Route-Flow Fusion: Making traffic measurement useful

Van Jacobson, Haobo Yu, Bruce Mah
{van,haoboy,bmah}@packetdesign.com
Packet Design Inc.

NANOG 35, Los Angeles, CA
October 2005

Mostly an ISP has to operate blindfolded ...

For years we've made do with interface counters that tell how many bytes have crossed a particular link but say nothing about who they came from, where they're going or why they're there.

That's changing: Recent advances in router hardware, decreasing storage cost and increasing processor speed have made it technically feasible for an ISP to routinely look at traffic flows.

Traffic Matrices

Current state of the art combines PoP or AS border NetFlow with prefix/AS info from a BGP passive peering to automatically synthesize traffic matrices (see [Telkamp] for an excellent introduction.)

It's possible to buy off-the-shelf commercial products to do this (e.g., Adlex Flowtracker, Network Signature BENTO).

(part of) a customer-transit outbound traffic matrix

(% of total traffic)	level3	cogent	qwest	wiltel	row total
All	63.9	18.3	16.9	0.5	
ucb	5.5	3.4	1.8	0	10.7
ucla	7.4	1.0	1.3	0.2	9.9
ucsd	5.9	0.5	1.4	0.1	7.9
csunet	4.7	1.4	1.2	0	7.4

Outbound transit traffic demand of top four (of 112) customers. Table was automatically computed from 24 hours of CENIC NetFlow and BGP data.

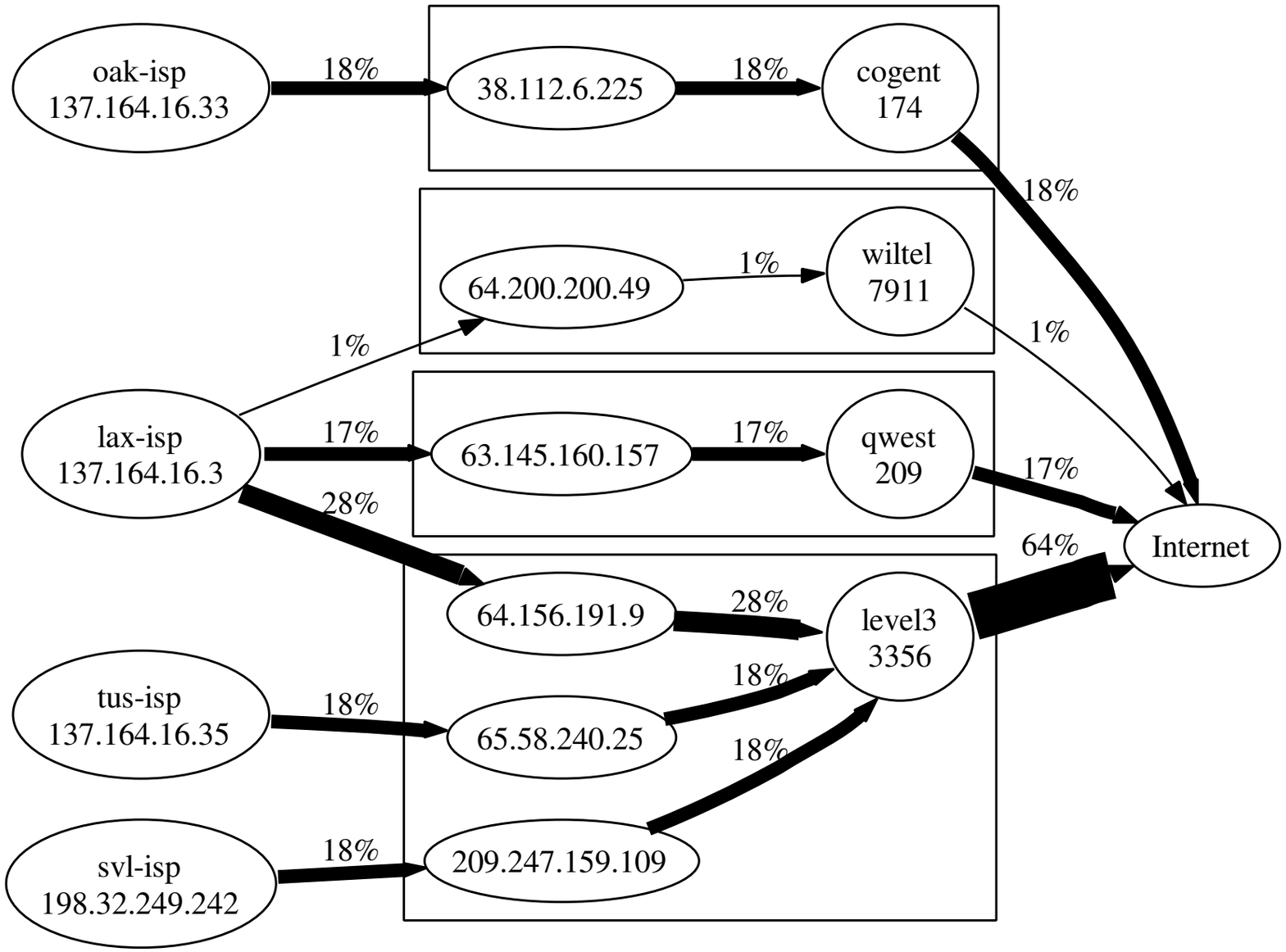
Note: All data shown in this presentation is courtesy of CENIC (www.cenic.org) and used by permission.

