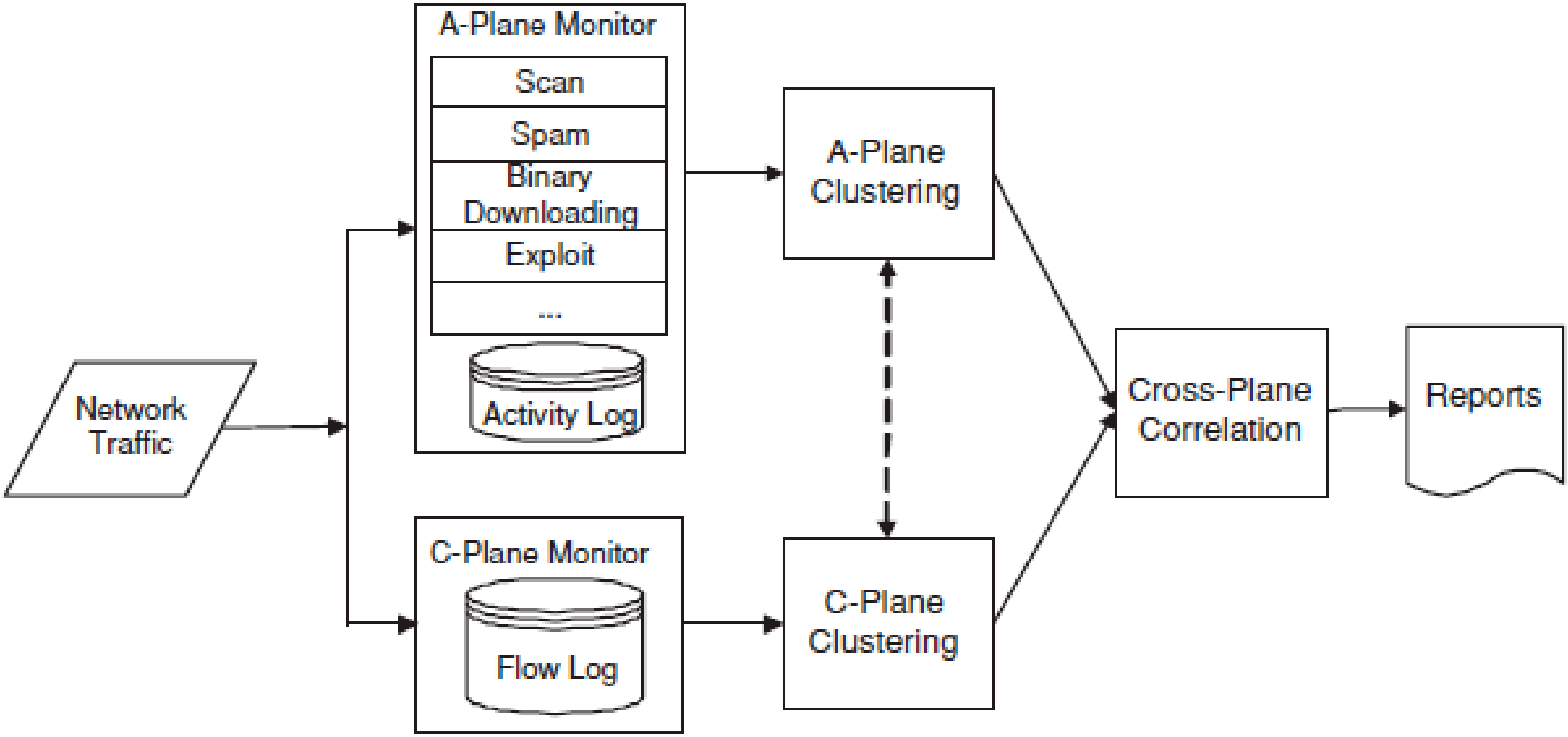
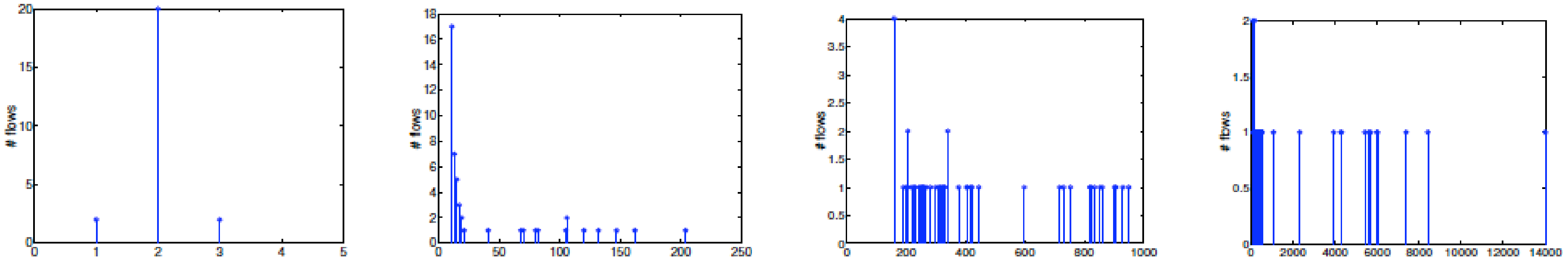
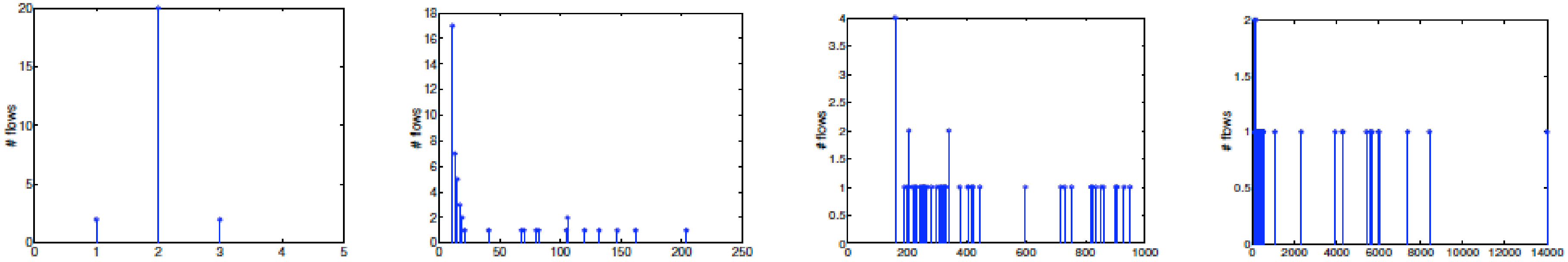


