

Quality-of-Service: An End System Perspective

Lars Eggert

MIT Communications Futures Program – Workshop on
Internet Congestion Management, QoS, and Interconnection

Cambridge, MA, USA
October 21-22, 2008



Nokia Research Center

NOKIA

Different shades of QoS

"My application..."

...needs a private circuit, period."

i.e., a specific required bit rate, delay, jitter, packet drop or bit error rate

...is a bit different from other apps."

i.e., would like some sort of preferential or special treatment



Who needs virtual circuits, anyway?

session-based QoS...

- is meaningless to users (flowspec?)

- is difficult to manage for ISPs (esp. inter-ISP)

- limits multiplexing + needs boxes = more \$/bit

- incentives aren't aligned (otherwise we'd have it)

- administers scarcity vs. eliminating scarcity

claim: most of the “killer applications” for QoS don't need it

(or actually: *soon* – or at least *eventually* – won't need it)

QoS can be some what helpful for cutting edge apps for a limited time

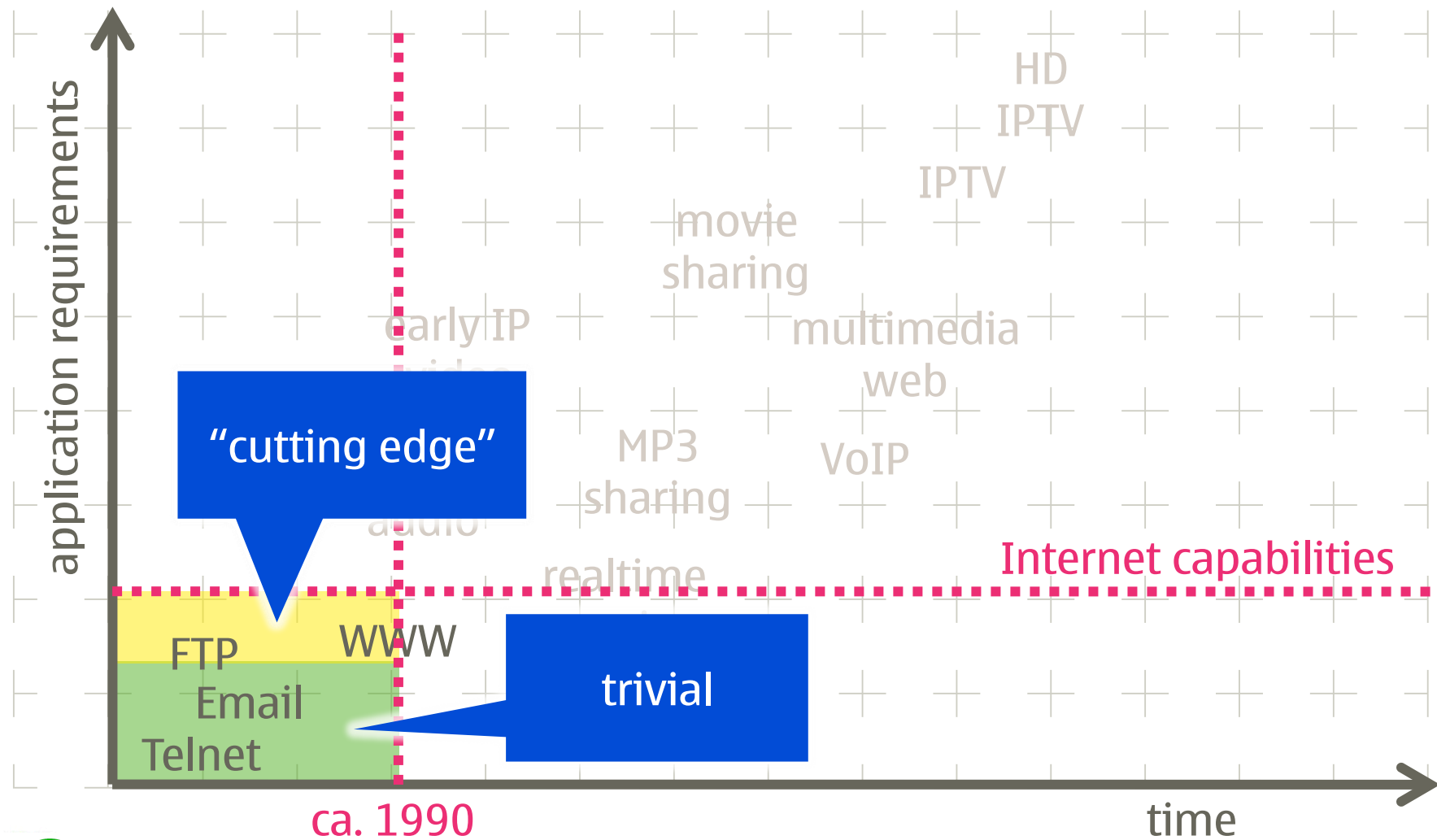
successful apps seem to be able to do without

