Service Delivery Architectures and Platforms

Bokor, László
Szabó, Csaba A.
Szabó, Sándor
Service Delivery Architectures and Platforms
írta Bokor, László, Szabó, Csaba A., és Szabó, Sándor

Publication date 2015
Szerzői jog © 2015 Bokor László, Szabó Csaba A., Szabó Sándor
4. 4 CALL/SESSION CONTROL PROTOCOLS ................................................................. 44
  4.1. Essential link control functions ................................................................. 44
  4.2. Most important call/session control protocols ......................................... 44
  4.3. H.323 ...................................................................................................... 44
  4.4. H.323 ...................................................................................................... 44
  4.5. H.323 architecture (1) ............................................................................ 45
  4.6. H.323 architecture (2) ............................................................................ 45
  4.7. H.323 Terminal ....................................................................................... 45
  4.8. Gateway ................................................................................................. 46
  4.9. Gatekeeper ............................................................................................. 46
  4.10. Multipoint Control Unit (MCU) ............................................................... 46
  4.11. BICC ................................................................................................. 46
  4.12. BICC (Bearer Independent Call Control) ................................................. 46
  4.13. BICC architecture (1) ........................................................................... 47
  4.14. BICC architecture (2) ........................................................................... 47
  4.15. MGCP/MEGACO/H.248 ................................................................. 47
  4.16. MGCP/Megaco/H.248 ......................................................................... 47
  4.17. MGCP/Megaco/H.248 architecture (1) .................................................. 48
  4.18. MGCP/Megaco/H.248 architecture (2) .................................................. 48
  4.19. SESSION INITIATION PROTOCOL (SIP) .................................................. 48
  4.20. Session Initiation Protocol (SIP) .............................................................. 48
  4.21. SIP architecture .................................................................................... 49
  4.22. User Agent ............................................................................................ 49
  4.23. Proxy server (1) .................................................................................... 49
  4.24. Proxy server (2) .................................................................................... 49
  4.25. Registrar/Location server ..................................................................... 50
  4.26. Redirect server ...................................................................................... 50
  4.27. SIP and the IMS ................................................................................. 50
  4.28. Identification with SIP ......................................................................... 51
  4.29. Public User Identities .......................................................................... 51
  4.30. Private User Identities .......................................................................... 52
  4.31. Identification example (1) ..................................................................... 52
  4.32. Identification example (2) ..................................................................... 52
  4.33. Identification example (3) ..................................................................... 53
  4.34. Identification example (4) ..................................................................... 53
  4.35. Identification example (5) ..................................................................... 53
  4.36. Forking proxies (1) ................................................................................ 54
  4.37. Forking proxies (2) ................................................................................ 54
  4.38. Forking proxies (3) ................................................................................ 55
  4.39. SIP call between different domains (1) .................................................... 55
  4.40. SIP call between different domains (2) .................................................... 56
  4.41. SIP protocol .......................................................................................... 56
  4.42. SIP protocol structure (1) ...................................................................... 57
  4.43. SIP protocol structure (2) ...................................................................... 57
  4.44. Typical SIP request ............................................................................... 57
  4.45. Typical SIP response ............................................................................. 58
  4.46. SIP methods (1) .................................................................................... 58
  4.47. SIP methods (2) .................................................................................... 58
  4.48. SIP methods (3) .................................................................................... 59
  4.49. SIP responses (1) .................................................................................. 59
  4.50. SIP responses (2) .................................................................................. 59
  4.51. SIP responses (3) .................................................................................. 60
  4.52. SIP header fields (1) ............................................................................... 60
  4.53. SIP header fields (2) ............................................................................... 60
  4.54. SIP header fields (3) ............................................................................... 61
  4.55. SIP header fields (4) ............................................................................... 61
  4.56. SIP header fields (5) ............................................................................... 61
  4.57. SDP (SESSION DESCRIPTION PROTOCOL) ........................................ 62
  4.58. Session descriptions ............................................................................. 62
  4.59. SDP example ......................................................................................... 62
9. Authentication, Authorization, Accounting (AAA) ........................................ 89
  9.1. AAA: Introduction .................................................................................. 89
  9.2. AAA Framework on the Internet .......................................................... 90
  9.3. AAA on the Internet (1) ......................................................................... 90
  9.4. AAA on the Internet (2) ......................................................................... 90
  9.5. Diameter (1) ......................................................................................... 90
  9.6. Diameter (2) ......................................................................................... 90
  9.7. Diameter (3) ......................................................................................... 90
  9.8. Diameter (4) ......................................................................................... 90
  9.9. Format of a Diameter message (1) ....................................................... 90
  9.10. Format of a Diameter message (2) ...................................................... 90
  9.11. Diameter Base Protocol Commands (1) ............................................. 90
  9.12. Diameter Base Protocol Commands (2) ............................................ 90
  9.13. Diameter Base Protocol Commands (3) ............................................. 90
  9.15. AAA in the IMS (1) ............................................................................ 90
  9.16. AAA in the IMS (2) ............................................................................ 90
  9.17. The Cx and Dx interfaces .................................................................... 90
  9.18. Diameter messages during registration .............................................. 90
  9.19. List of commands defined by the Diameter Application for the Cx interface .................. 90
  9.20. Location Information Request and Answer (LIR, LIA) .................... 90
  9.21. Registration Termination Request and Answer (RTR, RTA) ............. 90
  9.22. The user profile .................................................................................. 90
  9.23. Structure of the user profile .................................................................. 90
  9.24. The Sh interface (1) ............................................................................ 90
  9.25. The Sh interface (2) ............................................................................ 90
  9.26. Accounting ......................................................................................... 90

8. Quality of Service (QoS) in the IMS ......................................................... 87
  8.1. QoS on the Internet .............................................................................. 87
  8.2. Integrated Services (1) ......................................................................... 87
  8.3. Integrated Services (2) ......................................................................... 87
  8.4. Resource ReSerVation Protocol (RSVP) - 1 ....................................... 87
  8.5. Resource ReSerVation Protocol (RSVP) - 2 ....................................... 87
  8.6. Resource ReSerVation Protocol (RSVP) - 3 ....................................... 87
  8.7. Differentiated Services (DiffServ) ......................................................... 87
  8.8. The main idea of DiffServ (1) .............................................................. 87
  8.9. The main idea of DiffServ (2) .............................................................. 87
  8.10. How DiffServ works .......................................................................... 87
  8.11. Marking packet in the edge router ..................................................... 87
  8.12. Marking, classification and "conditioning" ........................................... 87
  8.13. DiffServ - "behavior" of nodes ............................................................. 87
  8.14. Placing of "DiffServ Code Points" (DSCPs) ....................................... 87
  8.15. DiffServ - expedited forwarding (EF) ............................................... 87
  8.16. DiffServ - assured forwarding (AF) .................................................. 87
  8.17. Assured forwarding (AF) sub-classes .............................................. 87
  8.18. DiffServ - advantages and problems ................................................. 87
  8.19. QoS in the IMS (1) ............................................................................ 87
  8.20. QoS in the IMS (2) ............................................................................ 87
  8.21. QoS in the IMS (3) ............................................................................ 87
  8.22. QoS in the IMS (4) ............................................................................ 87
  8.23. QoS in the IMS (5) ............................................................................ 87
  8.24. PDP context activation ...................................................................... 87
  8.25. Secondary PDP context activation ..................................................... 87
  8.26. QoS in the network ............................................................................ 87

7. Service Delivery Architectures and Platforms ......................................... 85
  7.12. GPRS Roaming (6) ............................................................................ 85
  7.11. GPRS Roaming (5) ............................................................................ 85
  7.10. GPRS Roaming (4) ............................................................................ 85
  7.9. GPRS Roaming (3) ............................................................................. 85
  7.8. GPRS Roaming (2) ............................................................................. 85
  7.7. GPRS Roaming (1) ............................................................................. 85

6. Quality of Service (QoS) in the IMS ......................................................... 83
  6.12. GPRS Roaming (6) ............................................................................ 83
  6.11. GPRS Roaming (5) ............................................................................ 83
  6.10. GPRS Roaming (4) ............................................................................ 83
  6.9. GPRS Roaming (3) ............................................................................. 83
  6.8. GPRS Roaming (2) ............................................................................. 83
  6.7. GPRS Roaming (1) ............................................................................. 83

5. Resource ReSerVation Protocol (RSVP) ................................................. 81
  5.12. GPRS Roaming (6) ............................................................................ 81
  5.11. GPRS Roaming (5) ............................................................................ 81
  5.10. GPRS Roaming (4) ............................................................................ 81
  5.9. GPRS Roaming (3) ............................................................................. 81
  5.8. GPRS Roaming (2) ............................................................................. 81
  5.7. GPRS Roaming (1) ............................................................................. 81

4. Resource ReSerVation Protocol (RSVP) ................................................. 77
  4.12. GPRS Roaming (6) ............................................................................ 77
  4.11. GPRS Roaming (5) ............................................................................ 77
  4.10. GPRS Roaming (4) ............................................................................ 77
  4.9. GPRS Roaming (3) ............................................................................. 77
  4.8. GPRS Roaming (2) ............................................................................. 77
  4.7. GPRS Roaming (1) ............................................................................. 77

3. Resource ReSerVation Protocol (RSVP) ................................................. 73
  3.12. GPRS Roaming (6) ............................................................................ 73
  3.11. GPRS Roaming (5) ............................................................................ 73
  3.10. GPRS Roaming (4) ............................................................................ 73
  3.9. GPRS Roaming (3) ............................................................................. 73
  3.8. GPRS Roaming (2) ............................................................................. 73
  3.7. GPRS Roaming (1) ............................................................................. 73

2. Resource ReSerVation Protocol (RSVP) ................................................. 69
  2.12. GPRS Roaming (6) ............................................................................ 69
  2.11. GPRS Roaming (5) ............................................................................ 69
  2.10. GPRS Roaming (4) ............................................................................ 69
  2.9. GPRS Roaming (3) ............................................................................. 69
  2.8. GPRS Roaming (2) ............................................................................. 69
  2.7. GPRS Roaming (1) ............................................................................. 69

1. Resource ReSerVation Protocol (RSVP) ................................................. 65
  1.12. GPRS Roaming (6) ............................................................................ 65
  1.11. GPRS Roaming (5) ............................................................................ 65
  1.10. GPRS Roaming (4) ............................................................................ 65
  1.9. GPRS Roaming (3) ............................................................................. 65
  1.8. GPRS Roaming (2) ............................................................................. 65
  1.7. GPRS Roaming (1) ............................................................................. 65
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.27. Accounting</td>
<td>106</td>
</tr>
<tr>
<td>9.28. Types of charging</td>
<td>106</td>
</tr>
<tr>
<td>9.29. Accounting tasks (1)</td>
<td>107</td>
</tr>
<tr>
<td>9.30. Accounting tasks (2)</td>
<td>107</td>
</tr>
<tr>
<td>9.31. Accounting tasks (3)</td>
<td>107</td>
</tr>
<tr>
<td>9.32. Accounting tasks (4)</td>
<td>108</td>
</tr>
<tr>
<td>9.33. Accounting tasks (5)</td>
<td>108</td>
</tr>
<tr>
<td>9.34. Accounting in the IMS</td>
<td>108</td>
</tr>
<tr>
<td>9.35. IMS Offline charging architecture</td>
<td>108</td>
</tr>
<tr>
<td>9.36. Offline charging (1)</td>
<td>109</td>
</tr>
<tr>
<td>9.37. Offline charging (2)</td>
<td>110</td>
</tr>
<tr>
<td>9.38. Online charging architecture</td>
<td>110</td>
</tr>
<tr>
<td>9.39. Online charging (1)</td>
<td>111</td>
</tr>
<tr>
<td>9.40. Online charging (2)</td>
<td>111</td>
</tr>
<tr>
<td>9.41. Immediate Event Charging</td>
<td>112</td>
</tr>
<tr>
<td>9.42. Event Charging with Unit Reservation</td>
<td>112</td>
</tr>
<tr>
<td>10. 10 IMS SERVICES</td>
<td>113</td>
</tr>
<tr>
<td>10.1. Impact of SIP on application development</td>
<td>113</td>
</tr>
<tr>
<td>10.2. IMS services and application building blocks</td>
<td>113</td>
</tr>
<tr>
<td>10.3. PRESENCE</td>
<td>114</td>
</tr>
<tr>
<td>10.4. Presence</td>
<td>114</td>
</tr>
<tr>
<td>10.5. User profile</td>
<td>114</td>
</tr>
<tr>
<td>10.6. Presence service</td>
<td>114</td>
</tr>
<tr>
<td>10.7. Presence service (1)</td>
<td>115</td>
</tr>
<tr>
<td>10.8. Presence service (2)</td>
<td>115</td>
</tr>
<tr>
<td>10.9. Presence service (3)</td>
<td>116</td>
</tr>
<tr>
<td>10.10. Presence service (4)</td>
<td>116</td>
</tr>
<tr>
<td>10.11. Presence service (5)</td>
<td>117</td>
</tr>
<tr>
<td>10.12. Presence service (6)</td>
<td>117</td>
</tr>
<tr>
<td>10.13. Presence service (7)</td>
<td>118</td>
</tr>
<tr>
<td>10.14. Presence service (8)</td>
<td>119</td>
</tr>
<tr>
<td>10.15. Presence service in the IMS</td>
<td>120</td>
</tr>
<tr>
<td>10.16. Presence architecture in the IMS (1)</td>
<td>120</td>
</tr>
<tr>
<td>10.17. Presence architecture in the IMS (2)</td>
<td>120</td>
</tr>
<tr>
<td>10.18. Presence architecture in the IMS (3)</td>
<td>121</td>
</tr>
<tr>
<td>10.19. Presence architecture in the IMS (4)</td>
<td>122</td>
</tr>
<tr>
<td>10.20. Functional description of the Presence architecture</td>
<td>123</td>
</tr>
<tr>
<td>10.21. Presence Service architecture</td>
<td>123</td>
</tr>
<tr>
<td>10.22. Functional units in the Presence architecture</td>
<td>123</td>
</tr>
<tr>
<td>10.23. Presence User Agent</td>
<td>124</td>
</tr>
<tr>
<td>10.24. Presence proxy</td>
<td>124</td>
</tr>
<tr>
<td>10.25. Presence user agent in IMS</td>
<td>125</td>
</tr>
<tr>
<td>10.26. Relationship between Presence agent and other IMS entities</td>
<td>125</td>
</tr>
<tr>
<td>10.27. Watcher application and Presence Server in the IMS</td>
<td>126</td>
</tr>
<tr>
<td>10.28. Presence service - IMS (1)</td>
<td>126</td>
</tr>
<tr>
<td>10.29. Presence service - IMS (2)</td>
<td>127</td>
</tr>
<tr>
<td>10.30. Presence service - IMS (2)</td>
<td>127</td>
</tr>
<tr>
<td>10.31. Presence data model (1)</td>
<td>128</td>
</tr>
<tr>
<td>10.32. Presence data model (2)</td>
<td>128</td>
</tr>
<tr>
<td>10.33. Presence access list (1)</td>
<td>129</td>
</tr>
<tr>
<td>10.34. Presence access list (2)</td>
<td>129</td>
</tr>
<tr>
<td>10.35. Acquiring presence information</td>
<td>130</td>
</tr>
<tr>
<td>10.36. The registration mechanism of IMS watcher</td>
<td>130</td>
</tr>
<tr>
<td>10.37. Notifying the Presence Server about the IMS registration</td>
<td>131</td>
</tr>
<tr>
<td>10.38. Notifying the presence server about the user's status</td>
<td>131</td>
</tr>
<tr>
<td>10.39. Updating presence information</td>
<td>132</td>
</tr>
<tr>
<td>10.40. Instant message service</td>
<td>132</td>
</tr>
<tr>
<td>10.41. Network implementations (1)</td>
<td>132</td>
</tr>
<tr>
<td>10.42. Network implementations (2)</td>
<td>133</td>
</tr>
<tr>
<td>10.43. PUSH-TO-TALK OVER CELLULAR</td>
<td>134</td>
</tr>
</tbody>
</table>
11.4. Development history of smartphones: IBM Simon ........................................ 167
11.5. Development history of smartphones: Nokia 9000 Communicator .................. 168
11.6. Development history of smartphones: Ericsson R380 .................................. 169
11.7. Development history of smartphones: Kyocera 6035 .................................... 169
11.8. History: BlackBerry 5810 .............................................................................. 170
11.9. Development history of smartphones: Palm Treo 600 .................................. 171
11.10. Development history of smartphones: Nokia N70 ...................................... 172
11.11. Development history of smartphones: iPhone (2G) ...................................... 173
11.12. iPhone evolution ......................................................................................... 174
11.13. HTC Dream (G1) ......................................................................................... 175
11.15. LG Optimus 7 ............................................................................................ 177
11.16. Operating system market share .................................................................... 178
11.17. Smartphone market share ........................................................................... 180
11.18. Mobile app store timeline ........................................................................... 180
11.19. Average price of apps ................................................................................ 180
11.20. App Store and Android Market .................................................................... 181
11.21. Global mobile revenue ............................................................................... 182
11.22. Mobile operators ........................................................................................ 183
11.23. Mobile app revenues .................................................................................. 183

12. 12 BLACKBERRY APPLICATION DEVELOPMENT ...................................... 184
   12.1. What is BlackBerry? .................................................................................... 184
   12.2. BES (BlackBerry Enterprise Server) ....................................................... 185
   12.3. Operation of BES (BlackBerry Enterprise Server) ................................ 186
   12.4. BIS (BlackBerry Internet Service) ........................................................... 186
   12.5. Services setting on BlackBerry device .................................................... 187
   12.6. Development environment .................................................................... 187
   12.7. Environment for development (BlackBerry JDEs) - 1 ......................... 188
   12.8. Environment for development (BlackBerry JDEs) - 2 ......................... 189
   12.9. Environment for development (BlackBerry Eclipse Plug-in) - 1 ............. 189
   12.10. Environment for development (BlackBerry Eclipse Plug-in) - 2 .......... 190
   12.11. Development ......................................................................................... 190
   12.12. Application signature and verification .................................................. 191
   12.13. The process to sign your application (1) ............................................. 191
   12.14. The process to sign your application (2) ............................................. 192
   12.15. Install application to the BlackBerry device ....................................... 193
   12.16. Application sharing, distribution .......................................................... 195
   12.17. BlackBerry App World .......................................................................... 195
   12.18. BlackBerry App World - Upload application ....................................... 196

13. 13 IPHONE APPLICATION DEVELOPMENT ........................................ 197
   13.1. The iPhone and the iOS ........................................................................... 197
   13.2. What do you need to start developing for iPhone? ............................... 198
   13.3. Development environment - Xcode (1) ............................................... 199
   13.4. Development environment - Xcode (2) ............................................... 200
   13.5. Developer license ................................................................................... 200
   13.6. The process to get the developer license (1) ....................................... 201
   13.7. The process to get the developer license (2) ....................................... 202
   13.8. App Store ............................................................................................... 202
   13.9. App Store - Upload application ............................................................. 203

14. 14 ANDROID APPLICATION DEVELOPMENT ...................................... 204
   14.1. What is Android? ...................................................................................... 204
   14.2. Android architecture (1) ................................................................. 205
   14.3. Android architecture (2) ................................................................. 205
   14.4. Obtaining the required tools (1) ......................................................... 206
   14.5. Obtaining the required tools (2) ......................................................... 206
   14.6. Creating your First Android Application ........................................ 207
   14.7. Anatomy of an Android Application ................................................ 208
   14.8. Main elements of applications ............................................................. 209
   14.9. GUI (1) ................................................................................................. 210
   14.10. GUI (2) ............................................................................................... 210
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.11. GUI settings</td>
<td>210</td>
</tr>
<tr>
<td>14.12. Overview</td>
<td>211</td>
</tr>
<tr>
<td>14.13. Signing an application</td>
<td>212</td>
</tr>
<tr>
<td>14.15. Google Play</td>
<td>213</td>
</tr>
<tr>
<td>14.16. Android Market registration</td>
<td>214</td>
</tr>
<tr>
<td>14.17. Upload an application to Android Market</td>
<td>214</td>
</tr>
<tr>
<td>15.15 WINDOWS PHONE APPLICATION DEVELOPMENT</td>
<td>215</td>
</tr>
<tr>
<td>15.1. Predecessors: Pocket PC and Windows Mobile (1)</td>
<td>215</td>
</tr>
<tr>
<td>15.2. Predecessors: Pocket PC and Windows Mobile (2)</td>
<td>215</td>
</tr>
<tr>
<td>15.3. Present: Windows Phone</td>
<td>216</td>
</tr>
<tr>
<td>15.4. Windows Phone 7 overview</td>
<td>217</td>
</tr>
<tr>
<td>15.5. Windows Phone 7 Architecture</td>
<td>218</td>
</tr>
<tr>
<td>15.6. Runtimes</td>
<td>219</td>
</tr>
<tr>
<td>15.7. Development environment</td>
<td>219</td>
</tr>
<tr>
<td>15.8. Configuration of the development environment</td>
<td>220</td>
</tr>
<tr>
<td>15.9. Visual Studio 2010</td>
<td>220</td>
</tr>
<tr>
<td>15.10. Expression Blend 4</td>
<td>221</td>
</tr>
<tr>
<td>15.11. Deploying the application (1)</td>
<td>221</td>
</tr>
<tr>
<td>15.12. Deploying the application (2)</td>
<td>222</td>
</tr>
<tr>
<td>15.13. Deploying the application (3)</td>
<td>223</td>
</tr>
<tr>
<td>15.14. Market (1)</td>
<td>223</td>
</tr>
<tr>
<td>15.15. Market (2)</td>
<td>224</td>
</tr>
<tr>
<td>15.16. Market (3)</td>
<td>224</td>
</tr>
<tr>
<td>15.17. Summary</td>
<td>225</td>
</tr>
<tr>
<td>15.18. Windows Phone</td>
<td>225</td>
</tr>
</tbody>
</table>
Service Delivery Architectures and Platforms

1. 1 INTRODUCTION

1.1. Objective of the course

• To give an overview of next generation network architectures and session control protocols, fixed-mobile convergence and mobile evolution.

• Students who have successfully finished the course
  • will be able to understand network capabilities and the technical background of fixed and mobile telecommunication services,
  • will be familiar with service delivery platforms,
  • will obtain starting points for development of services in next generation networks, including mobile platforms.

1.2. Why it is worth taking this course?

• Students can
  • obtain theoretical knowledge,
  • with practical additions, and
  • learn about innovative projects.

1.3. Course outline

1. Introduction to NGN and IMS
2. Service Delivery Platforms
3. IMS Architecture
4. Call and Session Control Protocols
5. Registration and Session Establishment
6. Media Transport
7. Roaming
8. Quality of Service
9. AAA
10. Services
11. Integration with Mobile Platforms
12. Lab 1. Blackberry Application Development
13. Lab 2. iPhone Application Development
14. Lab 3. Android Application Development
15. Lab 4. Windows Phone Application Development

1.4. Contents of this lecture

• Current trends in telecommunications industry
• The concept of NGN - Next Generation Networks
• Introduction to IMS - IP Multimedia Subsystem
• Appendix: NGN-IMS Case Study
• Recommended literature

1.5. CURRENT TRENDS IN TELECOMMUNICATIONS INDUSTRY

1.6. The challenges in traditional telco industry

• Service providers - billions of losses
  • The rise of VoIP providers in long distance calls market.
  • Wireless solutions gradually occupy the market of long-distance calls.
  • The same market movements in local calls.
• Manufacturers - billions of losses
  • The new, previously unknown IP PBX and IP phone manufacturers significantly reduce the market share of well-known manufacturers.

1.7. Changes in the business environment

• Globalization effect
  • e-Commerce
  • Global information society (e-Gov, VPN, convergent networks)
  • Telecommunication market without borders
• IT networks and technologies
  • IP traffic forecasts, high capacity and scalability, QoS, QoP
• User expectations
  • Good quality, mobility, security, multimedia, simple solutions, proportionate costs

1.8. Convergence
1.9. Convergence

- Business
  - Dynamically changing market, technological and regulatory conditions.
  - Methods for increasing profitability:
    - Expand the market
    - Reduce staff turnover
    - Improve the economy
    - Strategic alliances
  - Services
    - Unified Messaging
    - Global Call Center

- Network
  - Private and public networks.
  - Wireline and wireless networks.

- Platform
  - Circuit-switched and packet-switched.
1.10. Market trends

• The traditional voice service market is declining.
• New services are needed to make up for lost revenues.

1.11. Market trends (1)
1.12. Market trends (2)

- Market features:
  - New IP multimedia services to fixed and mobile networks.
  - Rise of Voice over IP (VoIP).
  - Fixed-mobile substitution (FMS) - voice traffic is shifted to mobile networks.
  - The focus of communications moves to open, IP-based networks.
  - The IP-based core network enables new services, promotes the FMC.
  - The IP Multimedia Subsystem (IMS) is a key element in the service providers' network.

1.13. The focus is on services instead of networks
• Physical networks are becoming transparent.
• The barrier of entry for new services will disappear.
• The value chain is horizontally separated.
• Reduction of income.
• The "ownership" of users is lost.
• The mobile value chain is out of the hands of service providers.