Figure 2: Architecture overview of our BotMiner detection framework.



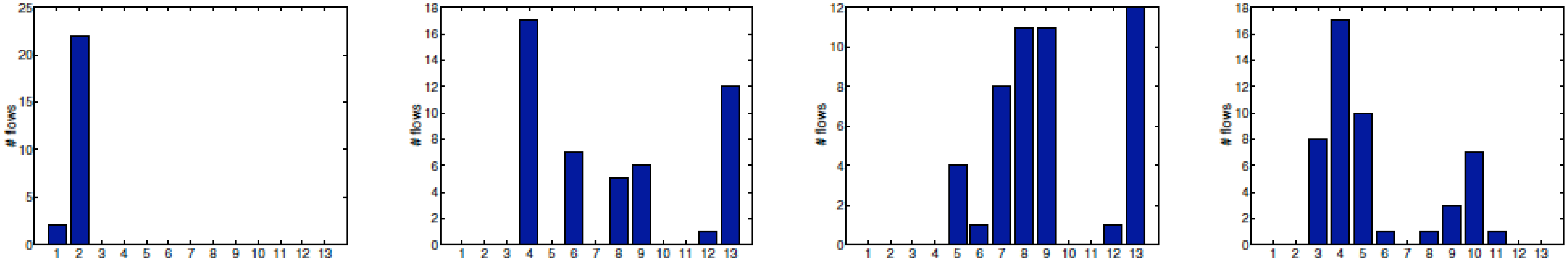
(a) fph (b) ppf (c) bpp (d) bps

Figure 4: Visit pattern (shown in distribution) to `Google` from a randomly chosen normal client.



(a) fph (b) ppf (c) bpp (d) bps

Figure 4: Visit pattern (shown in distribution) to `Google` from a randomly chosen normal client.



(a) Scaled fph (b) Scaled ppf (c) Scaled bpp (d) Scaled bps

Figure 5: Scaled visit pattern (shown in distribution) to `Google` for the same client in Figure 4.

