the Internet's 'core'

Top IP, prefixes and AS

Andre Broido
broido@caida.org, CAIDA.SDSC.UCSD
kc claffy
kc@caida.org, UCSD/SDSC/CAIDA
9 dec 00

IP graph

- Nodes: IP addresses
- Links: pairs of IP addresses
  - TTL distance 1
  - Observed in a skitter path
  - Nonresponding hops changed to IP contexts
  - No attempt to remove bogus addresses
  - Graph is directed

Combinatorial core

- How can we obtain the core?
- Outdegree 0:
  - nodes with no outgoing links.
- eestinations have outdegree 0
- Strip nodes of outdegree 0
- Stub network's routers now have outdegree 0
- Repeat (recursive) stripping of outdegree 0

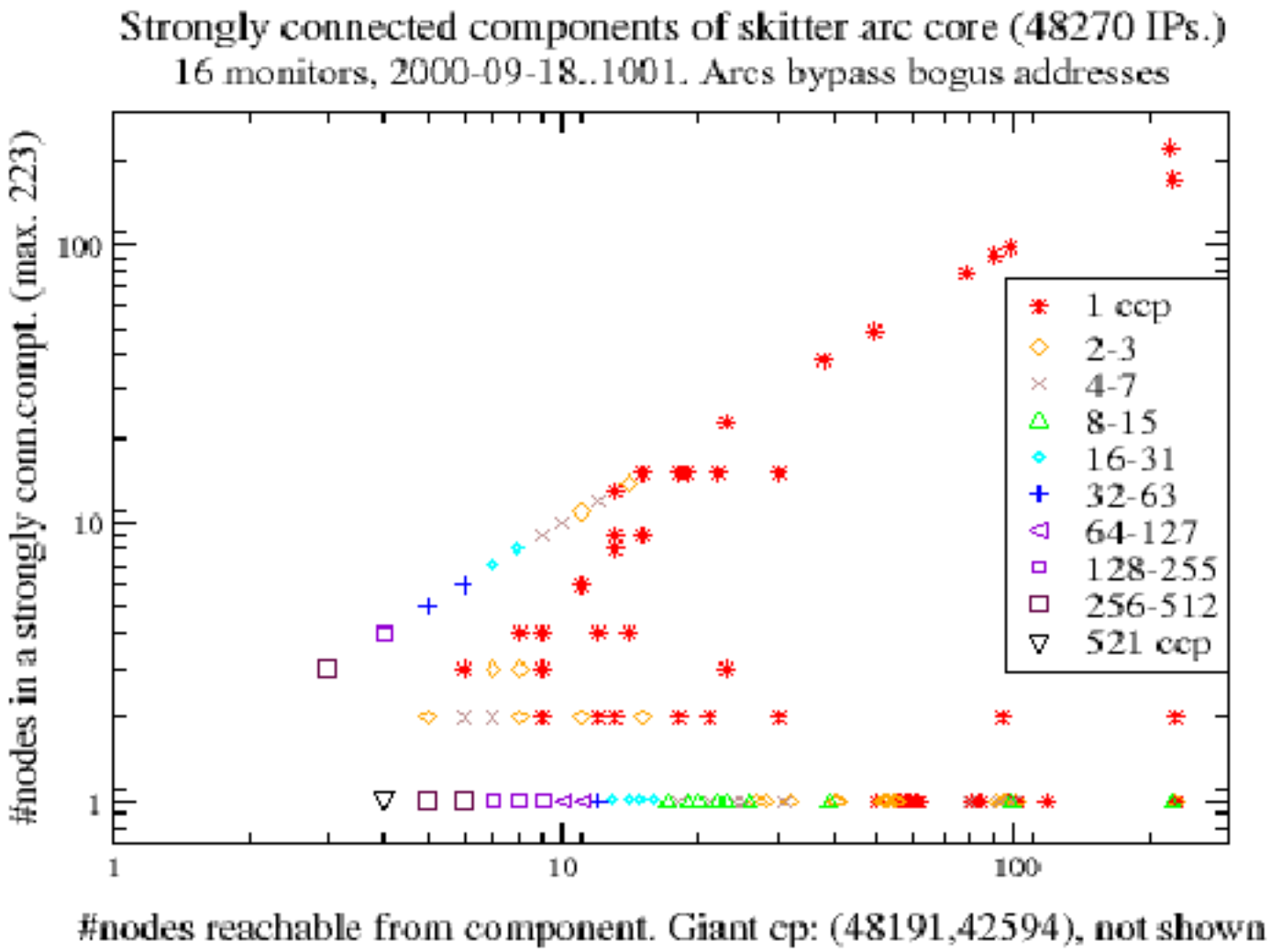
Skitter data of Jul.15–18

- 826265 nodes
- 60329 (7%) in combinatorial core
- Stripped in 24 iterations
- 52330 nodes in giant connected component
- Almost 90% of the core

Giant component

- Bidirectional connectivity
- Connected components of the core
- Giant component
- 200 times larger than any other connected component