1.14. Challenges for service providers

• VoIP threatens the main source of income, the voice call. Basic VoIP service is cheap, it becomes the default service.
• Compensation: high-level SIP*-based services:
  • Rich VoIP services
  • Personalized services
  • FMC VoIP services
• Differences between fixed and mobile operators will disappear. Examples: France Telecom, Telecon Italia, Telefonica.
• An appropriate service platform have to be chosen for both sides (service integration).
• The IP Multimedia Subsystem can be this common platform.
*) SIP - Session Initiation Protocol, to be dealt with during this course

1.15. THE CONCEPT OF NGN - NEXT GENERATION NETWORK
1.16. NGN 3-level network architecture

- Application servers: independent service layer.
- Session servers: call control, soft switches, SIP protocol.
- Routers: signaling messages and content (media) transport.
- Media gateway, Media servers: data processing, conversions.

1.17. NGN

Integrated platform for packet-based services.

- Management tasks are treated separately:
  - Mobility, security, authentication, authorization, accounting (AAA).
- Different access networks can be treated uniformly, regardless of whether:
  - the applied technology is wired or wireless,
  - the provider owns the network or it is an independent network.
• Integrated architecture, services, standard interfaces.
• Flexible, economical, faster application development.

1.18. Main features of NGN architecture

• Layered
  • Network services and session control are clearly separated from the transport elements.
  • Clear separation of access networks from the services.
  • These separations make it possible that the service development and connections, transport considerations can be independent.

• Open service interfaces
  • Allows service providers and third parties to develop and introduce new services easily.

• Distributed network intelligence
  • In contrast to ISDN, the NGN concept makes it possible to separate the service intelligence from network elements. The network intelligence can be divided among the appropriate network sites, typically it is located at the edge of the network.

1.19. Structure of NGN networks, ITU-T Example

1.20. NGN All-IP architecture
1.21. INTRODUCTION TO IMS - IP MULTIMEDIA SUBSYSTEM

1.22. What is IMS?

- IMS = IP Multimedia Subsystem (Packet Switched domain)
- Multimedia call control in packet switched network
- IMS introduces the IP-based services to the mobile world
  - The user have to be authenticated only once.
  - Access charging, service charging and content charging.
  - Managing multimedia sessions.
- IMS introduces new, rich media services
  - Presence, Conferencing, Push, Chat, Push-to-talk, ...
  - Makes is possible to deliver services to the users on third-party network.
- IMS is another step towards the world of the IETF Internet
  - The IMS is more than IETF SIP: it is not only a protocol, it is an architecture.

1.23. The vision of the IMS network

- IMS is the key element of 3G (4G) networks, services can be accessed from mobile and fixed networks
- Why do we need IMS if most of the services can be accessed anyway?
  - Provides QoS
  - Provides accounting
  - Supports service integration
  - Supports fixed-mobile convergence
• Enables web applications

Benefits of IMS:
• Easy service development, unified look, attractive services
• Advanced QoS and accounting
• IP core
• Session management, roaming
• New service capabilities

1.24. Expectations towards IMS

• Compete with traditional internet services:
  • QoS, security, accounting
  • Integrated multimedia services
• The IMS is the least common multiple
• A universal service platform
  • Flexible application development
  • Openness towards third party developers
  • Unified look
• Can be accessed from fixed and mobile networks

1.25. Standardization (1)

• ETSI - TISPAN and 3GPP;
• ITU (International Telecommunication Union) - raises specifications to global standards;
• ATIS (Alliance for Telecommunications Industry Solutions);
• IETF - IPv6, MPLS, and SIP (Session Initiation Protocol) extension;
• TMF (Telecom Management Forum) and OSS/J - standard OSS (Operational Service System) components;
• MSF - NGN VoIP;

1.26. Standardization (2)

• OMA (Open Mobile Alliance) and Parlay - Mobile services and DRM (Digital Mobile Radio);
• MEF (Metro Ethernet Forum) - Ethernet transport-networks;
• DSL Forum - DSL and QoS architectures;
• IEEE 802.11x - Wi-Fi hotspots;
• W3C - WEB services and security;
1.27. Controlling IP-based services in the IMS

- The IP network enables communication between endpoints
- IMS controls SIP sessions

1.28. IMS: Controlling GPRS/IP service

- There is an IP multimedia overlay network over the GPRS network
- Signaling and data transmission over GPRS network
- Controlling IP Services (QoS, security, accounting)

1.29. IMS - flexible services

- Service enablers, for example presence and Group servers
- Push to Talk and community services.
1.30. Integration of various communication services by IMS

1.31. IMS services

1.32. IMS architecture principles

- IMS does not define specific services, only enablers
- "Built-in" support for multimedia over IP, VoIP, IM, presence
- Flexible multimedia transmission over IP
- Horizontal architecture
- Uses IETF standards
- Modular design, open interfaces
1.33. IMS impact on the telco services value chain

- GSM value chain:
  - The network and the services are not separated
  - Minimal access to third parties
  - The operator is responsible for developing service packages

- IMS value chain:
  - Loose vertical integration of networks, services and applications
  - Standardized interfaces: easy to integrate
  - All participants focus on their competence

1.34. APPENDIX: NGN-IMS CASE STUDY

1.35. NGN-IMS case study

- BT 21 CN
• Setting up NGN network till 2011
• Cost savings, simplification
  • IP core network
  • Less network layer
  • IMS-based services

1.36. BT21CN

1.37. The current situation

• It is expensive and complicated to connect multiple complex networks
• Telecom service providers want to lower their costs
• The number of mobile and wireless subscribers rises
• There are more and more roaming subscribers
• Global standards
• Quick and planned transition to NGN

1.38. The aim of BT21CN

• Make it easier to create new services
  • Quicker
  • Involving multiple developers
• Make it easier to purchase and the use of services
  • "Enable customers"
• Make it easier to introduce and operate services
  • Continuous automation
• 30-40% cost reduction

1.39. What does this mean in practice?

• New services
  • Open APIs and service development platforms
  • Mobility support
  • Reusable components and capabilities
  • Broadband
• Cost reduction
  • Attitude change: smaller number of networks and systems with more features and services
  • Convergence: converged IP and MPLS core network
  • One network with lots of services instead of lots of networks with one service

1.40. NGN transition

• According to Mick Reeve, the leading expert of British Telecom, major transformation is taking place in Europe in the next 10-15 years which stations are predicted as follows:
  • There are six main platforms where the telecommunication of the world takes place: PSTN, DPCN (Data Packet Core Network), ATM + IP, MSH-SDH (Mesh Synchron Digital Hierarchy) and PDH.
  • Firts, DPCN will disappear, after that ATM and IP will go to a common IP channel under the supervision of the Call Server.
  • Then, PDH, MSH-SDH and PSTN will disappear and a new consolidated state will be formed where all traffic is integrated into an IP-MPLS WDM "channel" under the "Class 5 Call Server's" supervision.
  • IMS will be the key element of the integrated, intelligent worldwide network's control plane.

1.41. 21CN vision
1.42. Current networks

1.43. Platform consolidation

1.44. Platform consolidation
1.45. Platform consolidation

1.46. Platform consolidation

1.47. Platform consolidation
1.48. 21CN - key milestone overview

1.49. BT21CN 2009

• 31 March 2009 BT hits key 21CN milestones
  • Exchanges serving 10 million UK homes and businesses are now enabled for next generation broadband. Over the last 12 months, BT has succeeded in increasing the footprint of our ADSL2+ service from 5% to 40% of the UK - and rollout continues.

• 5 March 2009
  • Also BT has deployed more than 600 Ethernet nodes in the UK - confirming its position leading Ethernet availability across the UK - and rollout continues.

• BT launches IVPN service for large organisations

• BT announces the launch of its BT Intelligent Virtual Private Network (iVPN) service in 172 countries, allowing global organisations to better manage and improve the performance of their IT network.

1.50. BT21CN 2009

• 2 Feb 2009 Wholesale launch ‘Pay-as-you-grow’ managed broadband service with roadmap to 21CN
  • BT Wholesale launches a ‘pay-as-you-grow’ managed broadband service called BT Plusnet Partner which is aimed at resellers and virtual ISPs. The service allows smaller ISPs, start-ups and brand extenders to easily add broadband to their product portfolio or expand into new broadband markets quickly and cost effectively, without the need for infrastructure investment.

• 12 Feb 2009 BT provides 21CN update as part of its third quarter and nine months results to December 31, 2008
  • "We continued the roll out of 21CN supported next generation broadband and Ethernet services during the quarter. Progress on rolling out Ethernet to date means that BT Wholesale now has the largest Ethernet footprint in the UK market and, by the end of this financial year, we will be twice as large as BT’s nearest wholesale."

1.51. 21C High Level Network Architecture
1.52. Common Intelligence Vision

1.53. BT 21C - The first fully converged network of the world
1.54. Recommended literature


- The relevant IETF, ITU and 3GPP standards (exact references to RFC or Recommendation numbers are given on the slides)

2. 2 SERVICE DELIVERY PLATFORMS

2.1. Service Delivery Platform (SDP)

- The penetration of broadband networks is playing a key role in bringing about major changes in the way that network-based services are provided.

- New schemes such as Software as a Service (SaaS) and Platform as a Service (PaaS) let users access a variety of services over network.

- Applications, computers, storage, and other resources combined over the network enable the rapid delivery of diverse services.

- In response to this shift to a "service economy", it will become increasingly important for the infrastructure of the network society to enhance the value of the network and build a service platform that can deploy new services promptly.

2.2. The old and the new architecture

IMS - IP Multimedia Subsystem, to be dealt with in detail during this course

2.3. Why do we need SDP?

- Today's service providers are facing well-known challenges, e.g.:
• ARPU (Average Revenue Per User) erosion as voice becomes a commodity.

• Aggressive competitors offering a broad range of services.

• In response, they must deliver new high-revenue services.

• Unfortunately, there is a major obstacle barring their path: their traditional network integration and marketing processes mean that new service deployment takes a very long time and costs a huge amount.

• The concept of SDP has been developed as a solution to this problem. An SDP allows service providers to define, develop and deploy new services far faster than they have been able to in the past.

2.4. Why do we need SDP? (cont.)

• This is achieved in two ways:
  
  • An SDP includes tools that allow for very easy definition and development of new services.
  
  • It also provides a single environment within which network integration occurs once, so new services do not require major new IT integration.

  • Once the cost of service deployment has been massively reduced, it allows the service provider to take an entirely different perspective on return on investment and business planning.

  • An SDP is more than a single product or component within the network. It is a suite of interconnected products that enable flexible service creation, modification and subscriber personalization.

2.5. SDP

• SDPs combine commercial off-the-shelf ("COTS") hardware (industry-standard servers) with interoperable libraries of common function code, or "software building blocks." These building blocks shortcut development and allow fairly junior IT professionals to make the kinds of modifications to services that once could only be made by experienced programmers.

• By employing published open interfaces across the entire SDP landscape, SDPs are designed to easily integrate with existing OSS and billing systems, minimizing the customization required to enable flow through provisioning.

• Key Functional Areas:
  
  • Service Creation Environment
  
  • Service Execution Environment
  
  • Common Service Support Functions

2.6. Service Creation Environment (SCE)

• Service Creation Environment (SCE) is used to rapidly create new services, or improve and customize existing services. This (typically graphical) interface offers developers an easy way to identify and combine the appropriate pre-defined software code building blocks required for the creation of a new service.
2.7. Service Creation Environment (SCE)

- In addition, this same environment executes subscriber service personalization. Subscribers typically have a much simpler and more limited interface to the SCE.

2.8. Service Execution Environment (SEE)

- The Service Execution Environment provides the back end for the Service Creation Environment. Together with the SCE, it enables rapid commercial development and deployment of applications.

- The Service Execution Environment is another way of referring to the native resource pool responsible for implementing changes and enhancements to services. It comprises either:
  - a Java Platform, Enterprise Edition (J2EE) environment optimized across a server farm, or
  - a purpose-built platform utilizing proprietary software.

- Benefits of JavaEE:
  - Flexible
  - Scalable
  - Open platform
  - Simplifies integration with legacy platforms
  - Allows new servers to be added at any time without requiring system downtime or service interruption.
2.9. Common Service Support Functions

- They refer to the basic software elements used to construct a service.
- The Service Creation Environment presents the Common Service Support Functions to the developer in a simple and often visual format.
- This makes it easy to develop services without possessing an intimate understanding of the underlying code. An example would be SIP servlets.
- Thus, the Common Service Support Functions provide developers with a generic library for common underlying software functions that is easily extensible across platforms in the event of great subscriber demand.

2.10. SDP architecture

2.11. SDP and IMS: an architectural point of view

- SDPs can be considered as a framework for quickly creating applications that sit on top an IMS network
3. 3 IMS ARCHITECTURE

3.1. Architecture

- 3GPP IMS does not standardize nodes, but functions. This means that the IMS architecture is a collection of functions linked by standardized interfaces. Implementers are free to combine two functions into a single node (e.g., into a single physical box). Similarly, implementers can split a single function into two or more nodes.

- The IP Multimedia Core Network Subsystem contains:
  - one or more user databases, called HSSs (Home Subscriber Servers) and SLFs (Subscriber Location Functions);
  - one or more SIP servers, collectively known as CSCFs (Call/Session Control Functions);
  - one or more ASs (Application Servers);
  - one or more MRFs (Media Resource Functions), each one further divided into MRFC (Media Resource Function Controllers) and MRFP (Media Resource Function Processors);
  - one or more BGCFs (Breakout Gateway Control Functions);
  - one or more PSTN gateways, each one decomposed into an SGW (Signaling Gateway), an MGCF (Media Gateway Controller Function), and an MGW (Media Gateway).

3.2. IMS architecture
3.3. Home Subscriber Server (HSS)

- The Home Subscriber Server (HSS) is the central repository for user-related information.
- Technically, the HSS is an evolution of the HLR (Home Location Register), which is a GSM node.
- The HSS contains all the user-related subscription data required to handle multimedia sessions:
  - User profiles
  - Service-related information
  - Location information
  - Authentication and authorization information (AKA - Authentication and Key Agreement)
  - Cryptographic keys
  - S-CSCF allocated to users

3.4. HSS Modules

- Request Dispatcher
  - This module receives diameter message from underlying framework and routes it to appropriate application.
- Cx/Dx Application
  - This is a 3GPP authentication application that supports IMS AKA authentication scheme ref no.
- Sh Application
  - This application is used by the application server for retrieving and updating user service profile.
- DB Interaction Manager
  - This module manages interaction with the backend data store.
- Subscriber profile and Service Profile Management GUI
• Admin interface for subscriber related data.

3.5. HSS network diagram

3.6. Subscriber Locator Function (SLF)

• Networks with a single HSS do not need a Subscription Locator Function (SLF). On the other hand, networks with more than one HSS do require an SLF.

• The SLF is a simple database that maps users' addresses to HSSs. A node that queries the SLF, with a user's address as the input, obtains the HSS that contains all the information related to that user as the output.

• Both the HSS and the SLF implement the Diameter protocol (RFC 3588) with an IMS-specific Diameter application.

3.7. Call/Session Control Function (CSCF)

• The CSCF (Call/Session Control Function), which is a SIP server, is an essential node in the IMS. The CSCF processes SIP signaling in the IMS. There are three types of CSCFs, depending on the functionality they provide. All of them are collectively known as CSCFs, but any CSCF belongs to one of the following three categories.
• Proxy Call Session Control Function (P-CSCF)
• Interrogating Call Session Function (I-CSCF)
• Serving Call Session Control Function (S-CSCF)

3.8. P-CSCF (1)

• The P-CSCF is the first point of contact (in the signaling plane) between the IMS terminal and the IMS network.

• Bearer traffic is not passed through this portion of the IMS, as this is a signaling and control network.

• From the SIP point of view the P-CSCF is acting as an outbound/inbound SIP proxy server. This means that all the requests initiated by the IMS terminal or destined for the IMS terminal traverse through the P-CSCF. The P-CSCF forwards SIP requests and responses in the appropriate direction (i.e., toward the IMS terminal or toward the IMS network).

• P-CSCF is responsible for authentication. Once the P-CSCF authenticates the user (as part of security association establishment) the P-CSCF asserts the identity of the user to the rest of the nodes in the network. This way, other nodes do not need to further authenticate the user, because they trust the P-CSCF.

• Additionally, the P-CSCF verifies the correctness of SIP requests sent by the IMS terminal. This verification keeps IMS terminals from creating SIP requests that are not built according to SIP rules.

3.9. P-CSCF (2)

• It can also compress and decompress SIP messages using SigComp, which reduces the round-trip over slow radio links.

• The P-CSCF also generates charging information toward a charging collection node.

• The P-CSCF may include a PDF (Policy Decision Function). The PDF may be integrated with the P-CSCF or be implemented as a stand-alone unit. The PDF authorizes media plane resources and manages Quality of Service over the media plane.

• An IMS network usually includes a number of P-CSCFs for the sake of scalability and redundancy.

• The P-CSCF may be located either in the visited network or in the home network. In the case when the underlying packet network is based on GPRS, the P-CSCF is always located in the same network where the GGSN (Gateway GPRS Support Node) is located.

3.10. I-CSCF

• The I-CSCF is a SIP proxy located at the edge of an administrative domain.

• I-CSCF serves as the gateway into each individual IMS network. It is the I-CSCF that determines whether or not to grant access to other networks forwarding SIP messages to the operator.

• Besides the SIP proxy server functionality, the I-CSCF has an interface to the SLF and the HSS. This interface is based on the Diameter protocol. The I-CSCF retrieves user location information and routes the SIP request to the appropriate destination (typically to an S-CSCF).

• It has also interfaces to the application servers as well, to be able to forward messages that are destined to services and not to users.
Additionally, the I-CSCF may optionally encrypt the parts of the SIP messages that contain sensitive information about the domain, such as the number of servers in the domain, their DNS names, or their capacity. This functionality is referred to as THIG (Topology Hiding Inter-network Gateway).

The I-CSCF is usually located in the home network, although in some special cases, such as an I-CSCF(THIG), it may be located in a visited network as well.

### 3.11. S-CSCF (1)

- The S-CSCF is the central node of the signaling plane.
- The S-CSCF is essentially a SIP server, but it performs session control as well.
- S-CSCF also acts as a SIP registrar. This means that it maintains a binding between the user location (e.g., the IP address of the terminal the user is logged on) and the user's SIP address of record (also known as a Public User Identity).
- Like the I-CSCF the S-CSCF also implements a Diameter interface to the HSS. The main reasons to interface the HSS are as follows:
  - To download the authentication vectors of the user who is trying to access the IMS from the HSS. The S-CSCF uses these vectors to authenticate the user.
  - To download the user profile from the HSS. The user profile includes the service profile, which is a set of triggers that may cause a SIP message to be routed through one or more application servers.
  - To inform the HSS that this is the S-CSCF allocated to the user for the duration of the registration.

### 3.12. S-CSCF (2)

- All the SIP signaling the IMS terminals sends, and all the SIP signaling the IMS terminal receives, traverses the allocated S-CSCF.
- One of the main functions of the S-CSCF is to provide SIP routing services.
- It decides to which application server(s) the SIP message will be forwarded, in order to provide their services.
- It provides routing services, typically using Electronic numbering (ENUM, RFC 2916) lookups.
- It enforces the policy of the network operator.
- The S-CSCF is always located in the home network.
- There can be multiple S-CSCFs in the network for load distribution and high availability reasons.

### 3.13. Application Server (AS)

- The AS (Application Server) is a SIP entity that hosts and executes services.
- Depending on the actual service the AS can operate the following modes:
  - SIP Proxy
  - SIP UA (User Agent)
  - SIP B2BUA (Back-to-Back User Agent): a concatenation of two SIP User Agent
• Application Servers communicate with the S-CSCF over the SIP-based ISC interface and with the HSS over the Diameter-based Sh interface.

• The AS has interfaces for configuration purposes.

• The AS can be located either in the home network or in an external third-party network to which the home operator maintains a service agreement. In any case, if the AS is located outside the home network, it does not interface the HSS.

3.14. Types of Application Servers (1)

• SIP AS (Application Server): this is the native Application Server that hosts and executes IP Multimedia Services based on SIP.

• OSA-SCS (Open Service Access - Service Capability Server): this application server provides an interface to the OSA framework Application Server. It inherits all the OSA capabilities, especially the capability to access the IMS securely from external networks. This node acts as an Application Server on one side (interfacing the S-CSCF with SIP) and as an interface between the OSA Application Server and the OSA Application Programming Interface.

• IM-SSF (IP Multimedia Service Switching Function): this specialised application server allows us to reuse CAMEL (Customized Applications for Mobile network Enhanced Logic) services that were developed for GSM in the IMS.

3.15. Types of Application Servers (2)

• The IM-SSF allows a gsmSCF (GSM Service Control Function) to control an IMS session.

• The IM-SSF acts as an Application Server on one side (interfacing the S-CSCF with SIP). On the other side, it acts as an SSF (Service Switching Function), interfacing the gsmSCF with a protocol based on CAP (CAMEL Application Part)
3.16. Media Resource Function (MRF)

- The MRF (Media Resource Function) provides a source of media in the home network. The MRF provides the home network with the ability to play announcements, mix media streams (e.g., in a centralized conference bridge), transcode between different codecs, obtain statistics, and do any sort of media analysis.

- The MRF is divided into two nodes:
  - A signaling plane node called the MRFC (Media Resource Function Controller). The MRFC acts as a SIP User Agent and contains a SIP interface towards the S-CSCF and controls the resources in the MRFP via an H.248 interface.
  - A media plane node called the MRFP (Media Resource Function Processor). The MRFP implements all the media-related functions, such as playing and mixing media.

- The MRF is always located in the home network.

3.17. Breakout Gateway Control Functions (BGCF)

- The BGCF is essentially a SIP server that includes routing functionality based on telephone numbers. The BGCF is only used in sessions that are initiated by an IMS terminal and addressed to a user in a circuit-switched network, such as the PSTN or the PLMN.

- The main functionality of the BGCF is:
  - to select an appropriate network where interworking with the circuit-switched domain is to occur;
  - or, to select an appropriate PSTN/CS gateway, if interworking is to occur in the same network where the BGCF is located.
3.18. IMS Application Layer Gateway (IMS-ALG)

- IMS supports two IP versions, namely IP version 4 (IPv4, specified in RFC 791) and IP version 6 (IPv6 specified in RFC 2460).
- At some point in an IP multimedia session or communication, interworking between the two versions may occur. In order to facilitate interworking between IPv4 and IPv6 without requiring terminal support, the IMS adds two new functional entities that provides translation between both protocols.
- These new entities are the IMS Application Layer Gateway (IMS-ALG) and the Transition Gateway (TrGW).
- The IMS-ALG acts as a SIP B2BUA by maintaining two independent signaling legs: one towards the internal IMS network and the other towards the other network. Each of these legs are running over a different IP version. Additionally, the IMS-ALG rewrites the SDP by changing the IP addresses and port numbers created by the terminal with one or more IP addresses and port numbers allocated to the TrGW. This allows the user plane traffic to be routed to the TrGW.

3.19. Transition Gateway (TrGW)

- The TrGW does the translation of IPv4 and IPv6 at the media level (e.g., RTP (Real-time Transport Protocol), RTCP (Real-time Transport Control Protocol)).
- The TrGWs effectively a NAT-PT/NAPT-PT (Network Address Port Translator-Protocol Translator). The TrGW is configured with a pool of IPv4 addresses that are dynamically allocated for a given session.

3.20. The IMS-ALG and the TrGW
3.21. PSTN/CS Gateway (1)

- The PSTN gateway provides an interface toward a circuit-switched network, allowing IMS terminals to make and receive calls to and from the PSTN (or any other circuit-switched network).

- The PSTN gateway is decomposed into the following functions:

  - Signaling Gateway (SGW): the Signaling Gateway interfaces the signaling plane of the CS network (e.g., the PSTN).

    - The SGW performs lower layer protocol conversion. For instance, an SGW is responsible for replacing the lower MTP transport with SCTP (Stream Control Transmission Protocol) over IP. So, the SGW transforms ISUP* (ISDN User Part) or BICC* (Bearer Independent Call Control Protocol) over MTP into ISUP or BICC over SCTP/IP.

*) Both BICC and ISUP are call control protocols in circuit-switched networks.

3.22. PSTN/CS Gateway (2)

- Media Gateway Control Function (MGCF):

  - The MGCF is the central node of the PSTN/CS gateway.

  - It implements a state machine that does protocol conversion and maps SIP either ISUP over IP or BICC over IP.

  - In addition to the call control protocol conversion the MGCF controls the resources in an MGW (Media Gateway).

  - The protocol used between the MGCF and the MGW is H.248

3.23. PSTN/CS Gateway (3)

- Media Gateway (MGW):

  - The Media Gateway interfaces the media plane of the PSTN or CS network.

  - On one side the MGW is able to send and receive IMS media over the Real-time Transport Protocol (RTP) (RFC 3550).

  - On the other side the MGW uses one or more PCM (Pulse Code Modulation) time slots to connect to the CS network.

  - The MGW performs transcoding when the IMS terminal does not support the codec used by the CS side. A common scenario occurs when the IMS terminal is using the AMR (3GPP TS 26.071) codec and the PSTN terminal is using the G.711 codec (ITU-T Recommendation G.711).

