cflowd and arts++

Cisco flow-export collection

dwm@caida.org

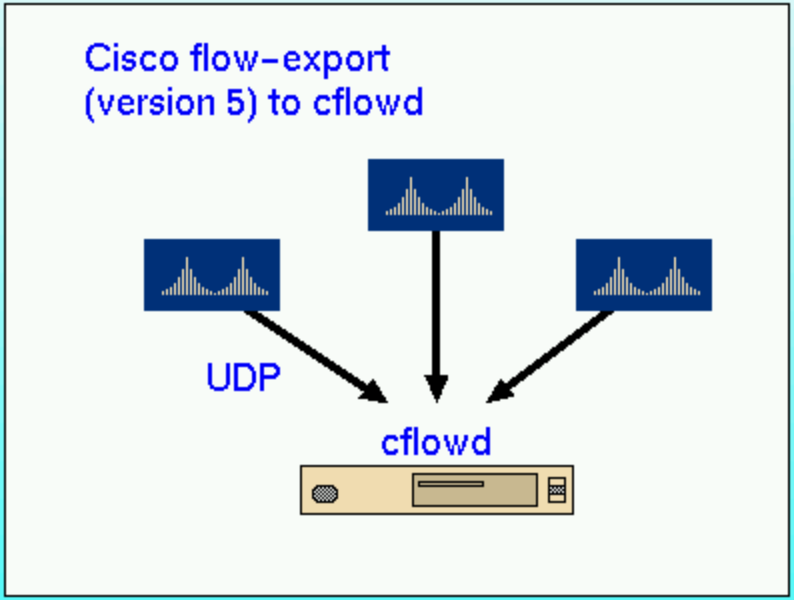
Background

- host software for collecting Cisco version 5 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 to Upcoming Release

- central collector included
- uses arts++ package
- arts++ adds signficant functionality

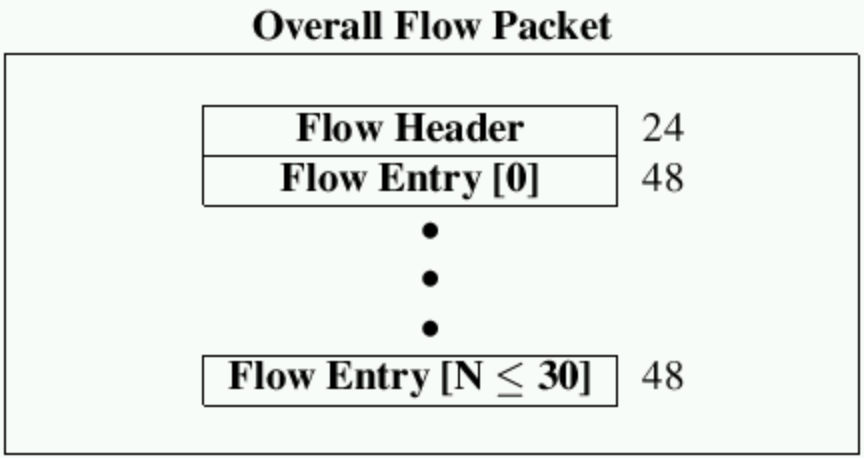
cflowd



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 table and protocol table aggregation across all flows

Why so much aggregation?