A traffic matrix is primarily an engineering tool. It's an $O(n^2)$ analysis that's extremely useful for optimization & capacity planning but:

- It can't answer operational questions like "who filled up this link?" (requires an $O(n^3)$ "A to B via C" analysis).
- It can't answer strategic planning questions like "where does customer traffic go when it leaves here?" (requires an O(n!) path analysis).

Conventional wisdom says this scaling makes most operational & strategic questions too expensive to answer. But conventional wisdom is wrong ...

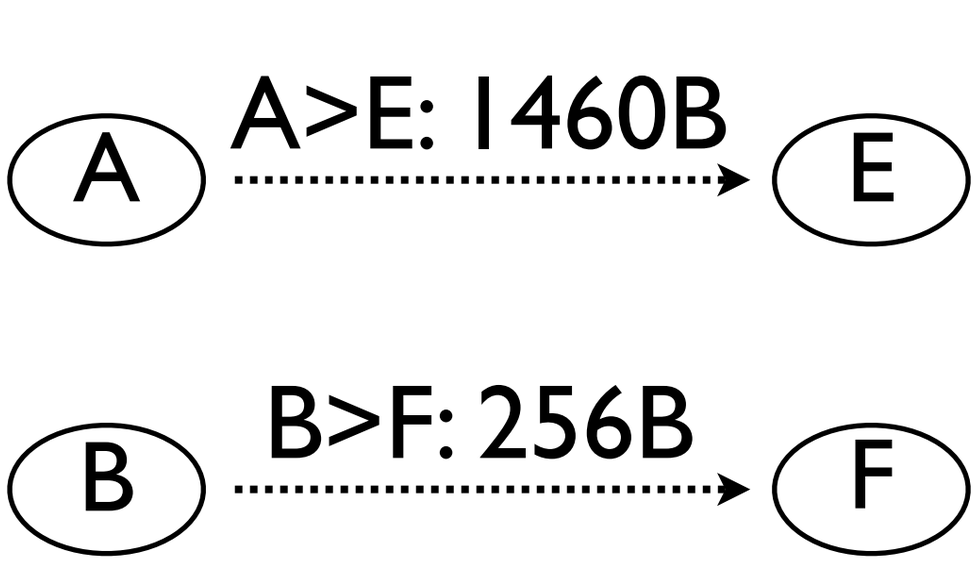
Basis of scaling is that the number of places where data *might go* is huge. But at any particular time the number of places where it actually *does go* is small.

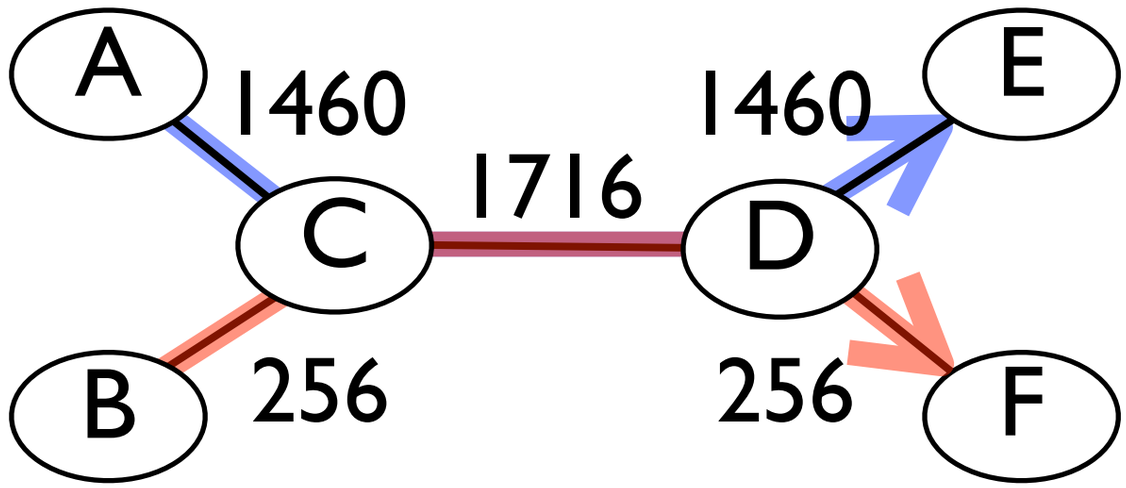


CENIC transit traffic

Observation 1

Routing can be used to convert a point measurement into a path measurement:





Notes

- This is not a new idea (see, for example, [Telkamp, slide 21]).
- Since measurement applies to every path segment, doesn't matter where on path you measure and multiple measurements cross-check each other.
- The technique works fine with partial data (unlike tomographic analysis) so incremental deployment immediately gives useful results.