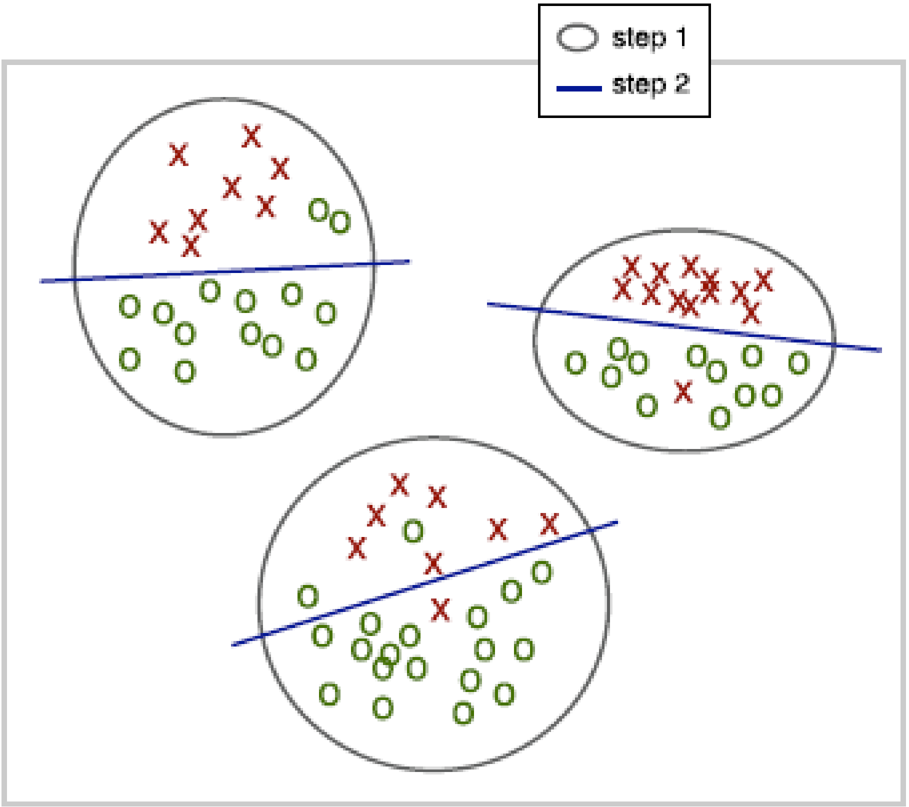


Figure 6: Two-step clustering of C-flows.

Trace	Size	Duration	Pkt	TCP/UDP flows	Botnet clients	C&C server
Botnet-IRC-rbot	169MB	24h	1,175,083	180,988	4	1
Botnet-IRC-sdbot	66KB	9m	474	19	4	1
Botnet-IRC-spybot	15MB	32m	180,822	147,945	4	1
Botnet-IRC-N	6.4MB	7m	65,111	5635	259	1
Botnet-HTTP-1	6MB	3.6h	65,695	2,647	4	1
Botnet-HTTP-2	37MB	19h	395,990	9,716	4	1
Botnet-P2P-Storm	1.2G	24h	59,322,490	5,495,223	13	P2P
Botnet-P2P-Nugache	1.2G	24h	59,322,490	5,495,223	82	P2P

Table 1: Collected botnet traces, covering IRC, HTTP and P2P based botnets. Storm and Nugache share the same file, so the statistics of the whole file are reported.

example of a typical P2P-based botnet, namely Storm worm [18, 23]. In order to issue commands to the bots, the botmaster publishes/shares command files over the P2P network, along with specific search keys that can be used by the bots to find the published command files. Storm bots utilize a pull mechanism to receive the commands. Specifically, each bot frequently contacts its neighbor peers searching for specific keys in order to locate the related command files. In addition to search

Trace	Size	Duration	Pkt	TCP/UDP flows	Botnet clients	C&C server
Botnet-IRC-rbot	169MB	24h	1,175,083	180,988	4	1
Botnet-IRC-sdbot	66KB	9m	474	19	4	1
Botnet-IRC-spybot	15MB	32m	180,822	147,945	4	1
Botnet-IRC-N	6.4MB	7m	65,111	5635	259	1
Botnet-HTTP-1	6MB	3.6h	65,695	2,647	4	1
Botnet-HTTP-2	37MB	19h	395,990	9,716	4	1
Botnet-P2P-Storm	1.2G	24h	59,322,490	5,495,223	13	P2P
Botnet-P2P-Nugache	1.2G	24h	59,322,490	5,495,223	82	P2P

Table 1: Collected botnet traces, covering IRC, HTTP and P2P based botnets. Storm and Nugache share the same file, so the statistics of the whole file are reported.

