Status of the "H.325 Project"

Status of the "ITU-T H.325 Project"

Istvan Sebestyen, Siemens AG

Dave Lindbergh, Polycom Inc.

May 2006

What is H.325??

- 3rd Generation of multimedia system and terminal standards from ITU-T Study Group 16
  - 1st Generation: H.320, H.321, H.322 (~1988-1992)
  - 2nd Generation: H.310, H.323, H.324 (1996)
  - SIP work started also in 1996
- Experience, Advancing Technology, the NGN
- Opportunity for a new generation:
  - Better integrated features
  - Greater extensibility
  - More flexible growth path for the future

Status of the "H.325 Project"

- Still in early conceptual phase
- Initiated by management of ITU-T Study Group 16
  - Chair, Vice Chair, some Rapporteurs (all in their ITU roles)

Why is ITU doing this?

- *Maybe we're not* – still in "exploration" phase
- Intention is to explore alternatives to fully realize the potential of NGN
  - Generate discussion, action as needed
- Standards work is "bottom-up"
  - Driven by contributions (means from *you*)
  - Too early to say if this will take off
- First "requirements" must be agreed
  - 2005-06: Call for Requirements issued

Standardized services & applications

Might include (but not limited to):

- VoIP
- Video Telephony
- Video Conferencing
- Video on Demand (VoD) (?)
- Data Conferencing
- Telemedicine (?)
- Distant learning (?)
- DRM (?)

Non-standardized services & applications

- Offered / invented by 3rd parties
  - Harness creativity of users, entrepreneurs
- Interactive gaming
- Home device control
- E-Business systems, electronic payments
- Calendars, scheduling, …
- *Who knows what will be invented???*
  - Applications must be easily **integrated** in H.325
    - e.g. through appropriate APIs

H.325 design principles

- Incremental development approach
  - Broad architecture, narrow core
  - Allow step-by-step development
  - Smaller initial investments
- Primarily intended for use on the NGN
- But can also run in pre-NGN environments
  - Simple interface stubs when NGN not present
  - Smooth the transition to the NGN
  - Network agnostic approach is traditional in SG16

Incremental development of H.325

- Broad architecture
  - Clean, flexible, extensible
- Initial small "core" subset of architecture
  - Simple point-to-point audio and video calls
  - Simple streaming mode
  - Simple broadcast mode
- Expand the core over time gradually as market requirements and the NGN network evolves

H.325 Growth Path

1. Simple core – drive adoption
2. Support all features in previous systems
   - POTS, H.32x series, SIP, etc
   - Interop with "legacy" systems (probably via GWs)
3. Support *anticipated* new functions & features
4. Allow for *unanticipated* future requirements

- Keep it simple
  - e.g. no "Babel" of required codecs and protocols
- Key idea: *Use flexibility to create simplicity*

Desired characteristics for H.325

- Exploit capabilities and protocols of the NGN
  - Avoid duplication of NGN functions
- Exploit advancing technology
  - Terminals are becoming more capable
  - Terminals can be more flexible
  - We may be able to “buy” simplicity

Build on NGN functionality

- Leverage "Core" NGN Release 1 services
  - Presence
  - Instant Messaging
  - MMS
- On non-NGN networks, implement as stubs
  - Emulate some or all of NGN functionality in H.325
  - Only when not on NGN

Design alternatives for "H.325"

- Alt. 1 – Streamlined traditional architecture
  - Like H.323 or SIP
  - Learn lessons, avoid past mistakes
- Alt. 2 – Software defined client architecture
  - Extensible “downloadable software plug-in” design
  - New approach - more flexibility, more risk
- Alt. 3 – Some combination of both

Flexibility thru Simplicity

- Goals:
  - Reduce “legacy support” burden
  - Reduce mode permutations
  - Simplify implementation & interoperability testing
  - Obtain flexibility for unanticipated requirements
- Models:
  - Software defined radio (SDR)
  - H.245 Generic Capabilities
  - UUIDs / GUIDs / OIDs
  - Java / Jini

Too many media codecs

- Existing multimedia terminals include a large and ever-increasing number of media codecs
  - G.711, G.723.1, G.728, G.729, G.722, G.722.1, G.722.2, G.729EV, H.261, H.263, H.264, etc., etc.… (more coming!)
  - Lots of complexity in terminals
  - Lots of cost (development effort, testing, royalties…)
  - Lots of mode permutations = interoperability problems

- Why did this happen?
  - For performance & feature reasons
    - Bitrate, quality, latency, features, cost…
  - And backward compatibility for continued interoperability
  - Both are good reasons…but bring problems

How to manage explosion of codecs?

