cflowd 2.0

Cisco flow-export collection

Daniel McRobb
dwm@caida.org

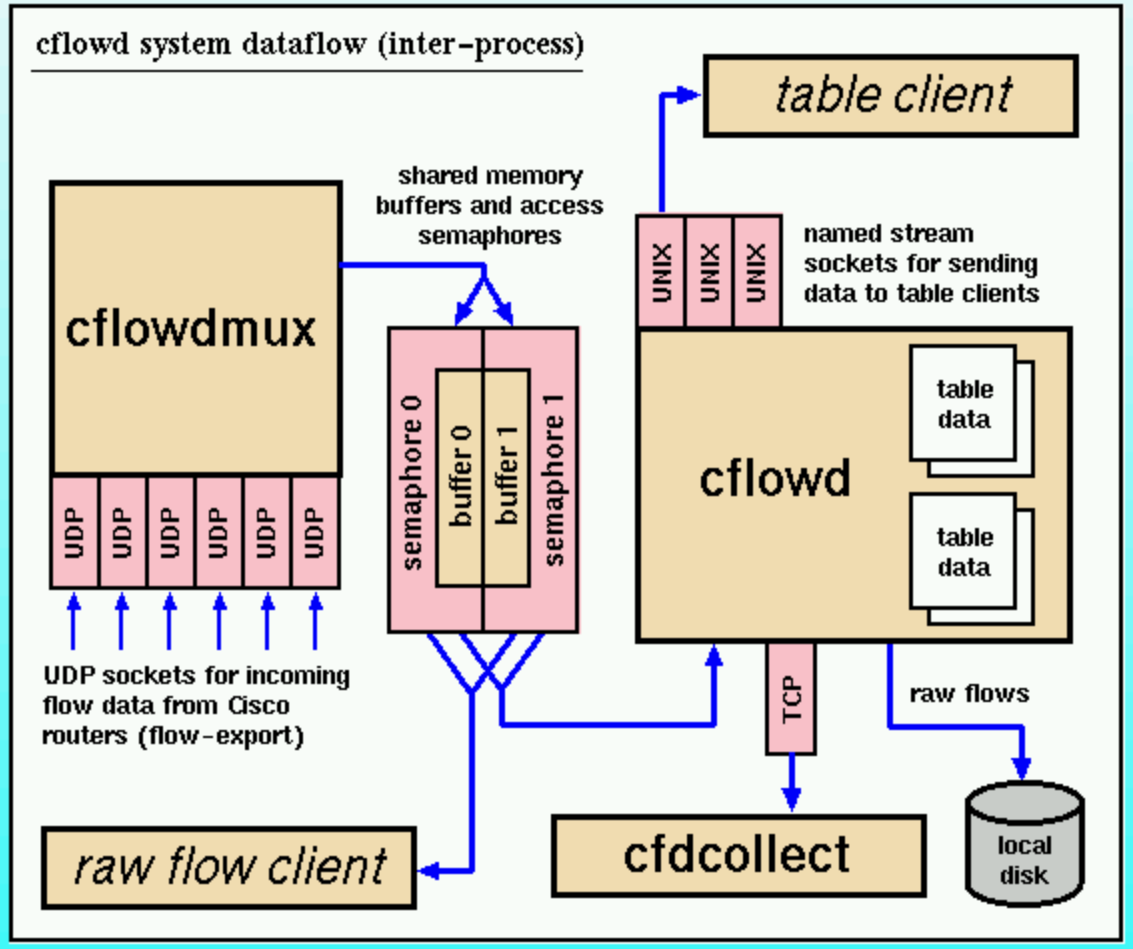
Background

- host software for collecting Cisco version 5 (and version 1) flow-export data
- aggregates data into tables for continuous collection of summary data (in time-series)
- stores raw flow data in rotating log files
- provides client/server collection of tabular data in time series