Small connected components

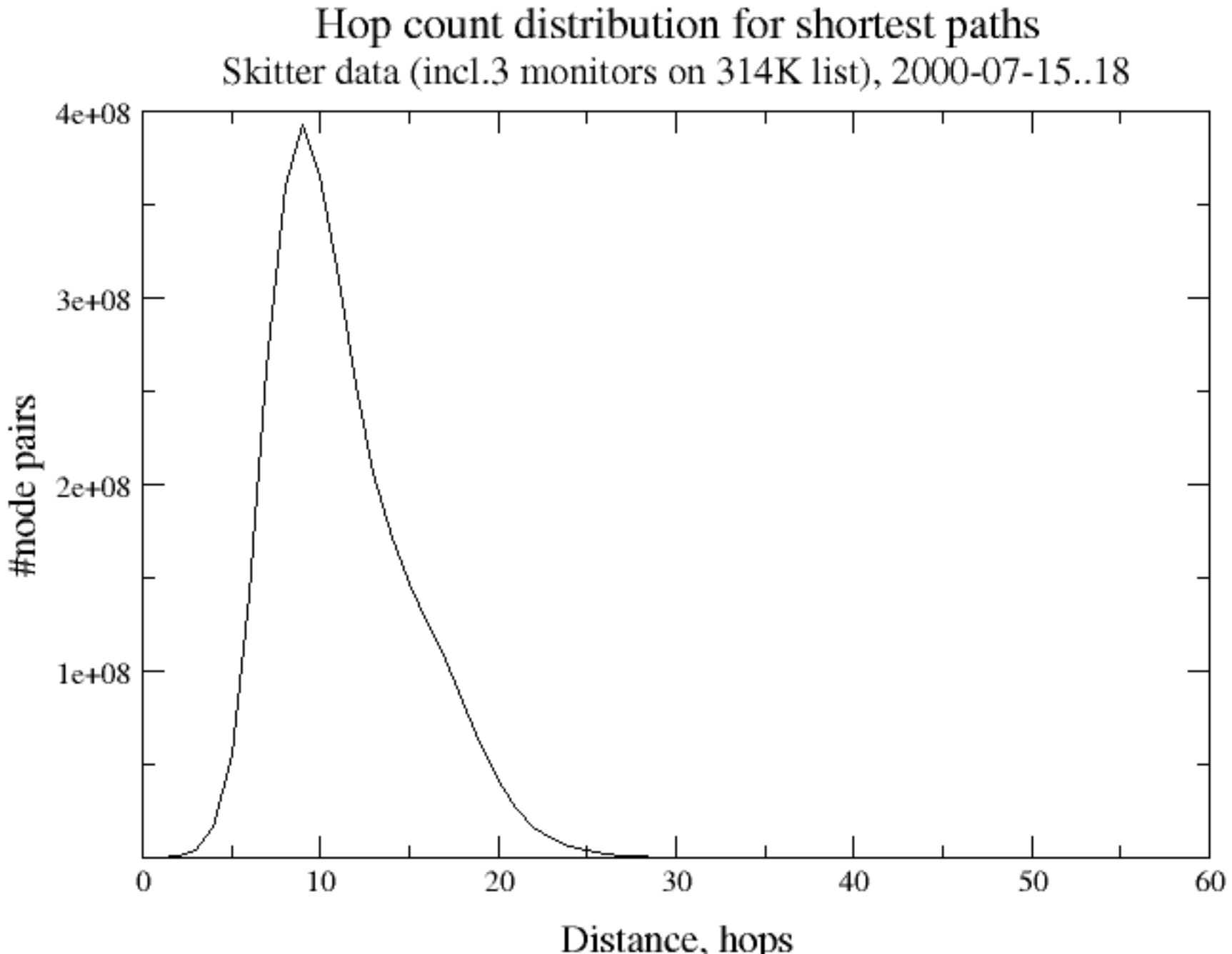


Giant component is 200 times larger than any other
Largest of small components are /24s, /25s

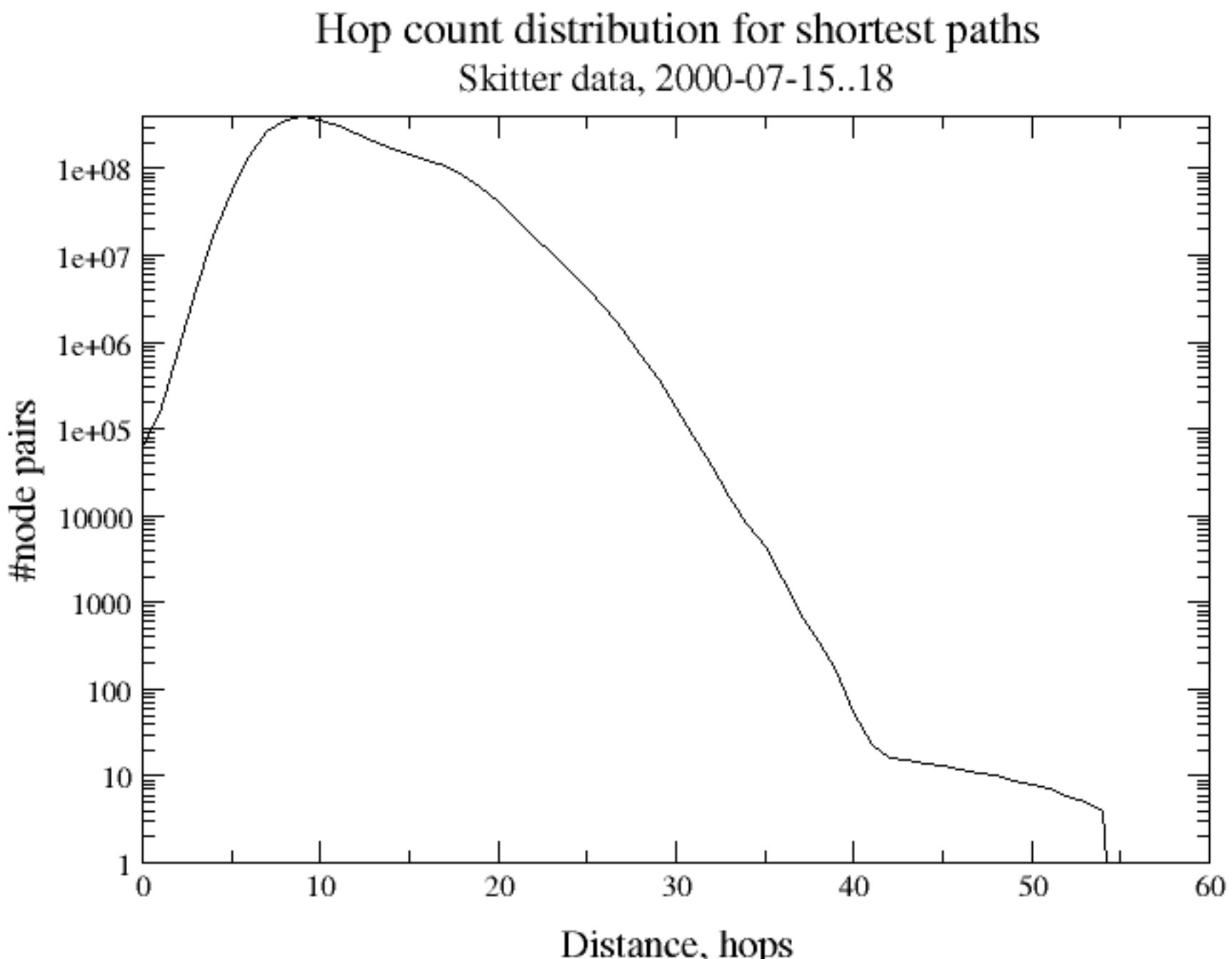
Hop metric

- Shortest path distance on IP graph
- Find the maximum distance from any given node to all core
- Center of the core: those with min maximum distance
- Take top 50 by this metric