3.24. A PSTN/CS Gateway interfacing a CS network
3.25. IMS-Based PES Architecture (1)

- IMS-based PES (PSTN Emulation System) provides IP networks services to analog devices. PES allows non-IMS devices to appear to IMS as normal SIP users.

- Analog terminal using standard analog interfaces can connect to IMS-based PES in two ways:
  - Via A-MGW (Access Media Gateway) that is linked and controlled by AGCF. AGCF is placed within the operators network and controls multiple A-MGWs. A-MGW and AGCF communicate using H.248 over the P1 reference point. POTS phone connect to A-MGW over the z interface. The signaling is converted to H.248 in the A-MGW and passed to AGCF. AGCF interprets the H.248 signal and other inputs from the A-MGW to format H.248 messages into appropriate SIP messages. AGCF presents itself as P-CSCF to the S-CSCF and passes generated SIP messages to S-CSCF or to IP border via IBCF (Interconnection Border Control Function). Service presented to S-CSCF in SIP messages trigger PES AS. AGCF has also certain service independent logic, for example on receipt of off-hook event from A-MGW, the AGCF requests the A-MGW to play dial tone.

3.26. IMS-Based PES Architecture (2)

- Via VGW (VoIP-Gateway) or SIP Gateway/Adapter on customer premises. POTS phones via VOIP Gateway connect to P-CSCF directly. Operators mostly uses Session Border Control between VoIP Gateway and P-CSCF for security and to hide network topology. VoIP Gateway link to IMS using SIP over Gm reference point. The conversion from POTS service over the z interface to SIP occurs in the customer premises VoIP...
Gateway. POTS signaling is converted to SIP and passed on to P-CSCF. VGW acts as SIP user agent and appears to P-CSCF as SIP terminal.

- Both A-MGW and VGW are stateless and unaware of the services. They only relay call control signaling to and from the PSTN terminal. Session control and handling is done by IMS components.

### 3.27. TISPAN IMS architecture with interfaces

![TISPAN IMS Architecture Diagram]

*3.28. Interfaces (1)*
### 3.29. Interfaces (2)

<table>
<thead>
<tr>
<th>Interface</th>
<th>IMS entities</th>
<th>Description</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go</td>
<td>PDF, GPRS</td>
<td>Allows operators to control QoS in the user plane and exchange charging correlation information between IMS and GPRS network.</td>
<td>COPS, Diameter</td>
</tr>
<tr>
<td>ISC</td>
<td>S-CSCF, AS</td>
<td>Reference point between S-CSCF and AS. Main functions are to: Notify the AS of the registered MPU, registration state and UE capabilities. Supply the AS with information to allow it to execute multiple services. Convey charging function addresses.</td>
<td>SIP</td>
</tr>
<tr>
<td>ICI, [Izi]</td>
<td>IBCFs, (TrGWs)</td>
<td>Used to exchange messages between an IBCF (TrGWs) and another IBCF (TrGWs) belonging to a different IMS network.</td>
<td>SIP, {RTP}</td>
</tr>
<tr>
<td>Mg</td>
<td>MGCF -&gt; I-, S-CSCF</td>
<td>ISUP signalling to SIP signalling and forwards SIP signalling to I-CSCF.</td>
<td>SIP</td>
</tr>
</tbody>
</table>

### 3.30. Interfaces (3)
<table>
<thead>
<tr>
<th>Interface</th>
<th>IMS entities</th>
<th>Description</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mm</td>
<td>CSCFs, other IP network</td>
<td>Used for exchanging messages between IMS and external IP networks.</td>
<td>SIP</td>
</tr>
<tr>
<td>Mp</td>
<td>MRFC, MRFP</td>
<td>Allows an MRFC to control media stream resources provided by an MRFP.</td>
<td>H.248</td>
</tr>
<tr>
<td>P1</td>
<td>AGCF, A-MGW</td>
<td>Used for call control services by AFG to control H.248 A-MGW and Residential Gateways.</td>
<td>H.248</td>
</tr>
<tr>
<td>P2</td>
<td>AGCF, CSCF</td>
<td>Reference point between AGCF and CSCF.</td>
<td>SIP</td>
</tr>
<tr>
<td>Rf</td>
<td>CSCFs, BCF, MRFC, MGCF, AS</td>
<td>Used to exchange offline charging information with CDF.</td>
<td>Diameter</td>
</tr>
<tr>
<td>Ro</td>
<td>AS, MRFC, S-CSCF</td>
<td>Used to exchange online charging information with OCF.</td>
<td>Diameter</td>
</tr>
</tbody>
</table>

### 3.31. Interfaces (4)

<table>
<thead>
<tr>
<th>Interface</th>
<th>IMS entities</th>
<th>Description</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sh</td>
<td>AS, HSS</td>
<td>Used to exchange User Profile information between an AS (SIP AS or OSA SCS) and HSS.</td>
<td>Diameter</td>
</tr>
<tr>
<td>Si</td>
<td>IM-SSF, HSS</td>
<td>Transports CAMEL subscription information including triggers for use by CAMEL based application services information.</td>
<td>MAP</td>
</tr>
<tr>
<td>Ut</td>
<td>UE&lt;-&gt;AS, PES&lt;-&gt;AS, AGCF</td>
<td>Facilitates the management of subscriber information related to services and settings.</td>
<td>HTTP, XCAP</td>
</tr>
<tr>
<td>z</td>
<td>POTS, VoIP Gateway</td>
<td>Conversion of POTS services to SIP messages.</td>
<td></td>
</tr>
</tbody>
</table>

### 3.32. Organizations behind IMS

- 3GPP (Third Generation Partnership Project)
  - 1998, GSM evolution. ARIB, TTC (Japan), CCSA (China), ETSI (EU), T1 Committee (USA), TTA (Korea).
- 3GPP2
  - ANSI/TIA/EIA-41, CDMA2000-based development. ARIB, TTC (Japan), CCSA (China), TIA (USA), TTA (Korea)
- IETF (Internet Engineering Task Force)
- UMTS networks are realised based on 3GPP releases.
3.33. 3GPP Releases (1)

- Release ’99
  - Frozen: 1999 December
  - Specified the first UMTS 3G networks
  - Incorporating a CDMA air interface
- Release 4
  - Frozen: March 2001
  - Separation of control and user planes.
  - Added features including an all-IP Core Network
  - TD-SCDMA
- Release 5
  - Frozen: March/June 2002
  - IMS - IP-based Multimedia Services
  - HSDPA - High Speed Downlink Packet Access

3.34. 3GPP Releases (2)

- Release 6
  - Frozen: September/December 2004
  - Second phase of IMS
• HSUPA
• Presence
• Instant Messaging
• PoC
• Access network independency
• DRM (Digital Rights Management)
• Developments to enhance user experience
• WLAN-3G cooperation

3.35. 3GPP Releases (3)

• Primary objectives of Release 6:
  • Improve capacity
  • Quality of Service (QoS), service enabler and delivery for multimedia packet-based services
  • All-IP network
  • Technology integration: 2G, 3G, WLAN, etc.
  • Collaboration with UMTS
    • billing, security, user authentication
  • The same session control layer (IMS) for all services.

3.36. 3GPP Releases (4)

• Release 7
  • Stage 1: December 2005; Stage 2: 2006; Stage 3: 2007
  • Uplink developments
  • Extended spectrum
  • Advanced Global Navigation Satellite System concept
  • IMS emergency call, e-call, etc..
  • HSPA+
  • AGCF (Access Gateway Control Function)
  • PES (PSTN Emulation Service)
• Release 8
  • 2008 Q4
  • First LTE release
  • SAE (System Architecture Evolution)
• EPC (Evolved Packet Core)
• New OFDMA, FDE and MIMO based radio interface
• Enhanced media session continuity
• IMS-centralised services

3.37. 3GPP Releases (5)

• Release 9
  • 2009 Q4
  • SAE enhancements
  • Femto cell support
  • Emergency call over GPRS and EPS (Evolved Packet System).
  • Enhancements to multimedia telephony
  • IMS media plane security
  • WiMAX and LTE/UMTS interoperability
  • Dual-Cell HSUPA
  • Dual-Cell HSDPA
  • Public Warning System (PWS)
  • Enhancements to services centralization and continuity.

• Release 10
  • 2011 Q1
  • LTE Advanced that meets the requirements of IMT-Advanced (International Mobile Telecommunications-Advanced) 4G
  • Multi-Cell HSDPA
  • Enhancements to the single radio voice call continuity (SRVCC)
  • Enhancements to IMS emergency sessions

3.38. 3GPP Releases (6)

• Release 11
  • 2012 Q3
  • Advanced IP Interconnection of Services
  • Service layer interconnection between national operators/carriers as well as third party application providers
  • USSD simulation service
  • Network-provided location information for IMS
• SMS submit and delivery without MSISDN in IMS.

• Release 12
  • 2014 Q2 (planned)
  • Service and Media Reachability for Users over Restrictive Firewalls (SMURFs).

### 3.39. UMTS Architecture R99

![UMTS Architecture R99 Diagram](image1)

### 3.40. UMTS Architecture - R4

![UMTS Architecture R4 Diagram](image2)

### 3.41. UMTS Architecture R5 - First appearance of IMS

![UMTS Architecture R5 Diagram](image3)
3.42. UMTS R6 IMS

3.43. 3GPP R7 Reference model
3.44. Architecture R8

3.45. 3GPP Releases
3.46. From GSM to LTE

*EUL (Enhanced Uplink) = HSUPA
3.47. Mobile network evolution

4. 4 CALL/SESSION CONTROL PROTOCOLS

4.1. Essential link control functions

- Call and connection management
- Registration, identification
- Call set-up
- Route management
- Features for users: call forwarding, caller identification etc.

4.2. Most important call/session control protocols

- H.323
- BICC (Bearer Independent Call Control)
- MGCP/Megaco/H.248
- SIP (Session Initiation Protocol)

4.3. H.323

4.4. H.323

- H.323 is a recommendation from the ITU Telecommunication Standardization Sector (ITU-T) that defines the protocols to provide audiovisual communication sessions on any packet network.
• It is widely implemented by voice and video conferencing equipment manufacturers, is used within various real-time applications on the internet, and is widely deployed worldwide by service providers and enterprises for both voice and video services over IP networks.

• Supports point-point and point-multipoint connections

4.5. H.323 architecture (1)

4.6. H.323 architecture (2)

• A typical H.323 network consists of zones connected through the World Wide Web

• Every zone has a Gatekeeper, a certain number of terminals, gateways, and a certain number of Multipoint Control Units in a local network.

• A zone can contain multiple number of LANs, the only requirement is that each zone can have only one Gatekeeper.

4.7. H.323 Terminal

• Multimedia endpoint which is capable of real-time, two-way communication

• It can communicate with other terminals, gateways and MCUs
• The communication type between two terminals can be control, audio, video and data

• Two terminals can communicate with each other:
  • Directly
  • Through the Gatekeeper

4.8. Gateway

• Allows cooperation between the different networks

• Works as a translator between different types of networks, in these cases, it builds up the session between two terminals

• For the translation, it has to know the used audio and video codecs

• Makes the necessary transformation in data formats and signaling formats

4.9. Gatekeeper

• Optional, but if present, it is the central intelligence

• If the network consists of multiple zones, then they are connected by Gatekeepers

• Tasks: addressing, admission control, authentication, bandwidth management, accounting, call forwarding

• If the network contains a Gatekeeper, the other elements have to register with it, and the elements are managed by the Gatekeeper

4.10. Multipoint Control Unit (MCU)

• A Multipoint Control Unit is responsible for managing multipoint conferences

• Maintains the audio, video, data sessions between the endpoints.

• Can be a separate element but usually it is integrated into the Gateway or the Gatekeeper.

• It is composed of two logical entities:
  • Multipoint Controller (MC): manages the signals between terminals, determines which audio and video streams have to be used
  • Multipoint Processor (MP): processes the streams

4.11. BICC

4.12. BICC (Bearer Independent Call Control)

• BICC is specified in ITU-T recommendation

• BICC is a signaling protocol based on N-ISUP that is used for supporting narrowband Integrated Services Digital Network (ISDN) service over a broadband backbone network.
• Separates the signaling plane and the media plane, so the signaling and media messages can go through different nodes

4.13. BICC architecture (1)

![BICC Architecture Diagram]

ISN: Interface Serving Node CSF: Call Service Function BCF: Bearer Control Function BF: Bearer Function
CMN: Call Mediation Node

4.14. BICC architecture (2)

• ISN (Interface Serving Node):
  • The network node that both receives and transmits signaling messages and media

• CMN (Call Mediation Node):
  • Located between two ISNs
  • Handles only signaling messages

• BF (Bearer Function):
  • Receives media packets and forwards them to the appropriate endpoint based on the instructions of the CSF

• CSF (Call Service Function):
  • Its main task is to manage calls
  • Cooperates with ISUP

• BCF (Bearer Control Function):
  • Receives requests from the CSF
  • Responsible for building up and tearing down routes

4.15. MGCP/MEGACO/H.248

4.16. MGCP/Megaco/H.248

• The protocol was the result of collaboration of the MEGACO working group of the Internet Engineering Task Force (IETF) and International Telecommunication Union Telecommunication Study Group
• Used between elements of a physically decomposed multimedia gateway

• It is a text-based, stateless protocol

• The Gateway and its control are separated

• Scalable

4.17. MGCP/Megaco/H.248 architecture (1)

4.18. MGCP/Megaco/H.248 architecture (2)

• MG (Media Gateway):
  • Converts various media streams between different telecommunication networks

• MGC (Media Gateway Controller):
  • Controls MG's resources
  • Responsible for call control and signaling

• SG (Signaling Gateway):
  • Receives and forwards signaling messages between nodes that use different protocols. So its main task is protocol conversion.

4.19. SESSION INITIATION PROTOCOL (SIP)

4.20. Session Initiation Protocol (SIP)

• Specified by the IETF as a protocol to establish and manage multimedia sessions over IP networks.

• SIP was accepted as a 3GPP signaling protocol and permanent element of the IP Multimedia Subsystem (IMS).

• SIP is an Application Layer protocol designed to be independent of the underlying Transport Layer.

• Widely used for controlling communication sessions such as voice and video calls over Internet Protocol. The protocol can be used for creating, modifying and terminating two-party (unicast) or multiparty (multicast) sessions. Sessions may consist of one or several media streams.

• Supports media negotiation and mobility.

• SIP employs design elements similar to the HTTP request/response transaction model.
4.21. SIP architecture

- Text-based protocol.

4.22. User Agent

- UAs (User Agents) are SIP endpoints that are usually handled by a user. In any case, user agents can also establish sessions automatically with no user intervention (e.g., a SIP voicemail). Sessions are typically established between user agents.

- User agents come in all types of flavors. Some are software running on a computer, others, like the commercial SIP phones, look like desktop phones, and others still are embedded in mobile devices like laptops, PDAs, or mobile phones.

- It can be UAS (User Agent Server) or UAC (User Agent Client)

- It behaves as UAS, if it receives requests from a UAC and responds them.

- It behaves as UAC, if it send requests to a UAS and processes the responses.

4.23. Proxy server (1)

- Proxy servers, typically referred to as proxies, are SIP routers.

- A proxy receives a SIP message from a user agent or from another proxy and routes it toward its destination.

- Routing the request involves relaying the message to the destination user agent or to another proxy along the path.

- Proxies are also useful for enforcing policy (for example, making sure a user is allowed to make a call). A proxy interprets, and, if necessary, rewrites specific parts of a request message before forwarding it.

- Most of the time the signals go through several proxies to reach the called party.

4.24. Proxy server (2)

There are two types of proxy servers:
Service Delivery Architectures and Platforms

- Stateless:
  - Simple and quick messaging
  - Doesn’t support transactions
  - Not able to retransmit the messages
  - Usage: for example load balancing

- Stateful:
  - Forking
  - Supports retransmission
  - Other functions: for example accounting

4.25. Registrar/Location server

- SIP registrars are logical elements, and are commonly co-located with SIP proxies.
- Receive the users’ registration requests, that contain the users’ location information. (e.g., IP address, port, user name).
- Stores the location information in the location database.

4.26. Redirect server

- Redirect servers are also used to route SIP messages, but they do not relay the message to its destination as proxies do. Redirect servers instruct the entity that sent the message (a user agent or a proxy) to try a new location instead.
- Collects the requested information from the location database.
- The redirect server allows proxy servers to direct SIP session invitations to external domains.

4.27. SIP and the IMS
4.28. Identification with SIP

- In a network of any kind, it must be possible to uniquely identify users.
- There are two types of identification:
  - Public User Identity
    - SIP URI (SIP Uniform Resource Identifier)
    - Tel URI (Telephone Uniform Resource Identifier)
  - Private User Identity

[fragile]

4.29. Public User Identities

- An IMS user is allocated with one or more Public User Identities.
- A Public User Identity is either a SIP URI or a TEL URI.
- In the IMS, Public User Identities are used to route SIP signaling.
- If we compare the IMS with GSM, a Public User Identity is to the IMS what an MSISDN (Mobile Subscriber ISDN Number) is to GSM.
• **SIP URI:**
  - Format: sip:Alice.Smith@domain.com
  - Can contain phone number: sip:+1-212-555-0293@domain.com;user=phone
  - This format is needed because SIP requires that the URI under registration be a SIP URI.
  - If the channel is secured, then the form of SIP URI is: sips:Alice.Smith@domain.com

• **Tel URI:**
  - TEL URI represents a phone number in international format:
    - tel:+1-212-555-0293
  - TEL URIs are needed to make a call from an IMS terminal to a PSTN phone.

### 4.30. Private User Identities

- Each IMS subscriber is assigned a Private User Identity. Unlike Public User Identities, Private User Identities are not SIP URIs or TEL URIs; instead, they take the format of NAI (Network Access Identifier, specified in RFC 2486). The format of a NAI is: Alice.Smith@domain.com

- Unlike Public User Identities, Private User Identities are not used for routing SIP requests; instead, they are exclusively used for subscription identification and authentication purposes.

- A Private User Identity performs a similar function in the IMS as an IMSI (International Mobile Subscriber Identifier) does in GSM.

- A Private User Identity need not be known by the user, because it might be stored in a smart card, in the same way that an IMSI is stored in a SIM (Subscriber Identity Module).

[fragile]

### 4.31. Identification example (1)

- SIP provides personal mobility. It means that users can be reached using the same identifier no matter where they are.

- For example, Alice can be reached at sip:Alice.Smith@domain.com regardless of her current location. This is her public URI, also known as her AoR (Address of Record).

- Nevertheless, when Alice is logged in at work her SIP URI is sip:asmith@ws1234.company.com

- And when she is working at her computer at the university her SIP URI is sip:alice@pc12.university.edu

- Therefore, we need a way to map Alice’s public URIs to her current URI (at work or at the university) at any given moment.

### 4.32. Identification example (2)

- To do this, SIP introduces a network element called the registrar of a particular domain. A registrar handles requests addressed to its domain.

- Every time Alice logs into a new location, she registers her new location with the registrar at domain.com, as shown in the figure. This way the registrar at domain.com can always forward incoming requests to Alice wherever she is.
4.33. Identification example (3)

- On reception of the registration the registrar at domain.com can store the mapping between Alice's public URI and her current location in two ways: it can use a local database or it can upload this mapping into a location server.

- If the registrar uses a location server, it will need to consult it when it receives a request for Alice.

- Note that the interface between the registrar and the location server is not based on SIP, but on other protocols.

[fragile]

4.34. Identification example (4)

- A given user can be available at several user agents at the same time.

- For instance, Alice can be reachable on her computer at the university: sip:alice@pc12.university.edu

- And on her PDA with a wireless connection: sip:alice@pda.com

- She has registered both locations with the registrar at domain.com.

4.35. Identification example (5)

- If the registrar receives a SIP message addressed to Alice's public URI it has to decide whether to route it to Alice's computer or to Alice's PDA.

- Alice has programmed the registrar to route SIP messages to her computer between 8:00 and 13:00 and to her PDA from 13:00 to 14:00. The registrar simply checks the current time and routes the SIP message accordingly.
• Being able to route SIP messages on the basis of any criteria is a very powerful tool for building services that are specially tailored to the needs of each user. Users typically choose to route SIP messages based on the sender, the time of the day, whether the subject is business-related or personal, the type of session (e.g., route video calls to the computer with the big screen), etc., the combinations are infinite.

4.36. Forking proxies (1)

• Sometimes it is useful to receive calls on several user agents at the same time. SIP proxy servers that route messages to more than one destination are called forking proxies.

• A forking proxy can route messages in parallel or in sequence.

4.37. Forking proxies (2)

• Redirect servers are also used to route SIP messages
4.38. Forking proxies (3)

- A user agent sends a SIP message to sip:Alice.Smith@domain.com
- The redirect server tells it to try the alternative address: sip:alice@pda.com
- Then the message arrives to Alice's PDA, according to the alternative address.

4.39. SIP call between different domains (1)
4.40. SIP call between different domains (2)

- (1) "A" calls "B" through a proxy server.
- (2) The proxy requests the redirect server about the location of "B's" proxy server.
- (3) The redirect server responds to the request.
- (4) "A's" proxy server informs "B's" proxy server that "A" initiated a call to "B".
- (5-6) "B's" proxy server requests the registrar about the location of "B".
- (7) "B's" proxy server informs "B" that it has an incoming call.
- (8-10) "B" responds "A" through the proxy server.

4.41. SIP protocol

- Each SIP transaction consists of a client request that invokes a particular method or function on the server and at least one response.
- SIP reuses most of the header fields, encoding rules and status codes of HTTP, providing a readable text-based format.
• SIP messages are identified by methods.

• Messages start with the start line, which is called the request line in requests and the status line in responses. The start line is followed by a number of header fields that follow the format name:value and an empty line that separates the header fields from the optional message body.

4.42. SIP protocol structure (1)

• Start line:
  • The start line in requests is referred to as the request line. It consists of a method name, the Request-URI, and the protocol version SIP/2.0. The method name indicates the purpose of the request and the Request-URI contains the destination of the request: INVITE sip:Alice.Smith@domain.com SIP/2.0
  • The start line of a response is referred to as the status line. The status line contains the protocol version (SIP/2.0) and the status of the transaction, which is given in numerical (status code) and in human-readable (reason phrase) formats: SIP/2.0 180 Ringing

• Message headers
  • Right after the start line, SIP messages (both requests and responses) contain a set of header fields. There are mandatory header fields that appear in every message and optional header fields that only appear when needed. A header field consists of the header field’s name, a colon, and the header field’s value. For example: To: Alice Smith <sip:Alice.Smith@domain.com>; tag=1234

4.43. SIP protocol structure (2)

• Message body
  • The message body is separated from the header fields by an empty line. SIP messages can carry any type of body and even multipart bodies using MIME (Multipurpose Internet Mail Extensions) encoding.

  • SIP uses MIME to encode its message bodies. Consequently, SIP bodies are described in the same way as attachments to an email message. A set of header fields provide information about the body: its length, its format, and how it should be handled.

  • For example, the header fields below describe the SDP session description. Content-Disposition: sessionContent-Type: application/sdpContent-Length: 193

  • The Content-Disposition indicates that the body is a session description, the Content-Type indicates that the session description uses the SDP format, and the Content-Length contains the length of the body in bytes.

4.44. Typical SIP request

INVITE sip:Alice.Smith@domain.com SIP/2.0
Via: SIP/2.0/UDP ws1.domain2.com:5060;branch=z9hG4bK74gh5
Max-Forwards: 70
From: Bob <sip:Bob.Brown@domain2.com>; tag=9hx34576sl
To: Alice <sip:Alice.Smith@domain.com>
Call-ID: 6328776298220188511@192.0.100.2
Cseq: 1 INVITE
4.45. Typical SIP response

SIP/2.0 200 OK
Via: SIP/2.0/UDP ws1.domain2.com:5060;branch=z9hG4bK74gh5;
    received=192.0.100.2
From: Bob <sip:Bob.Brown@domain2.com>;tag=9hx34576sl
To: Alice <sip:Alice.Smith@domain.com>;tag=1df345fkj
Call-ID: 632877629822018511@192.0.100.2
Cseq: 1 INVITE
Contact: <sip:alice@192.0.0.1>
Content-Type: application/sdp
Content-Length: 151

v=0
o=alice 2890844545 2890844545 IN IP4 192.0.0.1
s=--
c=IN IP4 192.0.0.1
t=0 0
m=audio 30000 RTP/AVP 0
a=rtpmap:0 PCMU/8000

4.46. SIP methods (1)

- **REGISTER**
  - The REGISTER method is usually the first method to be initiated by a device right after the device is turned on. The purpose of the REGISTER is to notify the network of the device’s location (IP address). This is so the network knows how to route messages to the device, so callers can find the subscriber wherever they are located.

- **INVITE**
  - The next most used method is the INVITE. This is used to establish a session within the IMS. Think of the INVITE as an invitation to another user to join in a conversation (or an e-mail, or an instant message, or whatever type of session is being set up).
  - Any time a device needs to establish a connection (virtual, of course), the INVITE is the method used to establish that connection.

4.47. SIP methods (2)

- **ACK**
When an INVITE has been sent, the sender awaits a response by the destination. However, a dialog is not established between the two entities until the originator of the INVITE sends the ACK method. This method is the final handshake required to establish the dialog and allow for the session to begin.

The ACK contains the same credentials as the INVITE and may also contain a message body carrying SDP or other content.