Changes from 1.3b2 to 2.0

- all tables are per input interface (not per router)
- support for version 1 flow-export
- central collector included
- uses arts++ package for data storage and aggregation
- real-time flow-matching API (example 'flowwatch' program included)
- interface matrix
- IP nexthop table

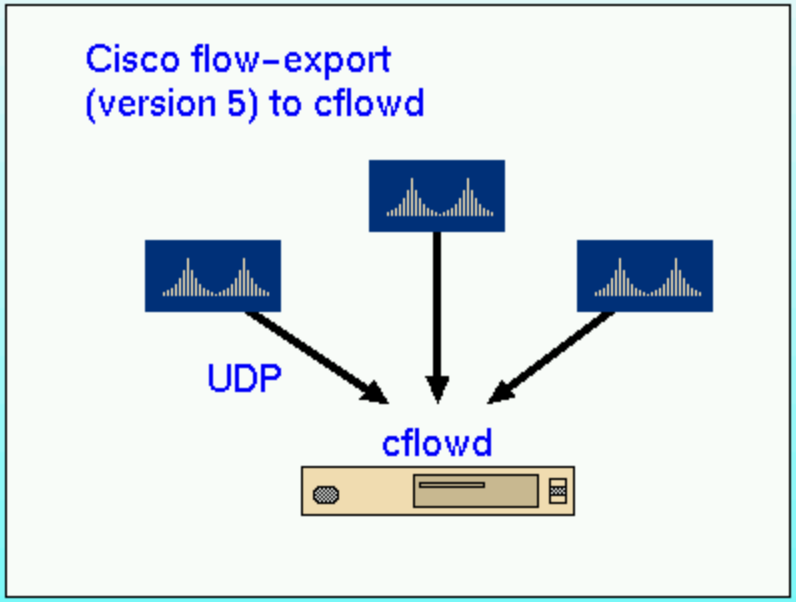
cflowd architecture



benefits of new architecture

- increased performance (shared memory packet queue vs. UNIX domain sockets for cflowdmux to cflowd IPC)
- hooks for real-time flow matching
- source code easier to maintain due to heavy use of STL and more modularization