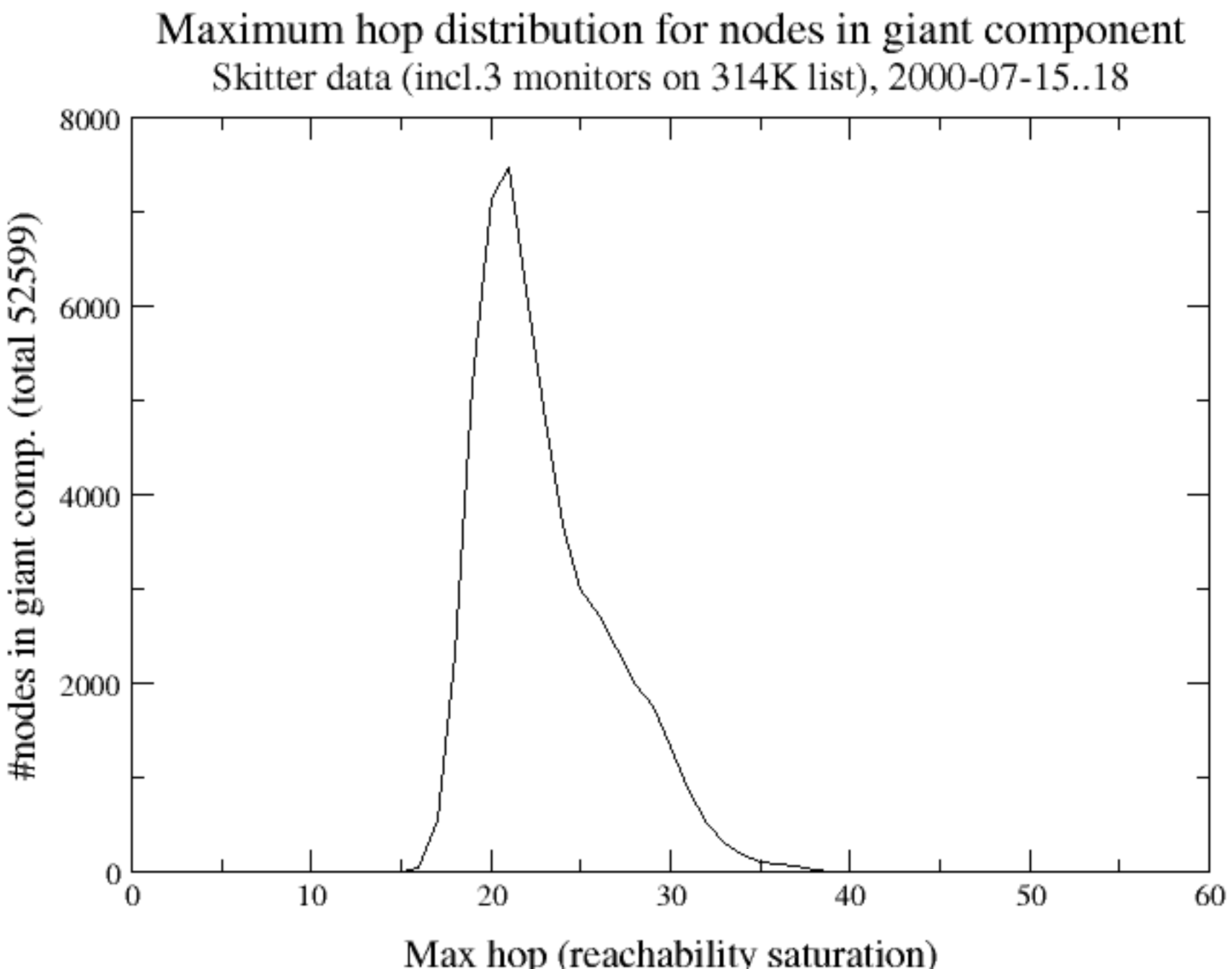
Neighbourhood size vs. hop distance



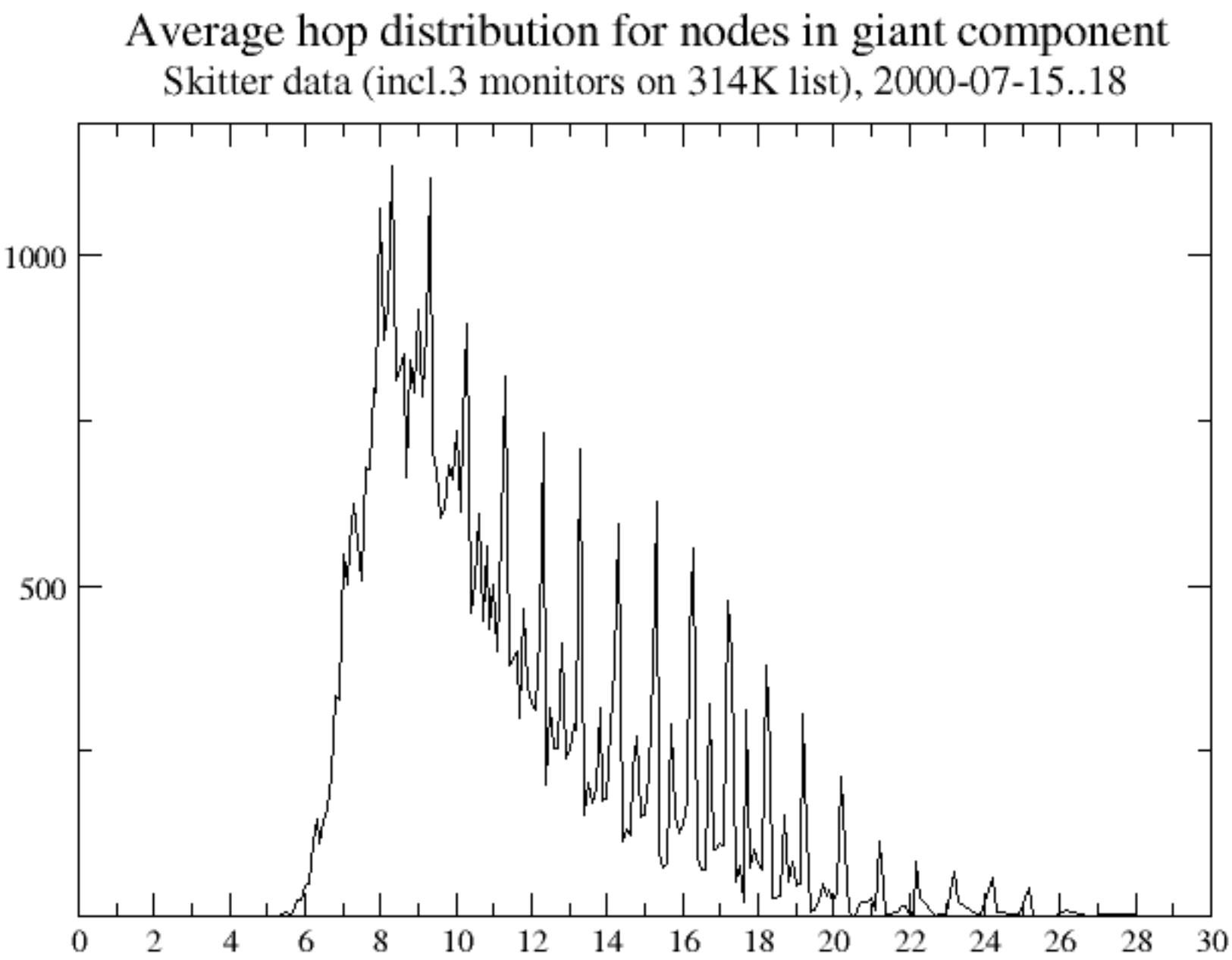
Neighbourhood size vs. hop distance



Maximum hop distance to the core