CANCEL

This method is used by a device when it wishes to cancel a request prior to receiving a response from the destination. For example, if a cell phone is attempting to set up a call and sends an INVITE, but the user then hangs up their phone immediately after dialing, the call is released and the session is ended. However, the session was never really established, because a response was never received by the originator.

4.48. SIP methods (3)

BYE

This is the simplest of methods. To release a session in progress, either device sends the BYE method to release the session. It can be initiated by either end of the session. Like all other methods used during a dialog, the BYE does require a response and acknowledgment prior to the session being released.

SUBSCRIBE

SUBSCRIBE is used by application servers to request updates from the S-CSCF and the HSS whenever a subscriber's registration changes. For example, a Presence server needs to know anytime a subscriber changes locations or activates another device in the network.

NOTIFY

NOTIFY is used by the S-CSCF to notify application servers (or any other entity) that have "subscribed" to event notification (registration updates) that a subscriber has changed his or her registration. The NOTIFY method will contain the changes that were made through the new registration.

4.49. SIP responses (1)

No request can be completed without a response. There are numerous types of responses. Each of the responses falls into one of six classifications, identified by a preceding number.

1xx: Provisional (e.g.: 100 Trying)

2xx: Successful (e.g.: 200 OK)

3xx: Redirect (e.g.: 300 Multiple Choices)

4xx: Client failure (e.g.: 403 Forbidden)

5xx: Server failure (e.g.: 503 Server Unavailable)

6xx: Global failure (e.g.: 600 Busy Everywhere)

The first digit of the number identifies the class of response, while the following two digits identify the specific response being given.

4.50. SIP responses (2)

100 Trying
• Indicates that the network is attempting to reach the destination (sent by a proxy to prevent retransmission by the requestor).

• 180 Ringing

• This is sent by the request recipient to indicate the request has been received and the device is alerting the subscriber.

• 200 OK

• This is sent to indicate that the receiver has accepted the request and the session can begin (once the ACK is sent by the requestor).

4.51. SIP responses (3)

• 401 Unauthorized

• This is sent by the S-CSCF to challenge a device when it first sends a REGISTER. The device will then send another REGISTER containing credentials per the registration process.

• 403 Forbidden

• This is used when a call is being rejected prior to a dialog being established. The dialog cannot be established until a 200 OK has been received as a response and the ACK has been sent as the final sequence in the handshake process. This response would be sent prior to the 200 OK.

• 407 Proxy Authentication Required

• Indicates that the user must first authenticate itself with the proxy server. If received by the user, the user may repeat the INVITE request with a suitable Proxy-Authorization field. This field should contain the authentication information of the user agent for the next outbound proxy or gateway.

4.52. SIP header fields (1)

• Via

• The Via header field keeps track of all the proxies a request has traversed. The response uses these Via entries so that it traverses the same proxies as the request did in the opposite direction.

• To

• The To header field contains the URI of the destination of the request. However, this URI is not used to route the request. It is intended for human consumption and for filtering purposes. For example, a user can have a private URI and a professional URI and filter requests depending on which URI appears in the To field. The tag parameter is used to distinguish, in the presence of forking proxies, different user agents that are identified with the same URI.

• From

• The From header field contains the URI of the originator of the request. Like the To header field, it is mainly used for human consumption and for filtering purposes.

4.53. SIP header fields (2)

• Contact

• The Contact header provides additional address information about a subscriber and is used to identify additional addresses that a request can be sent to in the event that the first request fails.
• Call-ID

• The Call-ID provides a unique identifier for all sessions, so proxies can correlate requests with responses. A single session may require a dialog with multiple entities, in which case the Call-ID is used by each of these entities as a reference.

• CSeq

• The CSeq header is used for ensuring proper sequencing of transactions during a dialog. The header, which consists of a decimal number followed by the method type, is used by endpoints to allow a device to track the proper response to the proper request.

4.54. SIP header fields (3)

• Max Forwards

• Max-Forwards is used to prevent looping of messages. When a message is created (either a request or a response), this header is set to 70. As the message passes through each proxy in the network, the value is decremented by one, until the value reaches 0. If a proxy receives a message with a Max-Forwards value of 0, the message is discarded.

• Content-Type

• This header identifies the type of content contained in the message body. For example, in an INVITE for a voice call, the message body will carry the SDP describing the session.

• Content-Length

• This header identifies the length of the message body, expressed in octets.

4.55. SIP header fields (4)

• Route and Record-Route

• Route header is used along with the Record-Route header when strict routing is implemented (as is the case in the IMS).

• When a request is being sent, the Record-Route header records the address of each of the entities in the call path. The response then inserts these addresses in the Route headers (there are typically multiple headers).

• The headers are listed in the order of the route. In other words, the addresses are shown in the same order they are routed through. The routers then use this for routing the responses to the next hop in the network.

• This form of strict routing ensures that a man-in-the-middle attack cannot be used to hijack a subscriber's registration, for example. It ensures that all requests and responses are sent through the same path used for the registration to reach the user.

4.56. SIP header fields (5)

• Authorization

• When a device sends a request such as an Invite or Register, it can include the Authorization header containing its credentials as part of the authorization process.

• If the Authorization header is not present, then the receiving endpoint within the IMS will send a 401 Not Authorized response. This is called the "challenge" and is sent by registrars and endpoints alike. The response will contain the WWW-Authenticate header describing the authentication scheme that is expected from the device.
• WWW-Authenticate
  
  • This header is used as a challenge to an entity sending a request. The challenge is carried in the response to a request with this header. The response will contain the Authenticate header containing the proper credentials.

4.57. SDP (SESSION DESCRIPTION PROTOCOL)

4.58. Session descriptions

• A session description is, as its name indicates, a description of the session to be established.

• It contains enough information for the remote user to join the session.

• In multimedia sessions over the Internet, this information includes the IP address and port number where the media needs to be sent and the codecs used to encode the voice and the images of the participants.

• Session descriptions are created using standard formats. The most common format for describing multimedia sessions is the Session Description Protocol (SDP), defined in RFC 2327.

• Note that although the "P" in SDP stands for "Protocol", SDP is simply a textual format to describe multimedia sessions.

[fragile]

4.59. SDP example

• The figure shows an example of an SDP session description that Alice sent to Bob.

• It contains, among other things, the subject of the conversation (Swimming techniques), Alice's IP address (192.0.0.1), the port number where Alice wants to receive audio (20000), the port number where Alice wants to receive video (20002), and the audio and video codecs that Alice supports (0 corresponds to the audio codec G.711 μ-law and 31 corresponds to the video codec H.261).

v=0
o=Alice 2790844676 2867992807 IN IP4 192.0.0.1
s=Let's talk about swimming techniques
c=IN IP4 192.0.0.1
t=0 0
m=audio 20000 RTP/AVP 0
a=sendrecv
m=video 20002 RTP/AVP 31
a=sendrecv

4.60. SDP architecture (1)

• SDP description consists of two parts:

  • session-level information

  • media-level information.

• Session-level information

  • The session-level information applies to the whole session and comes before the m= lines
• V: version
• O: user identifier
• S: subject of the session
• C: IP address
• T: time of the session

4.61. SDP architecture (2)

• Media-level information

• The media-level information is media stream-specific and consists of an m= line and a number of optional a= lines that provide further information about the media stream.

• The example has two media streams and, thus, has two m= lines. The a= lines indicate that the streams are bidirectional (users send and receive media).

• The format of all the SDP lines consists of type=value, where type is always one character long.

• Even if SDP is the most common format to describe multimedia sessions, SIP does not depend on it. SIP is session description format-independent. That is, SIP can deliver a description of a session written in SDP or in any other format.

4.62. SDP types
4.63. SDP - The Offer/Answer Model (1)

- In the SDP example, Alice sent a session description to Bob that contained Alice's transport addresses (IP address plus port numbers).

- Obviously, this is not enough to establish a session between them. Alice needs to know Bob's transport addresses as well.

- SIP provides a two-way session description exchange called the offer/answer model (which is described in RFC 3264).

- One of the users (the offerer) generates a session description (the offer) and sends it to the remote user (the answerer) who then generates a new session description (the answer) and sends it to the offerer.

- After the offer/answer exchange, both users have a common view of the session to be established. They know, at least, the formats they can use (i.e., formats that the remote end understands) and the transport addresses for the session.

4.64. SDP - The Offer/Answer Model (2)
The figure shows the answer that Bob sent to Alice after having received Alice's offer.

Bob supports the same audio and video codecs as Alice (G.711 μ-law and H.261). After this offer/answer exchange, all they have left to do is to have a nice video conversation.

5. 5 SESSION ESTABLISHMENT IN THE IMS

5.1. Contents

- Registration
- Session setup
- The Record-Route header
- The IETF SIP and the 3GPP SIP

5.2. Registration (1)

- Before an IMS terminal starts any IMS-related operation there are a number of prerequisites that have to be met.

  First, the IMS service provider has to authorize the end-user to use the IMS service. This typically requires a subscription or contract signed between the IMS network operator and the user.

  Second, the IMS terminal needs to get access to an IP-CAN (IP Connectivity Access Network) such as GPRS (in GSM/UMTS networks), ADSL (Asymmetric Digital Subscriber Line), or WLAN (Wireless Local Access Network).

  When these two prerequisites are fulfilled the IMS terminal needs to discover the IP address of the P-CSCF that will be acting as an outbound/inbound SIP proxy server.

  When the P-CSCF discovery procedure is completed the IMS terminal is able to send and receive SIP signaling to or from the P-CSCF.
5.3. Registration (2)

(1) The IMS terminal sends a Register message to the P-CSCF.

(2) The P-CSCF forwards the SIP REGISTER request to an I-CSCF in the home network.

(3-4) Messaging between I-CSCF and HSS (Diameter):

The HSS authorizes the user to roam the visited network and validates that the Private User Identity is allocated to the Public User Identity under registration.

Discovers whether there is an S-CSCF already allocated to the user.

5.4. Registration (3)

(1) The IMS terminal sends a Register message to the P-CSCF.

(2) The P-CSCF forwards the SIP REGISTER request to an I-CSCF in the home network.

(3-4) Messaging between I-CSCF and HSS (Diameter):

The HSS authorizes the user to roam the visited network and validates that the Private User Identity is allocated to the Public User Identity under registration.

Discovers whether there is an S-CSCF already allocated to the user.
(5) Then, the I-CSCF continues with the process by proxying the SIP REGISTER request to the chosen S-CSCF.

### 5.5. Registration (4)

(6-7) The S-CSCF then contacts the HSS for a double purpose: on the one hand, the S-CSCF needs to download authentication data to perform authentication for this particular user. On the other hand, the S-CSCF needs to save the S-CSCF URI in the HSS, so that any further query to the HSS for the same user will return routing information pointing to this S-CSCF. Users are authenticated by the S-CSCF with data provided by the HSS. These authentication data are known as authentication vectors.

(8-10) Then, the S-CSCF creates a SIP 401 (Unauthorized) response. This response includes a challenge in the WWW-Authenticate header field that the IMS terminal should answer.

(11-12) The response to the challenge (sometimes known as credentials) is included in a new SIP REGISTER request.

### 5.6. Registration (5)

(13-14) The I-CSCF sends a new Diameter UAR message, for the same reasons as explained before. The difference in this situation is that the Diameter UAA message includes routing information: the SIP URI of the S-CSCF allocated to the user. The HSS stored this URI when it received a Diameter MAR message (6). Therefore, no matter whether the I-CSCF is the same I-CSCF the first REGISTER request traversed or not, the second REGISTER request ends up in the same S-CSCF, the one that was allocated to the user at the time of the registration.

(15) The S-CSCF receives the REGISTER request that includes the user credentials. The S-CSCF then validates these credentials against the authentication vectors provided by the HSS in a Diameter MAA message (7).

### 5.7. Registration (6)

(16-17) If authentication is successful, then the S-CSCF sends a Diameter SAR message to the HSS for the purpose of informing the HSS that the user is now registered and to download the user profile. The user profile is an important piece of information that includes, among other things, the collection of all the Public User Identities allocated for authentication of the Private User Identity. It also indicates to the S-CSCF which of these Public User Identities are automatically registered in the S-CSCF in a set of implicitly registered Public User Identities.

(18-20) Last, but not least, the S-CSCF sends a 200 (OK) response to the REGISTER request, to indicate the success of the REGISTER request. The 200 (OK) response includes a P-Associated-URI header field that contains the list of URIs allocated to the user. It also contains a Service-Route header field that includes a list of SIP server URIs. The 200 (OK) response traverses the same I-CSCF and P-CSCF that the REGISTER request traversed.

Eventually, the IMS terminal gets the 200 (OK) response. At this stage the registration procedure is complete.

### 5.8. Registration (7)

- When the IMS terminal has completed its registration, it sends a SUBSCRIBE request for the reg event. This request is addressed to the same Public User Identity that the SIP User Agent just registered. The S-CSCF receives the request and installs that subscription.

- The S-CSCF sends a NOTIFY request to the user. This request includes an XML document in its body that contains a list of all the Public User Identities allocated to the user, along with the user's registration state. The IMS terminal now knows whether the user is registered or not, and with which Public User Identities the user is registered.
In case the S-CSCF has to shut down or if the operator needs to manually deregister a Public User Identity, the registration information will be changed. This will provoke the S-CSCF into informing each subscriber of the registration event of that user.

5.9. Session setup (1)

5.10. Session setup (2)

5.11. Session setup (3)
5.12. Session setup (4)

- We are assuming that both users are roaming to a network outside their respective home networks, such as when both users are outside their respective countries. This leads to having two different visited networks in the figures. We also assume that each of the users has a different business relationship with his respective operator; therefore, there are two different home networks in the figures. Additionally, we assume that the P-CSCF is located in a visited network.

- For the sake of simplicity, we assume that neither the calling party nor the called party have any services associated with the session.

- For the sake of clarity we refer to the originating P-CSCF and originating S-CSCF as the P-CSCF and S-CSCF that are serving the caller. Similarly, we refer to the terminating P-CSCF and terminating S-CSCF as the P-CSCF and S-CSCF that are serving the callee.

5.13. Session setup (5)

(1-14) INVITE and 100 Trying:

- The IMS Terminal Sends an INVITE Request to the associated P-CSCF.

- The message may contain service starting information and location information (e.g. Cell-Id).

- Invite message can contain SDP in which the terminals negotiate codecs.

- The P-CSCF receives the INVITE request and does not need to take any routing decision, because the Request-URI already contains a SIP URI that includes the IP address (or host name) of the IMS terminal.

- The P-CSCF inspects the content of the message:
  - Routing information
  - Codecs
  - Inserts charging headers.

- 100 Trying is a provisional response.

5.14. Session setup (6)
• The S-CSCF allocated to the caller receives the INVITE request and examines the P-Asserted-Identity header to identify the user who originated the INVITE request.

• The S-CSCF downloaded the user profile at registration. Among other information, the user profile contains the so-called filter criteria. The filter criteria contain the collection of triggers that determine whether a request has to traverse one or more Application Servers that provide services to the user.

• Once the S-CSCF has taken a determination of how to route the INVITE request, it adds a new value to the existing P-Asserted-Identity header field.

• The S-CSCF also policies the SDP for user-related information. This allows operators, for instance, to offer cheap subscriptions that do not allow the use of certain media streams (e.g., video) or certain premium codecs. In case the request does not fit with the policy the S-CSCF may not process the INVITE request and answer it with a 488 (Not Acceptable Here) response indicating the media types, codecs, and other SDP parameters which are allowed according to the policy.

5.15. Session setup (7)

• The INVITE request is received at the callee's IMS terminal. The INVITE carries an SDP offer generated at the caller's terminal. The SDP offer indicates the IP address and port numbers where the caller wants to receive media streams, the desired and supported codecs for each of the media streams, etc.

• We expect IMS terminals to be configured to answer this INVITE request automatically without requiring any interaction with the user. For instance, users may have configured their IMS terminals to automatically accept sessions that require support for the AMR audio codec and the H.263 video codec.

• The callee's IMS terminal could start its resource reservation process at this stage, because it knows all the parameters needed (all included in the SDP) to start resource reservation.

5.16. Session setup (8)

(15-20) 183 Session Progress:

183 (Session Progress) is a provisional response that the callee's IMS terminal sends back to the originating side.

The terminal inserts a Require header field with the value 100rel. This is required because the precondition extension requires that provisional responses be transmitted reliably.

The precondition extension requires that SDP be included in a provisional response and that support for the reliable provisional responses mechanism is given to guarantee provisional responses are received at the originating terminal.

(21-30) Prack and 200 OK

The "Reliability of Provisional Responses in SIP" (RFC 3262) guarantees that the reception of a provisional response is acknowledged by a PRACK request. If the callee does not receive a PRACK request to confirm the reception of the provisional response within a determined time, then it will retransmit the provisional response.

In parallel with the generation of the PRACK request the IMS terminal starts the resource reservation mechanism.

The 200 (OK) response is an answer to the PRACK request and should not be confused with a 200 (OK) for the INVITE that will occur later.

5.17. Session setup (9)
(31-40) Update and 200 OK:

The IMS terminal sends an UPDATE request that visits the same set of proxies as the PRACK request.

The UPDATE request contains another SDP offer, in which the caller's IMS terminal indicates that resources are reserved at his local segment.

Eventually, the callee's IMS terminal receives the UPDATE request (35). The IMS terminal will generate a 200 (OK) response.

(41-56) 180 Ringing, Prack and 200 OK:

When the callee's IMS terminal rings, it will also generate a 180 (Ringing) provisional response.

When the caller's IMS terminal receives the 180 (Ringing) response, (46) it will likely generate a locally stored ring-back tone to indicate to the caller that the peer terminal is ringing.

As the 180 (Ringing) response requires acknowledgement the caller's IMS terminal generates a PRACK request and the caller's terminal will answer it with a 200 (OK) response (52).

5.18. Session setup (10)

(57-67) 200 OK and Ack:

When the callee finally accepts the session the IMS terminal sends a 200 (OK) response (57), completes the INVITE transaction.

The caller's IMS terminal also sends an ACK request, (63) in Figure 5.20, to confirm receipt of the 200 (OK) response. The ACK request is routed back to the callee's IMS terminal.

- The session setup is complete, and both users can generate their respective audio and video media streams. These media streams are in general sent end to end (e.g., from the caller to the callee's IMS terminals and vice versa) via IP-CAN routers.

- If one of the parties wants to stop the session then it sends a BYE message, the other party responds with a 200 (OK) message.

5.19. The Record-Route header (1)
Session establishment through a proxy

5.20. The Record-Route header (2)

- When a SIP dialog is established (e.g., with an INVITE transaction), all the subsequent requests within that dialog follow the same path. In our example, all the requests after the INVITE (the ACK (5) and the BYE (6)) are sent end to end between the user agents.

- However, some proxies choose to remain in the signaling path for subsequent requests within a dialog instead of routing the first INVITE request and stepping down after the 200 (OK) response.

- The figure on the next slide shows a message flow where the proxy at domain.com remains in the path for all the requests sent within the dialog. The proxy requests to remain in the path by adding a Record-Route header field to the INVITE request (2).

5.21. The Record-Route header (3)
5.22. The Record-Route header (4)

- Alice obtains the Record-Route header field with the proxy’s URI in the INVITE request (2), and Bob obtains it in the 200 (OK) response (4).

- From that point on, both Bob and Alice insert a Route header field in their requests, indicating that the proxy at domain.com needs to be visited.

- The ACK (5 and 6) is an example of a request with a Route header field sent from Bob to Alice.

- The BYE (7 and 8) shows that requests in the opposite direction (i.e., from Alice to Bob) use the same Route mechanism.

5.23. The IETF SIP and the 3GPP SIP (1)

- IETF defines protocols like: SIP, SDP, RTP, Diameter

- The IETF SIP is user-centric, the most important objectives of the network element is the routing, and the call control is subsidiary.
• 3GPP defines the usage of IETF protocols in the 3GPP architecture

• The 3GPP SIP is network-centric, the operators want to control the access to the network, session establishment, termination of sessions, billing, etc.

5.24. The IETF SIP and the 3GPP SIP (2)

• The cause of the problems between 3GPP and IETF is that they try to enable inner network elements to control SIP between endpoints.

• The additional requirements of 3GPP for SIP protocol:
  • UMTS-AKA based authentication
  • Operator initiated session termination
  • Operator initiated re-authentication
  • Path, P-Access-Network-Info, etc.

5.25. Session establishment according to RFC 3261 (IETF)

IMS
5.26. Session establishment according to 3GPP

IMS
6. MEDIA TRANSPORT

6.1. Contents
• Transport protocol and their role in media communications
• Datagram Congestion Control Protocol (DCCP)
• Real-time transport protocol (RTP)
• RTP Control Protocol (RTCP)
• Media Transport in the IMS

6.2. Transport protocols

• Two types of media: media that tolerates a certain degree of packet loss and media that does not. Examples of the first type are audio and video and examples of the second type are web pages and instant messages.

• TCP can be used for reliable media transport while UDP offers unreliable media transport.

• UDP is not suitable for transporting large amounts of data traffic because it lacks congestion control mechanisms (i.e., congestion control would need to be implemented at the application layer, but it is seldom implemented at all).

6.3. Reliable Media Transport

• There are two transport protocols that provide reliable delivery of user data: TCP and SCTP (Stream Control Transmission Protocol, RFC 2960).

• TCP delivers byte streams, while SCTP delivers messages.

• TCP works best when the recipient of the data does not need to wait until all the data have arrived in order to start processing it. An Internet browser is a good example of such a case.

• When the application is interested in receiving all the data at once, SCTP is a better choice.

• SCTP provides features that are not present in TCP, such as better protection against DoS attacks, multi-homing, and multiple streams per SCTP association (connections are referred to as associations in SCTP).

• Although SCTP provides some advantages over TCP for some applications, it still is not widely used.

6.4. Unreliable Media Transport

• UDP is the transport protocol used to send media unreliably.

• Since lost packets are not retransmitted, applications introduce redundancy in the data to be transferred in order to be able to tolerate a certain level of packet loss.

• It has a big disadvantage when it comes to transporting large amounts of data: it does not provide any congestion control mechanism. UDP senders do not slow down sending data even when the network is severely congested, making the congestion even worse.

• The increase in the number of applications using UDP to send media, especially audio and video, made the IETF realize that we need an unreliable transport protocol that includes congestion control.

• DCCP (Datagram Congestion Control Protocol) supports different types of congestion control and has been designed with multimedia traffic in mind.

6.5. Datagram Congestion Control Protocol (DCCP)
• DCCP is an unreliable transport protocol that provides connection establishment and termination as well as negotiation of congestion control algorithms.

• DCCP connections are established with a three-way handshake, during which the characteristics of the connection are negotiated. In particular, DCCP peers negotiate the congestion control algorithm to be used.

• Congestion control algorithms are identified by their CCIDs (Congestion Control Identifiers). At present, there are three CCIDs to choose from:
  • sender-based congestion control
  • TCP-like congestion control
  • TFRC (TCP-Friendly Rate Control) congestion control

• DCCP is still a fairly new protocol and, so, is not widely used at present.

6.6. RTP and RTCP

• What is RTP ad what is not:
  • Carries media information in its payload
    • Media calls are handled by H.323 or SIP
  • Operated over a transport protocol
  • Supports transfer of media streams
  • Does not provide QoS

• RTCP:
  • A tool for monitoring QoS between endpoint

• Use UDP with a port couple:
  • RTP: an even numbered port
  • RTCP: the following odd numbered one

• RTP, RTCP: specified in RFC 1889

6.7. RTP services

• Identification of payload types (media or content types)

• Numbering

• Time stamp

• RTCP:
  • Supports (but does not implement) QoS assurance
  • Synchronization among media streams

• 5% of the total "bandwidth" (available data rate) is assigned to RTCP, the rest to RTP

6.8. RTP: packet header format (1)
Version (V, 2 bits)

Version number of RTP as defined in RFC 1889

Padding (P, 1 bit)

If 1: there is padding at the end of the RTP packet

Padding - last byte: how many bytes are padding bytes, should be dropped

Extension (X, 1 bit)

If 1: there will be a variable length header extension

If there is a header extension: the first 2 bytes specify its length

Extension follows the last valid field of fixed header

### 6.9. RTP: packet header format (2)

- CSRC count (CC, 4 bits)
  - Number of CSRC identifiers = number of multiplexed sources (specification of sources: in the CSRC field)
  - For a single source: CC = 0

- Marker (M, 1 bit)
  - Marking of significant events in the packet stream
  - examples:
    - Frame boundaries in various compression methods
• Beginning/end of active periods in the speech
• The actual interpretation is defined in "profile"

6.10. RTP: packet header format (3)

<table>
<thead>
<tr>
<th>V=2</th>
<th>P</th>
<th>X</th>
<th>CC</th>
<th>M</th>
<th>PT</th>
<th>Sequence number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronization Source Identifier (SSRC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contributing Source Identifiers (CSRC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Payload type (PT, 7 bit)
  • "profile", assigns payload formats to different types of media coding
• Sequence number (16 bit)
  • Allows for detecting lost packets and restoring original packet sequence
  • Its initial value is a random number (we are going to refer to it later) and incremented by 1 after every RTP packet sent
• Timestamp (32 bit)
  • Real time value of the first byte of RTP packet in the media stream

6.11. RTP: packet header format (4)

<table>
<thead>
<tr>
<th>V=2</th>
<th>P</th>
<th>X</th>
<th>CC</th>
<th>M</th>
<th>PT</th>
<th>Sequence number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronization Source Identifier (SSRC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contributing Source Identifiers (CSRC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• SSRC (32 bit)
  • Identifies the source of the RTP packet stream. A random value assigned by RTCP.
• CSRC (0...15-ször 32 bit)
  • Contributing source; identifies a component of the combined packet stream created by "RTP mixer"

6.12. RTP "profiles"

= Payload formats, corresponding to different coding methods.

