6lowpan Format Proposal

David Culler and Jonathan Hui

67th IETF Meeting
6lowpan WG
San Diego

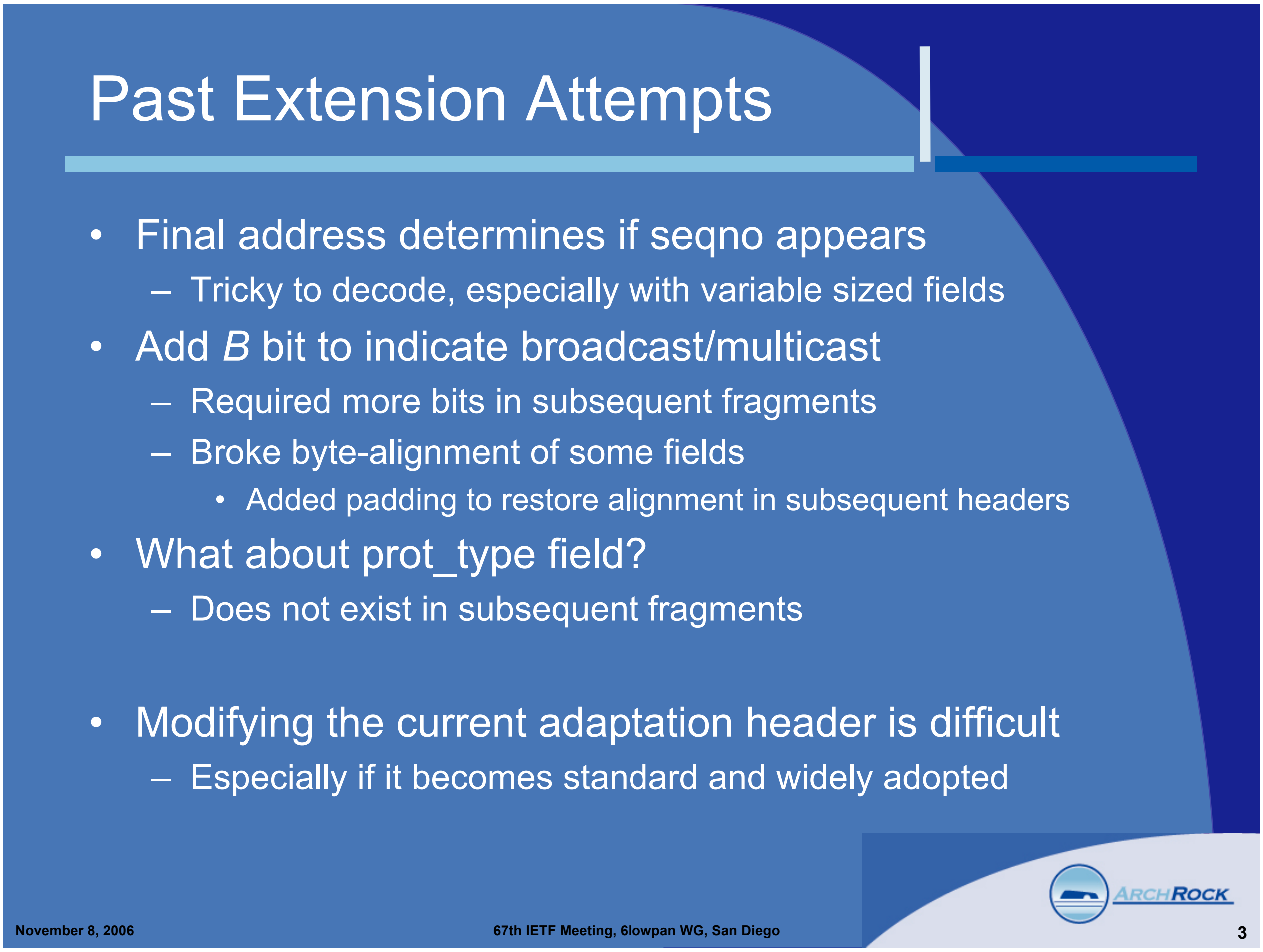
November 8th, 2006

Concerns

1. Inefficient parsing
   - Orthogonal concepts unnecessarily intertwined
   - Byte fields not byte aligned

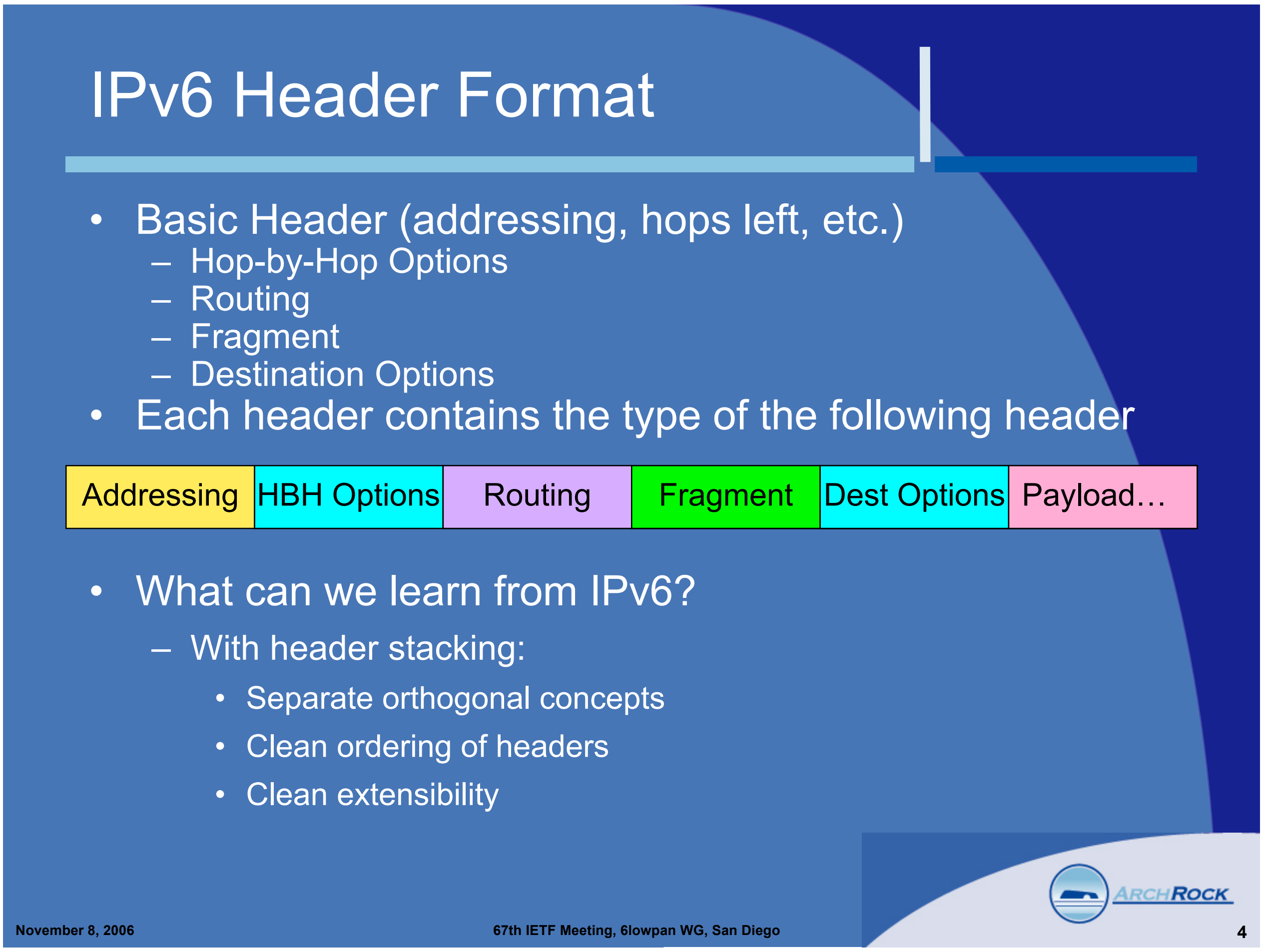
2. Unnecessary bits in simplest form
   - 1 hop, no frag packets contain frag and mesh bits

3. Leaves little room for extensions
   – No provision for additional routing parameters (except seqno)
   – Dangerous when there is no consensus
   – How do we add support for new protocols in the future?



Past Extension Attempts

- Final address determines if seqno appears
  - Tricky to decode, especially with variable sized fields
- Add *B* bit to indicate broadcast/multicast
  - Required more bits in subsequent fragments
  - Broke byte-alignment of some fields
    - Added padding to restore alignment in subsequent headers
- What about prot_type field?
  - Does not exist in subsequent fragments
- Modifying the current adaptation header is difficult
  - Especially if it becomes standard and widely adopted



IPv6 Header Format

- Basic Header (addressing, hops left, etc.)
  - Hop-by-Hop Options
  - Routing
  - Fragment
  - Destination Options
- Each header contains the type of the following header

Addressing	HBH Options	Routing	Fragment	Dest Options	Payload…

- What can we learn from IPv6?
  - With header stacking:
    - Separate orthogonal concepts
    - Clean ordering of headers
    - Clean extensibility

Current Header Format

<table>
<tr><td colspan="10"></td><td>1</td><td colspan="9"></td><td>2</td><td colspan="9"></td><td>3</td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="2">LF</td><td colspan="8">dgram_offset
prot_type</td><td>M</td><td>B</td><td colspan="4">rsv</td><td colspan="11">datagram_size</td><td colspan="5">dgram</td></tr>
<tr><td colspan="5">tag</td><td colspan="3">pad</td><td>O</td><td>F</td><td colspan="6">Hops Left</td><td colspan="8">Seqno</td><td colspan="8">Source</td></tr>
<tr><td colspan="8">Address</td><td colspan="16">Destination Address</td><td colspan="8">HC1</td></tr>
</table>