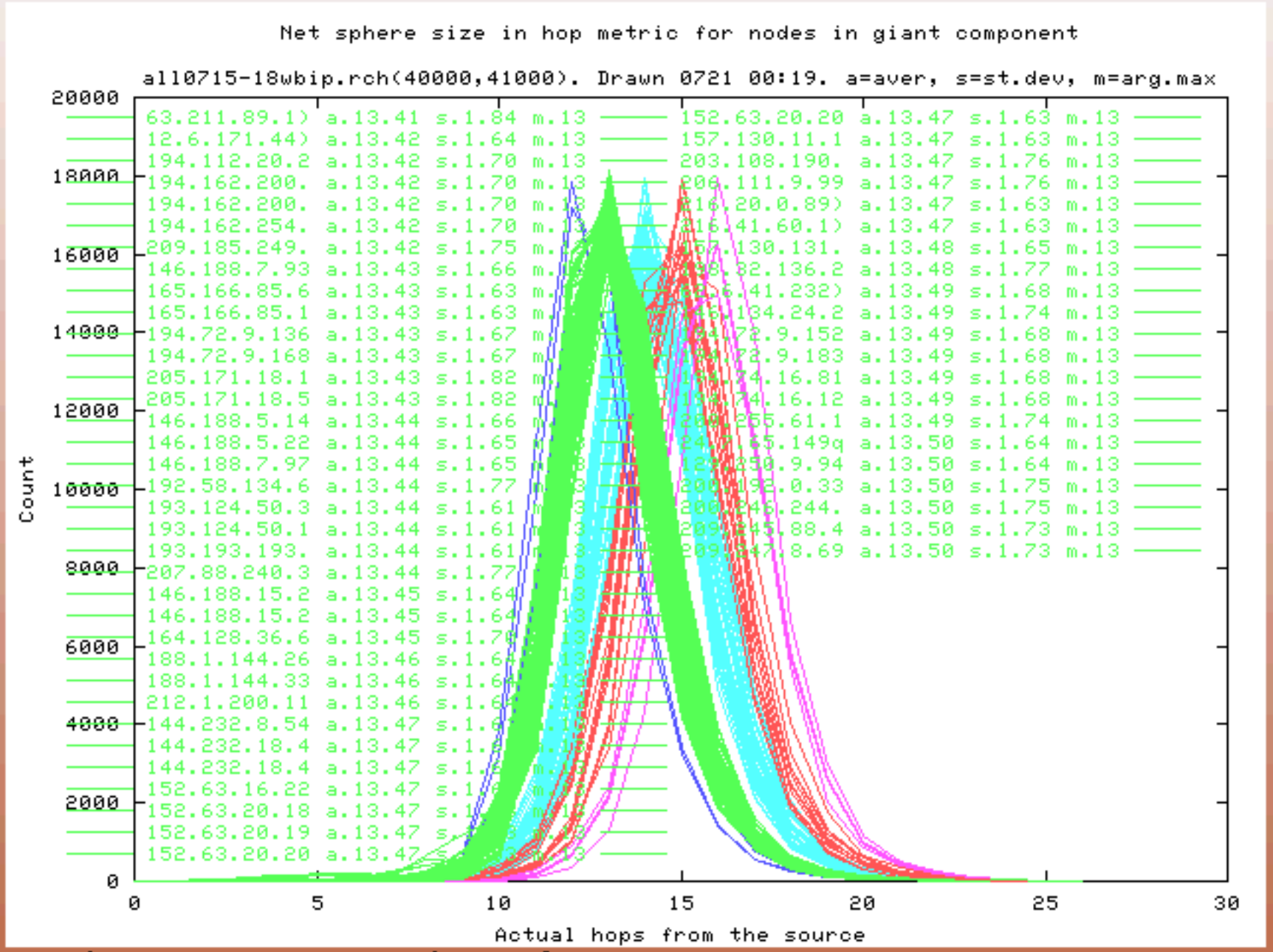
Average hop distance to the core



Top 3 prefixes

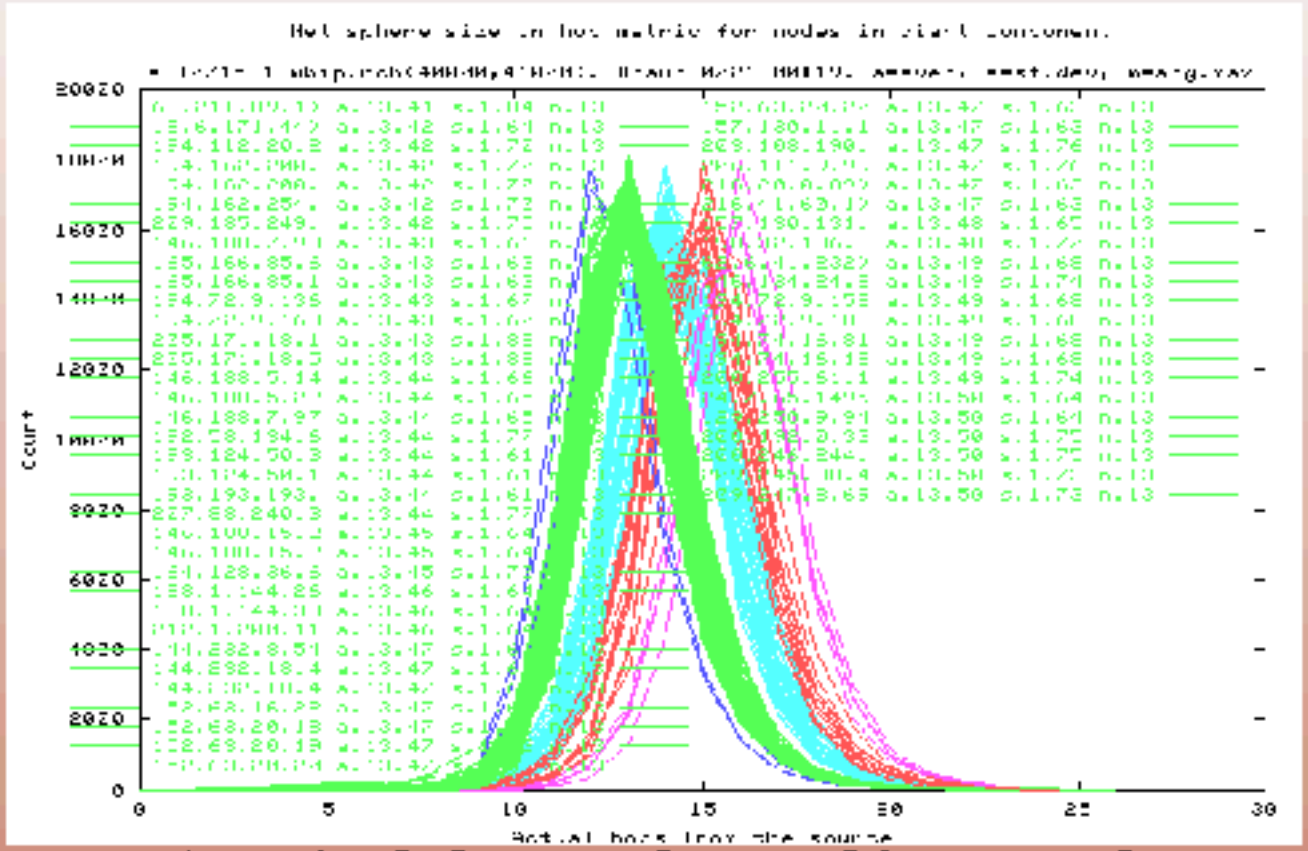
Prefix	Outdegree
157.130.0.0/16	1172
144.232.0.0/16	757
4.0.0.0/8	655