Examples for speech transmission (more are defined in the RFC):
Examples of video payloads: 26: JPEG, 33: MPEG2

6.13. SSRC, Synchronization source

- Identifies the source of the stream
  - Random number, unique within the session
  - E.g. microphone, camera or mixer
- If a source generates more than one streams within the session: each gets different SSRC
  - E.g. Multiple cameras
- The receiver selects packets into one stream based on SSRC


- The main purpose of RTP is to allow receivers to play out media at the proper pace, given that IP networks do not keep the timing relationship of the data being transported: that is, IP networks introduce jitter. They use RTP timestamps for this purpose.
- Receivers place incoming RTP packets in a buffer according to their timestamps and start playing them. If a packet with a particular timestamp needs to be played and still has not arrived, the receiver uses interpolation techniques to fill the gap.

6.15. Using RTP (2)

- The receiver in the figure needs to make an important decision: when to start playing media to the user.
- If it starts as soon as the packet with timestamp 0 is received, it runs the risk that the packet with timestamp 20 might not arrive in time to be played.
• On the other hand, if the receiver waits many seconds before playing the media, the delay would be so big that the users would be unable to maintain a normal conversation, and it would have to implement a much larger buffer.

6.16. Using RTP (3)

• Different implementations use different parameters to decide the lengths of their buffers: long buffers cause long delays but good quality while short buffers cause short delays but poor quality.

• The graph in the figure shows the delay experienced by packets sent from a sender to a receiver. In this example, most of the packets experience a delay of around 50 ms, some experience smaller delays, and a few experience much larger delays. A good tradeoff for the receiver would be to start playing packets 100 ms after they are sent. This way, only the few packets that appear in the tail of the distribution (which have a too long delay) would be discarded when they arrive.

6.17. Using RTP (4)

• In addition to timestamps, RTP packets carry sequence numbers.

• If the network drops too many packets at a particular time, peers may decide to use a different codec, one that provides better quality under heavy packet loss (i.e., a codec with more redundancy).

• RTP packets also carry binary sender identifiers and the payload type.

• Binary sender identifiers are used in conferences to identify the current speaker, and the payload type identifies the encoding and transport format of the data carried in the RTP packet.

• Payload types are numeric values that identify a particular codec and are typically negotiated (or simply assigned) using a session description protocol.

6.18. RTP Control Protocol (RTCP)

• It provides quality-of-service statistics, information to perform inter-media synchronization, and mappings between RTP binary sender identifiers and human-readable names.

• RTCP messages are sent by both RTP senders and RTP receivers. To develop quality-of-service statistics, RTP senders report (using RTCP) the number of RTP packets they have sent to the network and RTP receivers report the number of packets they have received.
• RTP senders use RTCP to provide a mapping between the timestamps of their media streams and a wall clock. This way, receivers can synchronize the play-out of different media streams. A common example of this type of inter-media synchronization is lip sync; that is, audio-video synchronization.

• RTCP messages also provide mappings between RTP binary sender identifiers and human-readable names.

• When SDP is used, RTP packets are normally sent a port with an even number and RTCP messages are sent to the consecutive odd port.

### 6.19. Secure RTP (SRTP)

• SRTP (RFC 3711) provides confidentiality, message authentication, and replay protection to RTP and RTCP traffic. The figure shows which portions of an RTP packet are authenticated and which are encrypted.

• Peers using SRTP to exchange media use a key management protocol to come up with a master key, which is used to generate session keys. Session keys are typically refreshed periodically so that attackers do not have access to large amounts of traffic encrypted under the same key.

### 6.20. Media Transport in the IMS

• The IMS uses RTP over UDP to transport media unreliably. DCCP may be used in the future, but at present it is not mature enough and widespread enough for the IMS.

• Regarding security, the IMS does not provide any kind of security at the media level. It is assumed that the traffic on the radio access is encrypted at lower layers and that the core IMS network is trustworthy. So, SRTP is not supported.

• When it comes to reliable transport protocols the natural choice would be to use TCP. It has been a stable protocol for many years and is supported virtually everywhere.

### 7. 7 ROAMING

#### 7.1. Roaming basics (1)

• IMS reuses the same concept of having a visited and a home network.

• In order to use a visited network, the visited network operator has to have a signed roaming agreement with our home network operator.

• In these agreements both operators negotiate some aspects of the service provided to the user, such as
  • price of calls,
  • quality of service,
• how to exchange accounting records.

7.2. Roaming basics (2)

• When the IP-CAN (IP Connectivity Access Network) is GPRS, the location of the P-CSCF is subordinated to the location of the GGSN. In roaming scenarios, GPRS allows location of the GGSN either in the home or in the visited network (the SGSN is always located in the visited network).

• Most of the IMS nodes are located in the home network, but there is a node that can be either located in the home or the visited network. That node is the P-CSCF (Proxy-CSCF).

7.3. Roaming and IMS

• In the IMS, both the GGSN and the P-CSCF share the same network. This allows the P-CSCF to control the GGSN over the so-called Go interface.

• As both the P-CSCF and the GGSN are located in the same network the Go interface is always an intra-operator interface, which makes its operation simpler.

• The SGSN is always located in the visited network.

• The IMS allows two different configurations, depending on whether the P-CSCF is located in the home or the visited network.

7.4. Roaming configurations (1)

• The figure shows a configuration where the P-CSCF (and the GGSN) is located in the visited network. This configuration represents a longer-term vision of the IMS, because it requires IMS support from the visited network (i.e., the GGSN has to be upgraded to be at least 3GPP Release 5-compliant).

7.5. Roaming configurations (2)

• The figure shows a near-term configuration where both the P-CSCF and the GGSN are located in the home network.
• This configuration does not require any IMS support from the visited network.
• Particularly, the visited network does not need to have a 3GPP Release 5-compliant GGSN.
• The visited network only provides the radio bearers and the SGSN.
• So, this configuration can be deployed from the very first day of the IMS.

7.6. Roaming configurations (3)

• This configuration has a severe disadvantage with respect to the configuration where the P-CSCF and GGSN are located in the visited network.
• Since the media plane traverses the GGSN and the GGSN is located in the home network the media are first routed to the home network and then to their destination.
• This creates an undesired trombone effect that causes delays in the media plane.

7.7. GPRS Roaming (1)

• Roaming in the case of IMS does not necessarily differ from normal GPRS roaming from inter-PLMN network point of view because IMS traffic is transferred on top of GTP tunnel, as any other data.
• When an IMS subscriber is located at his/her home GPRS/IMS network he/she will normally access IMS services through a home network Access Point (GGSN).

7.8. GPRS Roaming (2)

• In a roaming scenario, the subscriber either uses a visited network Access Point or a home network Access Point.
• The home network can restrict use of visited network Access Point.
• The figure on the next slide shows the usual scenario where the user has chosen to use a home network AP to access IMS services.
• This is a normal GPRS roaming situation where GGSN is in the home network and GTP tunneling is used across the inter-operator backbone network (e.g., GRX).
• The visited network doesn't even have to have an IMS domain in this case.
7.9. GPRS Roaming (3)

• The figure on the next slide presents the scenario where the roaming subscriber uses the visited IMS network.
• In this case, the IP address assigned to the IMS terminal is from the visited network-addressing domain.
• I-CSCF is a contact point between visited and home networks.
• In the figure, I-CSCF is only used in home network.
• Thus, P-CSCF of visited network discusses directly with I-CSCF of home network.

7.10. GPRS Roaming (4)

• It is likely that the scenario where GGSN is in the home network is the preferred model also for IMS roaming, at least in the first phase.
• Therefore, it should be beneficial for operators to concentrate on this scenario, for now.
Using the visited IMS network can offer some benefits, but implementing it is likely feasible later, when the level of real-time traffic increases and therefore issues such as improved QoS control and optimizing routing will become more important.

8. 8 QUALITY OF SERVICE (QOS) IN THE IMS

8.1. QoS on the Internet

- Although the Internet has been traditionally a best-effort network, the ability to provide a certain level of QoS for certain packet flows is essential for some applications.

- QoS is not only about requesting a better treatment for certain flows; users also want to know if the network will be able to provide them with the requested QoS. If there is a long delay or a high packet loss rate, some users may prefer to exchange instant messages instead of having a VoIP (Voice over IP) conversation.

- There are two models to provide QoS on the Internet: the Integrated Services model and the Differentiated Services (DiffServ) model.

8.2. Integrated Services (1)

- The Integrated Services architecture (specified in RFC 1633) was designed to provide end-to-end QoS.

- Endpoints request a certain level of QoS for their packet flows and, if the network grants it, their routers treat those flows accordingly.

- There are two different services available in this architecture:
  - Controlled load service
  - Guaranteed service

8.3. Integrated Services (2)

- Controlled load services: ensures that packets are treated as if the network was under moderate load. Flows using this service are not affected by network congestion when this appears. Nevertheless, the network does not guarantee a certain bandwidth or a certain delay. This service can be seen as a better-than-best-effort service.

- Guaranteed service: guarantees a certain bandwidth or a certain delay threshold. In practice, it is not common to see this service in use because the controlled load service is often good enough and is easier to manage.

8.4. Resource ReSerVation Protocol (RSVP) - 1

- The Integrated Services architecture uses RSVP (RFC 2205) as the resource reservation protocol.

- Endpoints send RSVP messages requesting a certain QoS (e.g., a certain bandwidth) for a flow.

- RSVP messages need to follow the same path as the packets of the flow. (e.g., RTP packets carrying voice).

- An RSVP reservation consists of a two-way handshake.

- The PATH message is sent from endpoint A to endpoint B, and the network routes it as for any other IP packet. At a later time, when endpoint A sends RTP packets with voice to endpoint B they will follow the same path as the PATH message did.
8.5. Resource ReSerVation Protocol (RSVP) - 2

- PATH messages store the nodes they traverse. This allows RESV messages to be routed back to endpoint A, following the same path as the PATH message followed but in the opposite direction.

- A PATH message is sent and a RESV message is received, as shown in figure.

8.6. Resource ReSerVation Protocol (RSVP) - 3

- Resource reservation actually takes places when routers receive RESV messages.

- The packets of the flow follow the same path as the PATH message as long as there are no routing changes in the network. If, as a consequence to a change in the network topology, packets from endpoint A to endpoint B start following a different path, a new resource reservation is needed. RSVP tackles this using soft states.

- Reservation soft states created by RESV messages are kept in routers only for a period of time. If they timeout before they are refreshed by a new RESV message, routers just delete them.

- Endpoints periodically exchange PATH and RESV messages while the flow (e.g., the RTP packets) is active.

- A router can reject a resource reservation request either because the router does not have enough resources or because the user is not allowed to reserve them.

8.7. Differentiated Services (DiffServ)

- The main problem with the integrated services architecture is that the network needs to store a lot of state information. When a packet arrives at a router, the router needs to check all the reservations it is currently handling to see whether the packet belongs to any of them. This means that routers need to store state information about every flow and need to perform lookups before routing any packet.

- The DiffServ architecture (RFC 2475 and RFC 3260) addresses some of the problems in the integrated services architecture.

- DiffServ routers know what treatments a packet can get. These treatments are referred to as Per-Hop Behaviors (PHBs).
PHBs are identified by 8-bit codes called Differentiated Services Codepoints (DSCPs). IP packets are marked at the edge of the network with a certain DSCP so that routers in the path apply the correct PHB to them.

8.8. The main idea of DiffServ (1)

- Solution is needed for the network core, which carries large volumes of aggregated traffic
- DiffServ assigns resources to a small number of traffic classes:
  - Such traffic classes are e.g.:
    - Premium
    - Regular
- Instead of notifying the routers about the QoS demands of a flow (e.g. by using RSVP),
- A single bit in the packet header will do that!

8.9. The main idea of DiffServ (2)

- Well, but how this idea would do the job?
  - Who should set the single "premium" bit and where?
    - Do it at the "administrative" boundary
  - What shall a router do, how it should behave differently, when it gets a packet in which this bit is set to premium?
    - Follows different behaviors, as defined in IETF DiffServ WG
- But let us see first how DiffServ works!

8.10. How DiffServ works

8.11. Marking packet in the edge router
8.12. Marking, classification and "conditioning"

- Observing the traffic (meter) and packets not in conformance are either
  - Shaped or
  - Dropped

8.13. DiffServ - "behavior" of nodes

- Per-Hop-Behavior (PHB): defines the way of forwarding packets belonging to a specific traffic class
  - Implemented in DiffServ capable routers
  - No cooperation between endpoints, as opposed to IntServ
  - PHBs are defined by 6 bits in the ToS field (IPv4)
    - IPv6: similarly, using Traffic Class
  - These 6 bits are called "DiffServ Code Points" (DSCP)
Remaining 2 bits: Explicit Congestion Notification

Two main PHB types:

- "expedited forwarding" (EF)
- "assured forwarding" (AF)

Plus a default PHB: best effort, everything is assigned to it which does not comply with the previous two

8.14. Placing of "DiffServ Code Points" (DSCPs)

IPv4:

<table>
<thead>
<tr>
<th>Version</th>
<th>Hdr length</th>
<th>Type of Service (ToS)</th>
<th>Total length</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 bit</td>
<td>4 bit</td>
<td>8 bit</td>
<td>16 bit</td>
</tr>
</tbody>
</table>

6 bit DiffServ-re:
64 különböző PHB

IPv6:

Version | Traffic class | Flow label

8.15. DiffServ - expedited forwarding (EF)

Simplest PHB:
- Forwarding packets occurs
  - With minimum delay
  - With low packet loss rate

It is good if the forwarding rate of the EF traffic is restricted only by the link data rate

Schedulers should give priority to EF traffic against other traffic types ("expressz" forwarding) thus guaranteeing the QoS parameters

Applications: beszéd, videó

EF traffic cannot be too large, otherwise QoS parameters cannot be guaranteed
  - Strict admission control needed
  - Service provider restricts the amount of EF traffic in the network (usually 30% of the total)

8.16. DiffServ - assured forwarding (AF)

12 sub-classes ($AF_{x:y}$)
• See the table on the next slide

• Thus total 14 kinds of service, according to the 6 bits of the DSCP field:
  • Assured Forwarding DSCP: according to the table
  • Expedited Forwarding DSCP: 101 110
  • Best Effort (Default PHB): 000 000

**8.17. Assured forwarding (AF) sub-classes**

<table>
<thead>
<tr>
<th>Drop Precedence</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>010 000</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Medium</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>High</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**8.18. DiffServ - advantages and problems**

• Since DiffServ is working in the third layer, can be based on any 2nd layer - IP-capable - infrastructure

• No signaling is needed

• No per-packet flow status management is needed in the nodes

• Complexity is shifted to the edges of the network (to access networks)

Problems, e.g.:

• Cannot ensure QoS alone
  • Different DiffServ domains, service provider
  • DiffServ-incapable nodes in the path

**8.19. QoS in the IMS (1)**

• One of the main features of the IMS is the highly configurable QoS.

• The most important factors for the end user are the bandwidth and the quality of the connection.

• The IMS allows operators controlling QoS, and thus discriminating certain groups of users.

• The IMS supports several end-to-end QoS models, terminals may use:
  • Link-layer resource reservation protocols (e.g., PDP context activation)
  • IntServ
  • DiffServ

• The most common model is to have terminals use link-layer protocols and to have the GGSN map link-layer resource reservation flows to DiffServ codes in the network.
8.20. QoS in the IMS (2)

- Terminals need to be able to map the media streams of a session into resource reservation flows. A terminal that establishes an audio and a video stream may choose to request a single reservation flow for both streams or to request two reservation flows, one for video and one for audio. Requesting a reservation flow may consist of creating a secondary PDP context or sending RSVP PATH messages, for instance.

- The P-CSCF instructs the terminal to perform resource reservation. To do so the P-CSCF uses the SRF (Single Reservation Flow) semantics (specified in RFC 3524) of the SDP grouping framework.

- The SDP grouping framework (RFC 3388) allows us to group media streams and to describe the semantics of the group. For example, LS (Lip Synchronization) semantics indicate that the play-out of media streams in the group need to be synchronized. LS semantics are typically used to group an audio and a video stream.

- The a=group line carries the semantics of the group (LS, SRF) and the identifiers of the streams (the a=mid line in the streams).

8.21. QoS in the IMS (3)

- SRF semantics indicate that all the streams in the group should use the same resource reservation flow. Consequently, the two audio streams of the session description in the figure would use the same PDP context (assuming a GPRS access), while the video stream would use its own PDP context.

```
v=0
o=-- 289083124 289083124 IN IP6 1080::8:800:200C:417A
t=0 0
c=IN IP6 1080::8:800:200C:417A
a=group:SRF 1 2
a=group:SRF 3
m=audio 20000 RTP/AVP 0
a=mid:1
m=audio 20002 RTP/AVP 0
a=mid:2
m=video 20004 RTP/AVP 31
a=mid:3
```

8.22. QoS in the IMS (4)
8.23. QoS in the IMS (5)

- When the access network is GPRS a resource reservation flow is a PDP context.
- The information stored by the network regarding this PDP context includes the terminal's IP address and the PDP context's QoS characteristics including its traffic class. There exist four traffic classes:
  - Best effort
  - Interactive
  - Streaming
  - Conversational
- The PDP contexts used for SIP signaling are always conversational.

8.24. PDP context activation

8.25. Secondary PDP context activation

- IMS terminals establish additional PDP contexts to send and receive media, as shown in the figure. The number of additional PDP contexts, referred to as secondary PDP contexts, depends on the instructions
received from the P-CSCF in the form of a=group:SRF lines. Secondary PDP contexts use the same IP address as the primary PDP context, but may have different QoS characteristics.

![Diagram showing IMS Terminal, SGSN, and GGSN with activation of secondary PDP context request and accept steps.]

8.26. QoS in the network

- The GGSN receives traffic from a given terminal over a PDP context, assigns it an appropriate DSCP (Differentiated Services Codepoint), and sends it out into a DiffServ-enabled network, as shown in the figure. In short, the GGSN implements the DiffServ edge function.

![Diagram showing PDP Context, GGSN, and DiffServ Network interaction.]

9. 9 AUTHENTICATION, AUTHORIZATION, ACCOUNTING (AAA)

9.1. AAA: Introduction

- The term AAA has been traditionally used to refer to Authentication, Authorization, and Accounting activities. All of those activities are of crucial importance for the operation of an IP network, although typically they are not so visible to the end-user.

- Authentication: the act of verifying the identity of an entity (subject).

- Authorization: the act of determining whether a requesting entity (subject) will be allowed access to a resource (object) (e.g., network access, certain amount of bandwidth, etc.).

- Accounting: the act of collecting information on resource usage for the purpose of capacity planning, auditing, billing, or cost allocation.

9.2. AAA Framework on the Internet
• At the beginning of 1997 IETF defined the Remote Authentication Dial In User Service protocol (RADIUS, RFC 2058) as the protocol to perform AAA functions on the Internet.

9.3. AAA on the Internet (1)

• A user has established an agreement to access the Internet with an operator that provides a collection of dial-up access servers. A computer equipped with a modem dials up a Network Access Server. A circuit-switched connection is established between the computer (actually, the modem in the computer) and the Network Access Server.

• The Network Access Server does not contain a list of users who can access the network, since there may be a large collection of servers that are geographically widely spread and it would not be feasible to manage the list in all access servers.

• Instead, the Network Access Server is configured to request authentication and authorization from an AAA server, using an AAA protocol like RADIUS.

• Once the user is authenticated and authorized the user can get access to the network.

9.4. AAA on the Internet (2)

• The RADIUS protocol performs relatively well in small-scale configurations and for the particular application that it was designed for; that is, a user dials into a dial-up server, the dial-up server requests authentication and authorization from an AAA server.

• RADIUS offers problems in large environments where congestion and lost data can appear.

• RADIUS runs over UDP and, therefore, lacks congestion control.

• RADIUS lacks some functionality that is required in certain applications or networks, such as the ability of the AAA server to send an unsolicited message to the access server.

• For all of these reasons the IETF has come up with an improved version of RADIUS, named Diameter (RFC 3588).

• For IMS the modern Diameter has been selected as the protocol to perform AAA functions.

9.5. Diameter (1)

• Diameter is specified as a base protocol and a set of Diameter applications that complement the base protocol functionality.

• The base protocol contains the basic functionality and is implemented in all Diameter nodes, independently of any specific application.
• Applications are extensions to the basic functionality that are tailored for a particular usage of Diameter in a particular environment.

• For instance, there is an application tailored for Network Access Server configurations, another for Mobile IPv4, another for Credit Control, and even one for SIP.

9.6. Diameter (2)

• Diameter runs over a reliable transport that offers congestion control.

• The Diameter base protocol defines different functional entities for the purpose of performing AAA functions. These are as follows:

  • Diameter client: a functional entity, typically located at the edge of the network, which performs access control. (e.g., NAS, Foreign Agents)

  • Diameter server: a functional entity that handles authentication, authorization, and accounting requests for a particular realm.

  • Proxy: a functional entity that, in addition to forwarding Diameter messages, makes policy decisions relating to resource usage and provisioning. A proxy may modify messages to implement policy decisions, such as controlling resource usage, providing admission control, and provisioning.

  • Relay: a functional entity that forwards Diameter messages, based on routing-related information and realm-routing table entries. A relay is typically transparent. It can modify Diameter messages only by inserting or removing routing-related data, but cannot modify other data.

9.7. Diameter (3)

• Redirect agent: a functional entity that refers clients to servers and allows them to communicate directly.

• Translation agent: a functional entity that performs protocol translation between Diameter and other AAA protocols, such as RADIUS.

• Diameter node: a functional entity that implements the Diameter protocol and acts either as a Diameter client, Diameter server, relay, redirect agent, or translation agent.

9.8. Diameter (4)
• Diameter is a peer-to-peer protocol, rather than the common client/server protocol.

• This means that, unlike protocols that follow the client/server model, in Diameter any of the peers can asynchronously send a request to the other peer.

• Unlike client/server protocols, a Diameter client is not the functional entity that sends a request and a Diameter server is not the functional entity that sends an answer to the request. Instead, a Diameter client is a functional entity that performs access control, whereas a Diameter server is the functional entity that performs authentication and authorization.

• In Diameter, both a Diameter client and a Diameter server can send or receive requests and responses.

9.9. Format of a Diameter message (1)

![Diagram of Diameter message format]

9.10. Format of a Diameter message (2)

• A Diameter message consists of a 20-octet header and a number of Attribute Value Pairs (AVPs).

• The length of the header is fixed; it is always present in any Diameter message. The number of AVPs is variable, as it depends on the actual Diameter message.

• An AVP is a container of data (typically authentication, authorization, or accounting data).

• Structure of the AVP:
9.11. Diameter Base Protocol Commands (1)

- Diameter messages are either requests or answers.
- A request and its corresponding answer are identified by a common Command-Code in the Diameter header.
- As a request and its corresponding answer share the same Command-Code address space, we need to refer to the Command-Flags to find out if the command is a request or an answer.

<table>
<thead>
<tr>
<th>Command-Name</th>
<th>Abbreviation</th>
<th>Command-Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abort-Session-Request</td>
<td>ASR</td>
<td>274</td>
</tr>
<tr>
<td>Abort-Session-Answer</td>
<td>ASA</td>
<td>274</td>
</tr>
<tr>
<td>Accounting-Request</td>
<td>ACR</td>
<td>271</td>
</tr>
<tr>
<td>Accounting-Answer</td>
<td>ACA</td>
<td>271</td>
</tr>
<tr>
<td>Capabilities-Exchange-Request</td>
<td>CER</td>
<td>275</td>
</tr>
<tr>
<td>Capabilities-Exchange-Answer</td>
<td>CEA</td>
<td>275</td>
</tr>
<tr>
<td>Device-Watchdog-Request</td>
<td>DWR</td>
<td>280</td>
</tr>
<tr>
<td>Device-Watchdog-Answer</td>
<td>DWA</td>
<td>280</td>
</tr>
<tr>
<td>Disconnect-Peer-Request</td>
<td>DPR</td>
<td>282</td>
</tr>
<tr>
<td>Disconnect-Peer-Answer</td>
<td>DPA</td>
<td>282</td>
</tr>
<tr>
<td>Re-Auth-Request</td>
<td>RAR</td>
<td>258</td>
</tr>
<tr>
<td>Re-Auth-Answer</td>
<td>RAA</td>
<td>258</td>
</tr>
<tr>
<td>Session-Termination-Request</td>
<td>STR</td>
<td>275</td>
</tr>
<tr>
<td>Session-Termination-Answer</td>
<td>STA</td>
<td>275</td>
</tr>
</tbody>
</table>

9.12. Diameter Base Protocol Commands (2)

- Abort Session Request/Answer (ASR, ASA)
• It might be necessary for a Diameter server to stop the service provided to the user (e.g., network access), because, say, there are new reasons that were not anticipated when the session was authorized. Among others, lack of credit, security reasons, or just an administrative order may be reasons to abort an ongoing Diameter session. When a Diameter server decides to instruct the Diameter client to stop providing a service the Diameter server sends an Abort-Session-Request (ASR) message to the Diameter client. The Diameter client reports the execution of the command in an Abort-Session-Answer (ASA).