- NGN may exacerbate problem if not addressed
- Interoperability will still be needed w/legacy:
  - ITU-T codecs
  - MPEG entertainment codecs
  - 3G mobile codecs
  - Internet de-facto codecs (Real, MS, QuickTime…)
- Application & device convergence is real
  - Smart PDA/phones, Game consoles, Set-top boxes, Digital cameras, Internet appliances, PCs…

H.325 media codec architecture (1)

- Goal: *Dramatically fewer required media codecs*
- Allow codecs to improve, optimize for applications
  - Without obsolescing devices
  - Without adding lots of complexity
- Software-defined decoders (downloadable “plug-ins”)
  - From far-end terminals, from servers, from network
  - Devices could (may) cache decoders after downloading
  - Decoders are less compute intense than encoders
  - Inefficiency of generic software less serious with decoders

H.325 media codec architecture (2)

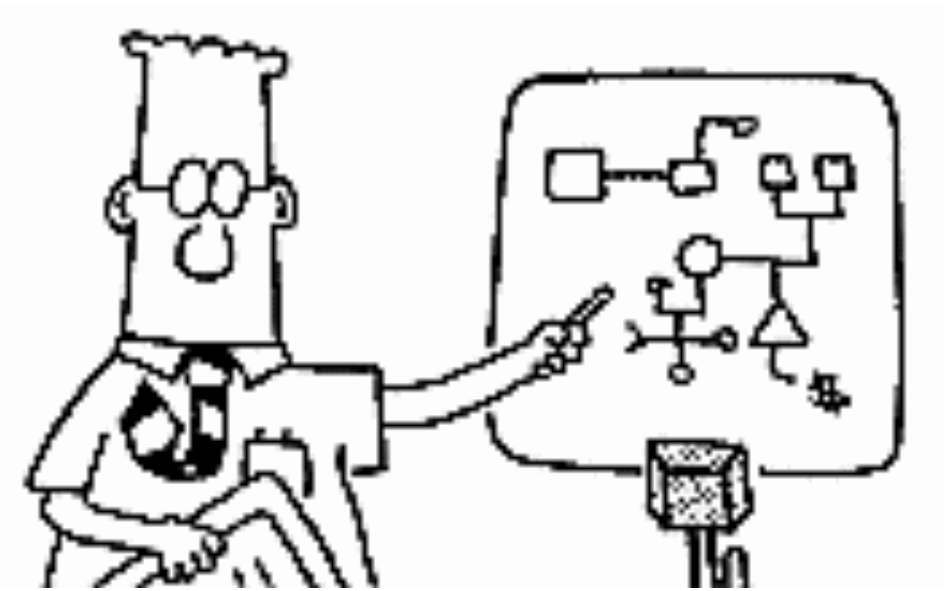
- Define new NGN-optimal codecs only if needed
  - Packet loss-friendly, minimal latency, etc.
- *Minimal* set of "built-in" media decoders
  - For "fall back" purposes only
  - Good enough for the applications, but simple and inexpensive
  - Not necessarily the latest & best – just a fallback
- Interworking with "legacy codecs" is critical
  - But use NGN Gateways
  - Avoid extra terminal complexity

H.325 protocol characteristics

- Goal: *Future-proof* next-generation systems
- Allow line protocols to evolve over time
  - Without obsolescing devices
  - Without locking in limited functions
- “Plug-in" (downloadable) protocol elements
  - From far-end terminals, from servers, from network
  - Devices could (may) cache protocol elements after downloading
  - True extensibility

End of management's dream

Now, we are collecting the Members' input…and discussion has started



Where to find out more?

- TD 132/WP3 (ITU-T SG13, 2006-01)
  - Draft ITU-T H.325 Requirements (Version 0.2)
- TD 133/WP3 (ITU-T SG13, 2006-01)
  - Summarized ITU-T H.325 Requirements (based on Version 0.2)
- Slides from this Workshop & other presenters
  - *http://www.itu.int/ITU-T/worksem/h325/200605/programme.html*

End

- Thank you - Questions?