“My
application...”

...needs a
private circuit,
period.”



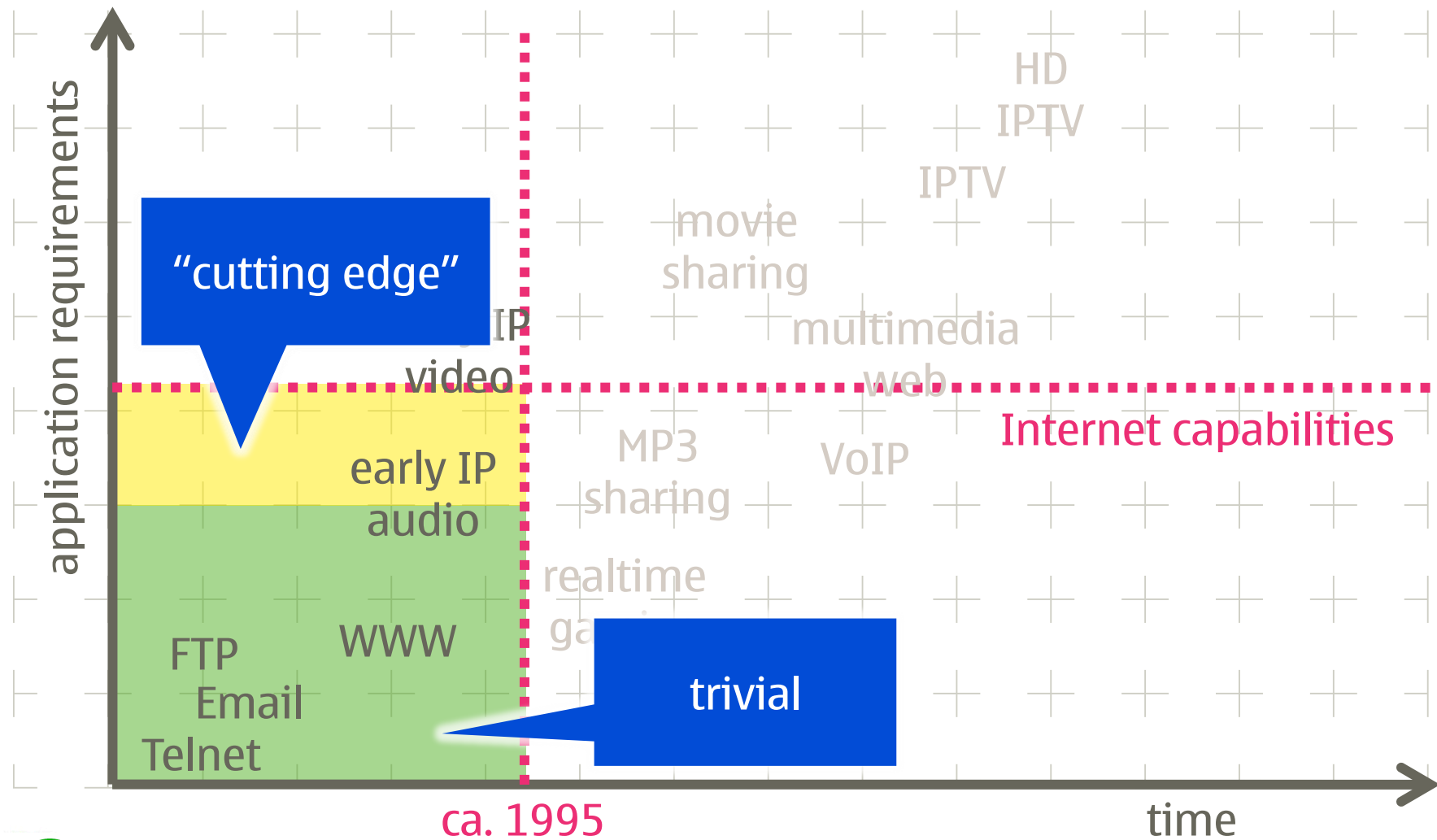
The cutting edge is a moving target – ca. 1990