Trace	Pkts	Flows	Filtered by F1	Filtered by F2	Filtered by F3	Flows after filtering	C-flows (TCP/UDP)
Day-1	5,178,375,514	23,407,743	20,727,588	939,723	40,257	1,700,175	66,981 / 132,333
Day-2	7,131,674,165	29,632,407	27,861,853	533,666	25,758	1,211,130	34,691 / 96,261
Day-3	9,701,255,613	30,192,645	28,491,442	513,164	24,329	1,163,710	39,744 / 94,081
Day-4	14,713,667,172	35,590,583	33,434,985	600,901	33,958	1,520,739	73,021 / 167,146
Day-5	11,177,174,133	56,235,380	52,795,168	1,323,475	40,016	2,076,721	57,664 / 167,175
Day-6	9,950,803,423	75,037,684	71,397,138	1,464,571	51,931	2,124,044	59,383 / 176,210
Day-7	10,039,871,506	109,549,192	105,530,316	1,614,158	56,688	2,348,030	55,023 / 150,211
Day-8	11,174,937,812	96,364,123	92,413,010	1,578,215	60,768	2,312,130	56,246 / 179,838
Day-9	9,504,436,063	62,550,060	56,516,281	3,163,645	30,581	2,839,553	25,557 / 164,986
Day-10	11,071,701,564	83,433,368	77,601,188	2,964,948	27,837	2,839,395	25,436 / 154,294

Table 2: C-plane traffic statistics, basic results of filtering, and C-flows.

Trace	Step-1 C-clusters	Step-2 C-clusters	A-plane logs	A-clusters	False Positive Clusters	FP Rate
Day-1 (TCP/UDP)	1,374	4,958	1,671	1	0	0 (0/878)
Day-2 (TCP/UDP)	904	2,897	5,434	1	1	0.003 (2/638)
Day-3 (TCP/UDP)	1,128	2,480	4,324	1	1	0.003 (2/692)
Day-4 (TCP/UDP)	1,528	4,089	5,483	4	4	0.01 (9/871)
Day-5 (TCP/UDP)	1,051	3,377	6,461	5	2	0.0048 (4/838)
Day-6 (TCP)	1,163	3,469	6,960	3	2	0.008 (7/877)
Day-7 (TCP)	954	3,257	6,452	5	2	0.006 (5/835)
Day-8 (TCP)	1,170	3,226	8,270	4	2	0.0091 (8/877)
Day-9 (TCP)	742	1,763	7,687	2	0	0 (0/714)
Day-10 (TCP)	712	1,673	7,524	0	0	0 (0/689)