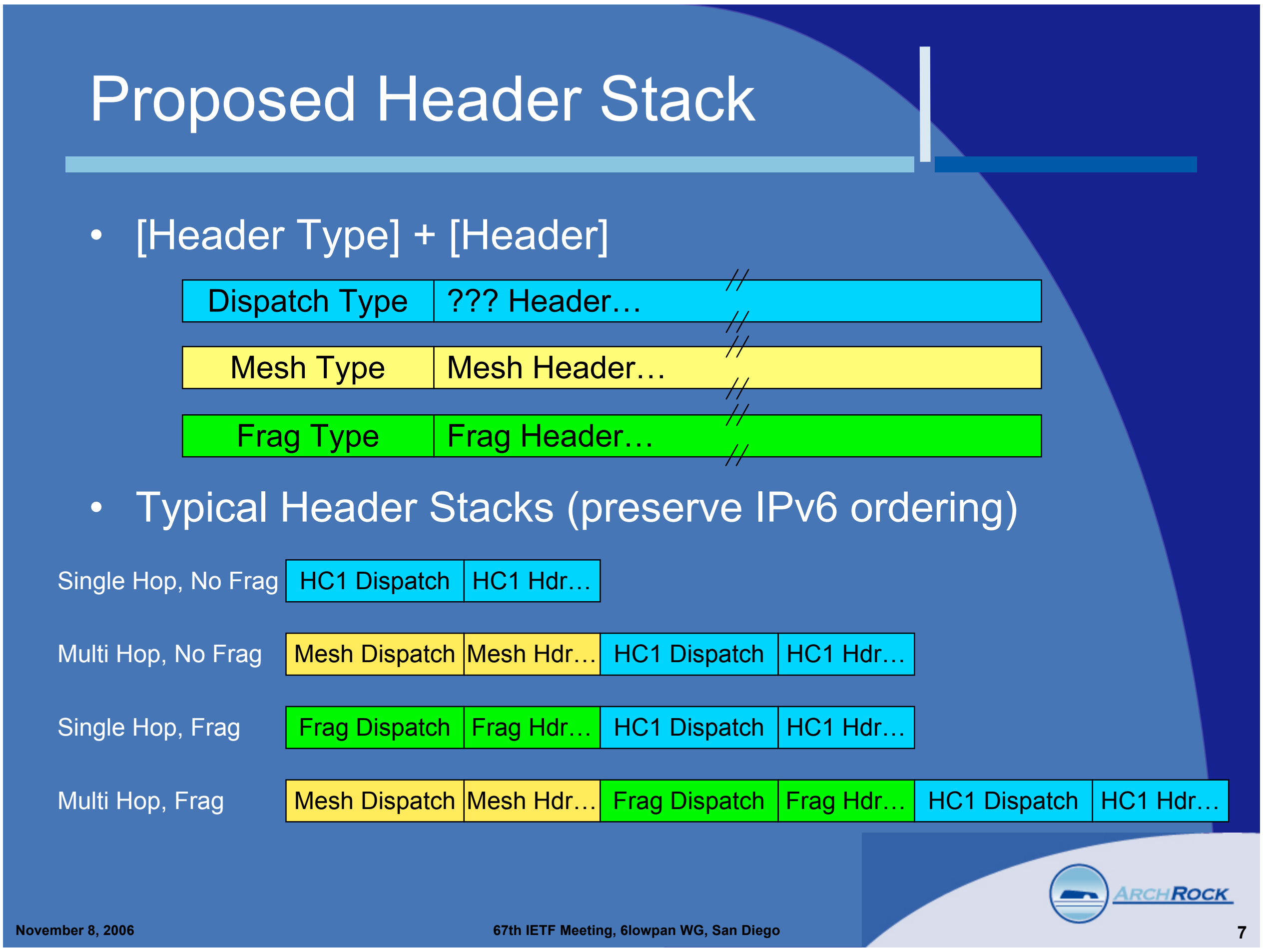
	6lowpan	IPv6
🟨	L2 Addressing	→ Addressing
🟦	L2 Mesh Options	→ Hop-by-Hop Options
🟩	L2 Fragmentation	→ Fragmentation
🟪	Upper Layer Protocol	→ Upper Layer Protocol

➔ ***Why not use IPv6 to guide header format?***

Goals

- **Preserve current functionality**

- Reduce complexity
- Better byte alignment
- Reduce header size
- Clean extensibility



