

# Appendix D

## Lines of Code Comparison

In this Appendix, we present the details of the Lines-of-Code analysis, the results of which were discussed in Chapter 3.

Table D.1 presents the output of CLOC for two open-source projects that have implemented OSPF-TE and RSVP-TE. The former is from the Quagga project [63]. Note that we have only accounted for the ospfd implementation, not any of the other routing protocols that are part of Quagga. Also the Quagga suite does not include RSVP-TE. We use the IST-Tequila project implementation for RSVP-TE [64, 65].

### quagga-0.99.18/ospfd

Language	files	blank	comment	code
C	26	6650	4276	32046
C/C++ Header	24	559	923	2198
make	1	12	1	25
SUM:	51	7221	5200	34269

### rsvpd.0.70-rc2/rsvpd

Language	files	blank	comment	code
C	78	7875	9866	45740
C/C++ Header	55	1297	2043	4243

make	7	125	71	302
Bourne	1	19	20	151
-----				
SUM:	141	9316	12000	50436
-----				

**Table D.1: IP/MPLS Control Plane Implementation**

The UNI is typically instantiated using a signaling protocol like RSVP-TE with suitable extensions [22]. Table D.2 shows an implementation of the OIF UNI from the IST-MUPBED project [66, 67].

**rsvp-agent**

Language	files	blank	comment	code
Bourne	8	3045	4334	23500
C++	21	1080	858	8006
m4	1	770	20	6206
Java	30	626	748	1499
C/C++				
Header	20	257	501	1177
Python	2	153	380	683
make	3	5	6	13
Bourne	1	1	0	2
-----				
SUM:	86	5937	6847	41086
-----				

**Table D.2: UNI Implementation**

Finally the GMPLS protocols for the transport network control plane include further extensions of the MPLS control plane protocols – OSPF-TE and RSVP-TE. Such extensions have been standardized [20] and Table D.3 shows implementations of the GMPLS protocols by the DRAGON project [68, 69]. Note that in accounting for the latter, we are ignoring the code required for the proprietary interfaces in the transport network vendor islands.

**dragon-sw/zebra/ospfd**

Language	files	blank	comment	code
C	30	7113	4102	35514
C/C++ Header	26	618	1014	2522
make	1	9	1	36
SUM:	57	7740	5117	38072

**dragon-sw/kom-rsvp**

Language	files	blank	comment	code
C++	85	4384	3745	31004
C/C++ Header	106	2395	2733	11851
Bourne	10	1299	1313	11146
C	2	128	128	821
Tcl/Tk	5	123	146	644
yacc	2	119	48	642
m4	1	20	57	476
make	12	92	3	344
Java	9	37	66	227
lex	2	37	46	109
ASP.Net	1	5	0	41
Perl	1	1	0	18
SUM:	236	8640	8285	57323

**Table D.3: GMPLS Control Plane Implementation**

In Table D.4, to implement the network function using our control solution requires only 4726 lines-of-code – *two orders of magnitude lesser* than the industry solution.

**nox/src/nox/coreapps/aggregation**

Language	files	blank	comment	code
Python	4	808	549	2634
C++	3	56	10	253
C/C++ Header	4	65	88	137
XML	1	5	6	54
make	1	19	0	46
SUM:	13	953	653	3124

**nox/src/nox/coreapps/circsw**

Language	files	blank	comment	code
C++	1	198	208	1243
C/C++ Header	2	138	88	459
make	1	9	0	19
XML	1	8	0	14
SUM:	5	353	296	1735

**Table D.4: Control-Function Implementation with Unified Control Architecture**

The base code of NOX is roughly 67,700 lines of code (Table D.5 and [51]) to which we have added 1100 lines to support circuit-switching - OpenFlow extensions for circuits, a discovery module and a circuit-API. The Quagga software suite is based on the zebra-daemon and related libraries. A lines-of-code estimate for this base code is shown in Table D.6, where we have *not* included the protocol implementations for all the protocols found in the software suite (OSPF, OSPFv6, ISIS, RIP, RIPnG, and BGP as well as vtysh, ospfclient, tools directories are not included).

Language	files	blank	comment	code
C++	161	6129	5647	37161
C/C++ Header	217	4806	11106	16190
Python	125	3623	5140	13516
make	60	522	63	2004
m4	26	117	14	1542
Bourne	40	180	231	1262
C	3	138	142	901
CSS	1	25	10	462
Bourne Again Shell	3	34	77	360
SQL	1	37	104	203
Perl	2	18	0	178
HTML	4	24	1	145
XSD	2	6	0	50
SUM:	645	15659	22535	73974

**Table D.5: NOX Base Lines of Code**

Language	files	blank	comment	code
C	95	9137	7098	48255
Bourne	10	4151	6911	25102
m4	7	1044	313	9275
C/C++ Header	52	1159	2164	3573
Bourne	8	67	85	351
make	10	75	55	241
awk	2	37	89	129
SUM:	184	15670	16715	86926

**Table D.6: Quagga Base Lines of Code**