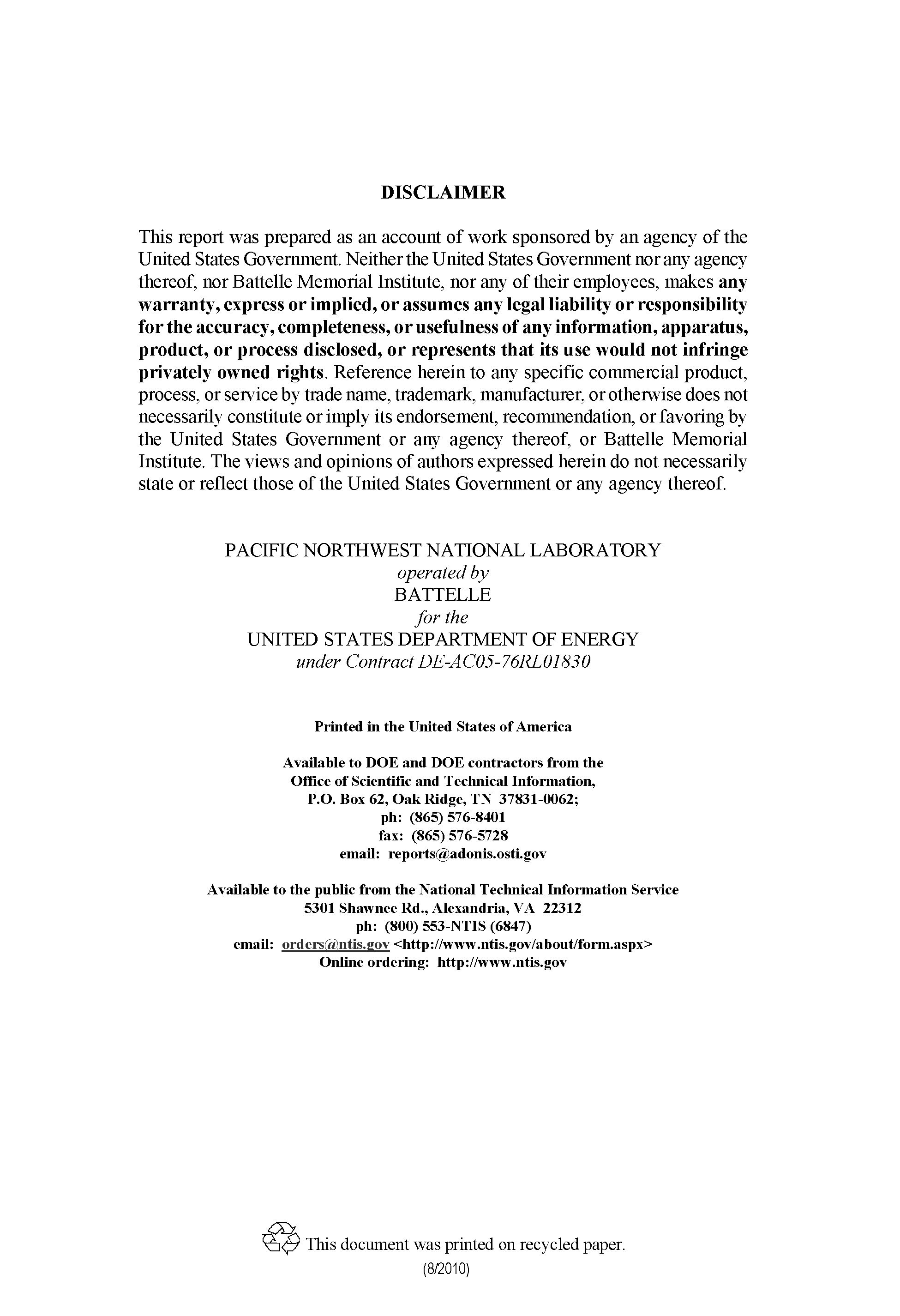
FLOWER Installation Guide

FLOWER Version 06 (flr06)

