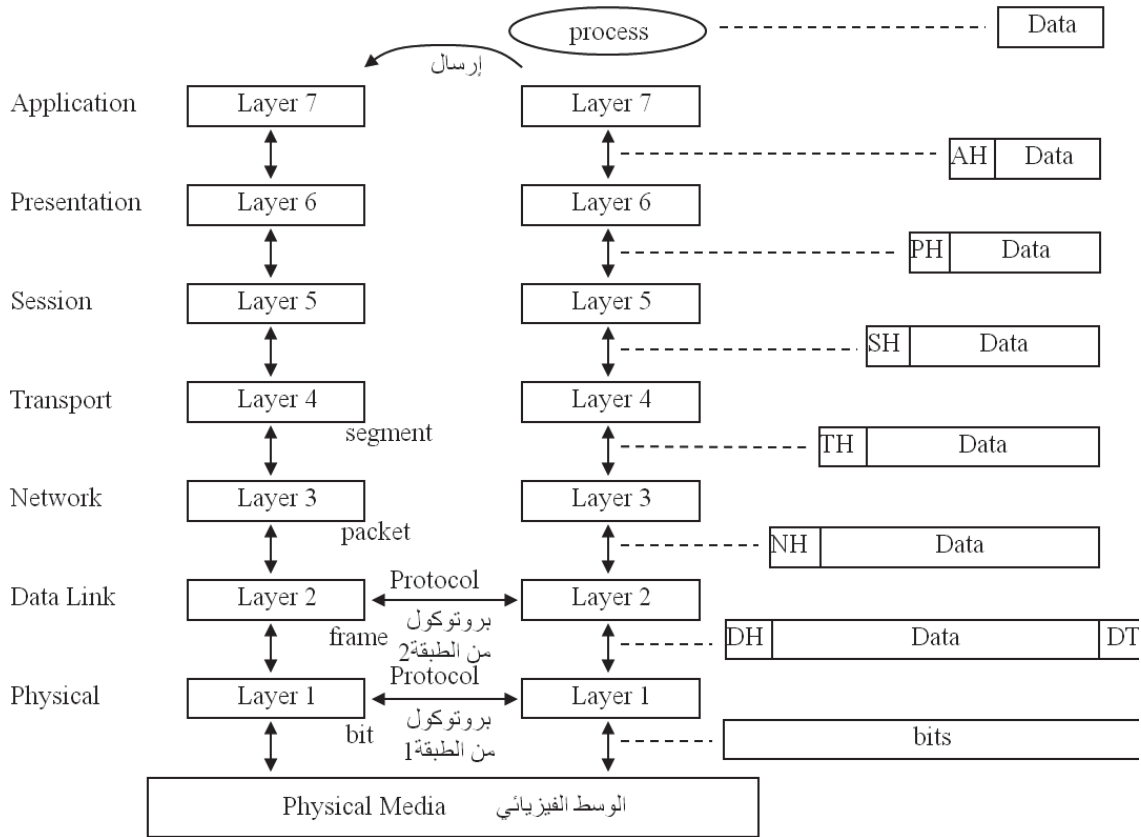


المحاضرة السادسة : الشبكات الحاسوبية 1

1 تحدثنا في المحاضرات السابقة عن تقسيم الطبقات لبروتوكولات الاتصالات:

- كل طبقة تعتبر الطبقة التي تسيقها موجودة وتستخدم وظائفها بشكل مباشر.
- كل طبقة (1+n) تستخدم وظائف الطبقة (n).
- كل طبقة تضيف ال header الخاص بها لل Data الواصلة إليها وتقصد بال header تعليمات التحكم الخاصة لعمل البروتوكولات في هذه الطبقة.

مثال :



لدينا في الرسم الموضح في الصفحة (1) process1 معين :

يطلب Application ولديه Data يريد إرسالها إلى process2 أخرى ، فكيف يتم ذلك ؟

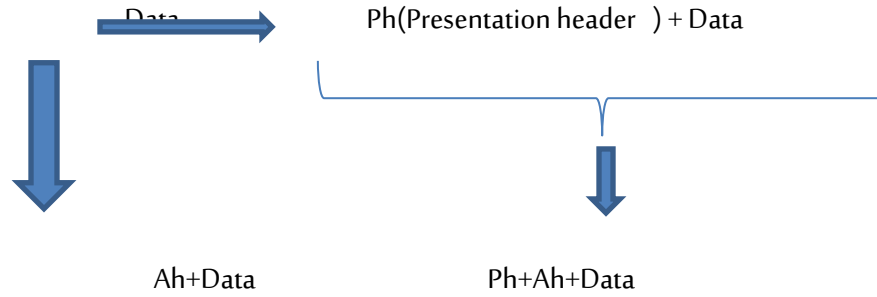
يقوم process1 بإرسال الData إلى طبقة Application ، تدخل الData إلى طبقة Application فتقوم الطبقة

بتروسيها بال header الخاصة بها فينتج لدينا :

$\xrightarrow{\text{Data}}$ Ah(Application header) + Data

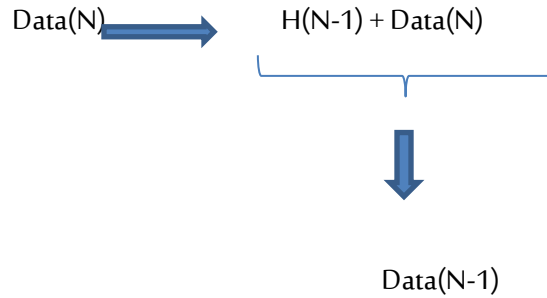
بعد ذلك ترسل طبقة ال Application ال Data والتي هي Ah + Data إلى طبقة ال Presentation بتغليف ال Data

بال header الخاص بها فينتج لدينا :



ونلاحظ أن ال Data التي كانت في طبقة Application تختلف عن ال Data في طبقة Presentation لأن ال Data الأولى هي المنقولة من process لكن ال Data الثانية هي عبارة عن Data المنقولة من process بالإضافة إلى Application header كما هو موضح بالشكل السابق .