Neighbourhood size vs. hop distance



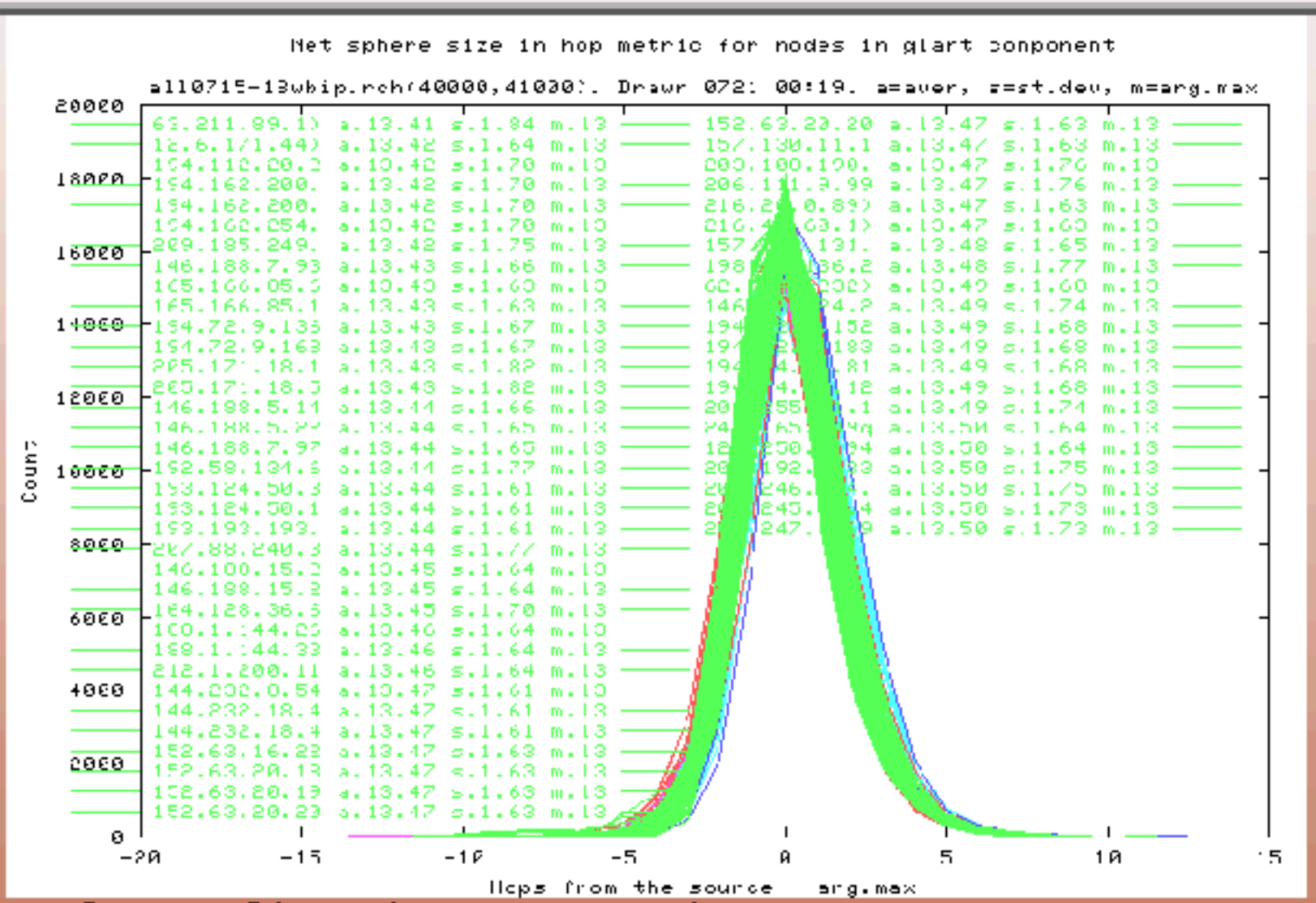
1000 nodes, 2% sample of core

Neighbourhood size vs. hop distance



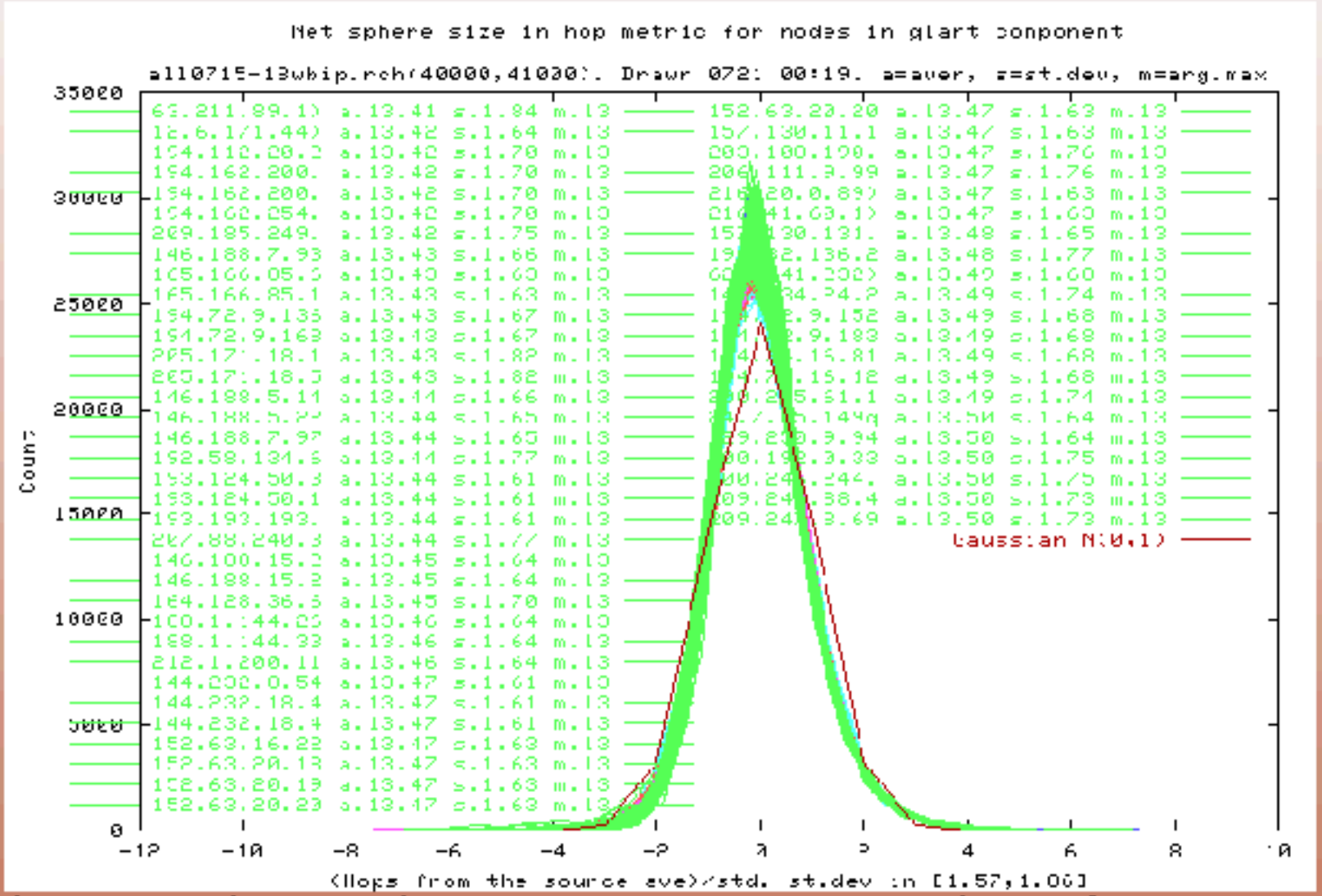
- ‘neighbourhood’: nodes at radius R
- hypothesis: there exists a mother function that fits all these curves
- such a function would provides a parameteric method for finding topological distance from internet core