Observation 2

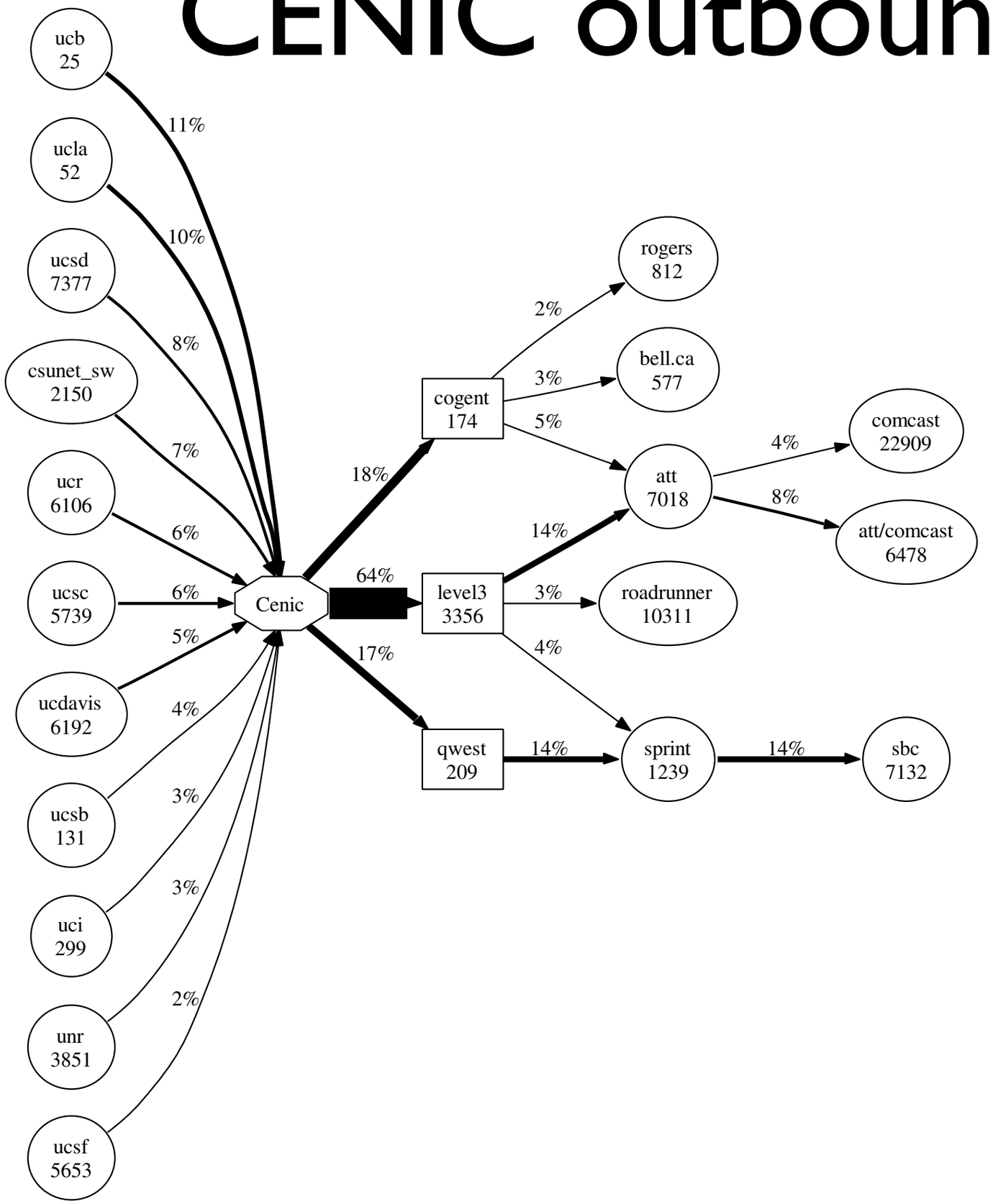
Routing contains all the meta-information needed to classify and aggregate flow information:

- IGP prefix and BGP prefix & last-hop AS# maps source and dest addresses to higher level units (network, organization, etc.).
- BGP first-hop AS# identifies customers, transit providers & peers (BGP community attributes tell you which is which).
- IGP & BGP next-hop show where external entities attach to internal topology.

Route-Flow fusion

- Separately record route (IGP & BGP) and flow (NetFlow, sflow, sniffer, MPLS TE/LDP) information.
- Do lazy, demand-driven data fusion of route and flow information rather than pre-computing the answers to particular questions.
- Result gives you aggregate data rate & traffic volume induced by selected flows on each link they traverse.