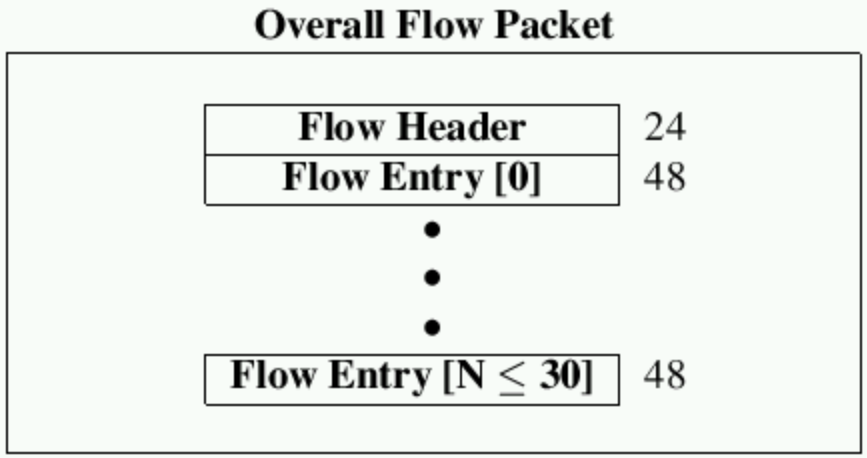
cflowd



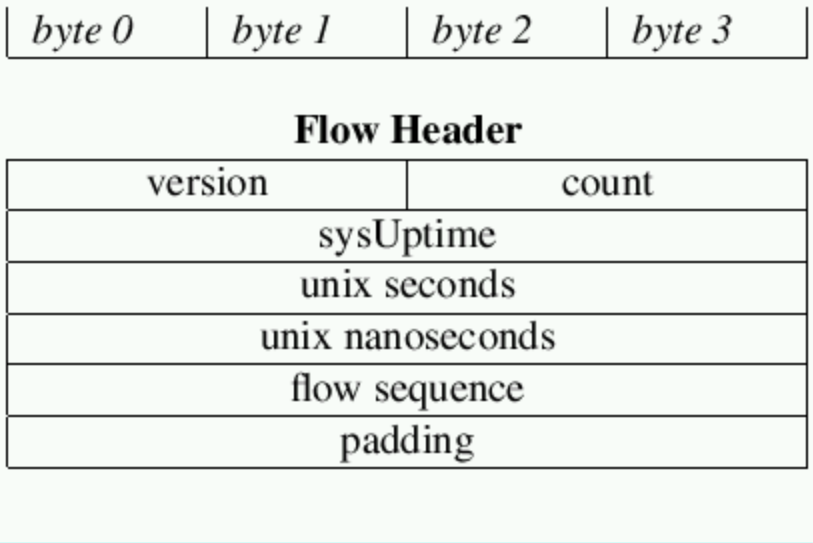
flow export version 5

- sends UDP packets to a specified host address and port number
- each UDP packet contains a number of flow entries

flow-export packets



flow header


byte 0
byte 1
byte 2
byte 3
Flow Header
version
count
sysUptime
unix seconds
unix nanoseconds
flow sequence
padding

flow entry

byte 0	*byte 1*	*byte 2*	*byte 3*

Flow Entry

source IP address			
destination IP address			
next hop IP address			
input intf index		output intf index	
packets			
bytes			
start time of flow			
end time of flow			
source port		destination port	
pad	TCP flags	IP protocol	TOS
source AS		destination AS	
src netmask	dst netmask	padding	

cflowd aggregation

- cflowd primarily designed to aggregate flow data into tabular data to be used for capacity planning
- AS matrix, net matrix, port matrix, protocol table, interface matrix and nexthop table aggregation across all flows

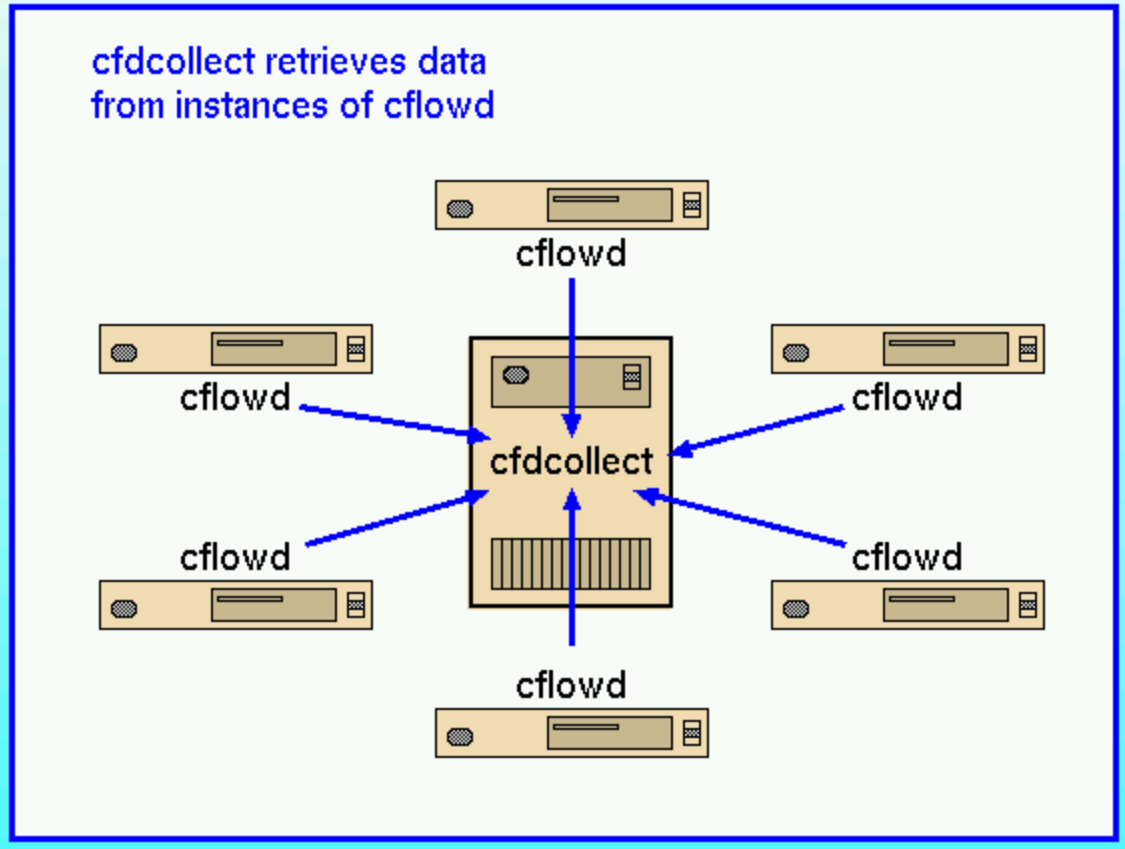
Why so much aggregation?

