Right-Sizing DICOM Routing for Mammography

A measured approach to capacity planning for XyDromatics Router and SynthIQ

A Synthology Healthcare Solutions Group technical whitepaper Version 1.0 · 2026-06-03

XyDromatics Router and SynthIQ are non-device software under FD&C Act §520(o)(1)(D).

Executive summary

Sizing a DICOM router is usually guesswork: a vendor quotes "enterprise-grade hardware," a hospital over-buys storage it never uses, and nobody can say what actually happens when four mammography units start pushing 3D tomosynthesis studies at once.

We took a different approach. On a dedicated performance rig — an isolated DICOM generator driving a system-under-test across three host sizes — we measured exactly what governs XyDromatics Router capacity. The findings are clear, and a few of them are counter-intuitive:

1. **Routine imaging is CPU-bound, and scales with cores.** ~700 study-instances per second on 4 vCPU, ~1,500 on 8 — roughly linear. Disk speed is irrelevant.
2. **You do not need high-IOPS storage.** Quadrupling disk IOPS (P15 → P30) produced *zero* throughput gain. The disk was never the bottleneck.
3. **3D mammography has a bottleneck that *moves*.** On a small host, DBT is memory-bound — it will exhaust RAM. Give it enough memory (64 GB) and the limit shifts to CPU. A DBT site must size memory *and* CPU together.
4. **High memory use under load is by design, not distress.** The router deliberately uses up to ~75% of host RAM as a managed ceiling under load, and recovers fully afterward. Knowing this prevents false alarms.
5. **Capacity grows by adding nodes, not bigger boxes.** Past a single node's comfortable band, a SynthIQ pool load-balances across routers and places large studies on nodes with memory to spare.

The result is a sizing matrix you can buy against with confidence, and a scale-out trigger that tells you exactly when to add the next node.

---

1. Why router sizing is misunderstood

Two myths drive most over-spending.

Myth 1: "Fast storage makes a fast router." Imaging is large, so the instinct is to buy the fastest disk. But a routing workload is dominated by network receive, in-memory parsing, and CPU-side transformation — not random disk I/O. In our testing the router used barely a quarter of even a modest disk's IOPS budget while CPU was pinned. **Premium SSD (P15-class) is sufficient.**

Myth 2: "Capacity is one number." A router's ceiling depends entirely on the *shape* of the traffic. A stream of small CT and CR images stresses completely different resources than a handful of 561 MB tomosynthesis volumes. Quoting "X studies/hour" without specifying the modality mix is meaningless. Real sizing needs a model, not a number.

2. Methodology

Credibility here comes from rigor, so the method matters.

- **Dedicated rig, isolated roles.** One VM generated DICOM load; a separate VM ran the router under test; they communicated over a private network. Nothing else competed for resources.
- **Three host sizes.** The system-under-test was measured at **4 vCPU / 16 GB**, **4 vCPU / 32 GB**, and **8 vCPU / 64 GB**, so we could see how the limits move with the box, not just read one data point.
- **Realistic send patterns.** Routine load used *study-mode* — many images per DICOM association, the way real modalities actually transmit — rather than the pessimistic one-association-per-image worst case.
- **Dual-host attribution.** Every high-load run sampled CPU on **both** the generator and the router. This is the discipline that makes the conclusions trustworthy: when 3D mammography began failing at high concurrency, the naïve reading was "the router is overloaded." The data said otherwise on the first pass — and then, on a corrected reading, confirmed it. (More below; it's a good illustration of why you measure instead of assume.)

3. The central finding: a bottleneck that moves

The headline result is a **two-regime model** for capacity:

Workload	On a small host (≤32 GB)	On a large host (64 GB)
Routine (CT/CR/DX/US/MR)	**CPU-bound**	CPU-bound
3D DBT / tomosynthesis	**Memory-bound**	**CPU-bound** (memory becomes ample)

For everyday imaging, the lever is always **vCPU**. For 3D mammography, the lever is **RAM until you have enough of it — then vCPU.** This is the single most important idea in the document: a tomosynthesis-heavy site that buys cores without memory will hit a wall, and one that buys memory without cores will hit a different wall. They scale together.

4. Routine imaging: CPU-bound and linear

Measured with realistic study-mode load:

Host	Sustained throughput	Success rate
4 vCPU / 16 GB	~700 instances/sec	100%
8 vCPU / 64 GB	~1,500 instances/sec	100%

Throughput scaled ~2.1× for a 2× increase in cores — close to linear. Two practical notes emerged:

- **The router's inbound-connection limit governs before the CPU saturates.** Peak throughput was reached with the cores only ~67% busy; a configurable concurrent-association cap bounds the rate first. The ~1,500/sec figure is the out-of-the-box number; sites that need more can raise the cap and trade into a higher, CPU-bound ceiling.
- **More isn't always faster.** Throughput peaks at moderate concurrency and *degrades* if a client opens far more simultaneous connections than the cap — excess connections simply queue. This is a client-tuning consideration, not a hardware limit.