Nokia Research Center

NOKIA

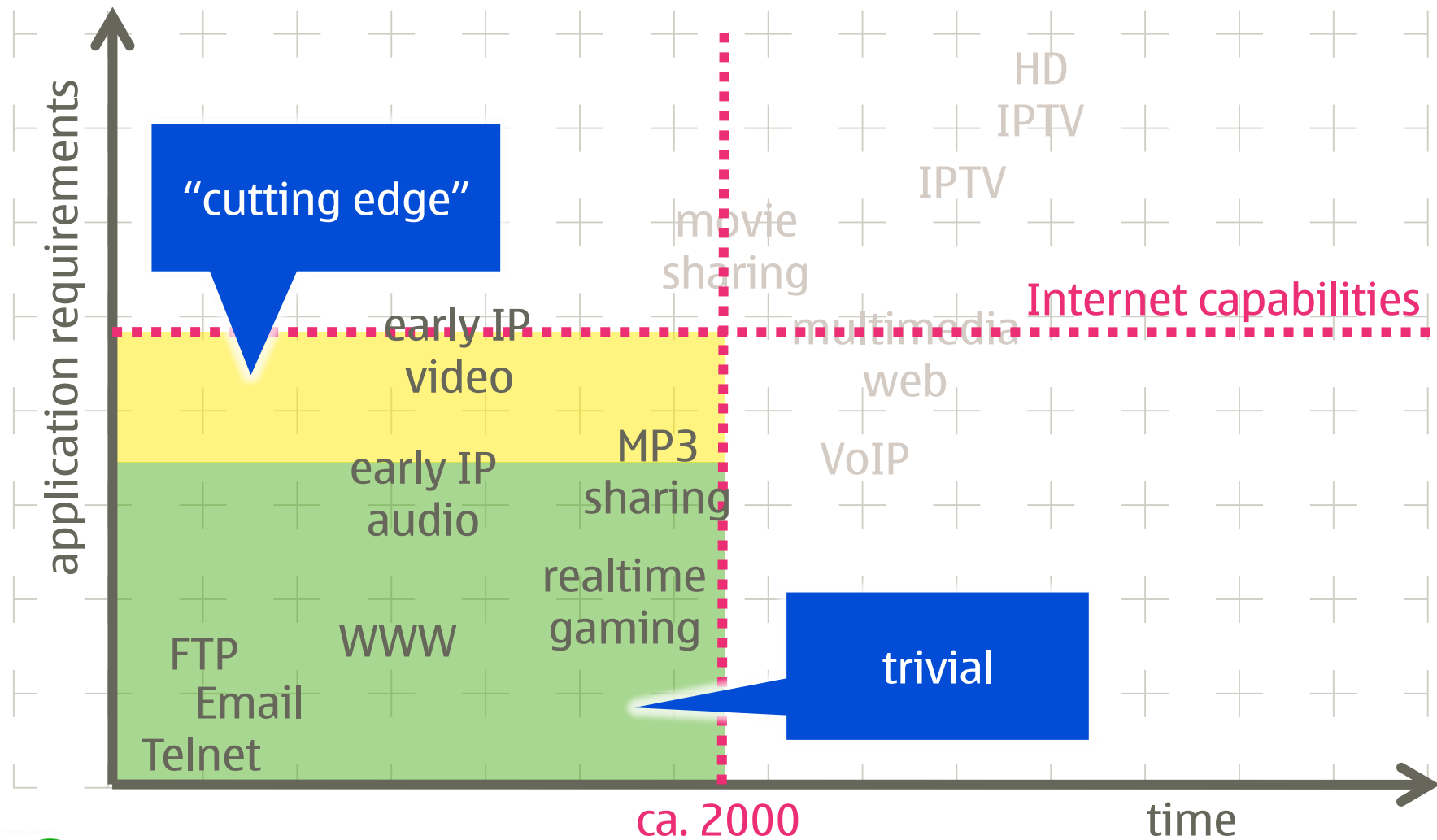
The cutting edge is a moving target – ca. 1995



