IPv4 header:
Recycle 16 bits?

draft-briscoe-intarea-ipv4-id-reuse-00

Bob Briscoe

IETF-80 Mar 2011

This work is partly funded by Trilogy, a research project
supported by the European Community www.trilogy-project.org

problem: protocol extensibility

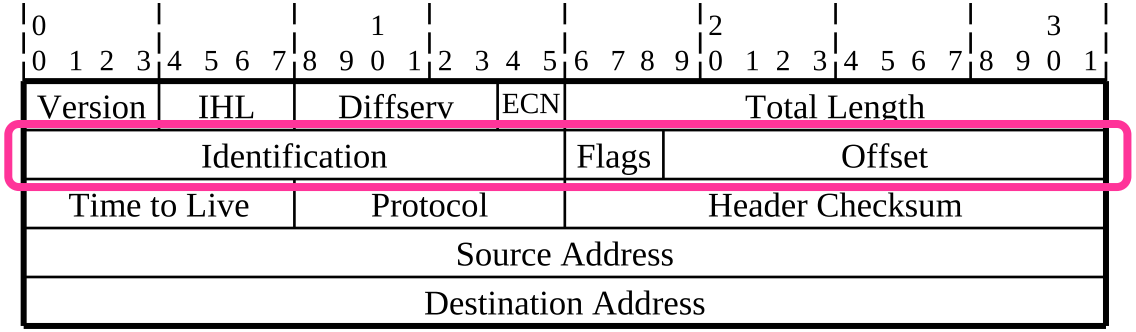
- new protocols need n bits in IP header (v4 and/or v6)
  - e.g. conex-abstract-mech, nat-reveal-option
- IPv4&6 extensibility mechanisms unusable in practice
  - any v4 option or v6 hop-by-hop ext hdr punted to slow-path

new extensibility design principle*

- put options where they will be ignored by existing kit
- kit with option code will know where to look

* ack: Rob Hancock

find a field ignored by existing kit



0 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

Version | IHL | Diffserv | ECN | Total Length

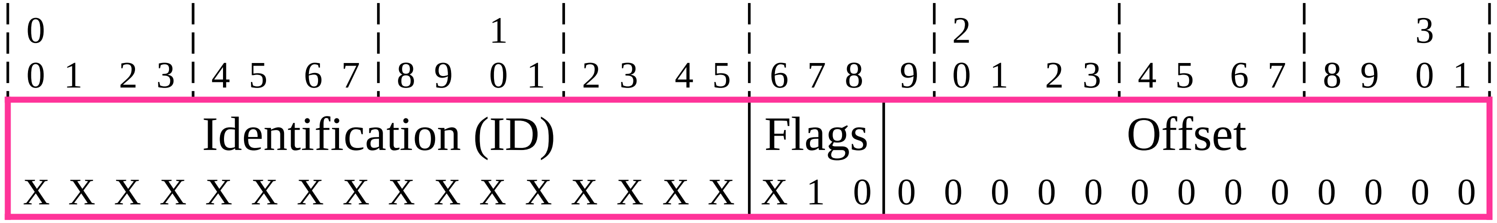
Identification | Flags | Offset

Time to Live | Protocol | Header Checksum

Source Address

Destination Address

- this draft proposes a new way to extend IPv4
  - similar ideas could apply to IPv6



0 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

Identification (ID) | Flags | Offset

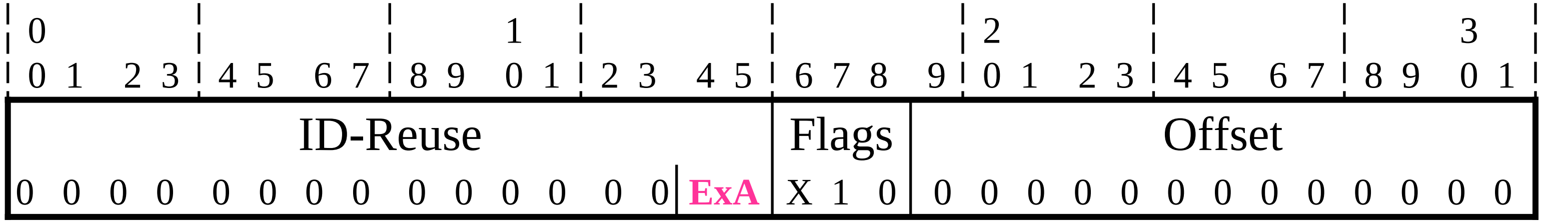
X X X X X X X X X X X X X X X X | X 1 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0

Fragmentation Fields that indicate an ‘atomic' Packet
“X” = “don’t care”

- large majority of IPv4 packets are atomic
  - unfragmented and unfragmentable
- ID field redundant in atomic packets [draft-ietf-intarea-ipv4-id-update]
- this draft proposes a process to manage re-use of the ID field

re-use ID field in atomic IPv4 packets

- frees up 16 bits for use by Internet community
  - propose IANA registry for re-using ID field
  - IETF can reassign whole field, subfields or codepoints within subfields
  - within constraints of previous use of ID for reassembly
- call the ID field “ID-Reuse” when packet is atomic
  - set currently unused ID-Reuse bits to zero