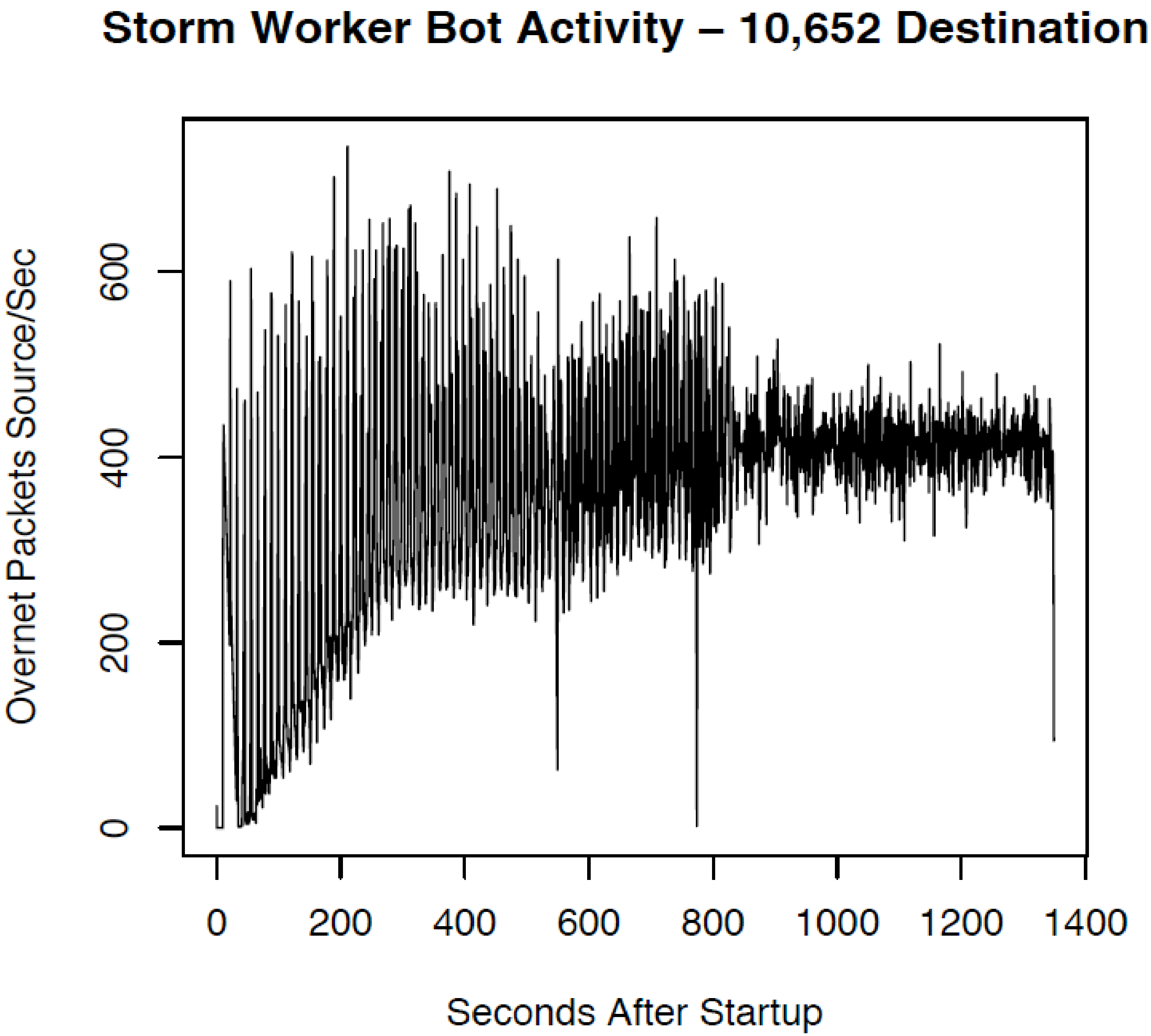
Table 3: C-plane and A-plane clustering results.

Trace	Step-1 C-clusters	Step-2 C-clusters	A-plane logs	A-clusters	False Positive Clusters	FP Rate
Day-1 (TCP/UDP)	1,374	4,958	1,671	1	0	0 (0/878)
Day-2 (TCP/UDP)	904	2,897	5,434	1	1	0.003 (2/638)
Day-3 (TCP/UDP)	1,128	2,480	4,324	1	1	0.003 (2/692)
Day-4 (TCP/UDP)	1,528	4,089	5,483	4	4	0.01 (9/871)
Day-5 (TCP/UDP)	1,051	3,377	6,461	5	2	0.0048 (4/838)
Day-6 (TCP)	1,163	3,469	6,960	3	2	0.008 (7/877)
Day-7 (TCP)	954	3,257	6,452	5	2	0.006 (5/835)
Day-8 (TCP)	1,170	3,226	8,270	4	2	0.0091 (8/877)
Day-9 (TCP)	742	1,763	7,687	2	0	0 (0/714)
Day-10 (TCP)	712	1,673	7,524	0	0	0 (0/689)