- data reduction
- scale of intended use (backbone-wide)
- disk and bandwidth requirements for raw flow data in a backbone network. We only transfer tabular data back to the central collector.

cfdcollect

- collects cflowd data at regular intervals
- produces time-series data for each of the tabular data types (AS matrix, net matrix, port matrix, protocol table, interface matrix, nexthop table)
- stores data in ARTS files (1 file per router per day)

centralized collection



arts++

- C++ class library for subset of ARTS data
- supports reading/writing of ARTS data via iostreams and UNIX file descriptors
- supports simple time-domain aggregation for several data types
- simple command-line utilities included for viewing ARTS data files and time domain aggregation

ARTS data files

- efficient data archival (binary, simple size-reducing techniques)
- data files are portable; arts++ class library is the interface
- extensible for additional data types
- versioning of data objects for different storage formats (typically used for space/CPU tradeoff)

ARTS data handled by arts++

- AS matrix (version 0)
- net matrix (version 2)
- port matrix (version 0)
- selected port table (version 0)
- port matrix (version 2)
- selected port table (version 0)
- interface matrix (version 0)
- IP nexthop table (version 0)
- protocol table (version 2)
- forward IP path (version 0)

AS matrix (version 0)

- counters for traffic (packets and bytes) from source ASes to destination ASes
- sparse matrix, having only entries for which traffic information is stored

AS matrix data

router: 192.172.226.1
ifIndex: 8
period: 08/17/1998 05:30:44 - 08/17/1998 05:37:42 EDT

Src AS	Dst AS	Pkts	Pkts/sec	Bytes	Bits/sec
195	195	70784	169.34	18301728	350272
87	195	55445	132.644	12702754	243115
3	195	53667	128.39	5130268	98186.9
1224	195	8764	20.9665	4746527	90842.6
3967	195	4514	10.799	3230531	61828.3
7220	195	2073	4.95933	3066911	58696.9
195	234	2151	5.14593	3056251	58492.8
7050	195	3546	8.48325	2382911	45606
33	195	3616	8.65072	2315453	44314.9
297	195	1613	3.85885	2197017	42048.2
194	195	10483	25.0789	2124627	40662.7
7224	195	1674	4.00478	1979142	37878.3

ARTS net matrix (version 2)

- counters for traffic (packets and bytes) from source networks to destination networks
- networks are identified by network number and netmask length
- sparse matrix, having only entries for which traffic information is stored

net matrix data

router: 192.172.226.1
ifIndex: 8
period: 12/17/1998 05:38:00 - 12/17/1998 05:48:03 EDT