0										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
ID-Reuse																Flags			Offset												
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**ExA**		X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

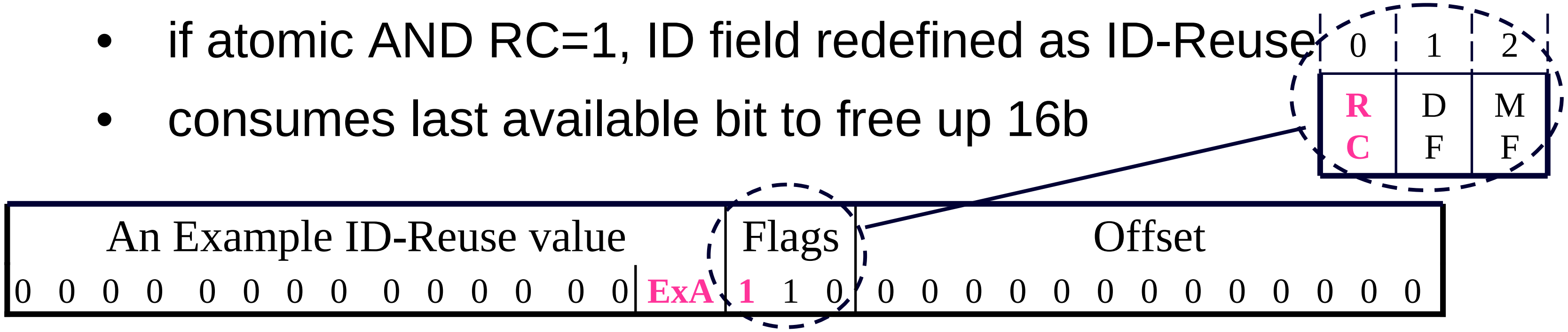
example registration: a new 2-bit field called ExA

disambiguation

An example ID-Reuse value	Flags	Offset
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	0 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0

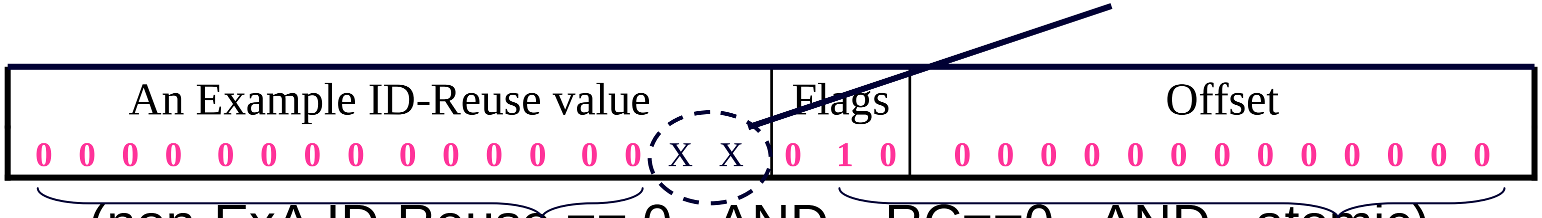
- how does an ExA implementation know whether this is:
  - a)an atomic packet using codepoint 10 in protocol ExA?
  - b)an atomic packet with arbitrary noise in the ID field?

- solution:
  - propose to redefine Reserved flag as ‘Recycled’ (RC) flag
  - if atomic AND RC=1, ID field redefined as ID-Reuse
  - consumes last available bit to free up 16b



incremental deployment tradeoff

- new problem
  - some pre-existing middleboxes (firewalls) discard RC = 1
- solution during initial deployment:
  - ExA implementation assumes packet using ExA protocol if



An Example ID-Reuse value	Flags	Offset
0 0 0 0 0 0 0 0 0 0 0 0 0 0 X X	0 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0

(non-ExA ID-Reuse == 0 AND RC==0 AND atomic)

(RC==1 AND atomic)

  - in this example, wrong with probability of 1 : $2^{(16-2)}$ = 1 : 2^{14}
  - protocol must not risk being wrong unless it does no harm

constraints on re-using IPv4 ID

- only in atomic packets

- IPsec authentication header interaction
  - ID immutable at least between IPsec endpoints
- Tunnel encapsulation
  - cannot rely on DF propagating to outer
  - cannot rely on ID field being copied to outer

conclusions

- consume last available bit to free up 16 in IPv4
- with non-trivial constraints
- principled incremental deployment
  - and a hack with a tradeoff and an added constraint

	middlebox traversal	new protocol recognition
RC=0	certain	uncertain
RC=1	uncertain	certain

discussion

- too constrained for those who want more bits?
- is this the most useful use of the Reserved flag?

IPv4 header:

Recycle 16 bits?

draft-briscoe-intarea-ipv4-id-reuse-00

Q&A