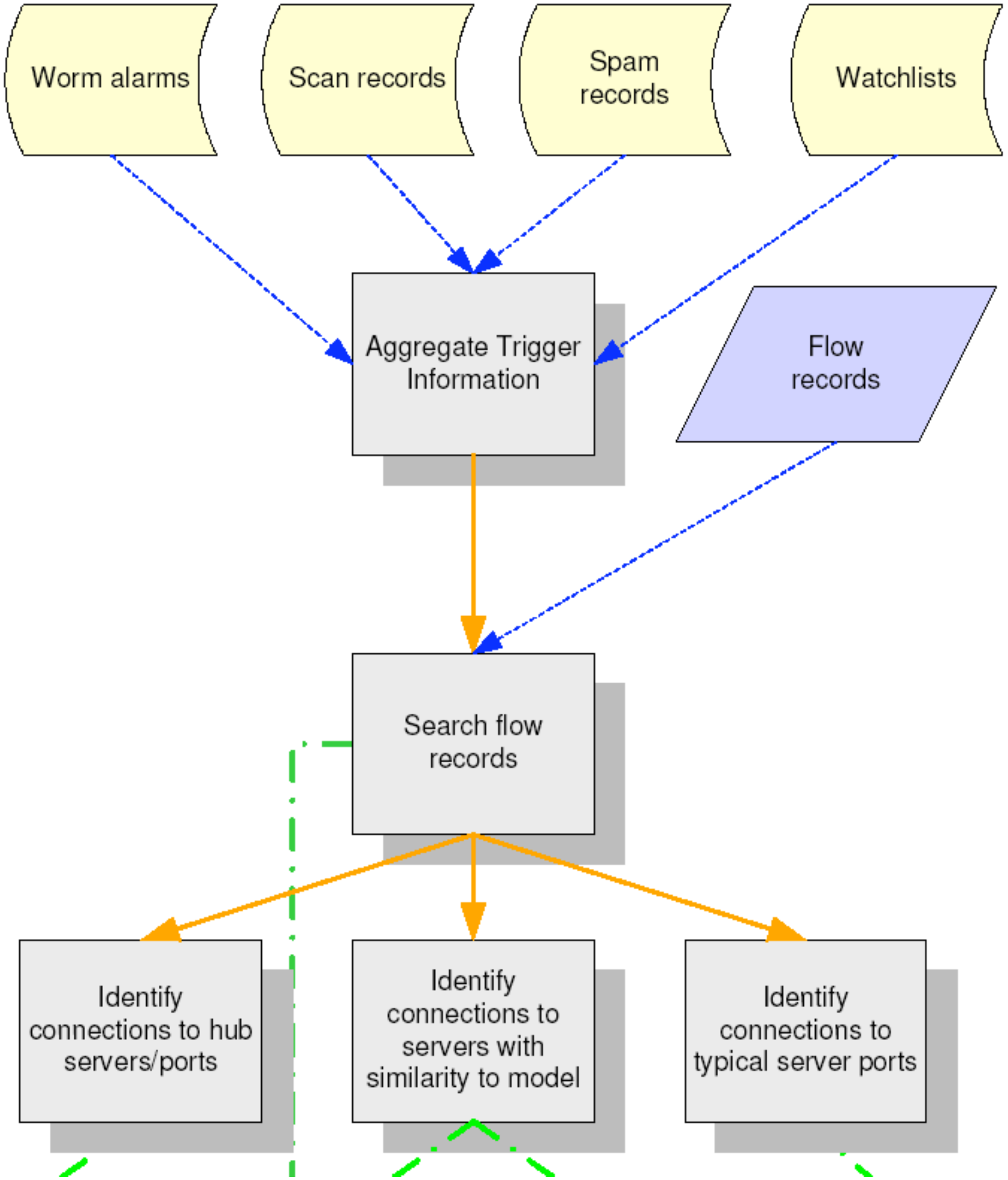
Table 3: C-plane and A-plane clustering results.