Neighbourhood size: shifting by arg.max



- just by aligning maxima -->
  you get something that looks like
  a mother function

Neighbourhood size: (hop−ave)/std



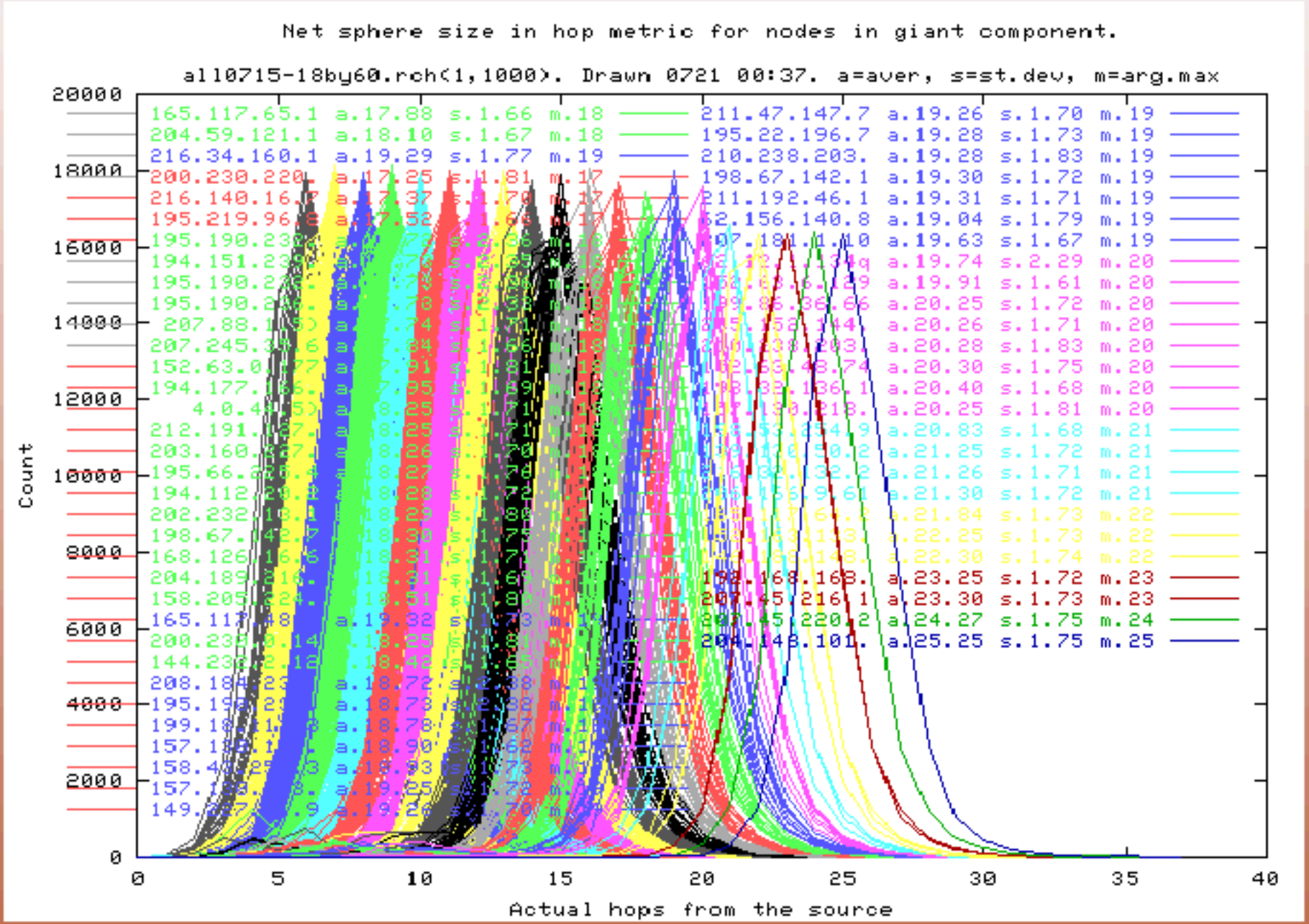
this analysis demonstrates a mother function f
s.t. the neighbourhood size = f((hop−ave)/std)

--> ask questions now!