CENIC outbound transit traffic

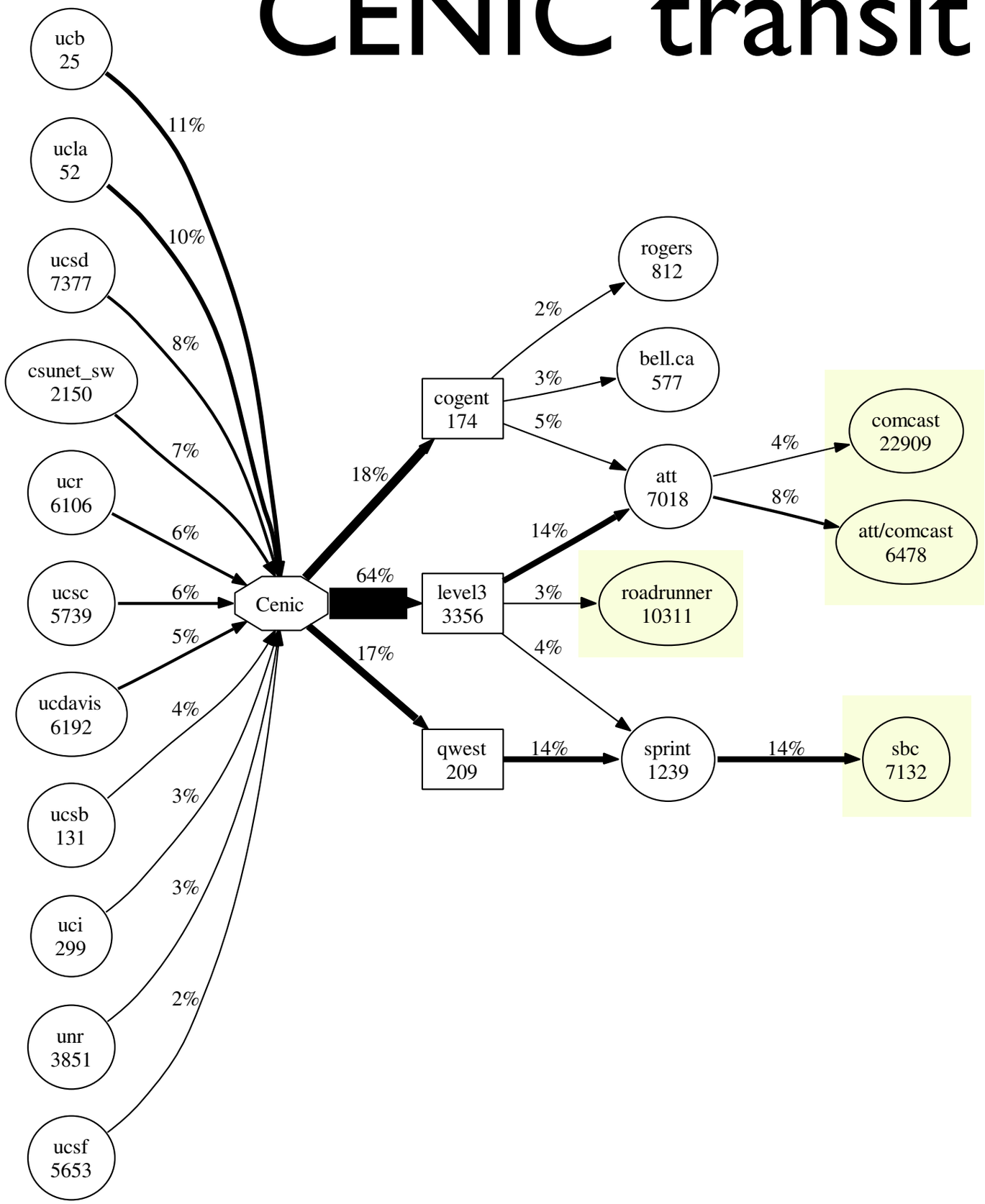


This is the same data as the traffic matrix on slide 4.

Customers are on the left. Transit providers are the rectangles. Edge thickness shows traffic volume. Edges carrying less than 1% of the traffic are pruned.

The **only** manual input needed to create this picture were two BGP community tags (customers and transits).

CENIC transit traffic (cont.)



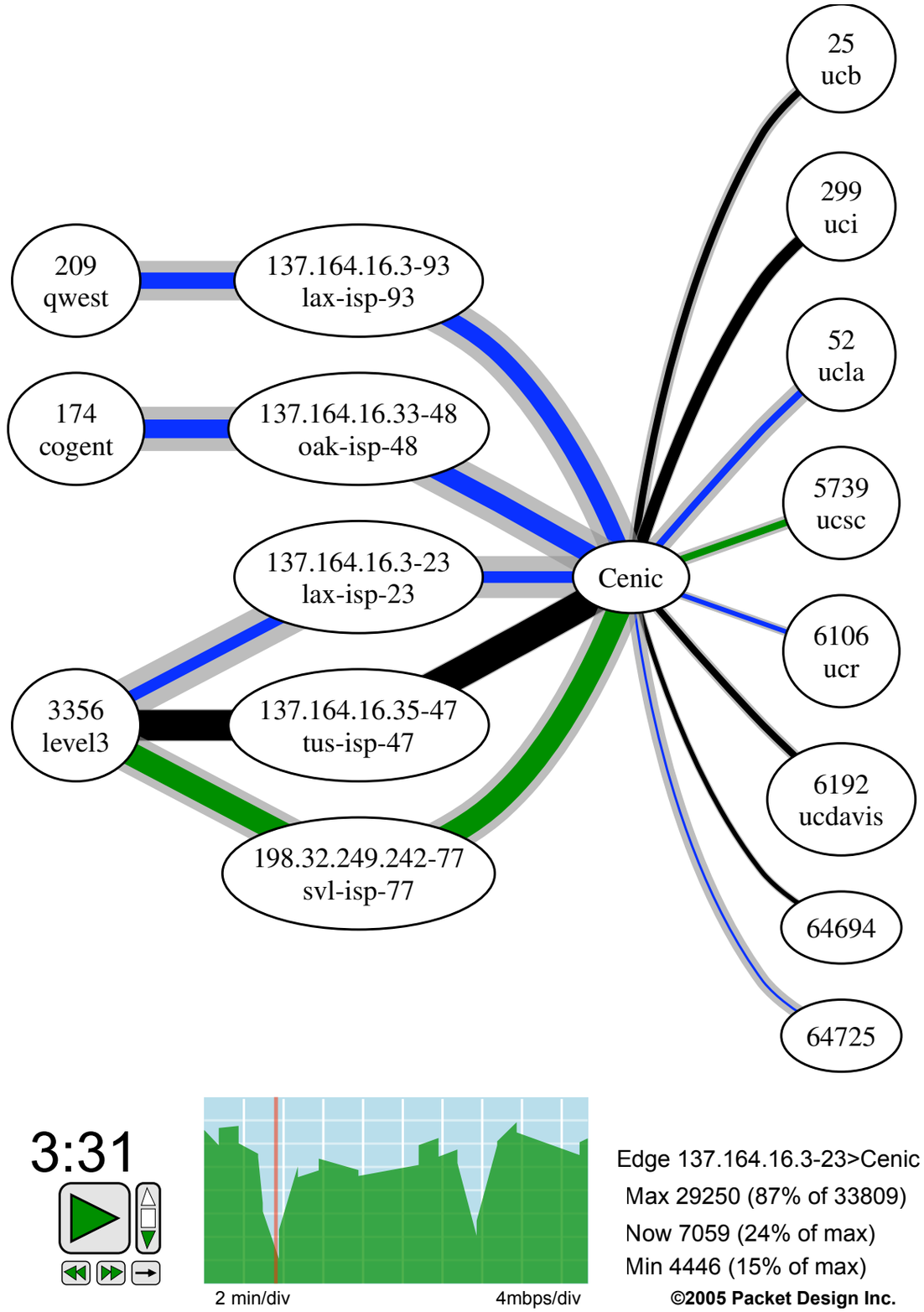
The computational cost of this view of the data is the same as a traffic matrix but this contains more operational and business information.