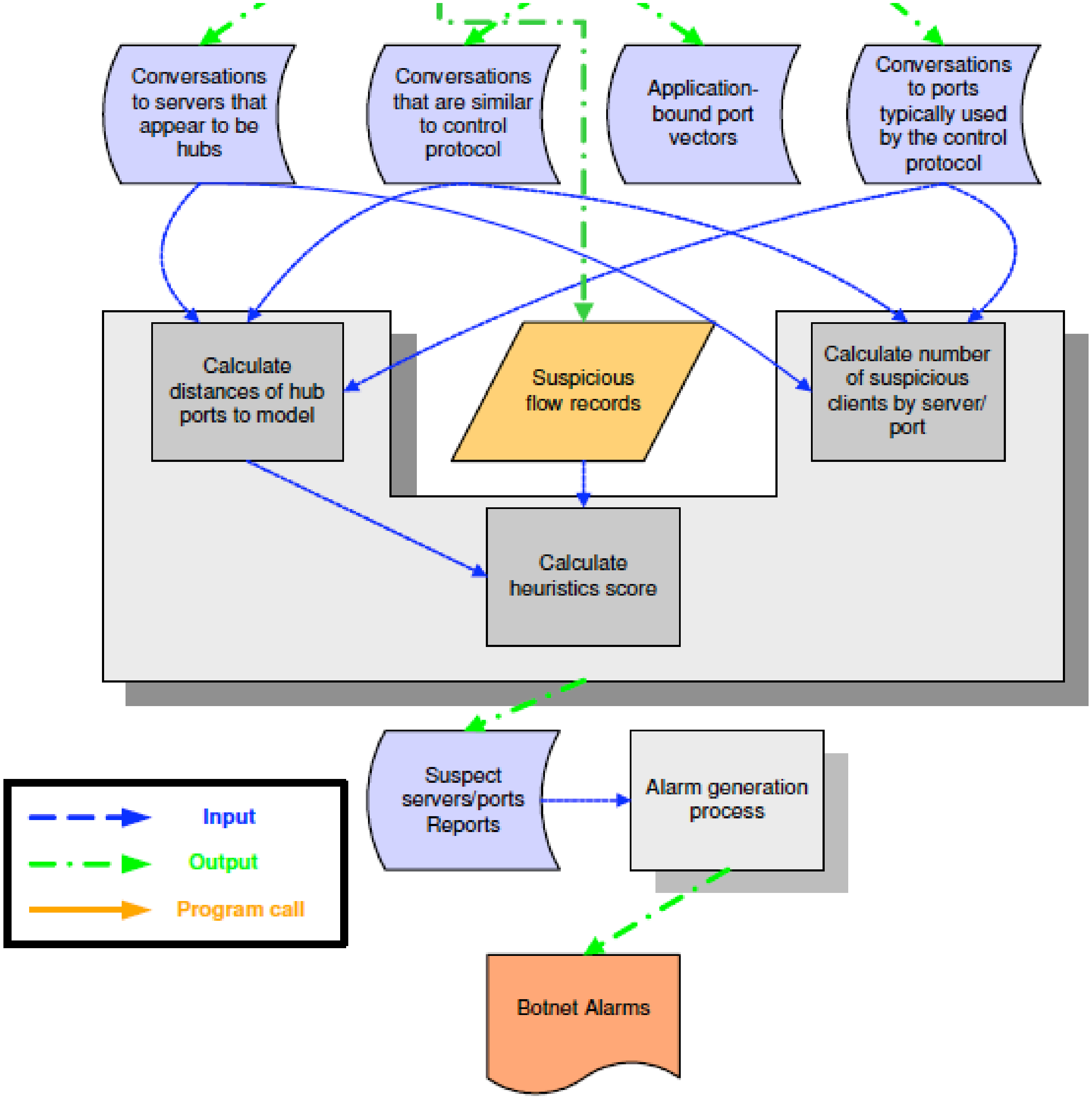
Botnet	Number of Bots	Detected?	Clustered Bots	Detection Rate	False Positive Clusters/Hosts	FP Rate
IRC-rbot	4	YES	4	100%	1/2	0.003
IRC-sdbot	4	YES	4	100%	1/2	0.003
IRC-spybot	4	YES	3	75%	1/2	0.003
IRC-N	259	YES	258	99.6%	0	0
HTTP-1	4	YES	4	100%	1/2	0.003
HTTP-2	4	YES	4	100%	1/2	0.003
P2P-Storm	13	YES	13	100%	0	0
P2P-Nugache	82	YES	82	100%	0	0

Table 4: Botnet detection results using BotMiner.

