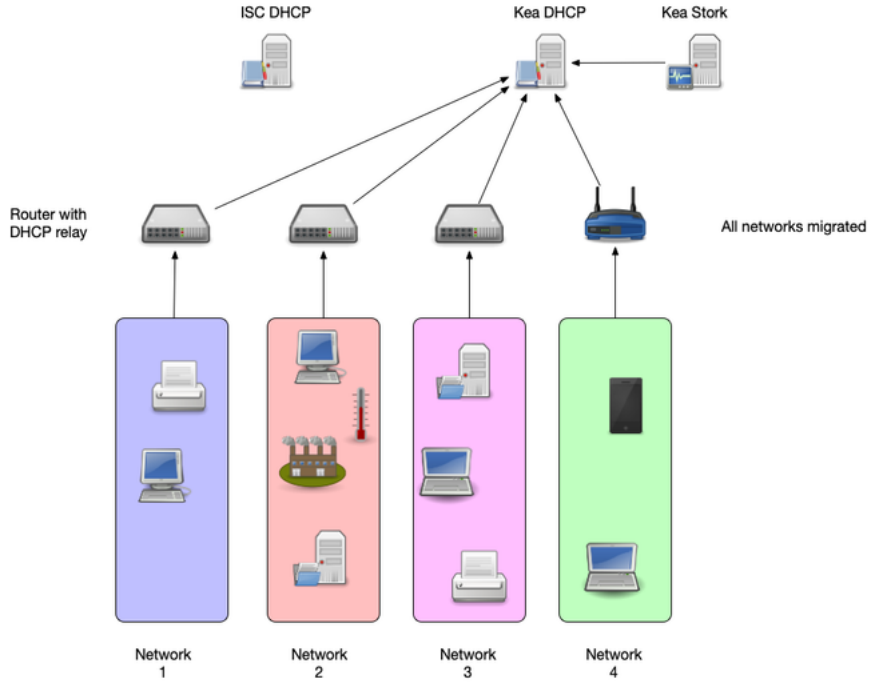
Proposed migration steps



Proposed migration steps

- Start with a ***friendly crowd***, low risk network (IT department WLAN)
- Switch networks one-by-one via DHCP relay configuration
    - On regression, switch back to ISC DHCP for this one network and investigate

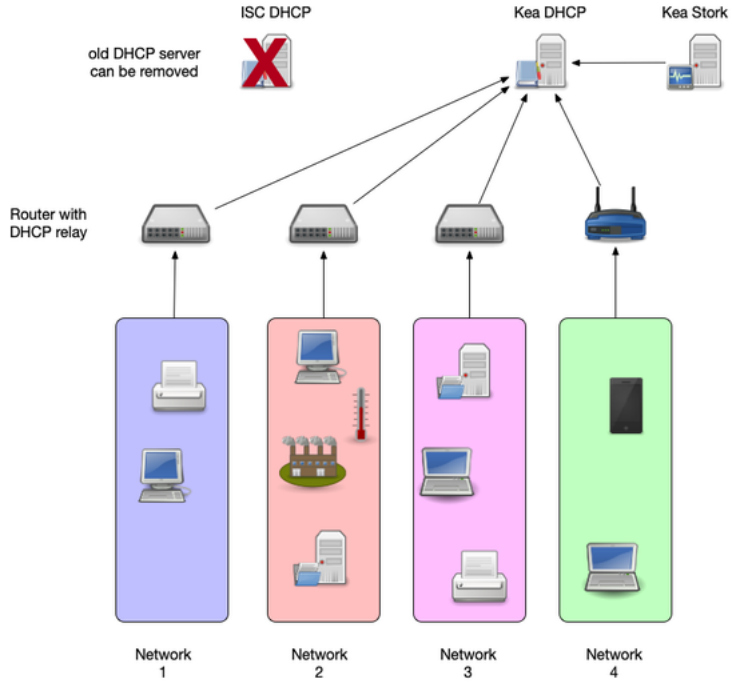
Proposed migration steps



Proposed migration steps

- Monitor leases on the ISC DHCP as well
  - At some point in time, there should be no DHCP requests coming to the old server(s)
  - Time to shut the old systems down

Proposed migration steps



Resources

- Alan Clegg – NANOG 76 - DHCP Migration to Kea
  - https://pc.nanog.org/static/published/meetings/NANOG76/19
- Kea High Availability vs ISC DHCP Failover
  - https://kb.isc.org/docs/aa-01617
- Kea HA Design Document
  - https://gitlab.isc.org/isc-projects/kea/-/wikis/designs/High-Availability-Design

Resources

- Available 3rd party hooks for Kea DHCP
  - https://gitlab.isc.org/isc-projects/kea/-/wikis/Hooks-available
- Using Host Reservations in Kea
  - https://kb.isc.org/docs/what-are-host-reservations-how-to-use-them

Questions?

References: RFCs, Books, recommended Webpages

Internet Standards

- DHCPv4
  - RFC 2131 - DHCPv4 Dynamic Host Configuration Protocol
  - RFC 2132 - DHCP Options and BOOTP Vendor Extensions
  - RFC 3396 - Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)
  - RFC 4361 - Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)
  - RFC 6842 - Client Identifier Option in DHCP Server Replies
- DHCPv6
  - RFC 8415 - Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

Books

- The DHCP Handbook - Understanding, Deploying, and Managing Automated Configuration Services (Ralph Droms, Ted Lemon) 1999
- IP Address Management - Principles and Practice (Timothy Rooney) 2011
- The TCP/IP Guide - A Comprehensive, Illustrated Internet Protocols Reference (Charles M. Kozierok) 2005
- Windows Server 2019 Inside Out (Orin Thomas)

Websites

- ISC Kea Documentation - https://kea.readthedocs.io/en/latest/
- ISC Knowledgebase - https://kb.isc.org/
- The TCP Guide - http://www.tcpipguide.com/
- Microsoft - Dynamic Host Configuration Protocol (DHCP) https://docs.microsoft.com/en-us/windows-server/networking/technologies/dhcp/dhcp-top