September 2018

DS Curtis



FLOWER Installation Guide

DS Curtis

September 2018

Prepared for

the U.S. Department of Energy

under Contract DE‑AC06‑76RL01830

with Battelle Memorial Institute

Pacific Northwest National Laboratory

Richland, Washington 99352

Acknowledgments

Copyright © (2011-2021) Battelle Memorial Institute. All Rights Reserved. PNNL-SA-120800

Battelle IPID: 17042-E

Prime Contract No.: DE-AC06-76RL01830 with Battelle Memorial Institute

Acronyms and Abbreviations

FLOWER FLOW analyzer

GCC GNU Compiler Collection

pcap packet capture files

PNNL Pacific Northwest National Laboratory

Contents

[Acknowledgments iii](#_Toc523929079)

[Acronyms and Abbreviations v](#_Toc523929080)

[1.0 Introduction 1.1](#_Toc523929081)

[2.0 System Requirements 2.1](#_Toc523929082)

[2.1 Hardware Requirements 2.1](#_Toc523929083)

[2.2 Operating System Requirements 2.1](#_Toc523929084)

[2.3 External Library Requirements 2.2](#_Toc523929085)

[2.4 PCAP Development Libraries on RedHat or CentOS 7.x 2.2](#_Toc523929086)

[2.5 PCAP Development Libraries on Ubuntu 16.04 2.2](#_Toc523929087)

[3.0 Install FLOWER 3.1](#_Toc523929088)

[3.1 Install FLOWER 3.1](#_Toc523929089)

[3.1.1 Install FLOWER on RedHat or CentOS 7.x 3.1](#_Toc523929090)

[3.1.2 Install FLOWER on Ubuntu 16.04 3.1](#_Toc523929091)

[3.2 Install FLOWER systemd Service 3.1](#_Toc523929092)

[3.3 Update root user environment for FLOWER 3.2](#_Toc523929093)

[3.4 Running FLOWER using systemd commands 3.2](#_Toc523929094)

[4.0 Configure FLOWER After Installation 4.1](#_Toc523929095)

[4.1 Configuring FLOWER 4.1](#_Toc523929096)

[5.0 Disable Operating System and Hardware Optimizations 5.1](#_Toc523929097)

[5.1 Disable Operating System and Hardware Optimizations 5.1](#_Toc523929098)

[5.2 Disable Network Optimizations on RedHat of CentOS 7.x 5.1](#_Toc523929099)

[5.3 Disable Network Optimizations on Ubuntu 16.04 5.1](#_Toc523929100)

[5.4 Ethtool is not working or does not work after reboots 5.1](#_Toc523929101)

[6.0 Troubleshooting 6.1](#_Toc523929102)

[6.1 FLOWER is not producing any files 6.1](#_Toc523929103)

[6.2 FLOWER is producing warning messages 6.1](#_Toc523929104)

[7.0 Reporting Bugs 7.1](#_Toc523929105)

# Introduction

The FLOWER (FLOW analyzER) application is designed to start at boot time from as a Linux systemd service running on RedHat 7.x, CentOS 7.x, or Ubuntu 16.04 to summarize network flows. FLOWER can also be used to read packets from a network interface or from packet capture (pcap) files that can be processed with the libpcap system library.

# System Requirements

## Hardware Requirements

The minimum and recommended hardware requirements outlined here are based on a typical small to medium sized installation. The exact requirements vary between deployments based on network traffic load.

|  |  |  |
| --- | --- | --- |
| **Resource** | **Minimum** | **Recommended** |
| CPU | Intel i3 | Intel i7 |
| Memory | 4 GB of system RAM | 16 GB of system RAM |
| Hard Disk | 50 GB of locally accessible, writable, disk space | 250 GB of locally accessible, writable, disk space |
| Network Interface | 1 Network Interface Card (NIC) with bandwidth of at least 1 Gbps | 1 Network Interface Card (NIC) with bandwidth of 10-40 Gbps |

## Operating System Requirements

FLOWER runs on Linux operating systems and versions using [systemd](https://en.wikipedia.org/wiki/Systemd).

|  |  |
| --- | --- |
| **Linux** | **Version** |
| CentOS | 7.x |
| RedHat | 7.x |
| Ubuntu | 16.04 |