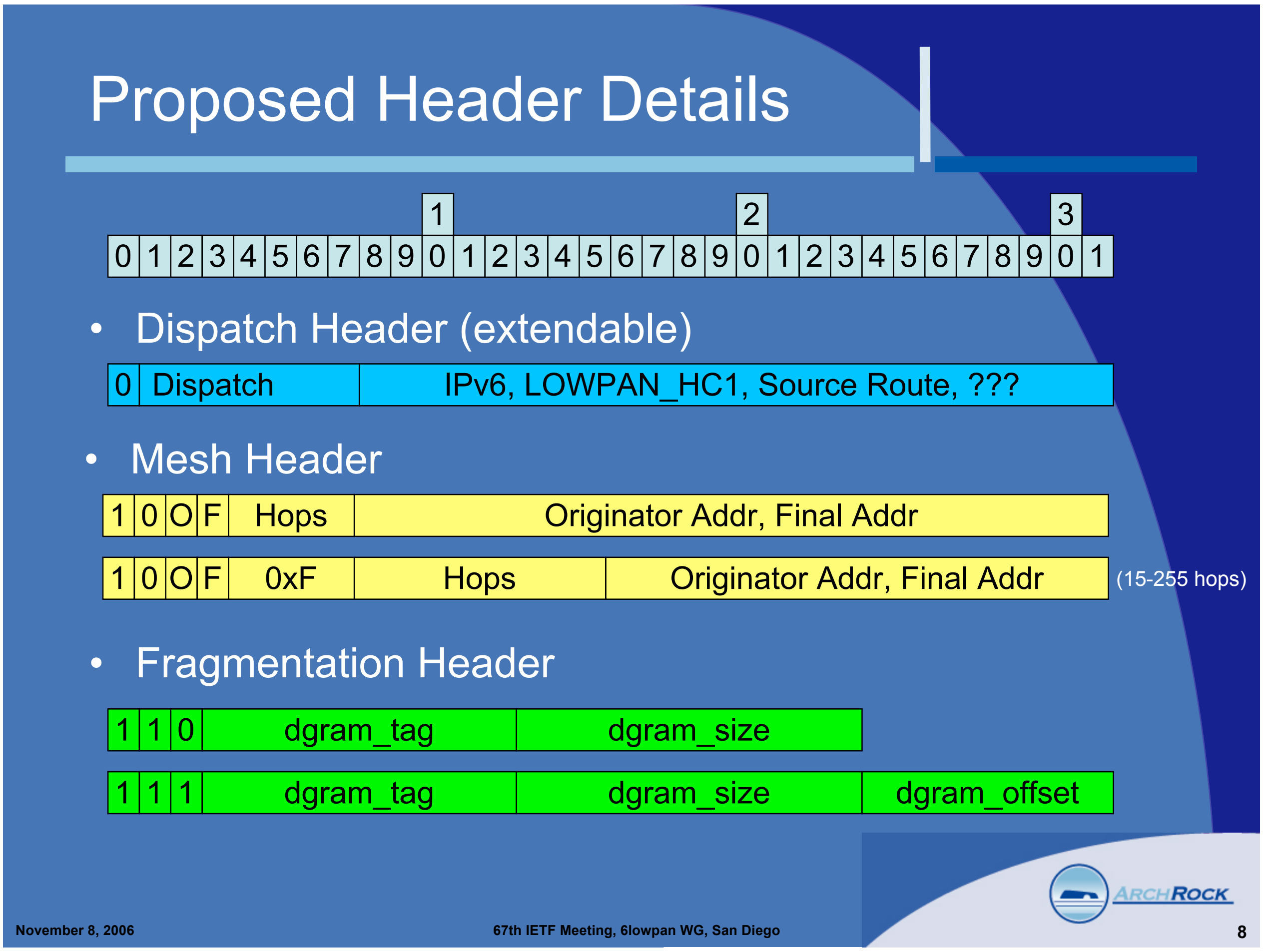
Proposed Header Stack

- [Header Type] + [Header]

Dispatch Type	??? Header…
Mesh Type	Mesh Header…
Frag Type	Frag Header…

- Typical Header Stacks (preserve IPv6 ordering)

Single Hop, No Frag	HC1 Dispatch	HC1 Hdr…				
Multi Hop, No Frag	Mesh Dispatch	Mesh Hdr…	HC1 Dispatch	HC1 Hdr…		
Single Hop, Frag	Frag Dispatch	Frag Hdr…	HC1 Dispatch	HC1 Hdr…		
Multi Hop, Frag	Mesh Dispatch	Mesh Hdr…	Frag Dispatch	Frag Hdr…	HC1 Dispatch	HC1 Hdr…



Proposed Header Details

										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1

- Dispatch Header (extendable)

0	Dispatch	IPv6, LOWPAN_HC1, Source Route, ???

- Mesh Header

1	0	O	F	Hops	Originator Addr, Final Addr		
1	0	O	F	0xF	Hops	Originator Addr, Final Addr	(15-255 hops)