Nokia Research Center

NOKIA

The cutting edge is a moving target – ca. 2000



Nokia Research Center

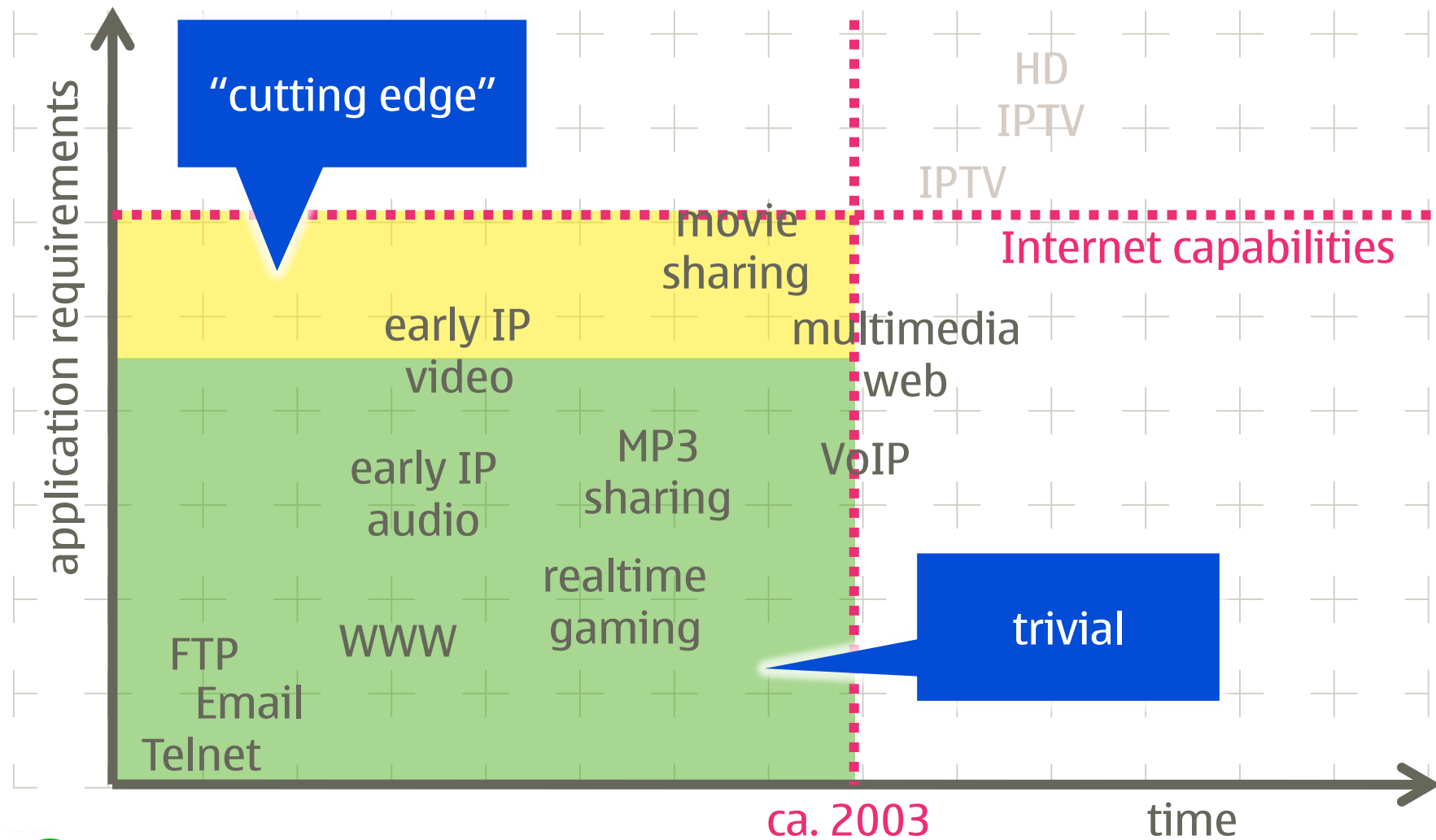