For example, note that a third of the total traffic goes to residential providers (comcast, roadrunner, sbc) or that 80% of the traffic sent to qwest is destined for sbc.

RFF can also show how traffic behaves over time.

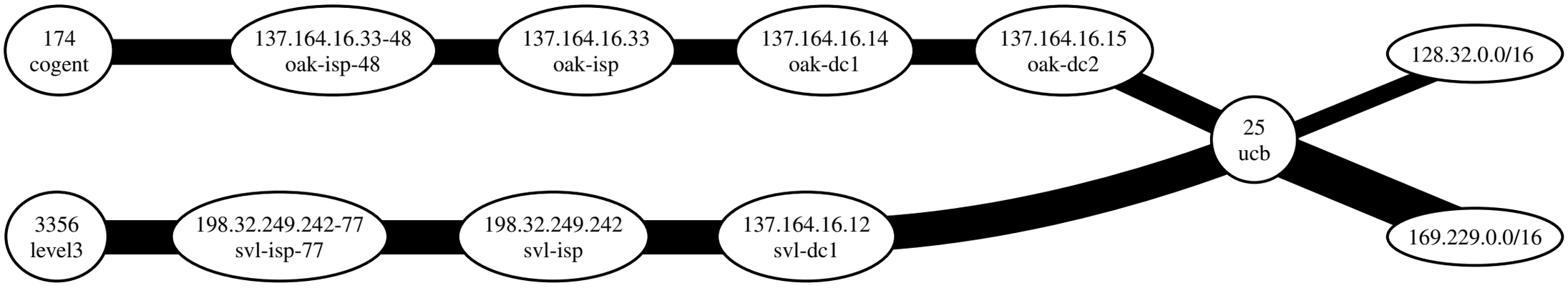
The same algorithm is used but rate vs. time *vectors* are distributed along the path rather than volume *scalars*.

This is a snapshot from an SVG animation of an inbound traffic anomaly that happened at 17:03 PST on February 3rd.

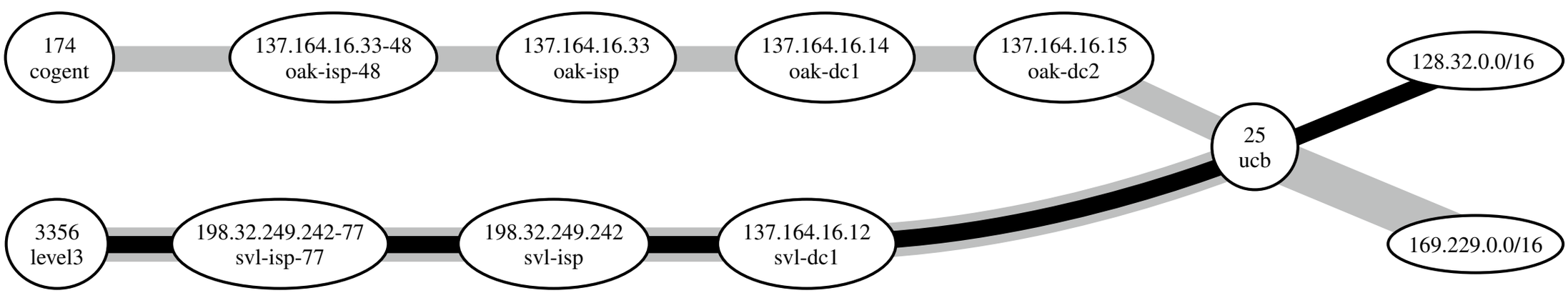


Prefix Anomalies

Inbound transit traffic to UCB seems ok - it's split 60:40 between transits & 75:25 between prefixes:

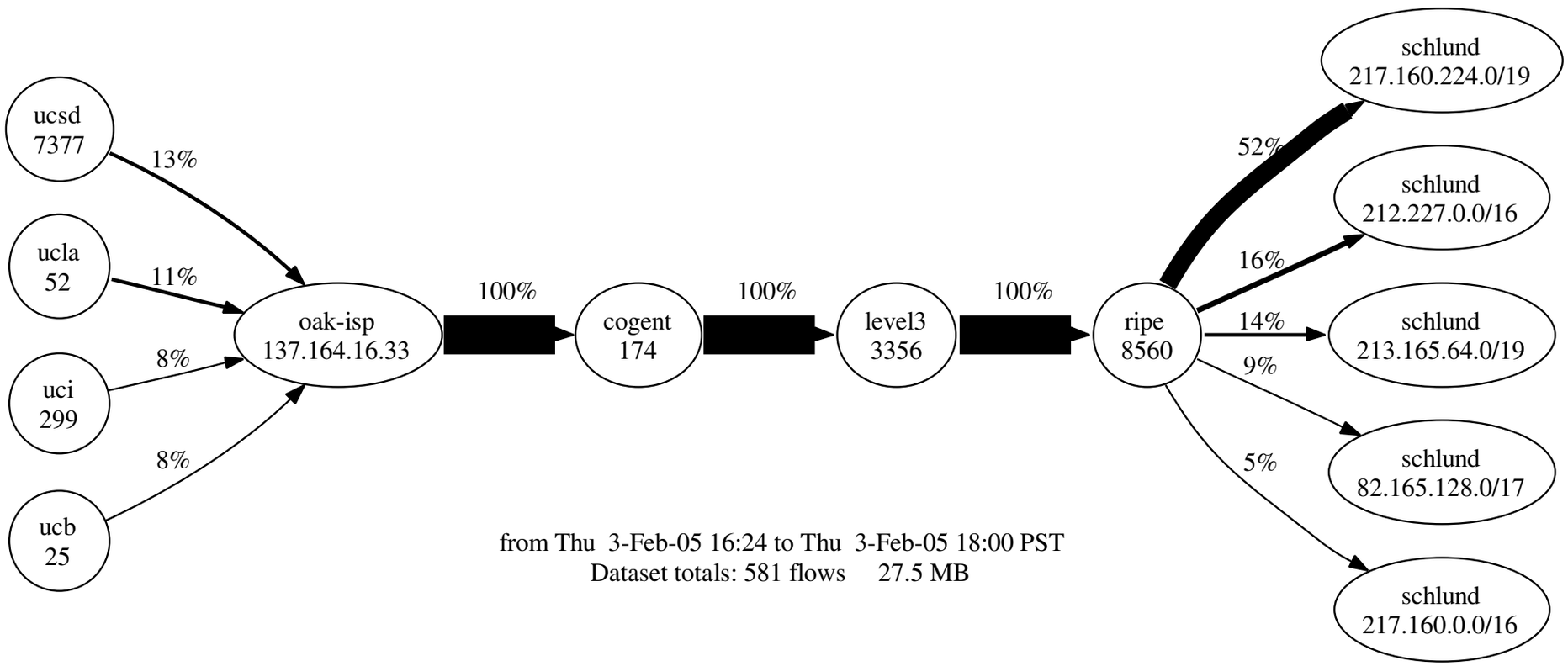


But clicking on 128.32/16 to highlight just its traffic shows it has no inbound from Cogent:



RFF lazy evaluation model easily supports ad-hoc queries.

For example this is all the traffic that goes via either Cogent>Level3 or Level3>Cogent.



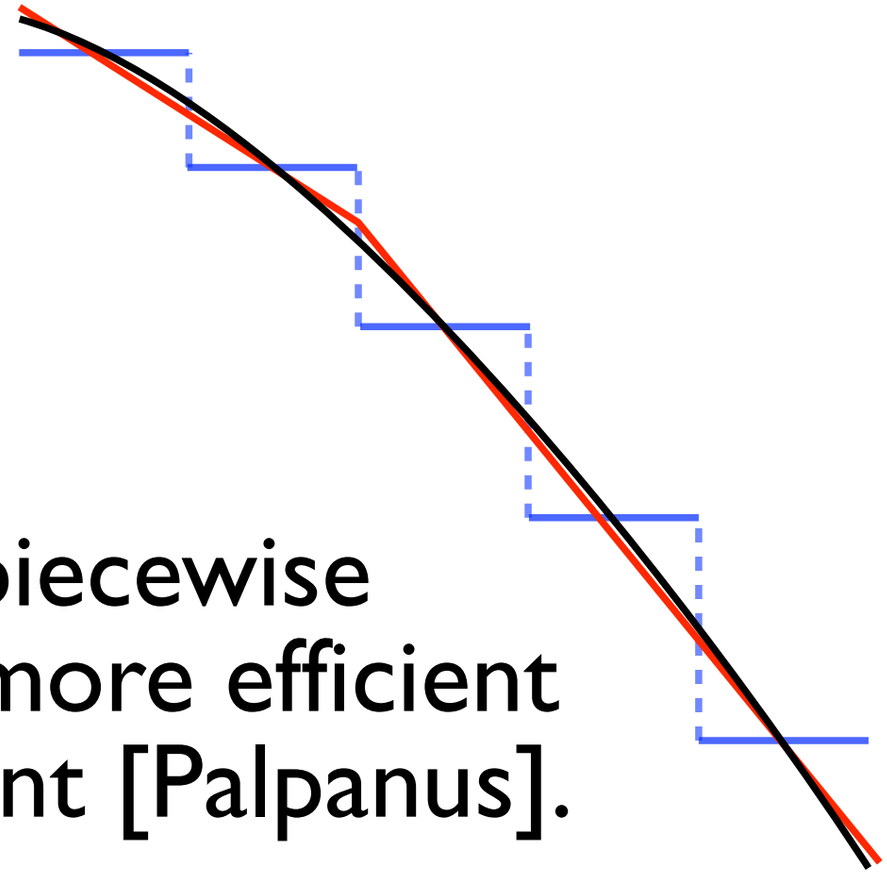
Note: this particular traffic was due to some stale config customization that was actually removed long before Cogent and Level3 de-peered.

Problem: NetFlow data volume is huge.

Typically 1% of link rate or 4GB/hour for a busy gig ethernet.

- Can aggregate by prefix or AS (reduces topological accuracy).
- Can increase active & inactive timeouts (reduces temporal accuracy).
- Can sample data (reduces both topological & temporal accuracy).

Underlying problem is that NetFlow's data model isn't a good match to data's behavior.



For data that contains trends, a piecewise linear approximation can be far more efficient than NetFlow's piecewise constant [Palpanus].

Segmenting (deciding how many lines & where) seems hard but there are fast and relatively simple *n*log*n* algorithms to do it [Keogh].