• Accounting Request/Answer (ACR, ACA)

• A Diameter node may need to report accounting events to a Diameter server that provides accounting services. Diameter provides the Accounting-Request (ACR) command, whereby a Diameter client reports usages of the service to a Diameter server. The command includes information that helps the Diameter server to record the one-time event that generated the command or the beginning or end of a service (e.g., access to a network).

9.13. Diameter Base Protocol Commands (3)

• Capabilities Exchange Request/Answer (CER, CEA)

• The first Diameter messages that two Diameter nodes exchange, once the transport connection is established, are the Capabilities-Exchange-Request (CER) and the Capabilities-Exchange-Answer (CEA). The messages carry the node's identity and its capabilities (protocol version, the supported Diameter applications, the supported security mechanisms, etc.).

• Device Watchdog Request/Answer (DWR, DWA)

• It is essential for Diameter to detect transport and application-layer failures as soon as possible, in order to take corrective action. The mechanism that Diameter provides to detect these failures is based on an application-layer watchdog. During periods of traffic between two Diameter nodes, if a node sends a request and no answer is received within a certain time period, that is enough to detect a failure either at the transport or application layer. However, in the absence of regular traffic it is not possible to detect such a potential failure. Diameter solves the problem by probing the transport and application layer by means of a Diameter node sending a DWR message. The absence of the receipt of a DWA message is enough to conclude that a failure has occurred.


• Session Termination Request/Answer (STR, STA)

• A Diameter client reports to the Diameter server that a user is no longer making use of the service by sending a Session-Termination-Request (STR) message. The Diameter server answers with a Session-Termination-Answer (STA) message. For instance, if the dial-up server reports that the dial-up connection has dropped, then the Diameter client sends the STR message to the Diameter server.

• Disconnect Peer Request/Answer (DPR, DPA)

• A Diameter node that has established a transport connection with a peer Diameter node may want to close the transport connection, (e.g., if it does not foresee more traffic toward the peer node). In this case the Diameter node sends a Disconnect-Peer-Request (DPR) to the peer node to indicate the imminent disconnection of the transport protocol. The DPR message also conveys the semantics of requesting the peer not to re-establish the connection unless it is essential (e.g., to forward a message).

• Re-Authentication Request/Answer (RAR, RAA)

• At any time, but especially in sessions that last a long time, the Diameter server may request a re-authentication of the user, just to confirm that there is no possible fraud. A Diameter server that wants to re-authenticate a user sends a Re-Auth-Request message to a Diameter client. The client responds with a Re-Auth-Answer message.
9.15. AAA in the IMS (1)

- Authentication and authorization are generally linked in the IMS.
- In contrast, accounting is a separate function executed by different nodes.
- There are three interfaces over which authentication and authorization actions are performed (namely the $C_x$, $D_x$, and $S_h$ interfaces).
- $C_x$: specified between a Home Subscriber Server (HSS) and either an I-CSCF or an S-CSCF.
- $D_x$: When more than a single HSS is present in a home network there is a need for a Subscription Locator Function (SLF) to help the I-CSCF or S-CSCF to determine which HSS stores the data for a certain user. The $D_x$ interface connects an I-CSCF or S-CSCF to an SLF running in Diameter redirect mode.
- $S_h$: specified between an HSS and either a SIP Application Server or an OSA Service Capability Server.

9.16. AAA in the IMS (2)

9.17. The $C_x$ and $D_x$ interfaces
• The $C_x$ interface is specified between an I-CSCF and an HSS or between an S-CSCF and an HSS.

• Similarly, the $D_x$ interface is specified between an I-CSCF and an SLF or between an S-CSCF and an SLF.

• For a particular user the I-CSCF and S-CSCF use the $C_x$ and $D_x$ interfaces to perform the following functions:
  
  • To locate an already allocated S-CSCF to the user.
  
  • To download the authentication vectors of the user. These vectors are stored in the HSS.
  
  • To authorize the user to roam in a visited network.
  
  • To record in the HSS the address of the S-CSCF allocated to the user.
  
  • To inform the HSS about the registration state of a user’s identity.
  
  • To download from the HSS the user profile that includes the filter criteria.
  
  • To push the user profile from the HSS to the S-CSCF when the user profile has changed.
  
  • To provide the I-CSCF with the necessary information to select an S-CSCF.

9.18. Diameter messages during registration

![Diagram of Diameter messages during registration]

9.19. List of commands defined by the Diameter Application for the Cx interface
9.20. Location Information Request and Answer (LIR, LIA)

- An I-CSCF that receives a SIP request that does not contain a Route header field that points to the next SIP hop (S-CSCF) needs to find out which S-CSCF (if any) is allocated to the user.

- On receiving the SIP request the S-CSCF sends a Location-Info-Request (LIR) to the HSS. The HSS replies with a Diameter LIA message that contains the address of the S-CSCF that is allocated to the user; therefore, the I-CSCF forwards the INVITE request to that S-CSCF.

9.21. Registration Termination Request and Answer (RTR, RTA)

- Due to administrative action the operator of the home network may wish to deregister one or more registered Public User Identities allocated to a user. When this happens the HSS sends a Registration-Termination-Request (RTR) message to the S-CSCF where the user is registered.
9.22. The user profile

- The user profile, which is stored in the HSS, contains a lot of information related to a particular user.
- The S-CSCF downloads the user profile when the user registers for the first time with that S-CSCF.
- If the user profile changes in the HSS while the user is registered to the network, then the HSS sends the updated user profile in a User-Data AVP included in a Diameter PPR message.
- A user profile is bound to a Private User Identity and to the collection of Public User Identities that are, in turn, associated with that Private User Identity.
- The user profile contains a plurality of service profiles. Each service profile defines the service triggers that are applicable to a collection of Public User Identities.

9.23. Structure of the user profile
9.24. The Sh interface (1)

- The $Sh$ interface is defined between a SIP AS or an OSA-SCS and the HSS.

- It provides a data storage and retrieval type of functionality, such as an Application Server downloading data from the HSS or an Application Server uploading data to the HSS.

- The $Sh$ interface also provides a subscription and notification service, so that the AS can subscribe to changes in the data stored in the HSS.

- The $Sh$ interface introduces the term user data to refer to diverse types of data. User data, in the $Sh$ interface context, can refer to any of the following:

  - Repository data: the AS uses the HSS to store transparent data.
  - Public Identifiers: the list of Public User Identities allocated to the user.
  - IMS User State: the registration state of the user in the IMS:
    - Registered
• Unregistered
• Pending
• S-CSCF name: contains the address of the S-CSCF allocated to the user.

9.25. The Sh interface (2)

• Initial Filter Criteria: contain the triggering information for a service.
• Location Information: contains the location of the user in the circuit-switched or packet-switched domains.
• User State: contains the state of the user in the circuit-switched or packet-switched domains.
• Charging Information: contains the addresses of the charging functions
• The $S_h$ interface defines eight new Diameter messages to support the required functionality of the interface:

<table>
<thead>
<tr>
<th>Command-Name</th>
<th>Abbreviation</th>
<th>Command-Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-Data-Request</td>
<td>UDR</td>
<td>306</td>
</tr>
<tr>
<td>User-Data-Answer</td>
<td>UDA</td>
<td>306</td>
</tr>
<tr>
<td>Profile-Update-Request</td>
<td>PUR</td>
<td>307</td>
</tr>
<tr>
<td>Profile-Update-Answer</td>
<td>PUA</td>
<td>307</td>
</tr>
<tr>
<td>Subscribe-Notifications-Request</td>
<td>SNR</td>
<td>308</td>
</tr>
<tr>
<td>Subscribe-Notifications-Answer</td>
<td>SNA</td>
<td>308</td>
</tr>
<tr>
<td>Push-Notifications-Request</td>
<td>PNR</td>
<td>309</td>
</tr>
<tr>
<td>Push-Notifications-Answer</td>
<td>PNA</td>
<td>309</td>
</tr>
</tbody>
</table>

9.26. Accounting

9.27. Accounting

• Defining the price of the services
• Creating and printing bills
• Collecting money
• Technology, marketing and service oriented
• Always changing
• Unique solutions

9.28. Types of charging

• Online charging: Online charging is a process where charging information for network resource usage is collected concurrently with that resource usage in the same fashion as in offline charging. However,
authorization for the network resource usage must be obtained by the network prior to the actual resource usage to occur. This authorization is granted by the Online Charging System (OCS) upon request from the network.

• In conclusion, online charging is a mechanism where charging information can affect, in real-time, the service rendered and therefore a direct interaction of the charging mechanism with the control of network resource usage is required.

• Offline charging: Offline charging is a process where charging information for network resource usage is collected concurrently with that resource usage. The charging information is then passed through a chain of logical charging functions. At the end of this process, CDR files are generated by the network, which are then transferred to the network operator's Billing Domain for the purpose of subscriber billing and/or inter-operator accounting.

• In conclusion, offline charging is a mechanism where charging information does not affect, in real-time, the service rendered.

9.29. Accounting tasks (1)

• Mediation

• To perform the rating calculations it is necessary to produce a Call detail record. A Call detail record (CDR, also known as Call Data Record) is “a record of a call setup and completion”, and its format varies among telecom providers or programs, which some allow to be configured by the user.

• EDR stands for Event Data/Detail Record. EDR records are used for systems that charge more than calls - content. e.g. buying ring tones).

• The generated CDR/EDR may not be in a form suitable for the particular rating system. In this case a piece of software, known as the mediation system, may be required to render the data into a form useful by the rating system.

• The mediation system is also useful for gathering data from various sources to aggregate into one record.

9.30. Accounting tasks (2)

• Rating

• To calculate the price of the service requested by the user depending on the following things:
  • Other services requested by the user
  • Discounts entitled to the user
  • Parameters of the requested service
  • Parameters and settings of the user
  • User's behavior

9.31. Accounting tasks (3)

• Billing

• Creating billing information from monthly data:
  • Requested services
• Discounts
• Forming the bill
• Creating the file to print/send
• Data to the A/R

9.32. Accounting tasks (4)

• Accounts/Receivable (A/R)
  • Financial management
  • Bill payments (banking transactions)
  • Pre-paid card (top-up) management
  • Warnings, notice
  • Credit limit check
  • Financial statements

9.33. Accounting tasks (5)

• Customer Relationship Management (CRM)
  • Defining subscribers, storing information
  • Defining services, selling services, storing parameters
  • Selling devices (installment)
  • Discounts

9.34. Accounting in the IMS

• IMS uses the Diameter protocol to transfer the accounting information that charging in the IMS is based on.

• The CSCFs inform the charging system about the type and the length of the sessions each user establishes. In addition, the routers (e.g., the GGSN) inform the charging system about media activity during those sessions.

• Charging systems use unique identifiers used to correlate the accounting data applying to a particular session received from different entities. So, the accounting records generated by a router and by a CSCF about the same session have the same unique identifier.

• The charging system assembles all the accounting information related to each user in order to charge them accordingly.

9.35. IMS Offline charging architecture
9.36. Offline charging (1)

- All the SIP network entities (except HSS and SLF) involved in the session use the $R_f$ interface to send accounting information to a CCF (Charging Collection Function) located in the same administrative domain.
- The CCF uses this information to create CDRs and sends them to the BS (Billing System) of its domain using the $B_i$ interface.
- The entities managing GPRS access (i.e., the SGSN and the GGSN) use the $Ga$ interface to report to the CGF (Charging Gateway Function), which uses the $B_p$ interface to report to the BS of its domain.
- The $R_f$ interface is based on the Diameter base protocol together with a vendor-specific Diameter Application for the $R_f/R_o$ interfaces. The $B_i$ and $B_p$ interfaces are based on a file transfer protocol (FTP), although the actual protocol is not standardized.
- The 3GPP Release 6 offline charging architecture is a little different. The CGF acts as the gateway between the 3GPP network and the billing domain. Additionally, a new entity called CDF (Charging Data Function) substitutes the CCF.
9.37. Offline charging (2)

9.38. Online charging architecture
9.39. Online charging (1)

- The S-CSCF uses the ISC interface, which is based on SIP.
- Application Servers and the MRFC use the $Ro$ interface, which is based on Diameter.
- Network operators configure those sessions to which online charging has to be applied by defining a filter criterion in the user profile.
- The filter criterion blindly sends all the SIP requests to the Application Server that is acting as a Session Charging Function (SCF).
- The SCF looks like any other AS to the S-CSCF, but it does not provide services for the user in the usual sense. Instead, the SCF reports accounting information to the Correlation Function using the $Rb$ interface.
- If the user runs out of credit during a session the Correlation Function informs the SCF through the $Rb$ interface and the SCF terminates the session by acting as a B2BUA and sending two BYE requests, one toward each terminal.

9.40. Online charging (2)

- The AS and the MRFC uses the $Ro$ interface to send charging information to the ECF (Event charging function).
• The AS or the MRFC receives the address or addresses of the ECF from the S-CSCF in the P-Charging-Function-Address header field of the SIP message.

• Online charging is based on credit units. Services are paid for by credit units, and users can enjoy a particular service as long as they have enough credit units in their accounts.

• There are two types of online charging in the IMS:
  • Immediate Event Charging (IEC)
  • Event Charging with Unit Reservation (ECUR)

9.41. Immediate Event Charging

• The ECF deducts a number of credit units from the user's account and then authorizes the MRFC or the AS to provide the service to the user.

9.42. Event Charging with Unit Reservation

• The ECF reserves a number of credit units in the user's account and authorizes the MRFC or the AS to provide the service to the user. In case a particular service costs more credit units than those originally reserved by the ECF, the MRFC or the AS can contact the ECF to request further credit unit reservations.

• When the service is over the MRFC or the AS report to the ECF the number of credit units that the user spent. At this point the ECF returns to the user's account all the credit units that were reserved but not used.
10. 10 IMS SERVICES

10.1. Impact of SIP on application development

10.2. IMS services and application building blocks

- Presence
- Push-to-Talk
• Instant Messaging
• Other services
  • Content sharing
  • Real-time video sharing
  • Games
  • Media Push
  • Location Based Services

10.3. PRESENCE

10.4. Presence

• Basic idea of Presence
  • Is the called party available at present? (chat, mms, e-mail, video...)
  • Or is he/she doing something else? (reading, sleeping, he/she is in a meeting)
  • Status?
  • Is he/she actually there?

• Creating profiles
  • For example the x group can send me any type of message at any time
  • But for the y group my status is busy and can only send e-mail
  • Calls can be rejected
  • Some status information are public
  • Some status information can only be seen by whom are entitled to see them.

10.5. User profile

10.6. Presence service
The aim of presence services is to publish and disseminate the user's presence information related to the actual availability through the network. Presence information can indicate whether the user is online or offline or can contain information about the user's current activities.

10.7. Presence service (1)

- PUA elements forward their own information to the Presence Agent (PA).
- The Presence Server (PS) is a functional entity in the system. The main task of the PS is to manage SUBSCRIBE and other system-wide messages.
- The other participants of the system are the watchers. Mainly they query presence information from PA but can query other watchers for presence information.

10.8. Presence service (2)

- The use of presence services:
  - Publication and dissemination of our current status.
  - Query status information of other users.
- The presence information can contain the following elements:
  - Actual status
  - Communication features
  - Capability of the terminal
  - Current activity
  - Location
  - Available services.
10.9. Presence service (3)

- The presence service participants can be classified according to several criteria, but basically we differentiate between two roles in the model:
  - Presence-entity: Source of presence information, presence message producer.
  - Watcher: Queries presence information.
- According to RFC3856 the SIP entities are the following:
  - Presence Agent (PA): Responsible for managing presence service subscriptions and other signaling messages.
  - Presence User Agent (PUA): Responsible for publishing presence information and sometimes for conversion.

10.10. Presence service (4)

- Service: Communication service, lehet instant messaging, phone network.
- Device: Physical communication equipment at the user's side.
- Person: The user.
- The user's presence status can be varied, according to the user's current status, location, communication ability.
- Presence information data components assigned to the user in the model:
  - presence-source URI
  - user
  - service
  - device.
10.11. Presence service (5)

- The primary purpose of PUBLISH request is managing SIP signal system compliant messaging and transmitting SIP-based event signals within the system.
- The PUBLISH method is not only used to publish presence messages but can be used to transmit any event signals.

Publishing presence information

10.12. Presence service (6)

- The watcher can claim presence information via the interface defined between the watcher and the PA.
- The presence sign-up process is realised through a SIP-based SUBSCRIBE method.
- There are two ways to acquire presence information:
  - Static: periodically
  - Dynamic: in case of changes
10.13. Presence service (7)

- Alice wants the list of watchers connected to her.
- The PUA sends a SUBSCRIBE request to the PA.
- The PA checks and authenticates the request and replies with 200 OK.
- The PA sends a NOTIFY request that indicates subscription.
- The NOTIFY contains the list of watchers. PA updates PUA’s information.
- NOTIFY can be sent when further changes occur, so Alice will be informed about new watchers or removed watchers.
PUA determines the list of watchers


- Alice wants to know her friends' status (Bob, Carol and David)
- Alice, as a watcher, sends a SUBSCRIBE request to the appropriate presence agent. (Message (1)(3)(5))
- Then Alice receives NOTIFY messages. (Message (7)(9)(11))
10.15. Presence service in the IMS

10.16. Presence architecture in the IMS (1)

- In the presence, service users can determine the information to be seen by other participants as well as the information sent to the subscribed watchers.

- The presence service is not only defined between presence entities and watchers, but presence functions can be extended to other services.

10.17. Presence architecture in the IMS (2)

- Besides the PA and RLS functions any application server is modeled as watcher.

- The Pen interface (i.e. presence interface) allows the application server to supply PUA functions, so this interface is very important in the dissemination of presence information of the presence entity.

- PUA can collect presence information from any source such as: HLR, MSC/VLR, SGSN, GGSN, vagy S-CSCF is.
10.18. Presence architecture in the IMS (3)

- Alice can also function as a watcher through the IMS terminal, so Alice can also subscribe to the messages of entities.
- The IMS allows Alice to subscribe to entities one by one, but for the sake of efficiency she sends a list of the chosen entities to the RLS.
10.19. Presence architecture in the IMS (4)

- The IMS presence application publishes the presence information related to the presence entity.

- First the IMS terminal sends a PUBLISH message to the P-CSCF and then the P-CSCF forwards it to the S-CSCF in the local network.

- After defining the initial filter criteria for the presence entity the S-CSCF forwards the PUBLISH request to the PA then it replies with 200 OK.

- The S-CSCF reaches PA via the appropriate application server, so it forwards the message to the application server.
10.20. Functional description of the Presence architecture

10.21. Presence Service architecture

10.22. Functional units in the Presence architecture

- Presence server
  - Stores presence information of agents
  - Presence information are acquired by proxies
    - The acquisition can be anonym
    - The user agent can make a list of authorized entities
    - The watcher proxies can filter presence information
  - Signaling to the acquiring proxy
• Gathering information from watcher proxies

• Agents
  • Provide presence information to the Presence Server
    • External presence information
    • Users’ presence information
    • Network presence information

10.23. Presence User Agent

• The presence user agent’s tasks are:
  • Gathering presence information
  • Gathering presence information and merging them into the appropriate format according to the Peu and Pep reference points.
  • Sending presence information to the presence server via the presence proxy according to the Pep and Peu reference points.
  • The Presence User Agent (PUA) can be found between the presence server and the user.

![Diagram of Presence User Agent](image)

10.24. Presence proxy

• Presence proxy
  • Acquiring and serving presence information
  • Determining the destination network
  • Informing other proxies
  • Authenticating, watcher, gathering proxy
  • Access control
• Server list
  • SIP application server
  • Adding to presence-server list

• Watcher application
  • UE (user equipment), Watcher application server, external networks

10.25. Presence user agent in IMS

Presence User Agent in the IMS

PUA- Presence User

10.26. Relationship between Presence agent and other IMS entities

• The architecture based on the S-CSCF and the HSS provides presence information to the presence server.
• The ISC interface is responsible for receiving presence information from the S-CSCF and forwarding to the presence agent.

• The network agent appears in the model as a separate functional entity whereby the different presence sources can be treated uniformly

10.27. Watcher application and Presence Server in the IMS

10.28. Presence service - IMS (1)

• Serving request through the application within the UE - Presence server role

• Presence information coming from the UE and other external sources must be in a proper format for the presence server

• The client must support the appropriate transmission method (IMS)

• Server: caching, querying and serving of incoming presence information
10.29. Presence service - IMS (2)

- Communication attributes and address-specific information are:
  - Communication status: online, offline
  - Communication type: telephone, sms, email, mms, ims
  - Contact: SIP url, email, instant message address: IM:nev@domainnev
  - Priority: setting priorities for communication types and addresses
  - Text messages

10.30. Presence service - IMS (2)

- The basic features of the server list:
  - SIP application server
  - Adding to the presence server list
10.31. Presence data model (1)

- The nested presence attributes of each layer are distinguished by unique identifiers.
- When sending multiple presence information multiple presence information have to be nested.
- In the model, the presence value selection related to the attribute values is done by presence watchers, according to the access rights assigned to the users.

10.32. Presence data model (2)

- The availability of the presence information can be determined through the authorization settings of the presence watchers.
- The list of presence messages accessible to a user or user group can be limited.
- The list is logically represented by the presence server, but may also appear as a separate entity within the network.
10.33. Presence access list (1)

Accessing to presence information is controlled by access lists.

There are 3 types of access lists:
- Personal list
- General list
- Blocked list

10.34. Presence access list (2)

- There are 3 types of access list that control the access of presence information:
  - Personal list
User authentication in the presence service

10.35. Acquiring presence information

- User A wants to acquire the presence information of user B but user B is in a different network.

10.36. The registration mechanism of IMS watcher

- The IMS watcher device can subscribe to the service from the same IM-CN network or from different IM-CN network. The methods are the same.
10.37. Notifying the Presence Server about the IMS registration

10.38. Notifying the presence server about the user's status
10.39. Updating presence information

The presence server selects the watchers to be updated and sends them a NotifyPresUp message. The message can contain a full exchange of presence or can only change the part of the existing presence.

10.40. Instant message service

10.41. Network implementations (1)

- In the central network segment, protected by firewall, the following components are included:
- Location/Presence server
- SIP server
- Device manager server
- Identity verification server
- In the model the network segment is responsible for integrating of different types of interfaces
- The system is connected to the traditional telephone network via a PSTN Gateway, the RADIUS server is used during authentication.

10.42. Network implementations (2)

- For outdoor positioning the network can use GPS data or Bluetooth information.
- The server can store location data in the location database.
- The location server can be a standard XML-RPC server.
10.43. PUSH-TO-TALK OVER CELLULAR

10.44. Push-to-talk (1)

- PoC (Push-to-talk over Cellular) is a walkie-talkie type of service. Unlike regular voice calls, which are full-duplex, PoC is a half-duplex service; that is, only one user can speak at a time.
- PoC sessions can have more than two participants.
- It does not require the deployment of new radio technologies.
- PoC can run on top of low-bandwidth and high-delay links. These links would be inappropriate for running other types of services, such as voice calls.
- There are several incompatible PoC specifications at present. Many are not based on the IMS, but consist of proprietary solutions implemented by a single vendor. As a result these PoC solutions generally cannot interoperate with equipment from other vendors.
- Many operators willing to provide PoC services felt uncomfortable with the situation just described and asked a few vendors for a standard solution based on the IMS. As a consequence, a group of vendors teamed up to develop an open PoC industry standard.

10.45. Push-to-talk (2)
10.46. Push-to-talk (3)

Telephone communication
- 2-way capacity used for the whole duration of the call

Push-to-Talk communication
- Communication is the background activity on virtual connection (i.e., possibility to talk at any instance, staying informed) BUT capacity is used only when someone talks, one way

10.47. Push-to-talk (4)

- Business users:
  - Service providers
  - Trade
  - Taxi
  - Car rental
  - Public transport
  - Airports
  - Harbors
  - Factories
  - Power plants
  - Stb.
10.48. PoC architecture (1)

10.49. PoC architecture (2)

- The User Equipment contains two logical elements:
  - PoC Client
  - XDMC (XML Document Management Client)
- The PoC client uses SIP to communicate with the SIP/IP Core over the POC-1 interface, and RTP and TBCP (Talk Burst Control Protocol) to communicate with the PoC server over the POC-3 interface.
- TBCP is a floor control protocol based on RTCP that is used to signal which user is allowed to speak at a given time.
• The XDMC uses SIP to communicate with the SIP/IP Core over the XDM-1 interface and XCAP to communicate with the Aggregation Proxy over the XDM-3 interface.

10.50. PoC architecture (3)

• The Aggregation Proxy acts as a single point for the XDMC to contact the network. The Aggregation Proxy performs user authentication and routes the XCAP messages from the XDMC to the appropriate server (the PoC XDMS or the Shared XDMS).

• The PoC XDMS manages documents that are specific to PoC (e.g., the members of a PoC group).

• Shared XDMS manages documents that are needed by PoC but that may be shared with other services (e.g., presence-related documents).