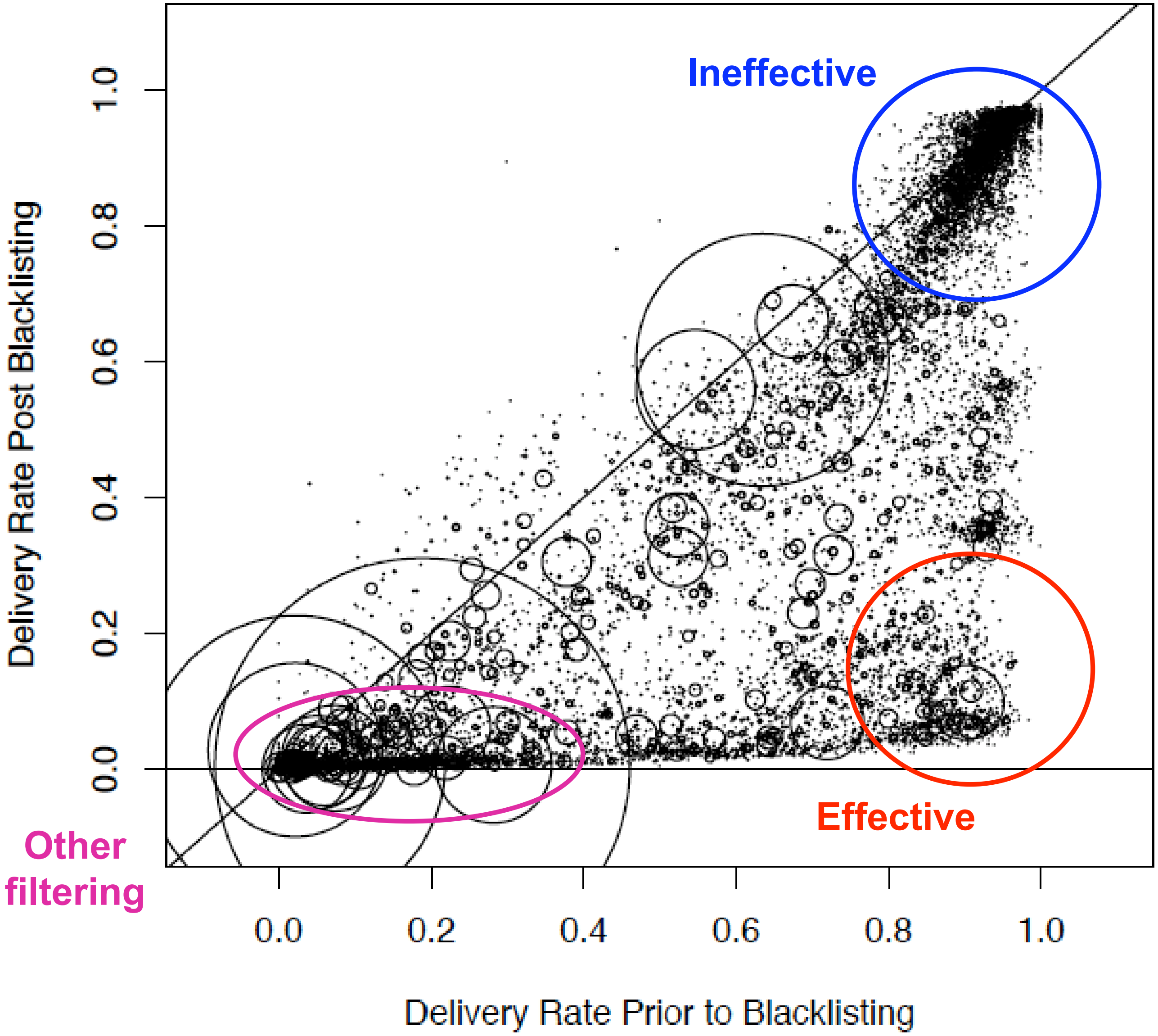

Storm Worker Bot Activity – 10,652 Destinations
Overnet Packets Source/Sec
0
200
400
600
0
200
400
600
800
1000
1200
1400
Seconds After Startup

features of chat-like protocols such as IRC. Karasaridis et al. [26] studied network flow level detection of IRC botnet controllers for backbone networks. The above two are similar to our work in C-plane clustering but different in many ways. First, they are used to detect IRC-based botnet (by matching a known IRC traffic profile), while we do not have the assumption of known C&C protocol profiles. Second, we use a different feature set on a new communication flow (C-flow) data format instead of traditional network flow. Third, we consider both C-plane and A-plane information instead of just flow records.


Worm alarms
Scan records
Spam records
Watchlists
Aggregate Trigger Information
Flow records
Search flow records
Identify connections to hub servers/ports
Identify connections to servers with similarity to model
Identify connections to typical server ports


Conversations to servers that appear to be hubs
Conversations that are similar to control protocol
Application-bound port vectors
Conversations to ports typically used by the control protocol
Calculate distances of hub ports to model
Suspicious flow records
Calculate number of suspicious clients by server/ port
Calculate heuristics score
Suspect servers/ports Reports
Alarm generation process
Botnet Alarms
Input
Output
Program call

Effects of Blacklisting on Delivery Rates



Ineffective
Effective
Other filtering
Delivery Rate Post Blacklisting
Delivery Rate Prior to Blacklisting