2008-10-22

Lars Eggert | © Nokia 2008

NOKIA

5

The cutting edge is a moving target – ca. 2003



Nokia Research Center

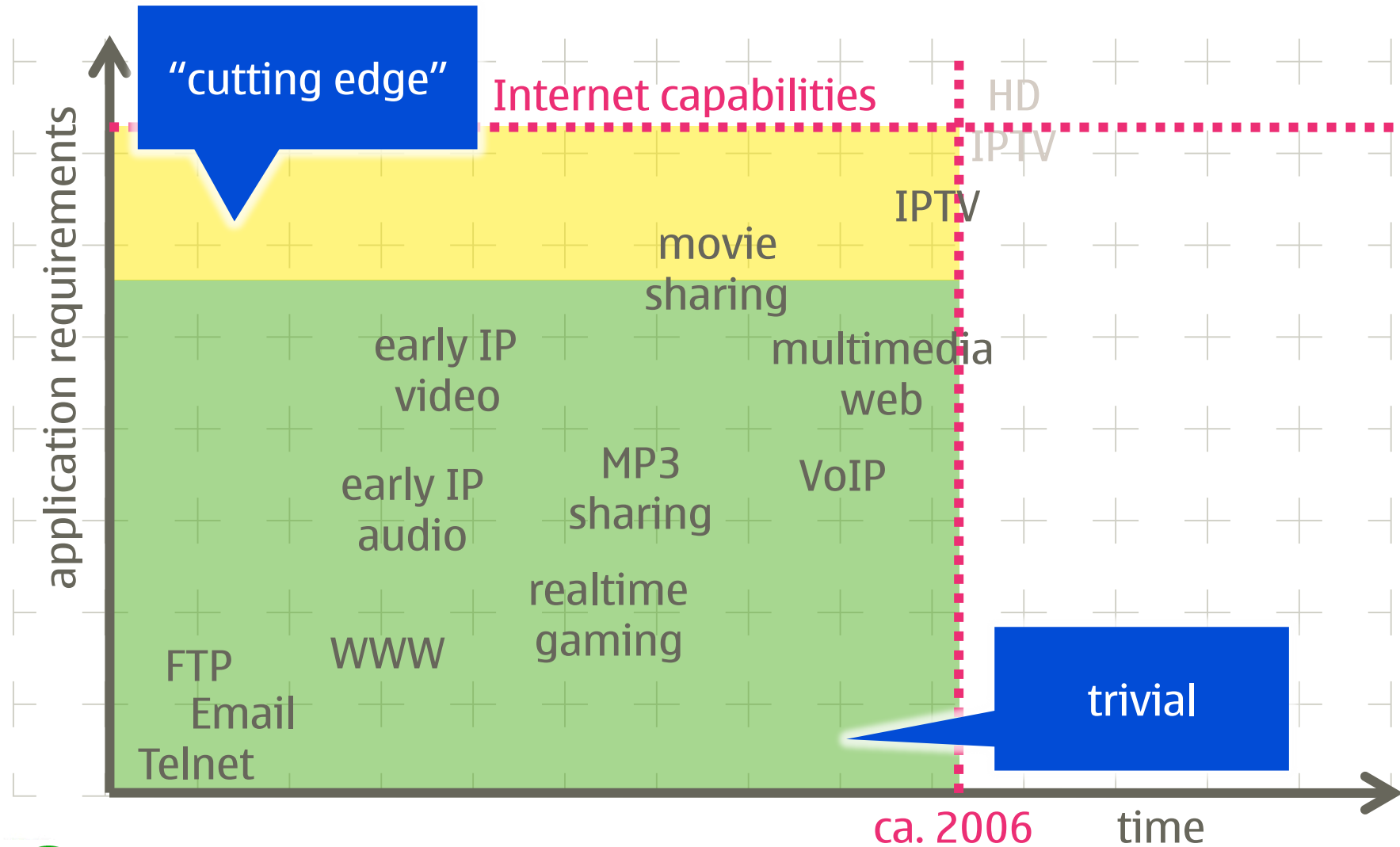