- Fragmentation Header

1	1	0	dgram_tag	dgram_size	
1	1	1	dgram_tag	dgram_size	dgram_offset

Comparison: Compactness & Functionality

										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1

- Single Hop, No Fragmentation

2 Bytes	LF	prot_type	M	B	rsv	(current)
1 Byte	0	Dispatch				(proposed)

- Multi Hop, No Fragmentation

3 Bytes	LF	prot_type	M	B	rsv	O	F	Hops Left	
2 Bytes	1	0	O	F	Hops	0	Dispatch		

- Single Hop, Fragmentation

5 Bytes	LF	prot_type	M	B	rsv	datagram_size	dgram	
	tag	pad						
4 Bytes	1	1	0	dgram_tag	dgram_size	0	Dispatch	

Comparison: Compactness & Functionality

										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1

- Multi Hop, Fragmentation

6 Bytes	LF	prot_type	M	B	rsv	datagram_size	dgram		
	tag	pad	O	F	Hops Left				
5 Bytes	1	0	O	F	Hops	1	1	0	dgram_tag	dgram_size
	0	Dispatch							

- Multi Hop > 14 hops

6 Bytes	LF	prot_type	M	B	rsv	datagram_size	dgram			
	tag	pad	O	F	Hops Left					
6 Bytes	1	0	O	F	0xF	Hops	1	1	0	dgram_tag
	dgram_size	0	Dispatch							

➔ Proposed format handles 255 hops vs 63

Comparison: Compactness & Functionality

- Byte savings in most common cases
- Equal in large diameter network
  - Proposed format supports >63 hops
- Keeps every bit of functionality
  - Fragmentation fields unchanged
  - Mesh fields unchanged (except hops left)

Comparison: Extensibility

- Deep Networks

1	0	O	F	Hops	Addrs	HC1 Dispatch	HC1 Hdr…

1	0	O	F	0xF	Hops	Addrs	HC1 Dispatch	HC1 Hdr…

- Mesh Protocols (LOAD, AODV, DYMO, Source route…)

Mesh Dispatch	Mesh Hdr…	HC1 Dispatch	HC1 Hdr…

Mesh Dispatch	Mesh Hdr…	Proto Dispatch	Proto Hdr…	HC1 Dispatch	HC1 Hdr…

- Other Upper Layer Protocols

Mesh Dispatch	Mesh Hdr…	HC1 Dispatch	HC1 Hdr…

Mesh Dispatch	Mesh Hdr…	HCX Dispatch	HCX Hdr…

- Anything Else…

Mesh Dispatch	Mesh Hdr…	HC1 Dispatch	HC1 Hdr…

Mesh Dispatch	Mesh Hdr…	XXX Dispatch	XXX Hdr…	YYY Dispatch	YYY Hdr…	ZZZ Dispatch	ZZZ

In a Nutshell

- Reformatted 6lowpan header format

- Preserved all current capabilities
- Strictly more compact
  - Especially in the most frequent cases
- Follows IPv6 header stacking methodology
  - Strictly more expressive and extensible
    - Can cleanly address current concerns expressed in WG (e.g., Mesh Delivery, B, …) and potential future concerns (e.g., diagnostics)
  - Easier to parse
    - Clean, orthogonal, byte aligned

- Questions / Comments?

Backup

Generalize Dispatch?

- Adds 1 byte to Mesh and Frag header each
- Other advantages:
  - Full 11-bits for frag offset
  - Full 6-bits for hops

										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1

IPv6 Dispatch	IPv6 Header

HC1 Dispatch	HC1 Header

Mesh Dispatch	O	F	Hops	Originator Addr, Final Addr

Frag Dispatch	dgram_tag	dgram_size	
dgram_offset			

- 1-hop, no frag: -1 byte
- Mesh or Frag: Same
- Mesh and Frag: +1 byte

Proposed Header Details

										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1

- Dispatch Header (extendable)

0	Dispatch	IPv6, LOWPAN_HC1, Source Route, ???

0	0x7F	Dispatch	IPv6, LOWPAN_HC1, ???