10.51. PoC Registration

• In order to use the PoC service, a terminal needs to register to the PoC service.

• When the terminal performs IMS registration, it adds the +g.poc.talkburst and the +g.poc.groupad feature tags to the Contact header field of the REGISTER request.

  • The +g.poc.talkburst feature tag indicates that the terminal can handle PoC sessions.

  • The +g.poc.groupad feature tag indicates that the terminal can handle group advertisement.

• On receiving these feature tags, the S-CSCF performs a third-party registration towards the PoC server of the domain.

10.52. PoC Server Roles

• A PoC server within a session can perform two roles:

  • Controlling PoC Function

  • Participating PoC Function

• A given PoC server in a given PoC session will be performing one or both roles.

• However, only one PoC server in a session performs the Controlling PoC Function. This server may or may not perform the Participating PoC Function as well. The rest of the PoC servers, assuming that there are several PoC servers involved in the session, will only perform the Participating PoC Function.

• The figure on the next slide shows a PoC session with one controlling and four participating PoC servers. Each of the PoC servers is in a different domain.

10.53. PoC session with a central controlling PoC server
10.54. Controlling and participating PoC server
10.55. PoC servers

- The controlling PoC server provides centralized PoC session handling:
  - Media mixing
  - Centralized floor control
  - Policy enforcement for participation in group sessions.

- The participating PoC server exchanges SIP signaling with the client and with the controlling PoC server and, optionally, relays media and floor control messages between them.

- When a participating PoC server chooses not to be on the media path, clients exchange media and floor control traffic directly with the controlling PoC server.

- The process to determine which one of the PoC servers involved in a session acts as the controlling PoC server depends on the type of the session.

10.56. PoC Session Types

- One-to-One PoC session
  - A PoC session between two users.
  - The controlling PoC server is the inviting user's PoC server and at the same time, it is also the participating PoC server.

- Ad-hoc PoC Group
  - A user selects a set of users in an ad-hoc fashion (e.g., picking them from the terminal's address book) and invites all of them into a multiparty PoC session.
  - The controlling PoC server is the inviting user's PoC server and at the same time, it is also the participating PoC server.

10.57. PoC session types

- Pre-arranged PoC Group
  - Like the ad-hoc PoC group, the pre-arranged PoC group also consists of a multiparty PoC session. Nevertheless, the users participating in the session are selected beforehand, not in an ad-hoc manner when it is established. That is, a prearranged PoC group includes a predefined set of users (e.g., the user's golf buddies).
  - The controlling PoC server is the PoC server hosting the pre-arranged PoC group.

- Chat PoC Group
  - Chat PoC groups are also multiparty PoC sessions. However, when a user joins a chat PoC group, no invitations are sent to other users.
  - Conversely, when a user joins a pre-arranged PoC group, all the users that belong to that PoC group are invited to the PoC session.
  - The controlling PoC server is the PoC server hosting the chat PoC group.

10.58. One-to-one PoC session
10.59. Ad-hoc PoC Group session
10.60. Pre-arranged PoC Group session
10.61. Chat PoC Group session

![Diagram of Chat PoC Group session]

10.62. Bringing new users into a PoC session (1)

- New users can be added to an ongoing PoC session in two ways:
  - The new user sends an INVITE request to the URI of the session
  - The controlling PoC server sends an INVITE request to the new user.
- Participants in a PoC session can have the controlling PoC server send an INVITE request to the new user by sending a REFER request to the controlling PoC server.
- The figure on the next slide shows how a controlling PoC server receives a multiple REFER and, as a consequence, invites two new users to an ongoing PoC session.

10.63. Bringing new users into a PoC session (2)
10.64. Session Establishment Types

- All the message flows we have discussed so far use so-called on-demand signaling. Nevertheless, PoC defines two session establishment types: using on-demand signaling and using a pre-established session.

- When on-demand signaling is used, terminals generate INVITE requests to create new PoC sessions or join chat PoC groups. On the terminating side, the terminating user's PoC server relays incoming INVITE requests to the user.

- The use of pre-established sessions results in an optimization that allows a more rapid session establishment at the terminating side.

  - The terminal using the pre-established session establishes a session (using an INVITE request) with its home PoC server, typically right after registration.

  - At a later point, when the home PoC server receives an INVITE for the user, there is no need to use any SIP signaling towards the terminal. The session that was established previously is used to deliver media and floor control messages to the terminal.

  - It is also possible to use pre-established session at the originating side. However, the pre-established session does not eliminate the need for SIP signaling at the originating side. It only replaces the typical INVITE transaction used to establish a new session with a REFER transaction that instructs the PoC server to generate an INVITE.

10.65. Pre-established session
10.66. TBCP message flow
10.67. INSTANT MESSAGING

10.68. Instant Messaging

- Instant messaging is one of today's most popular services. Many youngsters (and not-so young people) use the service to keep in touch with their relatives, friends, co-workers, etc.

- Millions of instant messages are sent every day. So, it will come as no surprise that such a popular service is already supported in the IMS.

- Instant messaging is the service that allows a user to send some content to another user in near-real time.

- Due to the real-time characteristics of instant messages the content is typically not stored in network nodes, as often happens with other services such as email.

- The content in an instant message is typically a text message, but can be an HTML page, a picture, a file containing a song, a video clip, or any generic file.

- The instant messaging service combines perfectly with the presence service, since presence allows a user to be informed when other users become available.

10.69. IM services (1)
Most popular IM services:

- Google Hangouts
- Skype
- iMessage
- Tencent QQ (China)
- AOL Instant Messenger
- Yahoo! Messenger
- ICQ
10.70. IM services (2)

Multi-protocol IM services:

- Digsby
- Pidgin
- Trillian
- GAIM
- Nimbuzz (mobile)
- Fring (mobile)
- Adium (Mac)
- Meebo (Web)
- Viber
- WhatsApp
10.71. Voice Instant Messaging

- Voice instant messaging with audio content
- Voice messages created in terminal and sent via IMS to recipients ("SIP-based Immediate Messaging")
- Enables convergence with SIP PC clients

10.72. "Voice Messaging" - High-level concept

10.73. Modes of Instant Messages

- There are two modes of operation of the instant message service, depending on whether they are stand-alone instant messages or part of a session of instant messages.
• The first mode is a pager-mode instant message as one that is sent as a stand-alone message, not having any relation with previous or future instant messages. This mode of instant messaging is referred to as "pager mode" because the model resembles the way a two-way pager works. The model is also similar to the SMS (Short Message Service) in cellular networks.

• The second mode is a session-based instant message as one that is sent as part of an existing session, typically established with a SIP INVITE request.

• Both models have different requirements and constraints; hence their implementation is different.

10.74. Pager-mode

• The IETF has created an extension to SIP that allows a SIP UA to send an instant message to another UA. The extension consists of a new SIP method named MESSAGE.

• The SIP MESSAGE method is able to transport any kind of payload in the body of the message, formatted with an appropriate MIME type.

• The proxy forwards the MESSAGE request like any other SIP request, even when the proxy does not support or understand the SIP MESSAGE method.

10.75. Congestion Control with MESSAGE

• One of the problems with SIP derives from the fact that any proxy can change the transport protocol from TCP to UDP, SCTP, or other transport protocols and vice versa.

• If a UA is sending a large instant message over a transport protocol that does not offer congestion control, the network proxies can become congested and stop processing other SIP requests.

• At the time of writing, SIP does not offer a mechanism for a UA to indicate that all proxies in the path must use a transport protocol that implements end-to-end congestion control. Consequently, a limit has been placed on the SIP MESSAGE method such that MESSAGE requests cannot exceed the MTU (Maximum Transmission Unit) minus 200 bytes. If the MTU is not known to the UAC this limit is 1300 bytes.

• A solution to sending SIP MESSAGE requests with large bodies is to use the content indirection mechanism. The UAC uses HTTP - or any other protocol that runs over a congestion-controlled transport protocol - to store the body of the SIP request in a server. Then, the UAC inserts a link to the URI where the payload is stored, instead of sending the whole body embedded in the SIP MESSAGE request.

10.76. Session-based IM

• The session-based instant message mode uses the SIP INVITE method to establish a session where the media plane is not audio or video, but an exchange of instant messages.
• When the UAC establishes a session to send and receive instant messages the actual media (the collection of instant messages) are sent over the Message Session Relay Protocol (MSRP).

• MSRP is a simple text-based protocol whose main characteristic is that it runs over transport protocols that offer congestion control, such as TCP, SCTP, and TLS over TCP. Explicitly, MSRP does not run over UDP or any other transport protocol that does not offer end-to-end congestion control. Due to this, the main characteristic of MSRP is not imposing a restriction on the size of an instant message.

• Another characteristic of MSRP is that it runs on the media plane. Therefore, MSRP messages do not traverse SIP proxies. MSRP supports instant messages to traverse zero, one, or two MRSP relays.

10.77. MSRP relay

• An MSRP relay is a specialized node in transiting MSRP messages between two other MSRP nodes (endpoints or other relays).

• MSRP relays, which are located in the media plane, must not be confused with SIP proxies, which are located in the signaling plane.

• When an endpoint wants to make usage of an MSRP relay, it first opens a TLS connection towards its relay, authenticates (by sending an AUTH request), and if authentication is successful the relay provides the endpoint with an MSRPS URL that the endpoint can use for its MSRP sessions.

10.78. End-to-end session establishment with MSRP relays

![Diagram showing end-to-end session establishment with MSRP relays](image-url)
10.79. IM Session
10.80. APPENDIX 1: SETTING OF PRESENCE INFORMATION

10.81. Setting of presence information (1)

10.82. Setting of presence information (2)

10.83. Setting of presence information (3)
10.84. Setting of presence information (4)

10.85. Setting of presence information (5)

10.86. Setting of presence information (6)
10.87. Setting of presence information (7)

10.88. Setting of presence information (8)

10.89. Setting of presence information (9)
10.90. Setting of presence information (10)

1. Country
2. City
3. Building
4. Position in the building

10.91. APPENDIX 2: OTHER SERVICES

10.92. Mobile Applications
10.93. Multi Party Network Gaming
10.94. Content sharing

- Can be used to transfer files between users or later between group of users
- First, direct file sharing between terminals
- Enables convergence with SIP PC clients
10.95. Real-time video sharing

- Share the moment instantly in real-time, 'See What I See'
- Enables spontaneous behavior
• Share the camera view or video clip whilst in an ongoing session - Enrich your voice call by sharing live video
  • Share the moment or show what you mean, in real-time.
  • Flexibility to add and remove video
• Unidirectional mobile video streaming between peer users
• Enables convergence with PC clients
• First step to rich call services
• Ideal service for EDGE or WCDMA networks

10.96. What is Media Push?

• A new kind of news service - content pushed to the terminal client only when preferred
• Several types of content can be delivered - Text, URL, picture, streaming link, video...
• Based on SIP and IMS technology - control point for the operator for authentication, subscriber provisioning, versatile charging models
• Huge possibilities to extend service easily - by utilizing Presence and location information
• End-to-end SIP service
• (http://www.ubiquitysoftware.com/pdf/Ubiquity_Nokia_Media_Push.pdf)

10.97. Instant Media Push - Image, Location

10.98. One day in Rabbitfield...
I wonder what Rabby is doing tonight?

Session set-up

Voice

Bunnyman, here

Are you sneaking out tonight?

Rabbit House, here...

Yes!!! How about we see Itchy & Scratchy?
Bunny & Rabby find out that “Rchy & Scrocxy” is on at Burne Memorial theatre. They book two tickets and pay through their phones...

Let's skip the details and continue!

No, I'm down at Mee's, stalking Homer

I'll show you...

Shall we meet outside the theatre then?
Service Delivery Architectures and Platforms

Positioning system

EAT MY SHORTS!
Check out this map, dude...

That's OK!
I'll come over,
but you
must
help me to
find it.

Voice

Get position
Correct position

Bunny's position

Map service
(web-sensor)

http://www.Maps.com
Web-WAP page

http://www.Maps.com
Web-WAP page

Voice
11. 11 INTEGRATION WITH MOBILE PLATFORMS

11.1. The beginning

- The first smartphones combined the functions of personal digital assistants (PDA) with mobile phones.
- Later models added the functionality of portable media players, low-end compact digital cameras, pocket video cameras, and GPS navigation units to form one multi-use device.
- First, mobile phones were able to make and receive calls, after the appearance of wireless data traffic they were able to send and receive text messages (SMS, MMS) and PDAs were able to send e-mails.
- After that PDA’s and mobil phone's features were combined and the result was a smartphone.

11.2. Definition

- There are several approaches, but there is no official definition:
• A miniature computer, which can function as a phone.

• Smart phones utilize a complete operating system that provides developers with a standardized interface and platform to develop on.

• "Smart phones differ from ordinary mobile phones in two fundamental ways: how they are built and what they can do."David Wood, Symbian Ltd.

11.3. Smartphone features

• Operating system that can run applications including:
  • Calendar
  • E-mail client
  • Web browser
  • E-book reader
  • Media player
  • Accelerometer
  • GPS
  • Camera
  • Touch screen or QWERTY keyboard
  • etc.

11.4. Development history of smartphones: IBM Simon

• First smartphone
• Released in 1994
• Combined the features of mobile phones and PDAs
• Touch screen
• 500 gram
• Price: $900
• Functions
  • Address book
  • Calendar
  • Calculator
  • Note pad
  • E-mail client
  • Fax
11.5. Development history of smartphones: Nokia 9000 Communicator

• Released in 1996
• It has 8 MB of memory, which is divided between applications (4 MB), program memory (2 MB) and user data (2 MB)
• QWERTY keyboard
• 640x200 resolution
• GEOS V3.0 operating system
• E-mail client
• Text-based browser
• 397 gram
• Price: $1000
11.6. Development history of smartphones: Ericsson R380

- Released in 2000
- The first device to use Symbian OS
- It was the first device marketed as a "smartphone"
- It was as small and light as a normal mobile phone
- Touch screen
- 164 grams
- The magazine Popular Science appointed the Ericsson R380 Smartphone to one of the most important advances in science and technology
- It can be considered the clear forerunner of the popular P800/P900 series of smartphones

11.7. Development history of smartphones: Kyocera 6035

- Released in 2001
- It was one of the first smartphones to appear in the American market
- It wasn't very famous outside the US
- Operating system: Palm OS
- 8MB memory
- Limited browser capability
• 208 gram
• Price: $500

11.8. History: BlackBerry 5810

• Released in 2002
• Introduces "always available" business lifestyle
• The device lacked a speaker and a microphone, so a headset was needed to make calls.
• 8MB memory
• Java-based operating system
• Push e-mail service
• 133 gram
• Price: $749
11.9. Development history of smartphones: Palm Treo 600

- Released in 2003
- It was developed by Handspring, and offered under the palmOne brand after the merger of the two companies.
- Third-party softwares could be installed
- 32MB memory
- 160x160 resolution
- Color touch screen
- 168 gram
- Price: $500
11.10. Development history of smartphones: Nokia N70

- Released in 2005
- First Nokia N series smartphone
- Supports 3G
- Symbian operating system
- 2 megapixels camera
- Memory card slot
- Color screen(256k)
- Push-to-talk
- MP3 player
- Supports Word, Excel, PowerPoint files
11.11. Development history of smartphones: iPhone (2G)

• Released in 2007

• Multi-touch capacitive touch screen (320x480)

• It didn't support installing applications (only with jailbreak), from 2008, after the launching of App Store and iPhone 3G, applications can be downloaded and installed

• There was no multitasking before iOS 4

• 4GB, 8GB, 16GB storage

• 2 megapixels camera

• EDGE, Wi-Fi, Bluetooth

• 135 gram

• Price: $499
11.12. iPhone evolution
11.13. HTC Dream (G1)

- Released in 2008
- It was the first phone on the market to use the Android mobile device platform
- The operating system is designed by Google, so services like Gmail, Google Calendar, Google Maps are available
- 320x480 pixels capacitive touch screen
- QWERTY keyboard
- 256MB memory
- 3 megapixels camera
- Accelerometer


- Released 2010
- The Wave was the first phone with Samsung's own Operating System, BADA, meaning "Ocean"
- 512MB memory
- Capacitive touch screen (800x480)
- 5 megapixels camera
- 720p high-resolution video capture
- GPS, Bluetooth, Wi-Fi, HSDPA, HSUPA
- Applications can be downloaded from Bada Store
- 118 gram
11.15. LG Optimus 7

- Released in 2010
- It is part of the first generation Windows Phone lineup
- 512MB memory
- 800x480 pixels capacitive touch screen
- 5 megapixels camera
- 720p high-resolution video capture
- Wi-Fi, Bluetooth, GPS, HSDPA, DLNA
- Applications can be downloaded from Windows Phone Marketplace
- 157 gram
11.16. Operating system market share
Service Delivery Architectures and Platforms
Source: StatCounter Global Stats

11.17. Smartphone market share

![Global Smartphone Marketshare 2005-2012](image)

Source: StatCounter Global Stats

11.18. Mobile app store timeline

![Mobile app store timeline](image)


<table>
<thead>
<tr>
<th></th>
<th>Apple App Store</th>
<th>Android Market</th>
<th>BlackBerry App World</th>
<th>Nokia Ovi Store</th>
<th>Windows Marketplace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of applications</td>
<td>~700,000</td>
<td>~675,000</td>
<td>~60,000</td>
<td>~120,000</td>
<td>~100,000</td>
</tr>
<tr>
<td>Rate of free applications</td>
<td>37%</td>
<td>57%</td>
<td>26%</td>
<td>26%</td>
<td>17%</td>
</tr>
</tbody>
</table>

[fragile]

11.19. Average price of apps

![Average price of apps](image)
11.20. App Store and Android Market

Android Market is the fastest growing mobile content platform since the beginning of 2011

Source: http://www.research2guidance.com

[fragile]
11.21. Global mobile revenue

Source: http://www.appannie.com/blog/game-of-phones
11.22. Mobile operators

11.23. Mobile app revenues
12. 12 BLACKBERRY APPLICATION DEVELOPMENT

12.1. What is BlackBerry?

• BlackBerry is a generic, a brand and NOT a producer
• The producer's name is RIM (Research In Motion)
• Initially, RIM has dealt with wireless solutions beyond the BlackBerry, however the BlackBerry was their most successful product
• The BlackBerry system is complete only if your device is used in conjunction with the appropriate services
• The two key BlackBerry services:
  • BES (BlackBerry Enterprise Server)
  • BIS (BlackBerry Internet Service)

• Devices:
  • The phones which can use services produced by RIM
  • Initially, one only could use them to send emails
  • The devices run the RIM’s proprietary operating system, which was named BlackBerry OS
  • The current version is 7.1 (2013 April)

12.2. BES (BlackBerry Enterprise Server)

• Primarily designed for corporate customers
• Very safe, well-administered, comprehensive mobile e-mail service
• Helps the employees to get e-mail access outside the office that satisfies all requirements
• The e-mails, the address book and the calendar are synced continuously
• The BES works with policy-based authorization system, the various device functions can be disabled at policy level
12.3. Operation of BES (BlackBerry Enterprise Server)

- In usual enterprise infrastructures generally there is an Exchange server, a proxy, which provides access to the Internet, and a web and/or database server
- All these elements are managed by a Windows server (this is the most common solution)
- BES needs to be installed in this environment, which usually communicates with the Exchange server via MAPI interface and can access to the internet via the proxy
- As the electronic mailbox receives an e-mail, which mailbox is stored on the Exchange server of the BlackBerry user, the BES gets informed about it immediately, and it notifies the Canadian or British BlackBerry server via the proxy and the server will forward the e-mail to the given country's mobile service provider
- Then the operator notifies the user's Blackberry device, which almost immediately receives the e-mail
- RIM uses 3DES algorithm for encryption
- The required keys can only be known by the BES and the BlackBerry device
- The keys need to be constructed during the server configuration
- For these reasons, the e-mail leaves the corporate infrastructure in an encrypted form and it will be encrypted until it appears on the device

12.4. BIS (BlackBerry Internet Service)

- It was produced for retail customers
- The user gets a mailbox in exchange for a monthly fee on the RIM's central BlackBerry server and reaches this mailbox from the device
- The user's POP3 or IMAP mailbox can be set to the BIS which sends a copy of the mail to the phone in 15 minutes but in practice immediately
- The BIS synchronizes with the customer's POP3 or IMAP mailbox at every operation
- Overall, it does NOT really offer more than just a POP3 or IMAP mailbox, but it is expensive
- However, there are applications that can NOT be used without BIS, such as:
  - BlackBerry Maps
  - BlackBerry App World
12.5. Services setting on BlackBerry device

- BlackBerry phones have a unique identifier, which is called PIN

- BIS service configuration:
  - After the purchase of the device you need to order the BIS service from your provider
  - Then you need to register the PIN
  - If everything is fine with the PIN and the SIM card is active, the BIS sends the service book to the phone
  - The service book is a package that activates the email setup options of the phone
  - Then you need to complete the setting procedure

- BES service configuration:
  - Need to install the BES into the corporate infrastructure
  - Then you need to register the PIN numbers of the device, which causes a common key creation
  - If everything is fine with the PIN, the BES sends the service book to the phone
  - Users do NOT have to deal with the email settings, because they are on the BES server

12.6. Development environment
Currently there are a number of available BlackBerry application development tools.

The two most common and most recommended:

- BlackBerry Java Development Environments (BlackBerry JDEs)
- BlackBerry Eclipse Plug-in

Both are valid for the following:

- The use of standard Java SDK
- They have a simulator, which have a lot of features such as:
  - Call event simulation
  - GPS coordinates setting
  - Battery power simulation
  - Different network properties settings
- They ensure appropriate testing and debugging options
- Easy to use
- Free

12.7. Environment for development (BlackBerry JDEs) - 1

- Download and install Java SE JDK
- At least Java SE JDK 6, update 14 is needed
12.8. Environment for development (BlackBerry JDEs) - 2

- If you have a 64 bit version of Windows, it is also required a 32 bit version of Java (!)


- Download BlackBerry JDE

- Select the appropriate JDE
  - The JDE version number matches the version number of the BlackBerry OS related (e.g., BlackBerry JDE 7.0 matches with BlackBerry OS 7.0)
  - The BlackBerry OS is backward compatible (e.g., BlackBerry OS 6.0 applications using BlackBerry OS 7.0 will work without changing the code, but this is not true vice versa)

- [http://us.blackberry.com/developers/javaappdev/javadevenv.jsp](http://us.blackberry.com/developers/javaappdev/javadevenv.jsp)

- Install BlackBerry JDE

- Starting the installer

- After installation and the run of the JDE the left panel window will show the sample.jdw workspace, at the right side the HelloWorldDemo.java file will be displayed in the editor window

- The sample.jdw workspace contains various BlackBerry examples

- The BlackBerry API reference documentation is available from the Help menu

12.9. Environment for development (BlackBerry Eclipse Plug-in) - 1

- Download and install Java SE JDK

  - As described by the BlackBerry JDE
• Download BlackBerry Eclipse Plug-in

• Select the appropriate Plug-in
  
  • You need to take the same considerations as you did in connection with the BlackBerry JDE

• If you download the Plug-in, you will download in fact an Eclipse installation package, which also contains the BlackBerry Plug-in which needed to develop

• http://us.blackberry.com/developers/javaappdev/javaplugin.jsp

• Install BlackBerry Eclipse Plug-in

• Start the installer

• After installation you do NOT need to do any extra settings in your new Eclipse to start the development

• After installation you can find the sample codes in the following folder: \ eclipse\ plugins\ net.rim.ejde.componentpack6.0.06.0.0.30 components samples_

12.10. Environment for development (BlackBerry Eclipse Plug-in) - 2

12.11. Development

• The programming language is Java, so only the BlackBerry specific things will bring new features for those who knows the Java language

• User interface:

  • No GUI descriptor xml file, as in case of Android and Windows Phone 7, and there is NO built-in graphical GUI design tool

  • In case of GUI design tools there are 3rd party solutions, which have a disadvantage that generate a lot of unnecessary code, they are NOT reliable, so the use of them is NOT recommended

  • For this reasons the making a complex GUI is very difficult (e.g.: To adjust a ButtonField to the bottom of the screen or to change the color of a LabelField, you need to override classes

• Tutorials and help to get started

  • http://www.blackberry.com/developers/startpages/java/index.html
• The BlackBerry developer forum can help to solve problems
  • http://supportforums.blackberry.com

12.12. Application signature and verification

• After the successful build of the code, the compiler will also generate a file with .cod extension
• If you use the default settings, you can find this file in the deliverables folder of your project folder
• You need to upload this .cod file to your BlackBerry device to install the application
• If the application uses one of the following packages, you need to sign the particular .cod file:
  • net.rim.blackberry.api.browser
  • net.rim.blackberry.api.invoke
  • net.rim.blackberry.api.mail
  • net.rim.blackberry.api.menuitem
  • net.rim.blackberry.api.options
  • net.rim.blackberry.api.pdap
  • net.rim.blackberry.api.phone
  • net.rim.device.api.crypto
  • net.rim.device.api.io.http
  • javax.microedition.rim

• If you install the application to your BlackBerry device without the signature, then at the start of the application, you will receive the cause of the failure in a pop-up window
• If you want to run and test your application on a simulator, then you do NOT need the signed .cod file, you can use any package

12.13. The process to sign your application (1)

• 1. step: Get the signing keys
  • You need to fill the form on the following page:https://www.blackberry.com/SignedKeys/
  • After you send the from, you will receive 3 email with the signing keys (RBB, RRT és RCR)
• 2. step: Install signing keys with the help of Eclipse
• Window → Preferences → BlackBerry Java Plug-in → Signature Tool → Install New Keys

• You need to type the received keys in the pop-up window (can only be installed one after the other)

• After the key selection you need to create a private key with an own password (you need to do this only once)

• After that you need to type the PIN number, which you filled at the first step, then you need to click to the Register button

• After a successful install you can find in the eclipse plugins net.rim.ejde vmTools folder the following 3 files:
  • sigtool.db
  • sigtool.sck
  • sigtool.set

12.14. The process to sign your application (2)
• 3. step: Sign the .cod file with the help of Eclipse

  • Right click on the project → BlackBerry → Sign with Signature Tool

  • Then the Signing Password window will pop up, here you need to type the password, received at the install of the signing keys

  • Subsequently, the Signature Tool automatically sends the signing request to the BlackBerry server

  • You need to sign the cod file after each build operation, after that you can install the application to the BlackBerry

12.15. Install application to the BlackBerry device
• There are several ways, some of these are:
  
• With Eclipse
  
• Using Javaloader
  
• With BlackBerry Desktop Manager
  
• Install with Eclipse
  
• Right click on the project → Debug As → BlackBerry Device
  
• Install using Javaloader
  
• In the command prompt you need to navigate to the folder containing javaloader.exe
  
• Then, the following command installs the application to the BlackBerry:
    
• javaloader -u load yourfilename.cod
  
• Install with BlackBerry Desktop Manager
  
• Download BlackBerry Desktop Manager:
    
• http://blackberry-desktop-software.en.softonic.com/blackberry
  
• In addition, you will need the signed .cod file and the automatically generated .alx file which you can find in the deliverables folder of the project folder
12.16. Application sharing, distribution

- You can distribute the application in any form such as email: only the signed .cod file and the .alx file is required
- The user can easily install the application to the device with the BlackBerry Desktop Manager which has a graphical user interface

- The easiest way of the widespread distribution is to use the BlackBerry App World, which was supported by all BlackBerry devices with 4.5.0 or newer BlackBerry OS version

12.17. BlackBerry App World

- Application distribution service and application by RIM which was released on April 1, 2009
- There are two options to download and install applications:
  - With App World which you can install to you BlackBerry, you can download from here:http://uk.blackberry.com/services/appworld/
  - With a PC via web if your computer is connected to the BlackBerry device:http://appworld.blackberry.com/webstore/
• Most of the phones have a pre-installed App World client,
• RIM employees are developing it continuously, the current version is 3.1
• Nearly 50,000 applications are available for download in the BlackBerry App World
• Currently available in 129 countries

12.18. BlackBerry App World - Upload application
• The first step is to create a vendor account(https://appworld.blackberry.com/isvportal/home/login.seam?pageIndex=1)
  • Accept BlackBerry App World contract
  • Type personal contact information
  • Type company contact information
  • Type PayPal account
• So the developer can get the money for the developed applications

• It is free to create a vendor account (earlier you had to pay 200$)

After that the RIM will check the data that you entered during the account creation procedure, then it will send an email about the account acceptance and additional tasks.