5. 3D mammography: where the bottleneck moves

A single 3D tomosynthesis volume is ~561 MB to over 1 GB — one such study is larger than thousands of routine images combined. This is the sizing-critical case.

On a small host, DBT is memory-bound. Each concurrent volume in flight costs memory: roughly ~5 GB for the first concurrent study (establishing the large buffers) and ~2 GB for each additional concurrent volume, on top of a ~2.6 GB baseline. On a 16 GB host without protection, two concurrent volumes exhaust memory. A 32 GB host handles ~4–6 concurrent volumes cleanly and **sheds the excess gracefully** rather than crashing — the router's built-in memory ceiling converting an out-of-memory crash into a managed back-off.

On a 64 GB host, memory stops being the limit — and CPU takes over. This was the key measured result:

Concurrent DBT volumes	Result	Router CPU (8 vCPU)
4	100% success	light
8	100% success	busy
12	100% (elevated latency)	~2× saturated
16	partial — begins shedding	~2.5× saturated

Crucially, **memory never approached the host's limit at any concurrency** — it plateaued comfortably below the ceiling even at the shedding point. Doubling RAM took 3D mammography *out* of the memory regime entirely; from there, the receive → parse → encrypt → forward pipeline saturated the processors.

The dual-host sampling is what made this conclusive: at the shedding point the *generator* was idle while the *router* was 2.5× CPU-saturated. The limit was genuinely the router's processors, not a test artifact — and not memory.

Bottom line for 3D mammography on 8 vCPU / 64 GB: a comfortable band of ~8 concurrent volumes — roughly one per mammography unit sending at once — with graceful degradation beyond.

6. Operator FAQ: "why is the router using most of my RAM?"

Under 3D load, the router's memory footprint rises toward ~75% of host RAM — even at light concurrency. **This is intentional.** The runtime's server garbage collector trades memory for throughput: it allows the heap to grow to a hard ceiling before reclaiming, so the footprint reflects allocation high-water plus not-yet-collected scratch memory, not the live working set (which is far smaller). It recovers fully after a burst.

The practical guidance: **high memory use is healthy as long as the node isn't shedding**, and you should **not co-locate other memory-hungry services** on a mammography router — it will use the RAM you provision for it, by design.

---

7. Growing capacity: scale out, don't scale up

A single router scales up to its limits — cores for routine volume, memory-then- cores for 3D mammography. Past that, the answer is not a bigger box. It is a **SynthIQ pool.**

SynthIQ load-balances DICOM traffic across multiple routers and routes **weight-aware**: it predicts a study's size from the modality and SOP class of its very first image, and steers large tomosynthesis studies to backends that have the memory headroom to hold them — keeping the pool balanced and protecting every node from the out-of-memory failure mode. Capacity grows by adding nodes, linearly, with no forklift upgrade.

You should add a pool node when, sustained: routine throughput approaches the node's ceiling; *or* concurrent 3D studies approach the node's comfortable band (CPU climbing past ~1.5× core count, latencies stretching); *or* the node begins to shed gracefully under DBT load. When two of these recur, scale out.

---

8. Recommended configurations

Site profile	Routine load	3D mammography	Recommended (per router)
Small	low–moderate	none	4 vCPU / 16 GB / standard SSD
Medium	moderate–high	2D FFDM + occasional DBT (≤4 concurrent)	8 vCPU / 32 GB
Large (mammography)	high	4–8 units, 2D + 3D DBT	**8 vCPU / 64 GB**

- **32 GB is the floor** for 561 MB-class tomosynthesis (≈4 concurrent volumes).
- **64 GB / 8 vCPU** is the recommended mammography spec — clearing the memory regime and providing the cores that then become the limit.
- **Standard Premium SSD** throughout; high-IOPS storage is unnecessary.
- Beyond a single node's band, **add a SynthIQ pool node.**

---

9. Conclusion

Capacity planning for DICOM routing does not have to be guesswork. By measuring across host sizes with isolated, attributable load, we can state plainly what drives capacity: **cores for routine imaging, memory-then-cores for 3D mammography, and never the disk.** Customers can buy the right box — not the most expensive one — and know in advance exactly when to add the next node.

That is the difference between a specification and a measurement. This whitepaper is the measurement.

XyDromatics Router and SynthIQ are non-device software under FD&C Act §520(o)(1)(D). Figures in this document were measured on a dedicated performance rig and are provided for planning guidance; production capacity depends on deployment specifics including modality mix, network, and configuration.