**NOTE**: In general, FLOWER will not run on a Linux system where selinux is enabled.

## External Library Requirements

FLOWER is a dynamically linked executable that requires external libraries to run.

|  |  |
| --- | --- |
| Library | Version |
| PCAP development library | 1.5 (or higher) for RedHat and CentOS |
| 0.8 (or higher) for Ubuntu |

## PCAP Development Libraries on RedHat or CentOS 7.x

To get the PCAP development libraries on your RedHat/CentOS systems you can run the following:

* Install the libpcap-devel package from your yum repository

# yum install libpcap-devel libpcap

## PCAP Development Libraries on Ubuntu 16.04

To get the PCAP development libraries on your Ubuntu systems you can run the following:

* Install libpcap-dev from your Ubuntu repository:

# apt-get install libpcap-dev

**NOTE**: Because Ubuntu does not have a version 1.0 of the shared object file for libpcap.so, we need to create one with the following command:

# ln –s /usr/lib/x86\_64-linux-gnu/libpcap.so \

/usr/lib/x86\_64-linux-gnu/libpcap.so.1

# Install FLOWER

## Install FLOWER

By default, FLOWER will be installed in the directory, /opt/flower, where:

* binaries located in /opt/flower/bin
* the configuration file will be /opt/flower/conf/flower.conf
* data files will be generated in /opt/flower/data

### Install FLOWER on RedHat or CentOS 7.x

To get the PCAP development libraries on your RedHat/CentOS systems you can run the following:

# rpm –i flower-6.0-0.x86-64.rpm –y

To install FLOWER in a directory other than /opt/flower you can use the relocate option with the rpm command to install FLOWER in another directory. If you would like to install FLOWER in /usr/local/flower you can run the following:

# rpm –r /usr/local/flower –i flower-6.0-0.x86-64.rpm –y

### Install FLOWER on Ubuntu 16.04

To get the PCAP development libraries on your RedHat/CentOS systems you can run the following:

# dpkg -i flower\_6.0-0\_amd64.deb -y

## Install FLOWER systemd Service

For FLOWER to start collecting data from a network interface, you need to configure the software to tell the flower binary which network interface to use and to start up the systemd service. If you installed FLOWER in the default location, /opt/flower, then you would run the following command as the root user or using sudo:

# /opt/flower/bin/create-flower-config.sh

**NOTE**: If you installed FLOWER in another directory, you will have to adjust the command above.

The create-flower-config.sh script will prompt you for the network interface to collect data from (e.g. eth0) and the ***sitename*** (e.g. the name you want for this installation of FLOWER software to write in each output file).

The script will create the /opt/flower/conf/flower.conf file, enable the flower as a daemon, enable the flower service, and start the flower software.

**NOTE**: If you make changes to the flower.conf file, you will need to restart the flower daemon and flower software.

## Update root user environment for FLOWER

You may want to update the PATH environment variable for the root user to make it easier to run flower manually. Add the directory where you installed FLOWER to the root user’s shell file. For example, if the root user is running the bash shell environment, and you installed FLOWER in the default location, you would add the following line to the ~root/.bashrc file:

export PATH=$PATH:/opt/flower/bin

**NOTE**: After adding the line, you should source the ~root/.bashrc file to update your environment.

## Running FLOWER using systemd commands

In general it is recommend to use systemctl and journalctl instead of using the service command.

Start FLOWER using the following command as the user root:

# systemctl start flower.service

Stop FLOWER using the following command as the user root:

# systemctl stop flower.service

Check the status of FLOWER using the following command as the user root:

# systemctl status flower.service -l

You can also use the journalctl command rather than looking for errors in /var/log/messages. Check for FLOWER errors using the following command as the user root:

# journalctl -u flower.service

# Configure FLOWER After Installation

## Configuring FLOWER

Please refer to the FLOWER\_Ops\_Guide\_FLR06 document on how to configure and run the flower binary.