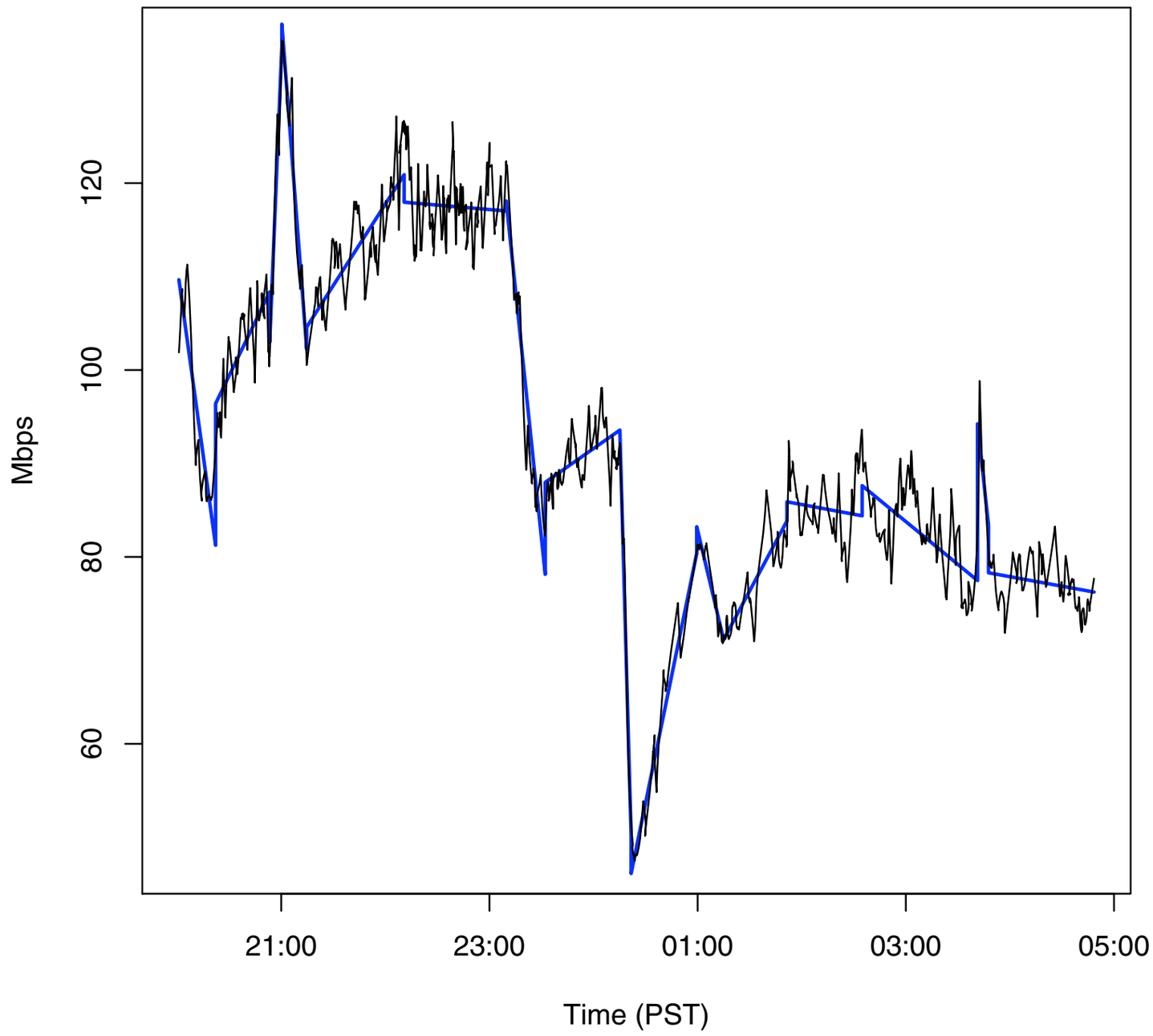
➡We find that least-squares segmentation gives good fidelity at average cost of around 16 bytes per prefix pair per hour.

Linear least-squares segmentation

Black line is 10 hours of NetFlow data (14 million records, 500MB) taken on CENIC PAIX GE link.

Blue line is LLS data segmentation (16 points, 64 bytes).

(Got 10,000,000:1 data compression.)