- Mesh Header

1	0	O	F	Hops	Originator Addr, Final Addr

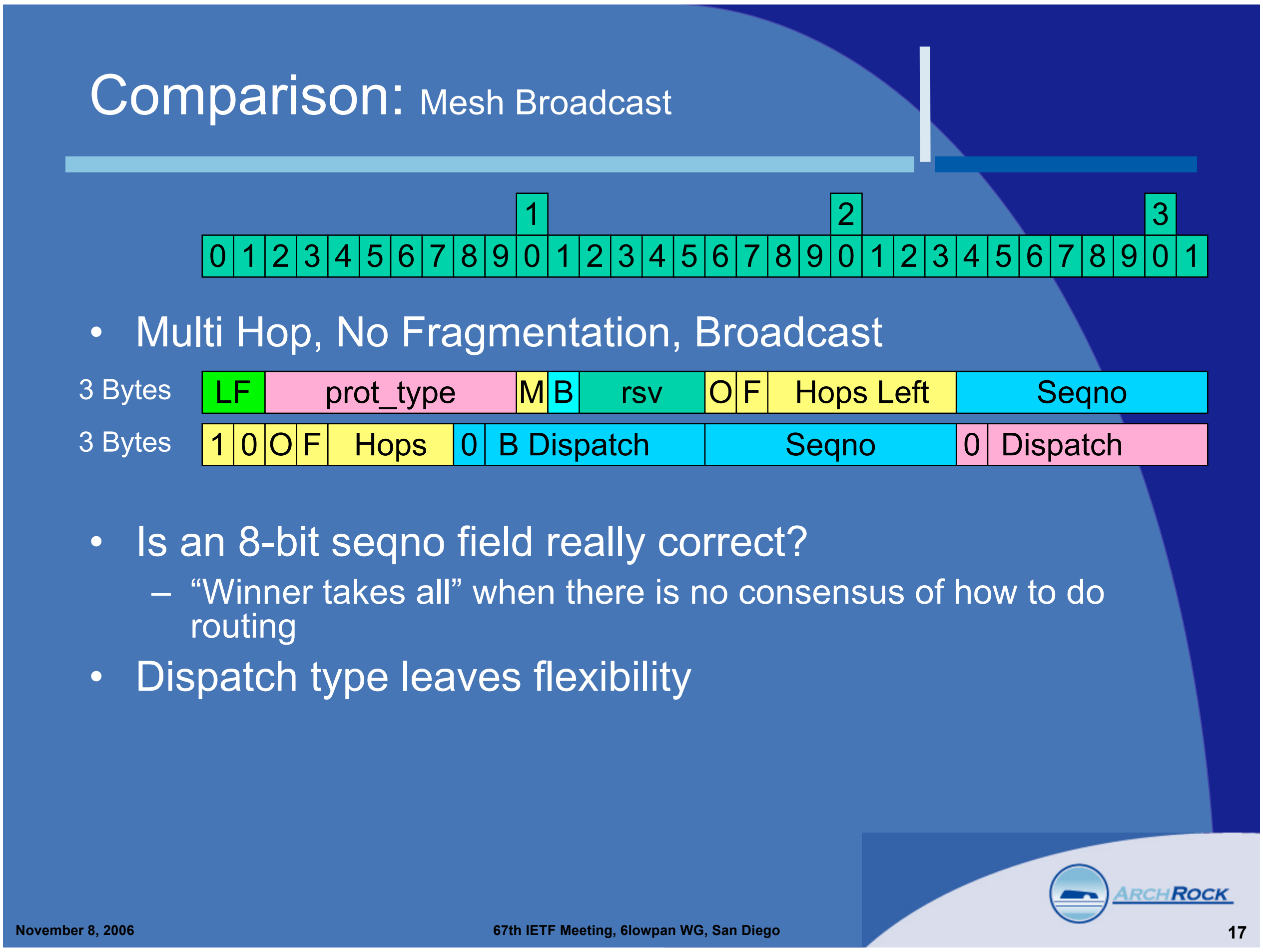
1	0	O	F	0xF	Hops	Originator Addr, Final Addr

(15-255 hops)