# Disable Operating System and Hardware Optimizations

## Disable Operating System and Hardware Optimizations

Newer versions of Linux have the ability to take advantage of Network Interface Card (NIC) optimizations such as TCP Segmentation Offload (TSO), Large Send Offload ([LSO](https://en.wikipedia.org/wiki/Large_send_offload)), and Generic Segmentation Offload (GSO). There is a good explanation of these NIC optimizations and how they work at [LoveMyTool.com](http://www.lovemytool.com/blog/2016/08/analyzing-tcp-segmentation-offload-tso-with-wireshark-by-paul-offord.html).

If you know you will be running FLOWER on interfaces on your system that have TSO, LSO, and GSO capabilities you will see errors in /var/log/messages or from the output of the command, systemctl status flower.service -l, so you will want to disable these optimizations. The messages will look something like the following:

Sep 05 16:14:35 host flower[16123]: FATAL ERROR\_MSG:

Sep 05 16:14:35 host flower[16123]: File: ./include/PacketParser.hpp(line:828)

Sep 05 16:14:35 host flower[16123]: Class: PacketParser

Sep 05 16:14:35 host flower[16123]: Function: setTotalBytes

Sep 05 16:14:35 host flower[16123]: Message: Error Code=RangeError

Sep 05 16:14:35 host flower[16123]: Adjust TotalBytes

Sep 05 16:14:35 host flower[16123]: Expected total bytes (7354) to be less than or equal to bytes on the wire (1982)

## Disable Network Optimizations on RedHat of CentOS 7.x

To disable the networking optimizations, you can add the following line to the file, /etc/sysconfig/network-scripts/ifcfg-<interface\_name>:

ETHTOOL\_OPTS="-K ${DEVICE} rx off tx off sg off tso off ufo off gso off gro off lro off rxvlan off txvlan off ntuple off rxhash off"

## Disable Network Optimizations on Ubuntu 16.04

To disable the networking optimizations, you can add the following line to the file, /etc/network/interfaces:

pre-up /sbin/ethtool –K <interface\_name> rx off tx off sg off tso off ufo off gso off gro off lro off rxvlan off txvlan off ntuple off rxhash off

## Ethtool is not working or does not work after reboots

It has been observed that ethtool does not work consistently, especially after reboots. In addition, some of the options such as udp-fragmentation-offload (ufo) may not work on some cards. If you run ethtool and get an error, remove that argument and try again.

# Troubleshooting

## FLOWER is not producing any files

If the flower process is not producing files but is still running, there are a few things you can do to find out what might be wrong. If the --use-ring=1 command line or configuration file option is being used then files should always be produced even if there are no packets on the wire.

Find out how many system resources flower is being used with the following commands:

ps --no-heading -C flower

lsof -p `ps --no-heading -C flower | awk '{print $1}'`

free –mt

grep flower /var/log/messages\*

Find the raw packets that might have caused the problem by creating a pcap file in the output directory using the command:

kill -SIGSEGV $(ps --no-heading -C flower | awk '{print $1}')

You can use Wireshark to analyze any \*.pcap files left in the FLOWER data directory that was specified in the flower.conf file or specified as a command line option.

## FLOWER is producing warning messages

Refer to section 5.0 for disabling optimizations if FLOWER is producing error messages in /var/log/messages that look like the following:

Oct 23 03:36:00 host flower: FATAL ERROR:

Oct 23 03:36:00 host flower: File: ./include/PacketParser.hpp(line:832)

Oct 23 03:36:00 host flower: Class: PacketParser

Oct 23 03:36:00 host flower: Function: setTotalBytes

Oct 23 03:36:00 host flower: Message: Error Code=RangeError

Oct 23 03:36:00 host flower: Adjust TotalBytes

Oct 23 03:36:00 host flower: Expected total bytes (2863) to be less than or equal to bytes on the wire (1982)

# Reporting Bugs

Please submit bug reports to flower-support@pnnl.gov and include as much information as possible to describe the problem.

Please include the output from the commands in the “Troubleshooting” section and the output from flower –v up to the copyright message. For example:

flower version: 6.1.1

Compiled on: Aug 23 2018, 14:38:24 (2018-08-23-14:37)

Compiled with: 8.1.0

Optimize Level: -O2

Debug: Release

Boost library version: 106700

pcap library version: libpcap version 1.5.3

Data Guide version: flr06