2008-10-22

Lars Eggert | © Nokia 2008

NOKIA

6

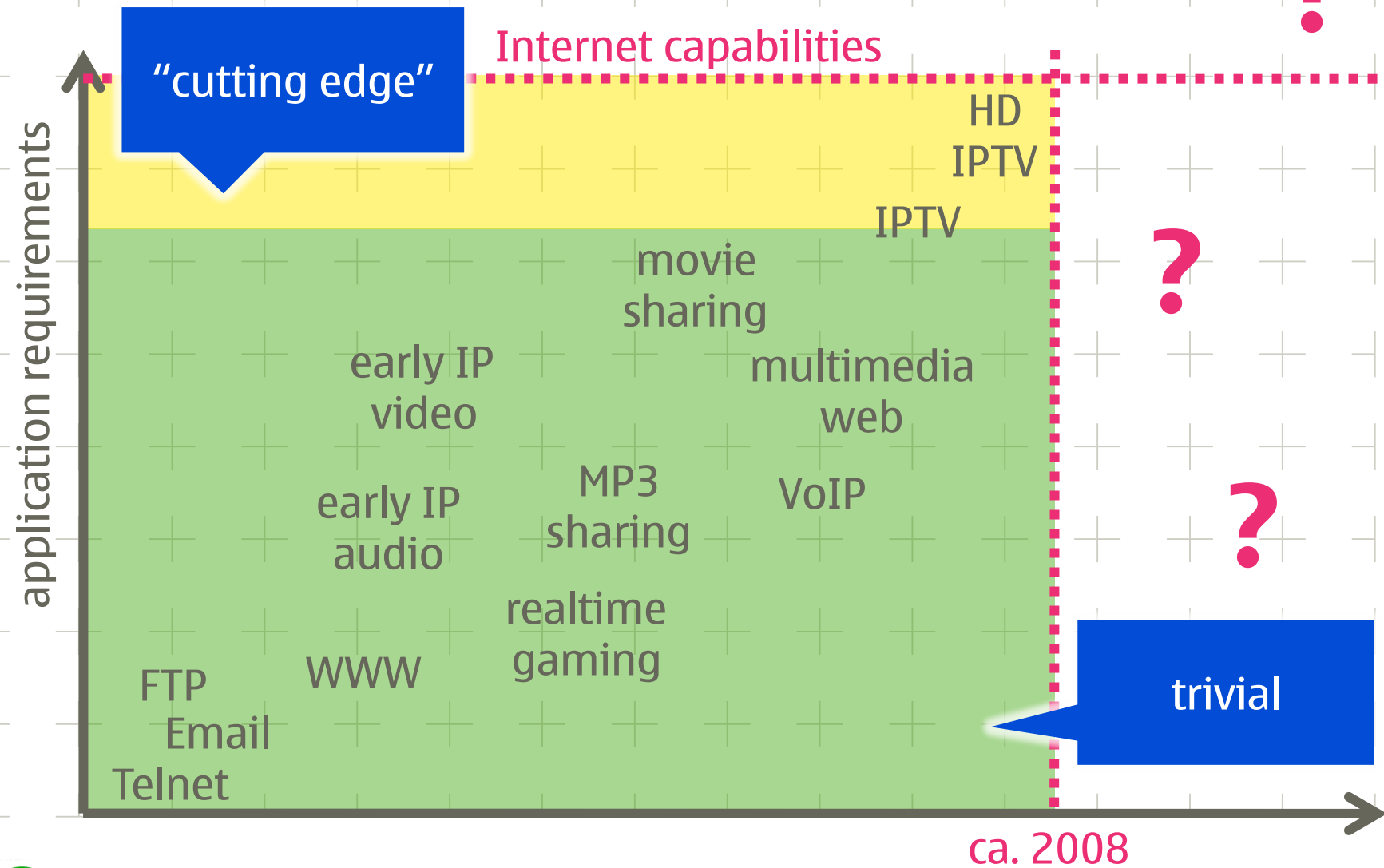
The cutting edge is a moving target – ca. 2005