- Fragmentation Header

1	1	0	dgram_tag	dgram_size

1	1	1	dgram_tag	dgram_size	dgram_offset



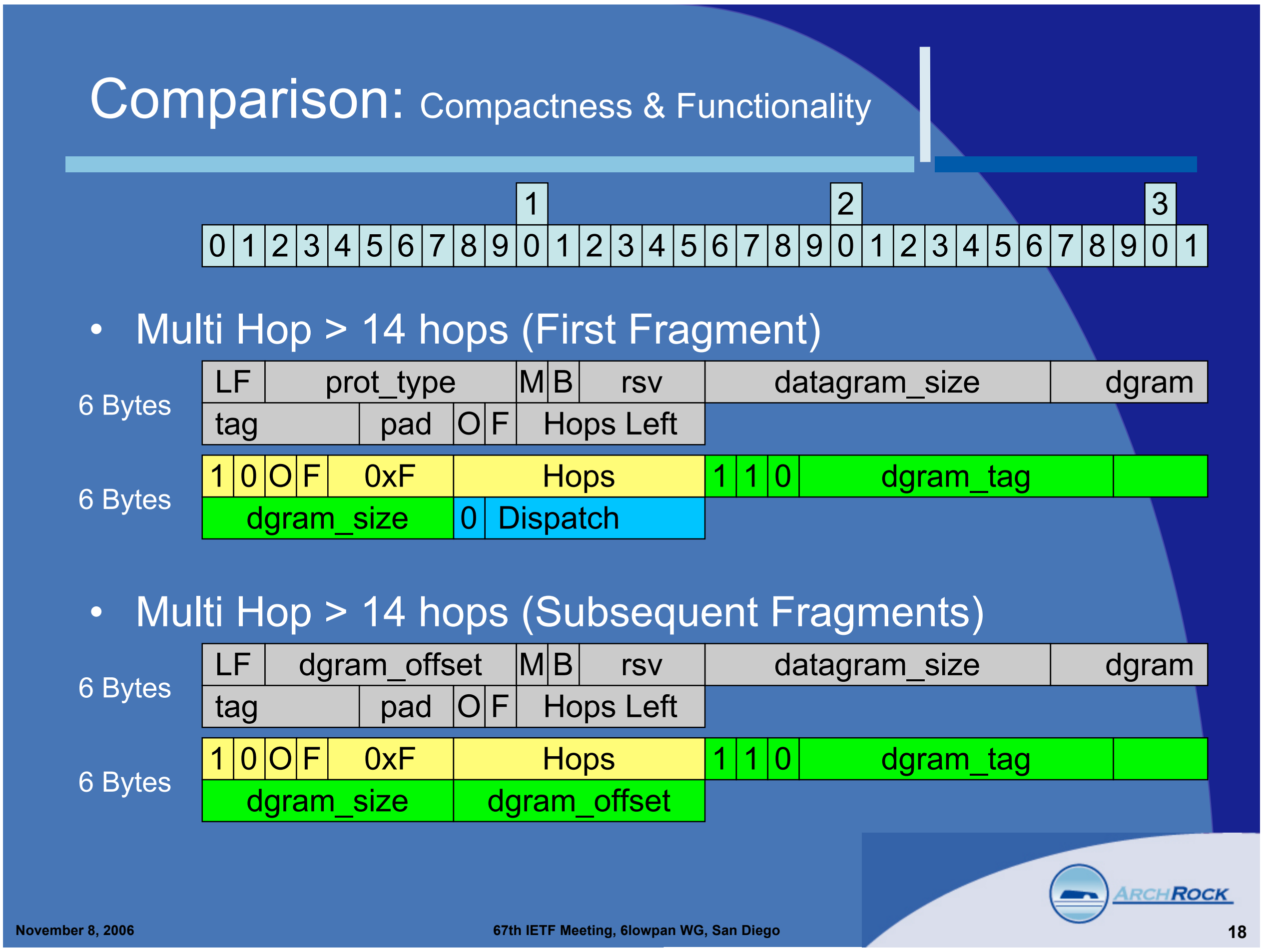
Comparison: Mesh Broadcast

<table>
<tr><td></td><td colspan="10"></td><td>1</td><td colspan="9"></td><td>2</td><td colspan="9"></td><td>3</td><td></td></tr>
<tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
</table>

- Multi Hop, No Fragmentation, Broadcast

<table>
<tr><td>3 Bytes</td><td colspan="2">LF</td><td colspan="8">prot_type</td><td>M</td><td>B</td><td colspan="4">rsv</td><td>O</td><td>F</td><td colspan="6">Hops Left</td><td colspan="8">Seqno</td></tr>
<tr><td>3 Bytes</td><td>1</td><td>0</td><td>O</td><td>F</td><td colspan="4">Hops</td><td>0</td><td colspan="7">B Dispatch</td><td colspan="8">Seqno</td><td>0</td><td colspan="7">Dispatch</td></tr>
</table>

- Is an 8-bit seqno field really correct?
  - "Winner takes all" when there is no consensus of how to do routing
- Dispatch type leaves flexibility