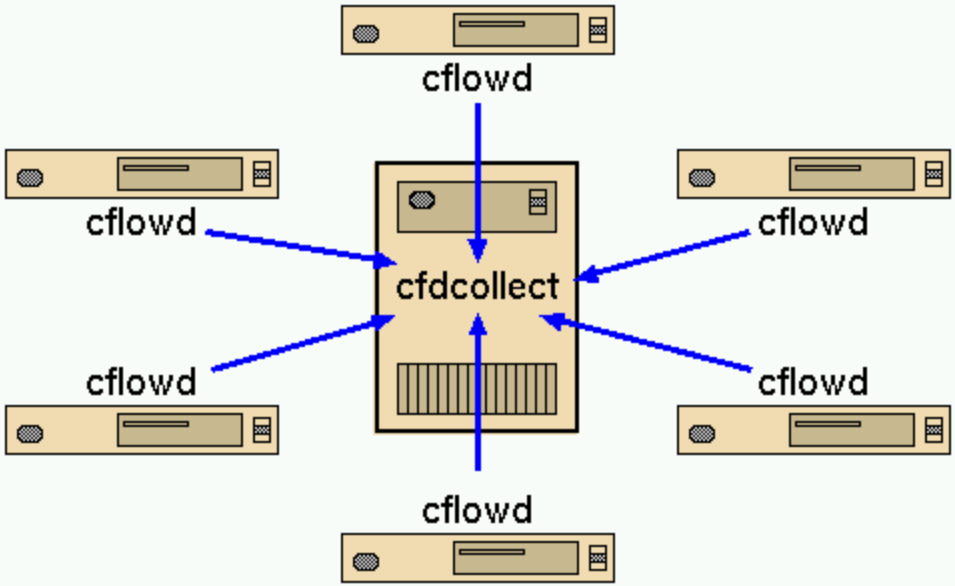
- data reduction
- scale of intended use (backbone–wide)
- Add a cflowd host, you may add it to central collection of tabular data.
- unreliable transport from Cisco to cflowd encourages deploying multiple cflowd hosts; use them.
- disk and bandwidth requirements for raw flow data in a backbone network. Only transfer tabular data back to the central collector.

cfdcollect

- permits collection of cflowd data at intervals
- end result is time-series data for each of the tabular data types (AS matrix, net matrix, port table and protocol table)
- stores data in ARTS files

centralized collection

cfdcollect retrieves data
from instances of cflowd



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; always written in network byte order, the 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 table (version 2)
- protocol table (version 2)
- forward IP path (version 0)

ARTS 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

ARTS AS matrix example data

router: 192.172.226.1
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
10487	195	1976	4.72727	1871791	35823.8
3549	195	1583	3.78708	1198146	22931
701	195	2276	5.44498	1118054	21398.2
195	1227	1659	3.9689	930102	17801
5050	195	2817	6.73923	696150	13323.4
3561	195	3199	7.65311	688489	13176.8
3847	195	904	2.16268	582805	11154.2
2895	195	3155	7.54785	574185	10989.2

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

ARTS net matrix example data

router: 192.172.226.1
period: 08/17/1998 05:38:00 - 08/17/1998 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
164.58.0.0/16	224.2.131.215/32	17987	1913904
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
192.76.157.71/32	224.2.204.172/32	676	698131
192.76.157.71/32	224.2.204.172/32	641	662238
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 table (version 2)

- counters for input and output traffic (packets and bytes) versus transport layer port number
- input counters represent traffic destined for the port while output counters represent traffic sourced from the port
- table is sparse; there are no entries for ports on which no traffic was seen

ARTS port table example data

router: 192.172.226.1
period: 08/26/1998 20:02:47 - 08/27/1998 20:01:23 EDT

Port	InPkts	InBytes	OutPkts	OutBytes
3128	7835924	690104918	2552580	3318268707
22	1522482	73248190	2404976	3354537868
80	316789	24446994	2504631	2180188920
60172	1737458	1180790389	5	416
1032	1062	900137	1864767	1176626265
55554	4003647	1156760497	0	0
1029	656	536740	2122683	828264152
46484	496272	743639839	247848	9916573
44265	0	0	1061388	726464677
61000	942260	709575377	41	3029
1026	1606	1442215	1843914	663322545
22130	1842195	663190200	116	11293
1033	1135	974659	1822077	653833897
26406	1803749	649349640	0	0
23824	1728238	622165680	5	570
1071	1373	1354134	1689113	606949288

ARTS 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

ARTS protocol example data

router: 192.172.226.1
period: 08/26/1998 20:02:47 - 08/27/1998 20:01:23 EDT

Protocol	Pkts	Bytes
17	37969426	12196396017
6	18131797	10258492395
1	8319666	1003117441
2	14306	2277900
41	216	16416
46	171	4788
47	10	565

ARTS 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

ARTS forward IP path example 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)
```

Aggregation Utilities

Time domain aggregation:

- artsasagg
- artsnetagg
- artsportagg
- artsprotoagg

Simple display utilities

- artsdump
- artsases
- artsnets
- artsports
- artsprotos

Future Tools

- plotting utilities using XRT/PDS
- utilities to generate data files for JClass Chart

Open Questions

- what types of aggregation are useful to network service providers?
- are there desired applications for flow-export outside of capacity planning and usage/billing?