The application that you want to upload must fully comply with the conditions of the vendor guidelines.

After you shared your application the RIM will contact with you, and you will get the results and the additional tasks.

13. 13 IPHONE APPLICATION DEVELOPMENT

13.1. The iPhone and the iOS

• iPhone
  
  • Line of smartphones designed and marketed by Apple Inc.
  
  • The first iPhone was introduced on June 9, 2007 by Steve Jobs and was released in the same year on June 29.
  
  • The biggest novelty was the multi-touch capacitive touchscreen.
  
  • The newest iPhone model is the iPhone 5, Apple announced it on September 12, 2012.
  
  • Runs Apple's iOS mobile operating system, originally named "iPhone OS".

• iOS

  • Although the iPhone was already introduced on 2007, the operating system was named officially iPhone OS or short iOS on March 6, 2008.
  
  • The current version is 6.0, which was released on September 24, 2012.
  
  • Interestingly, that the multitasking feature was released only with the iPhone OS version 4.0.
13.2. What do you need to start developing for iPhone?

- Mac (MacBook Air, MacBook Pro, iMac, Mac Mini, Mac Pro)
  - If you don't have a Mac, you can hack
    - You can try to install a Hackintosh to your PC
    - You can try to install a virtual machine with Mac OS (e.g., VMWare)
  - But hacking is NOT recommended because the system will be slow and it does NOT support a lot of feature
- iPhone
  - It is NOT recommended to develop with iOS simulator
    - It is uncomfortable to test multi-touch with mouse on the simulator
    - The simulator has a big CPU performance, so the incidental performance problems will not come out
    - It is also true for the memory, if our application has a big memory usage, the iOS can close it
- Xcode
  - Apple's powerful integrated development environment
• Can easily download from App Store
• After the download the operating system will automatically install the XCode
• After launching the Xcode do NOT need any configuration setting

13.3. Development environment - Xcode (1)

• The Xcode developers tools package includes
  • XCode IDE
    • This is the development environment
    • Integrates all the tools you need to develop
  • Apple LLVM Compiler
    • Apple's next generation compiler technology
    • Identifying coding mistakes real time and gives solution for them
  • Instruments for Performance and Behavior Analysis
    • Our application monitored with various aspects, such as CPU usage and memory allocation size
    • Graphically displays the collected data
Service Delivery Architectures and Platforms

- iOS Simulator
  - Application for testing (mostly in the development stage)
  - It is NOT possible to test the accelerometer, but you can try the multi-touch

- The newest Xcode version is 4.5 which was released on September 19, 2012
- It is free

13.4. Development environment - Xcode (2)

13.5. Developer license

- What is the developer license?
  - You can get developer license if you enroll to Apple Developer Program
  - To do this, developers must register themselves in the iOS Dev Center
- When do you need it?
  - If you want to test your application not only on the simulator, but also on a real iPhone
  - If you want to upload your application to the App Store
- Which developer program is for you?
• iOS Developer Program
  • iOS SDK, testing on a real phone, Ad Hoc distribution, App Store distribution
  • 99$ / year

• iOS Developer Enterprise Program
  • iOS SDK, testing on a real phone, Ad Hoc distribution, in-house distribution
  • 299$ / year

• iOS Developer University Program
  • iOS SDK, testing on a real phone
  • Free

13.6. The process to get the developer license (1)

• 1. step: Start with the registration
  • Go to the iOS Dev Center page:http://developer.apple.com/devcenter/ios/index.action
  • Register → Get Started

• 2. step: Apple Developer Registration
  • Apple ID - You can complete the registration process with a new or an existing Apple ID
  • Personal Profile - Need to type security and personal informations
  • Professional Profile - Need to fill different datas (which platform we want to develop, the type of applications that we plan to write, etc.)
  • Legal Agreement - Need to accept terms and conditions
  • Email verification - Apple sends an email including a code to the given address, after you type this code, Apple will accept your Apple ID

• 3. step: Payment
  • Go to the iOS Developer Program page:http://developer.apple.com/programs/ios/
13.7. The process to get the developer license (2)

- 4. step: Apple Developer Program Enrollment
  - Enter Account Info - Individual or company (need to choose Individual)
  - Select Program - iOS, Mac, or Safari (need to choose iOS)
  - Review and Submit - Need to verify personal and billing informations.
  - Agree to License - Need to accept terms and conditions
  - Purchase Program - Need to click on the Add to cart button, which will navigate to the hunarian App Store page, the iOS Developer Program is already in the basket

- Activate Program - After the purchase Apple will activate your Apple ID and you also get informations about it by email

13.8. App Store

- Online application distribution platform for iOS developed and maintained by Apple Inc.
- The service allows users to browse and download applications from the iTunes Store
- The App Store opened on July 10, 2008
• There are two options to download and install applications:
  
  • Download directly to the device (iPhone, iPad, iPod touch)

  • Can be downloaded with a PC or Mac, then you can install with the iTunes application to the device (http://www.apple.com/hu/itunes/)

• More than 500,000 applications are available for download in the App Store, which include both free and non-free applications

13.9. App Store - Upload application

• 1. step: Login to iTunes Connect page
  
  • Go to iTunes Connect page: https://itunesconnect.apple.com/WebObjects/iTunesConnect.woa

  • Need to type Apple ID and password

  • You can login if you have a developer license

  • Manage your Applications → Add new App (need to fill the necessary fields)
• The Bundle ID need to match with the Bundle Identifier in the Info.plist file of the application

• Go to the application detail page, choose "Ready to Upload Binary"

• 2. step: Upload with the help of XCodes
  • XCode → Product → Edit Scheme → Choose IOS Device at Destination
  • XCode → Product → Archive
  • Then Xcode will automatically make a clean and a build, then finally it will archive the file
  • If it succeeds, the Organizer window will automatically pop up, where you have to find your application among the Archives
  • Submit → Need to type username and password which you have typed at iTunes Connect → Need to choose application and identifier from drop-down lists → Next → Finish
  • So we have sent our application to review, Apple will inform you about the result via email

14. 14 ANDROID APPLICATION DEVELOPMENT

14.1. What is Android?

• Android is a Linux-based operating system designed primarily for touchscreen mobile devices such as smartphones and tablet computers, developed by Google

• Initially developed by Android Inc, whom Google financially backed and later purchased in 2005

• Android was unveiled in 2007 along with the founding of the Open Handset Alliance, a consortium of 86 hardware, software, and telecommunication companies

• The goal was to develop an unified, open source mobile operating system

• Due to the open source platform, manufacturers can adjust the system appearance to their own style and can extend the system with their own packages.

• The application development is manufacturer and hardware independent.

• Supported functions:
  • Data storage: SQLite
  • GSM/EDGE, UMTS, WiFi, BlueTooth
  • Media: MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF
  • Messages: SMS/MMS
• Web: Built-in Webkit-based browser

• Hardware: Camera, GPS, Accelerometer, Digital compass, Proximity Sensor

• Multi-touch, Multi-tasking, Flash, Tethering, ...

14.2. Android architecture (1)

14.3. Android architecture (2)

• Applications
  • At this top layer, you will find applications that ship with the Android device (such as Phone, Contacts, Browser, etc.), as well as applications that you download and install from the Android Market. Any applications that you write are located at this layer.

• Application Framework
  • Exposes the various capabilities of the Android OS to application developers so that they can make use of them in their applications.

• Libraries
• These contain all the code that provides the main features of an Android OS. For example, the SQLite library provides database support so that an application can use it for data storage. The WebKit library provides functionalities for web browsing.

• Android Runtime

• At the same layer as the libraries, the Android runtime provides a set of core libraries that enable developers to write Android apps using the Java programming language. The Android runtime also includes the Dalvik virtual machine, which enables every Android application to run in its own process, with its own instance of the Dalvik virtual. Dalvik is a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU.

• Linux Kernel

• This is the kernel on which Android is based. This layer contains all the low-level device drivers for the various hardware components of an Android device.

14.4. Obtaining the required tools (1)

• SDK:
  • Java Development Kit (http://java.sun.com/javase/downloads/index.jsp)
  • Android SDK starter package (http://developer.android.com/sdk/index.html)

• Integrated Development Environment:
  • Eclipse IDE 3.5+ (http://www.eclipse.org/downloads/) + Android Development Tools (ADT) plugin for Eclipse
  • Or: IntelliJ IDEA, NetBeans, etc.

• ADT plugin:
  • Eclipse → Help → Install New Software...
    • Work with: https://dl-ssl.google.com/android/eclipse/
    • Installing ADT and DDMS
  • Setting SDK’s path: Eclipse → Window → Preferences → Android → SDK Location

14.5. Obtaining the required tools (2)
• The Android SDK doesn't contain the APIs, they can be downloaded with the "Android SDK and AVD Manager":

• To run an application, an Android device is needed:
  • This can be a real Android device, then you need:
    • USB cable and driver
    • On the device: Settings → Applications → Development → USB debugging
  • Or you can use virtual devices:
    • Any number of virtual devices can be created with different settings: API version- SD card capacity-
      Screen resolution- Memory capacity,- Cache partition- Pixel density- Etc.

14.6. Creating your First Android Application

• Using Eclipse, create a new project by selecting File → Project:
  • New → Project... → Android → Android Project
  • Project name: The name of the project
• Build target: API version
• Application name: A user-friendly name for your application
• Package name: The name of the package, you should use a reverse domain name for this
• Create Activity: The name of the first activity in your application
• Min SDK Version: The minimum version of the SDK that your project is targeting

14.7. Anatomy of an Android Application

• src
  • Contains the .java source files for your project.

• gen
  • Contains the R.java file, a compiler-generated file that references all the resources found in your project. You should not modify this file.

• Android 1.5
  • This item contains one file, android.jar, which contains all the class libraries needed for an Android application.
• assets
  • This folder contains all the assets used by your application, such as HTML text files, databases, etc.

• res
  • This folder contains all the resources used in your application. It also contains a few other subfolders: drawable-
    <resolution>, layout, and values.

• AndroidManifest.xml
  • This is the manifest file for your Android application. Here you specify the permissions needed by your
    application, as well as other features (such as intent-filters, receivers, etc.)

14.8. Main elements of applications

• Activity
  • An activity represents a single screen with a user interface. An application can have zero or more activities

• Service
  • A service is a component that runs in the background to perform long-running operations or to perform
    work for remote processes. A service does not provide a user interface.

• Content Provider
  • A content provider manages a shared set of application data. You can store the data in the file system, an
    SQLite database, on the web, or any other persistent storage location your application can access. Through
    the content provider, other applications can query or even modify the data.

• Built-in Content Providers:
  • Call list, Contacts, Browser history, bookmarks, Telephone settings

• Broadcast Receiver
• A broadcast receiver is a component that responds to system-wide broadcast announcements. Many broadcasts originate from the system—for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured. Applications can also initiate broadcasts—for example, to let other applications know that some data has been downloaded to the device and is available for them to use.

14.9. GUI (1)

• Despite the fact that the system is planned to be unified, hardwares are significantly different. One of the major differences are the screen size and pixel density.

![GUI descriptors](image)

14.10. GUI (2)

• The problem is solved by the GUI descriptor class, where the elements' size and position can be defined by relative values and ratios

• More fixed-sized version can be added from the same picture:
  • You can place the different resolution pictures to the /res/drawable-ldpi, −mdpi, −hdpi directory
  • When you reference the picture, you have to add the path:/res/drawable/<filename> and the application will use the appropriate picture for the actual device

• The display structure can be modified manually from the xml file or from source code in case of dynamic applications
  • An activity is linked to a layout file (/res/layout/<filename>.xml)

• Or can be edited in Eclipse editor

• Or you can use http://www.droiddraw.org/ that generates the xml file.

14.11. GUI settings
14.12. Overview

- The system is based on Linux, thanks to the Dalvik VM, the programming language is Java
  - The platform has rich Android-specific API support
• There is no language-level novelty for those who know Java

• Unique things can be found on: http://developer.android.com e.g., Activity lifecycle, AsyncTask

• Solutions for practical problems can be found on web pages:
  • e.g., http://stackoverflow.com

14.13. Signing an application

• All Android applications must be digitally signed before they are allowed to be deployed onto a device (or emulator). Unlike some mobile platforms, you need not purchase digital certificates from a certificate authority (CA) to sign your applications. Instead, you can generate your own self-signed certificate and use it to sign your Android applications.

• Eclipse uses a default debug keystore. If you are publishing an Android application, you must sign it with your own certificate.

• While you can manually generate your own certificates using the keytool.exe utility provided by the Java SDK, Eclipse has made it easy for you by including a wizard that walks you through the steps to generate a certificate.

• It will also sign your application with the generated certificate (which you can also sign manually using the jarsigner.exe tool from the Java SDK).

• Aspects:
  • Only the updates that signed with the same certificate can overwrite the previous version, otherwise it will create a new application
• Applications with the same certificate can access each other’s data.

• It is recommended to sign all your applications with the same key.


• You can distribute your application manually (web, e-mail, etc.)

• Unsuitable for widespread distribution: difficult to install, updates, paid applications?

• Google Play formerly known as the Android Market, is a digital distribution service operated by Google.

• It opened on 23 October 2008, and provides an online store for music, movies, books, magazines, and Android applications, as well as an online music player.

• The service is accessible through the internet and the Play Store mobile app, included with most Android and Google TV devices.

• Handles the paying, automatic updates, filtering.

• Users can rate and add comments on applications.

• In October 2012 Google announced that Google Play had 700,000 apps available to download, matching the number of apps in Apple's App Store.

14.15. Google Play

• Support for paid applications was introduced on 13 February 2009 for developers in the United States and the United Kingdom, with support expanded to an additional 29 countries on 30 September 2010.

• Number of countries where free applications can be

  • Downloaded: 135

  • Uploaded: 147

• Number of countries where paid applications can be

  • Downloaded: 128

  • Uploaded: 29
• You can make purchases through Google Wallet using a credit, debit, or gift card.

• Revenue earned from the Android Market is paid to developers (70% of the application price.) via Google Checkout merchant accounts, or via Google AdSense accounts in some countries.

• Transaction fee: If monthly sales under $3,000: 2.9% + $0.30 per transaction. Orders from buyers whose billing address is outside the country of your Google Checkout account will be assessed an additional 1% processing fee.

• Money can also be earned by placing ads in the application (e.g., AdMob)

14.16. Android Market registration

• To use the market, a Google account is needed. Use this account to enter to https://market.android.com/publish to start the registration process.

• The registration fee is $25.

14.17. Upload an application to Android Market

• After the registration you can upload your application.
• During the process, you have to upload the .apk file and give some information about the application (some of them are optional):
  • Screenshots, icons, banners, description, changelog
  • Pricing setup
  • In which country you want to make it available
  • Supported devices
• Finally, click on the Publish button to make it available.

15. 15 WINDOWS PHONE APPLICATION DEVELOPMENT

15.1. Predecessors: Pocket PC and Windows Mobile (1)

• What is Pocket PC?
  • A device type (and OS) name, which Microsoft used for classifying mobile devices.
  • The first version was the Pocket PC 2000, released on 19. April 2000. The operating system followed the style of Windows 98 but did not have mobile phone capabilities. The OS is based on the Windows CE 3.0 kernel which is an operating system core for embedded devices.

• What is Windows Mobile?
  • Windows Mobile is a family of mobile operating system developed by Microsoft for smartphones and Pocket PCs.
  • The framework is .NET Compact Framework, which is a version of .NET framework used in desktop environment but with limited resources, developed for mobile and embedded devices.
  • The first version was the Windows Mobile 2003 for Pocket PC, released on 23. June 2003, but afterwards the former Pocket PC operating system was also called Windows Mobile
  • The last official release was the Windows Mobile 6.5.3, released on 2. February 2010. Compared to the first Pocket PC it has a lot of new features and services, but the user experience was far behind its competitors (Android, iOS)

15.2. Predecessors: Pocket PC and Windows Mobile (2)
15.3. Present: Windows Phone

- Windows Phone is a family of mobile operating systems developed by Microsoft
  - It is the successor to its Windows Mobile platform, although incompatible with it.
  - Unlike its predecessor, it is primarily aimed at the consumer market rather than the enterprise market. It was first launched in October 2010 (Windows Phone 7)
  - The concept focuses on to enhance the user experience
- New user interface called Metro
- Wide variety of services
- Stable and fast devices
- Strictly defined hardware requirements:
  - 800 x 480 WVGA screen
  - DirectX 9 hardware acceleration
  - Standard sensor kit
    - A-GPS, accelerometer, compass, light sensor, proximity sensor
    - Developers can only use the A-GPS and the accelerometer
  - One digital camera at least
  - Common hardware controls and buttons:
    - Start, Search and Back buttons + volume control etc.
  - WI-FI
  - 256 MB (or more) RAM and 8 GB (or more) flash memory.
• Last official version: WP 8 Apollo (29. October 2012)

15.4. Windows Phone 7 overview
Source: MSDN

### 15.5. Windows Phone 7 Architecture

![Windows Phone 7 Architecture Diagram]

Source: MSDN
15.6. Runtimes

Source: MSDN

• All development is done in managed code, in a protected sandbox allowing for the rapid development of safe and secure applications.

• The Windows Phone Application Platform provides two frameworks for developing applications
  
  • Silverlight: Silverlight is the ideal framework for creating Rich Internet Application-style user interfaces.
  
  • XNA: The XNA Framework is composed of software, services, and resources focused on enabling game developers to be successful developing on Microsoft gaming platforms (WP7 and XBOX games).
  
  • From Windows Phone 7.5 Mango, developers can use both frameworks in one application.

15.7. Development environment

• The development is built on existing Microsoft technologies (Silverlight, XNA, C# language) and devices.

  • Visual Studio 2010 (2012) integrated development environment
    
    • Includes: designer, project-tracking, project-management tools, package manager and manifest generator
  
  • Expression Blend 4
    
    • It allows designers to create creative and unique UI design independently from programmers. The result is an UI described in XAML that developers can use in Silverlight applications.
  
  • Windows Phone 7 emulator for easier development
  
  • Documentations, training kits, design, guideline and application examples
15.8. Configuration of the development environment

- Download the Windows Phone SDK!
- Installing the SDK we get a complete development environment
  - Microsoft Visual Studio 2010 Express for Windows Phone
    - If the Visual Studio 2010 have been installed earlier then the SDK is going to be integrated with it
- Windows Phone Emulator
- Windows Phone SDK Assemblies
- Silverlight 4 SDK and DRT
- Windows Phone SDK Extensions for XNA Game Studio 4.0
- Microsoft Expression Blend SDK for Windows Phone 7
- Microsoft Expression Blend SDK for Windows Phone OS 7.1 (Mango)
- WCF Data Services Client for Window Phone
- Microsoft Advertising SDK for Windows Phone
- Ready to develop!

15.9. Visual Studio 2010
15.10. Expression Blend 4

15.11. Deploying the application (1)

- Application can be tested on the Windows Phone emulator

- If you want to test your application on a real device you have to do the follows:

  1. You need a Windows Live ID (e.g. MSN or Hotmail address)

  2. You have to register yourself as a developer to the APP HUB
     - http://create.msdn.com/
     - The fee is 99$/year. Then you can upload your applications to the Marketplace. Furthermore as a developer you can register (device unlock) three WindowsPhone 7 mobile phone. You can test your application only on registered phones.
     - For students, the registration is free for 12 months. On the https://www.dreamspark.com/ page you have to confirm that you are a student. Then you can register yourself as a developer on the APP HUB.

  3. Download the Zune software
15.12. Deploying the application (2)

1. You have to register your device
   - The maximum number of registered devices are three (only one for students)
   - Turn on the phone and check your date and time settings!
   - Connect your phone to your computer with USB. Check if Zune is started!
   - Start the Windows Phone Developer Registration Tool! (Installed with the SDK)
   - Check if the phone status is "Phone !
   - Type your Live ID, used when you registered to APP HUB, and your password then register your device!
   - Don't forget to delete the registration if you sell your phone!

1. Deploying the application to a real device! (version A)
   - Use Visual Studio 2010 and switch from Windows Phone Emulator to Windows Phone Device!
15.13. Deploying the application (3)

1. Deploying the application to a real device! (version B)
   - Use the Application Deployment utility, installed with the SDK!
   - Choose the destination device!
   - Type the path of the your application's XAP file!
     - XAP is a renamed ZIP file made by Visual Studio during the Build. You can find it in the Project/Bin/Debug folder.
     - If you rename the file to ZIP, you can look at its content.

1. You are ready, you can test it on your device!

15.14. Market (1)

- The finished application can be delivered to users via the Marketplace.
  - On the management page you can track the downloads, revenues, and you can also update the application.
- First, you need a Live ID and an APP HUB ID.
- In order to publish your application you have to send the XAP file to Microsoft for review and approval.
• If you have followed the requirements, your application will receive a certificate and will be signed. After this, it is publishable to the Marketplace.

15.15. Market (2)

• To get the certificate you have to login to the APP HUB
• Choose the my dashboard / Windows Phone / submit a new app menu
• Follow along the form and fill the required data.

• Name of the application
• Type of publication (public, private)
• Path of XAP file (maximum 255 MB)
• Category of the application
• Description of the application
• Your contact
• Screenshots
• Regions where you want you application to be reachable
• Price (0$ is free, paid application between 0.99$–499$)
• You can decide if the application gets the certificate, it will be published automatically or you want to do it manually

15.16. Market (3)

• Business models
  • Free applications (max 100 / APP HUB registration)
  • Paid applications (unlimited)
  • Free applications with built in commercials (Microsoft Advertising SDK)
• Trial versions of paid applications

• Marketplace
  • Country/Region where you choose to distribute your application:
  • A Global Publisher is a third-party company that works with developers in countries or regions where App Hub has not yet launched.

• Payments
  • The payments to developers occurs when the value of downloaded applications reaches $200. 70% of the money goes to the developer minus taxes

15.17. Summary

15.18. Windows Phone
Pivot control

Panorama control
• Use the new, spectacular controls to enhance visual experience

• Take advantage of the collaboration between WP7 and Microsoft cloud to provide better service to the user

• Create location-based services based on the information received from the cloud

• Or just enjoy the benefits of a service, like a lost phone’s location on the map, remote shut down, etc.