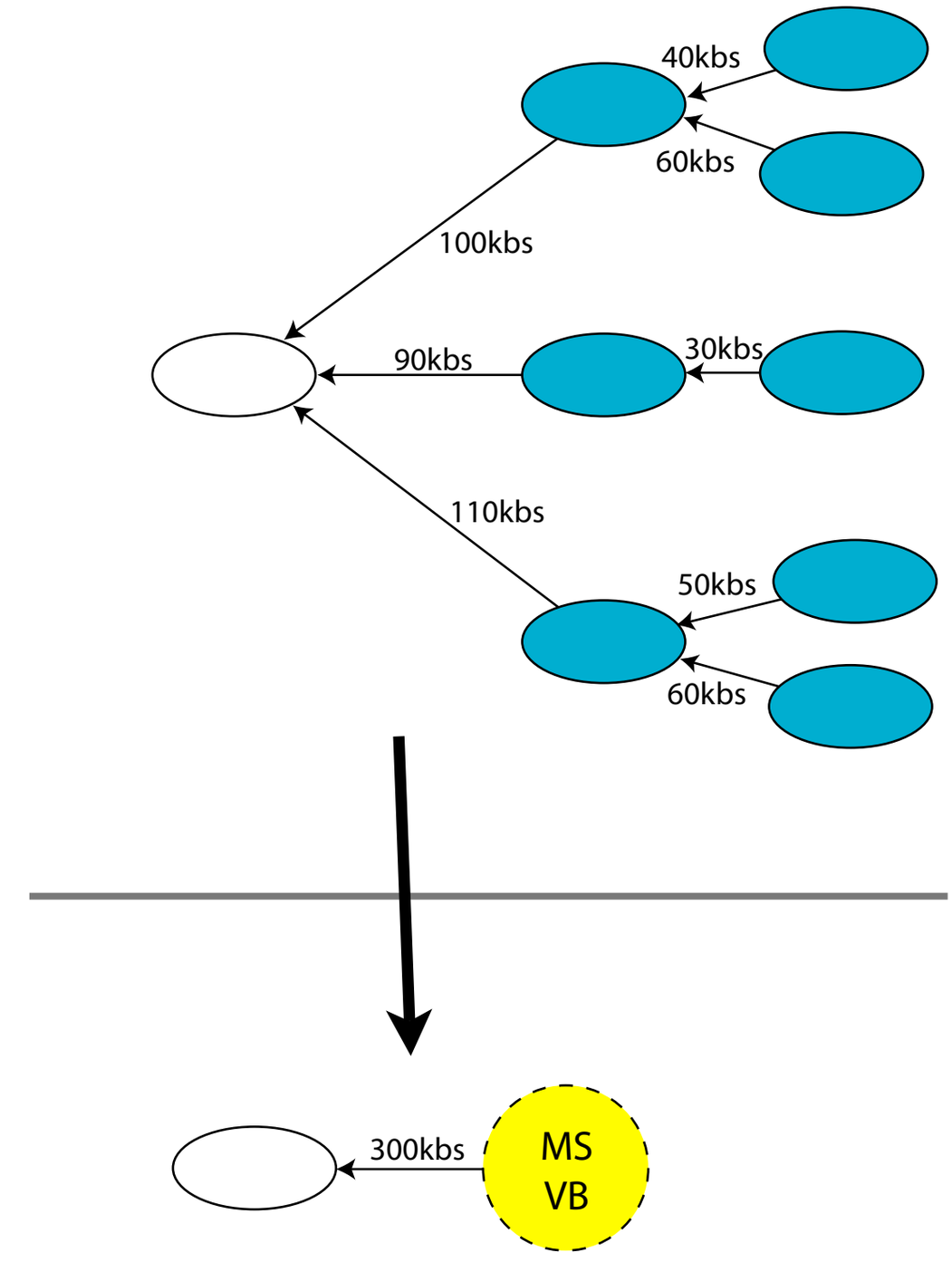
Problem: Number of "flows" (src/dst prefix pairs) is very large (200-500K / hour).

- Most of these (>90%) are various Microsoft viruses sending single packets to random IP addresses looking for other Windows machines to infect.
- Since they make a small (<10%) contribution to the total volume, a general statistical technique like Priority Sampling [Duffield] can be used to ignore them.

But it may be better to preserve more of the spatial & temporal information using *topological aggregation* . . .

Topological aggregation

If, say, the threshold for Microsoft Virus Background traffic were 150kb/s, all the blue nodes could be replaced by a single MSVB virtual node.



This preserves most of the topological information about the virus traffic and allows it to be analyzed as a separate entity.

Data volume problems redux

- Combination of concise flow representation and junk prefix aggregation results in quite tractable flow data volumes — 200 KB/hr or ~1GB/year per monitored GE link.
- Linear least-squares segmentation allows further reduction since old data can be converted to a lower fidelity representation that preserves averages and trends. [Palpanas]

Conclusion

- Lazy, demand-driven computation of flow-to-path traffic loading allows detailed view of almost all traffic behavior for about the same cost as computing a traffic matrix.
- A system to do this can completely self configure using the existing routing info.
- A year of complete flow data for a medium-sized ISP (200 full GE NetFlow feeds) fits on one disk.

We can build tools to do this today. Network operators & planners don't have to fly blind anymore.

Acknowledgments

- We're grateful to Randy Bush for (long ago) suggesting that tools to do this would be useful. We're sorry it took five years to implement his suggestion.
- We're deeply in debt to Darrell Newcomb of CENIC for data, excellent advice, cogent suggestions, willingness to test buggy code, ...

Bibliography

[Palpanas] Palpanas, T.; Vlachos, M.; Keogh, E.; Gunopulos, D. & Truppel, W. (2004),'Online Amnesic Approximation of Streaming Time Series', 20th International Conference on Data Engineering, IEEE.

[Telkamp] Telkamp, Thomas (2004), ‘Best Practices for Determining the Traffic Matrix in IP Networks’, NANOG 34, Seattle, WA, May 15, 2004.

[Keogh] Keogh, E.; Chu, S.; Hart, D. & Pazzani, M. (2001), An Online Algorithm for Segmenting Time Series, in 'Proceedings of IEEE International Conference on Data Mining', pp. 289-296.

[Duffield] Duffield, N.; Lund, C. & Thorup, M. (2005),'Sampling to estimate arbitrary subset sums'.

Measurement Notes

All but one of the visualizations in this presentation were generated from 24 hours of NetFlow data taken on February 3, 2005 on CENIC's six ISP interconnect routers. Slide 18 was generated from 12 hours of data taken June 3, 2004 on CENIC's PAIX router.

(see the "ISP Interconnects" & "Peering - Palo Alto" pages at intermapper.engineering.cenic.org for the topology and other details.)

Although CENIC accumulates customer, peer & transit NetFlow, presenter laziness caused us to analyze only the transit data.

IGP & BGP routes were extracted from a Route Explorer attached to CENIC's network.