Comparison: Compactness & Functionality

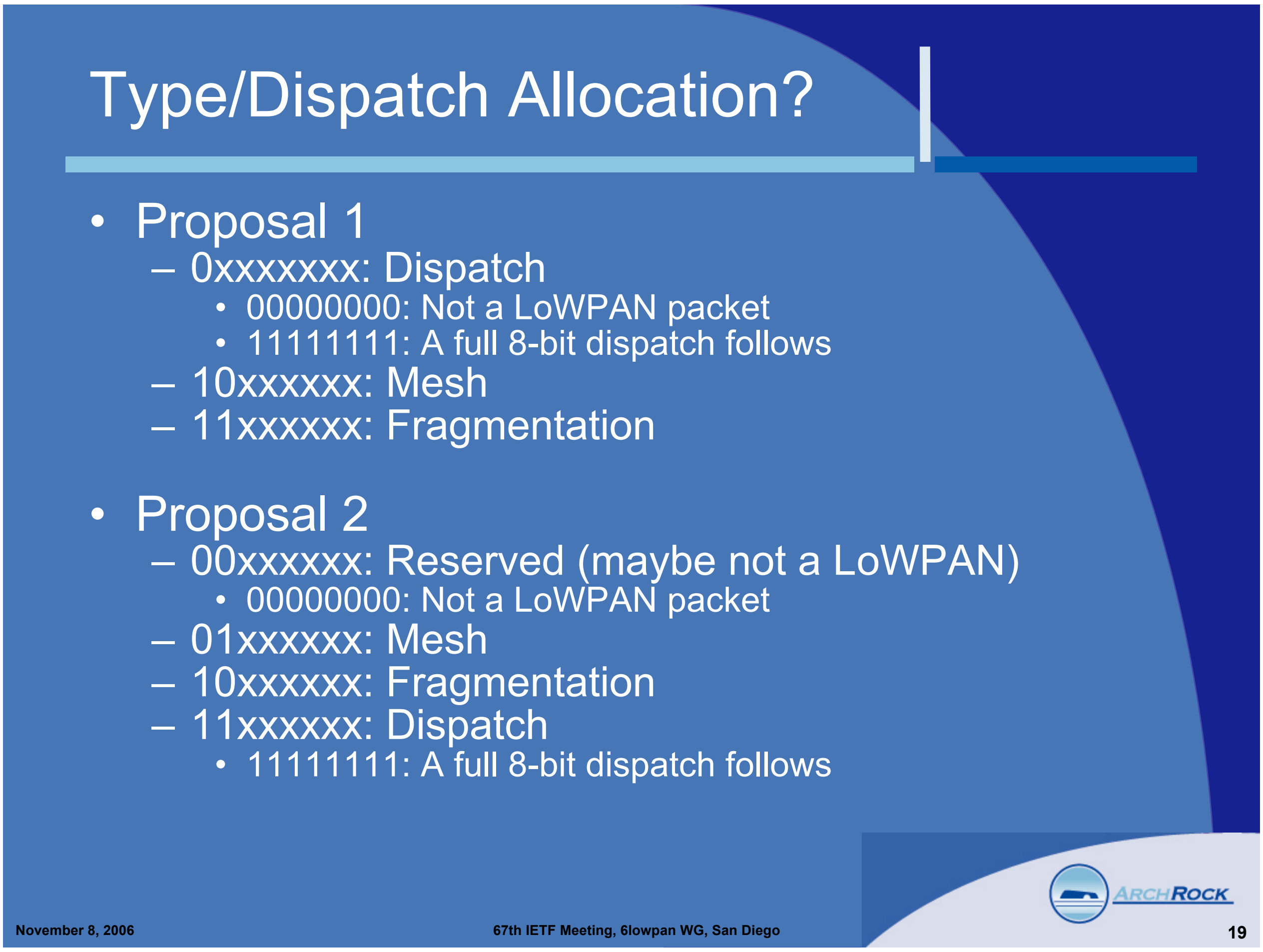
<table>
<tr><td></td><td colspan="10"></td><td>1</td><td colspan="9"></td><td>2</td><td colspan="9"></td><td>3</td><td></td></tr>
<tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
</table>

- Multi Hop > 14 hops (First Fragment)

<table>
<tr><td rowspan="2">6 Bytes</td><td colspan="2">LF</td><td colspan="8">prot_type</td><td>M</td><td>B</td><td colspan="4">rsv</td><td colspan="11">datagram_size</td><td colspan="5">dgram</td></tr>
<tr><td colspan="5">tag</td><td colspan="3">pad</td><td>O</td><td>F</td><td colspan="6">Hops Left</td><td colspan="16"></td></tr>
<tr><td rowspan="2">6 Bytes</td><td>1</td><td>0</td><td>O</td><td>F</td><td colspan="4">0xF</td><td colspan="8">Hops</td><td>1</td><td>1</td><td>0</td><td colspan="10">dgram_tag</td><td colspan="3"></td></tr>
<tr><td colspan="8">dgram_size</td><td>0</td><td colspan="7">Dispatch</td><td colspan="16"></td></tr>
</table>

- Multi Hop > 14 hops (Subsequent Fragments)

<table>
<tr><td rowspan="2">6 Bytes</td><td colspan="2">LF</td><td colspan="8">dgram_offset</td><td>M</td><td>B</td><td colspan="4">rsv</td><td colspan="11">datagram_size</td><td colspan="5">dgram</td></tr>
<tr><td colspan="5">tag</td><td colspan="3">pad</td><td>O</td><td>F</td><td colspan="6">Hops Left</td><td colspan="16"></td></tr>
<tr><td rowspan="2">6 Bytes</td><td>1</td><td>0</td><td>O</td><td>F</td><td colspan="4">0xF</td><td colspan="8">Hops</td><td>1</td><td>1</td><td>0</td><td colspan="10">dgram_tag</td><td colspan="3"></td></tr>
<tr><td colspan="8">dgram_size</td><td colspan="8">dgram_offset</td><td colspan="16"></td></tr>
</table>



Type/Dispatch Allocation?

- Proposal 1
  - 0xxxxxxx: Dispatch
    - 00000000: Not a LoWPAN packet
    - 11111111: A full 8-bit dispatch follows
  - 10xxxxxx: Mesh
  - 11xxxxxx: Fragmentation
- Proposal 2
  - 00xxxxxx: Reserved (maybe not a LoWPAN)
    - 00000000: Not a LoWPAN packet
  - 01xxxxxx: Mesh
  - 10xxxxxx: Fragmentation
  - 11xxxxxx: Dispatch
    - 11111111: A full 8-bit dispatch follows