Src Network	Dst Network	Pkts	Bytes
205.189.33.75/32	224.2.202.41/32	77168	62334623
130.149.0.0/16	224.2.231.31/32	30179	24064955
164.58.0.0/16	224.2.131.215/32	18146	4852665
128.223.83.204/32	224.2.246.13/32	12265	4397400
131.188.34.134/32	224.2.172.238/32	2568	2047808
133.82.241.137/32	224.2.172.238/32	6485	1683392
204.123.0.0/16	192.172.226.146/32	959	1427774
209.1.0.0/16	192.172.226.145/32	1231	1362956
205.216.162.0/23	192.172.226.145/32	828	1162127
129.89.143.30/32	224.2.172.238/32	3633	1019726
206.204.0.0/16	192.172.226.145/32	687	1016236
134.102.218.45/32	227.6.5.7/32	771	818813
192.76.157.71/32	224.2.204.172/32	781	810713
209.1.0.0/16	192.172.226.128/25	459	483147
203.8.105.0/24	192.172.226.128/25	4450	482959

ARTS port matrix (version 2)

- counters for (packets and bytes) from source transport layer port number to destination port number
- matrix is sparse; there are no entries for ports on which no traffic was seen

port matrix data

router: 192.172.226.1
ifIndex: 1
period: 10/04/1998 23:55:06 - 10/05/1998 23:56:09 CUT

srcPort	dstPort	Pkts	Pkts/sec	Bytes	Bits/sec
22	47164	1564950	17.3762	2133171600	189483
22	47762	897360	9.96369	1343973900	119381
1029	10000	1041240	11.5612	1070394720	95079.6
22	47591	378600	4.20372	567690210	50426.1
22	47365	31536	0.350155	47102464	4183.96
6000	22041	516041	5.72978	43601192	3872.95
979	22	461500	5.12419	39132700	3476.03
80	21777	28500	0.316445	37517700	3332.57
80	1193	28080	0.311782	37169280	3301.62
80	1799	29040	0.322441	36595200	3250.63
22	47169	26340	0.292462	36525240	3244.42
80	1137	25020	0.277806	34630200	3076.09
55678	53	483652	5.37015	30862702	2741.43
80	2006	22320	0.247827	27806760	2469.98

selected port table data

router: 192.172.226.1
ifIndex: 1
period: 10/04/1998 19:55:06 - 10/05/1998 19:56:09 EDT
selected ports: 1-2048,6000,7070

Port	InPkts	InBytes	OutPkts	OutBytes
22	1200549	88829504	3200619	4163947490
1029	0	0	1041240	1070394720
80	1260232	106367970	848827	765735423
53	501944	32169952	147542	30033536
6000	0	0	516041	43601192
0	527376	43382817	0	0
20	874012	34961888	0	0
513	177475	7224754	0	0
25	25514	3993236	41671	2807293
139	156654	6690843	0	0
515	2457	3002610	0	0
137	0	0	20257	2386046
110	39438	1778969	0	0
1017	0	0	20558	1502200
443	14730	1452375	0	0

protocol table (version 2)

- counters (packets and bytes) versus IP protocol (TCP, UDP, ICMP, IGMP, et. al.)
- sparse table, there are no entries for protocols that were not seen in the measured traffic

protocol table data

```
router:  192.172.226.1
ifIndex: 1
period:  10/04/1998 23:55:06 - 10/05/1998 23:56:09 CUT
Protocol            Pkts       Pkts/sec           Bytes       Bits/sec
--------  -------------  -------------  -------------  -------------
       6         8466492             94     5237854239          465261
      17         2207792             24     1175888232          104450
       1         5785338             64      324733568           28845

router:  192.172.226.1
ifIndex: 8
period:  10/04/1998 23:55:06 - 10/05/1998 23:56:09 CUT
Protocol            Pkts       Pkts/sec           Bytes       Bits/sec
--------  -------------  -------------  -------------  -------------
       6       154258552             1712   120841663280        10733967
      17        19487284              216     1921468060          170677
       1         3662818               40      222174954           19735
```

interface matrix

- counters (packets and bytes) for traffic from input interfaces to output interfaces