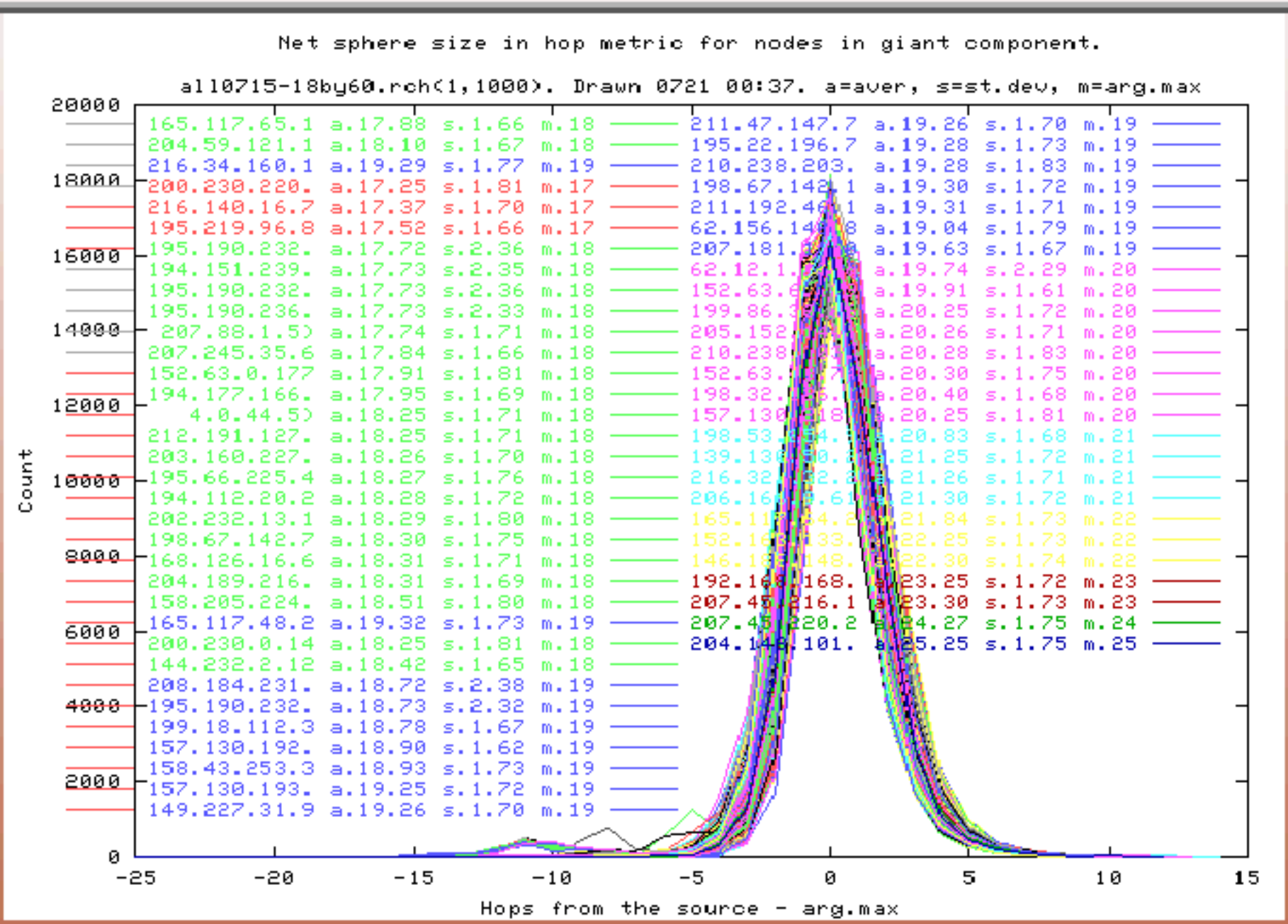
Comparing w/(discrete) Gaussian

- Generic shape is close
- Heavier tail

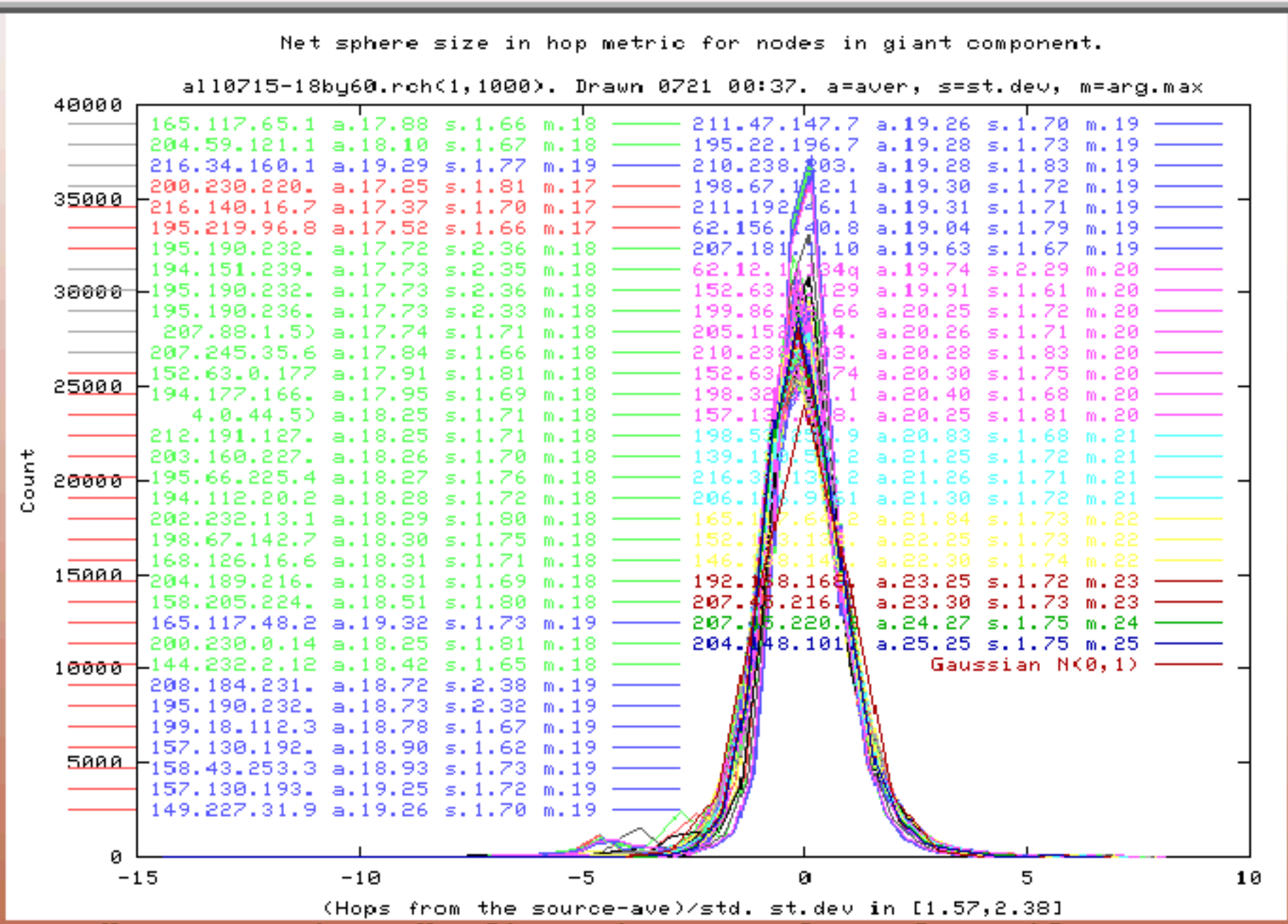
Neighbourhood size – all core



Neighbourhood size: shifting by arg.max

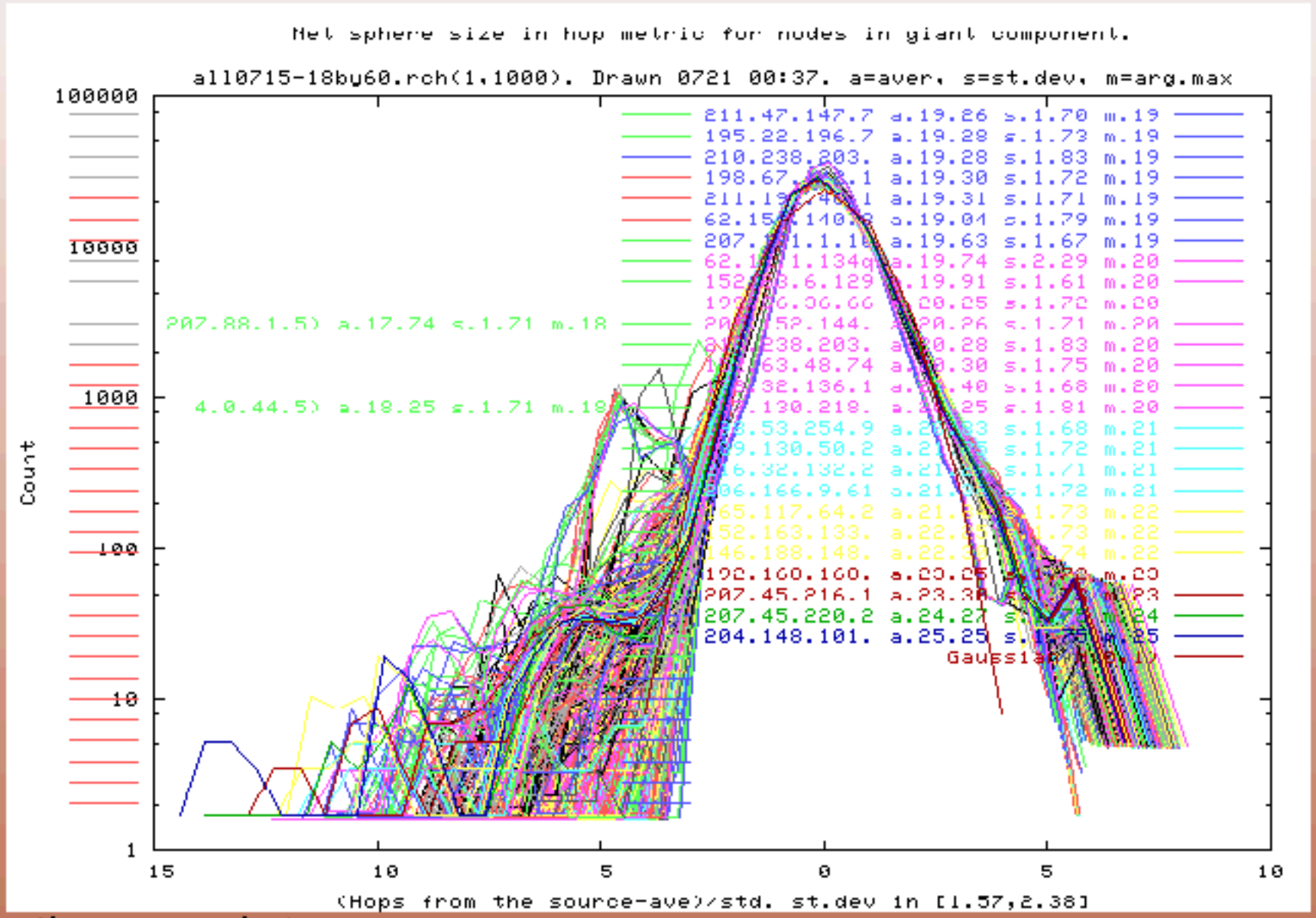


Neighbourhood size vs. (hop−ave)/std



- "Gaussian" fits into the family

Mother function covers 98% of reachable nodes



(log scale)

Policy-constrained dual AS graph

- Nodes = peering sessions
- Links = pairs of sessions w/common AS
- and followed by traffic
- in skitter or BGP data
- Derived from observed AS triples

Dual AS graph

Example:

- 1909 195 1740 – AS path (skitter,BGP)
- CAIDA SDSC CERF
- Nodes: 1909–195 and 195–1740
- Ordered AS pairs
- Link: from 1909–195 to 195–1740
- Directed links

Dual AS graph

- AS hop count = #sessions in a path
- easy to compare with AS hops
- If necessary, add: 0–AS and AS–0
- for traffic origination/termination
- (was not done for graphs shown here)

Dual AS graph

Idea:

- add rigidity to the graph
- constrain path by continuity
- and direction
- like differential equation
- but more choice at each step
- path continuation non–unique

General case:

- nodes: AS n–tuples, links: n+1–tuples

Skitter and BGP data of 2000-01-01..31

- Skitter accumulated over whole month
- Oregon and local (pinot) BGP added
- 6850 AS
- 36643 sessions (dual nodes)
- 169892 links between sessions
- No stripping
- 7676 sessions in the giant component
- Giant component reaches 34448 sessions
- max.max is 22 AS hops

Reachability table

```
195-145    8h 4.01a 0.88s 34448r 1 129  502  8970 15920 7482 1226 197 21
145-195    9h 4.22a 0.90s 34448r 1   9  442 6137  16108 9446 1833 369 95 8
1225-2914 9h 3.08a 0.89s 34448r 1 724 7413 16747  7832 1424  199  95 10 3
```

- 9 hops on dual graph = 10 AS hops
- 195-145 (SDSC-vBNS) and 145-195
- have different reachability profile

Conclusion

Neighbourhood size:

- mother function is invariant of graph
- shifted and stretched for each node
- shift and stretch = average and standard deviation of hop distance from a node

Likely implications

- there IS a core of the graph
- neighbourhood is small before we reach core
- starts growing as soon as we reach core
- "Gaussian" because minimums are a limiting case of addition

www.caida.org
UCSD/SDSC/CAIDA

Andre Broido
broido@caida.org
kc claffy
kc@caida.org