Nokia Research Center

NOKIA

The cutting edge is a moving target – ca. 2008



Nokia Research Center

NOKIA

Dead-obvious observations

the cutting edge moves

apps that were impossible become feasible, then commonplace
(watch what your bleeding edge users are doing!)

yesterday's cutting edge becomes today's bread-and-butter

adequately supported by the residual capacity of regular best-effort
Internet paths without special QoS (or with app-level approaches)

QoS is the perpetual panacea for the demanding Internet app *du jour*

without ever managing to really get deployed

c.f. multicast, IPv6, IPsec, MobileIP, ...



So what?

session-based QoS is a red herring

what's an "app that's a bit different" to do?

straightforward: app-level approaches

or end-system-level approaches (however: deployment)

pain → change → gain incentives align

plenty of examples

coding, prefetching, caching, parallel sessions

detect and limit/control self-interference

"My
application..."

...is a bit
different from
other apps."



So, nothing left to be done on the network side?

well, no – there are many things that'd benefit apps
(that aren't session-based QoS)

many all-time favorites, including

- turn on ECN

- use some AQM (avoid FIFO drop-tail)

- use reasonable buffer sizes (more $\neq$ better)

- consider “lower effort” PDB (RFC3662) & give incentives for use
- don't interfere with the apps ability to probe the path

- provide CE equipment that doesn't limit evolvability



Finally: a plug for related IETF work



TANA – Techniques for Advanced Networking Applications

1. an end-to-end congestion control algorithm that approximates less-than-best-effort “scavenger” service
2. a document discussing the tradeoffs surrounding the use of some or many parallel connections

next meeting (1st WG or 2nd BOF) to happen at IETF-73 (Nov 16-21, MSP)

Adding Acknowledgement Congestion Control to TCP

draft-floyd-tcpm-ackcc-04.txt proposed to the TCPM WG

(we’re always all ears for other neat ideas – talk to us!)



Nokia Research Center

NOKIA