interface matrix data

```
router:  192.172.226.1
ifIndex: 1
period:  01/13/1999 18:57:49 - 01/13/1999 19:02:46 EST
  SrcIntf  DstIntf           Pkts      Pkts/sec          Bytes       Bits/sec
  -------  -------  ------------  ------------  ------------  ------------
        1        5         91798       309.084       7250975        195312
        1        0           412       1.38721         44473       1197.93
        1        8           231      0.777778         42394       1141.93

router:  192.172.226.1
ifIndex: 5
period:  01/13/1999 18:57:49 - 01/13/1999 19:02:46 EST
  SrcIntf  DstIntf           Pkts      Pkts/sec          Bytes       Bits/sec
  -------  -------  ------------  ------------  ------------  ------------
        5        8        123520       415.892      57735894   1.55518e+06
        5        1         37246       125.407       8008465        215716
        5        0         10495       35.3367       6165813        166083
```

nexthop table

- counter (packets and bytes) for traffic from input interfaces to each IP nexthop

IP nexthop table data

router: 204.212.46.1
ifIndex: 5
period: 12/06/1998 18:58:37 - 12/06/1998 19:03:37 EST

NextHop	Pkts	Pkts/sec	Bytes	Bits/sec
204.212.46.2	35918	119	2008940	53571
204.212.46.6	617	2	154368	4116
204.212.46.3	310	1	27864	743
0.0.0.0	11	0	616	16

IP forward path (version 0)

- contains IP addresses of hops in forward path from a source to a destination
- contains an RTT value for the source to destination
- may be extended in the future to hold more information

forward IP path data

```
creation: 08/27/1998 03:19:34
Src: 192.172.226.24    (0x18e2acc0)
Dst: 208.201.105.6     (0x669c9d0)
Rtt: 95.623 ms
HopDistance: 14 (0xe)
IsComplete: true
NumHops: 13 (0xd)
        HopNum:    1 IpAddr: 192.172.226.1    (0x1e2acc0)
        HopNum:    2 IpAddr: 198.17.46.43     (0x2b2e11c6)
        HopNum:    3 IpAddr: 204.147.129.90   (0x5a8193cc)
        HopNum:    4 IpAddr: 198.32.128.12    (0xc8020c6)
        HopNum:    5 IpAddr: 204.70.4.93      (0x5d0446cc)
        HopNum:    6 IpAddr: 206.157.77.74    (0x4a4d9dce)
        HopNum:    7 IpAddr: 146.188.145.158 (0x9e91bc92)
        HopNum:    8 IpAddr: 146.188.146.6    (0x692bc92)
        HopNum:    9 IpAddr: 146.188.137.165 (0xa589bc92)
        HopNum:   10 IpAddr: 146.188.178.185 (0xb9b2bc92)
        HopNum:   11 IpAddr: 146.188.176.174 (0xaeb0bc92)
        HopNum:   12 IpAddr: 146.188.176.249 (0xf9b0bc92)
        HopNum:   13 IpAddr: 208.201.105.1    (0x169c9d0)
```

Time Domain Aggregation

Time domain aggregation reduces time granularity in time-series data.

Examples:

- convert 5-minute data to 1-hour data for a summary bar chart
- convert 1-hour data to one large aggregate for a pie chart

Aggregation Utilities

- artsasagg – for AS matrix
- artsnetagg – for net matrix
- artsportmagg – for port matrix
- artsprotoagg – for protocol table
- artsintfmagg – for interface matrix
- artsnexthopagg – for nexthop table

Simple display utilities

- artsdump
- artsases
- artsnets
- artsportms
- artsports
- artsprotos
- artsintfms
- artsnexthops

Supported platforms

- FreeBSD 2.2.x Intel
- FreeBSD 3.0 Intel
- Linux 2.0.35 Intel
- Sparc/Solaris 2.5.1
- Sparc/Solaris 2.6

Other platforms (field reports)

- BSDI BSD/OS 3.1
- BSDI BSD/OS 4.0
- Digital UNIX 4.0B

Future

- support for version 8 flow-export
- plotting utilities using XRT/PDS
- Web reporting tools (java-based)