فلو كان لدينا Data منقولة من طبقة N إلى طبقة N-1 فتكزن لدينا المعادلة التالية الخاصة بالنقل :



وهكذا تتابع الطبقات نقل ال Data حتى تصل إلى طبقة ال physical وتكون على شكل bits وكما ذكرنا كل طبقة تضيف ال header الخاص بها ومن ثم تنتقل هذه المعطيات من الطبقة الفيزيائية ال physical إلى الوسط الفيزيائي ما بين البروتوكولين (مرسل - مستقبل) لتستقبلهم الطبقة الفيزيائية عند البرتوكول الثاني بشكل بيتات فتتعرف الطبقة على ال header الخاص بها الموجود في ترويسة الطرد (وكما ذكرنا هو معلومات لازمة لأداء هذه الطبقة) .

وترسل ال Data إلى الطبقة العليا منها وهي طبقة ربط المعطيات (Data Link) تقوم هذه الطبقة بتحديد ال frame والتعرف على header وبدون trailer (حيث أن طبقة ربط المعطيات هي الوحيدة بين الطبقات التي تضيف trailer header أما باقي الطبقات تضيف فقط header ، ومما سبق نجد أن في طبقة Data Link يحصل التالي:

-1 ICI : (Interface Control Information) وهي معلومات التحكم المستخدمة عند طلب الوظائف بين الطبقات.

-2 SDU (Service Data Unit) Data وهي الـ Data.

❖ SAP (Service Access Point) وهي نقاط الخدمة بين الطبقات .

❖ PDU (Protocol Data Unit) وهو وحدة معطيات البروتوكول وهو عبارة عن Data مع header وله أنواع:

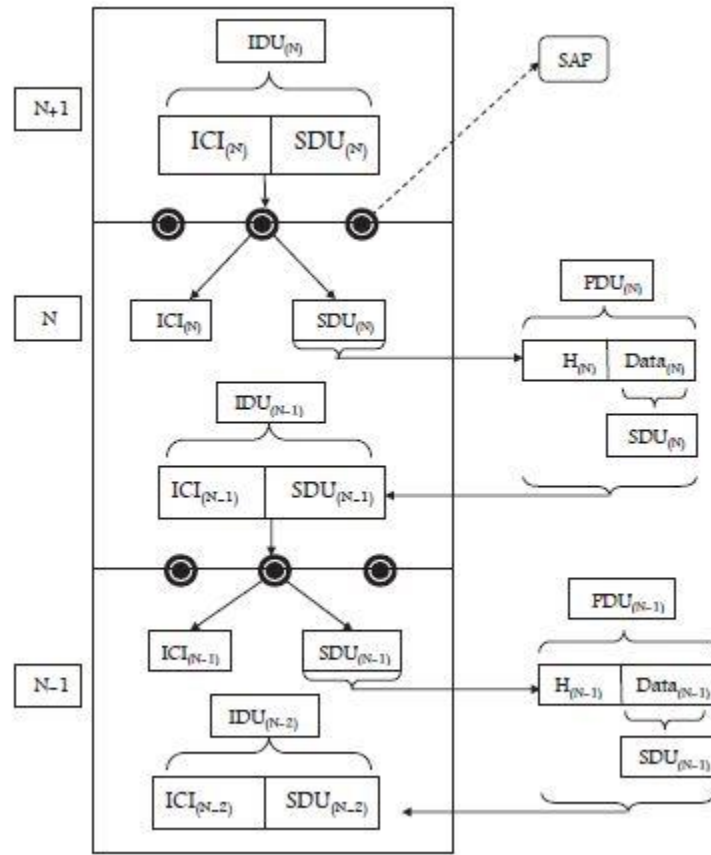
-1 في طبقة الـ Data Link وحدة المعطيات تسمى frame.

-2 في بروتوكول TCP وحدة المعطيات تسمى segment.

-3 في IP version 4 وحدة المعطيات تسمى datagram.

ملاحظة: كل طبقة تستطيع التفاعل (نقل المعلومات) مع الطبقة التي تليها والتي تسبقها ، لكنها فقط تستخدم وظائف الطبقة التي تسبقها (تحتها) .

لدينا المخطط التالي لكيفية نقل المعلومات بين الطبقات :



شرح المخطط :

هذه المخطط يشرح كيفية نقل المعطيات ضمن التطبيقات فأخذنا 3 طبقات $N+1$ و N و $N-1$ ، وكما ذكرنا في الأعلى إن ال SDU هي ال $Data$ التي ستنقل وكما نلاحظ في الطبقة $N+1$ نريد نقل ال $Data$ أو كما أسميناها SDU إل الطبقة N ولذلك نسميها $SDU(N)$ كما تقوم الطبقة $N+1$ بإضافة تعليمات التحكم المستخدمة في نقل هذه المعطيات وهي ICI ولأنها تعليمات التحكم الخاصة لعمل الطبقة N نسميها $ICI(N)$ تقوم الطبقة $N+1$ بتجميع $SDU(N)$ مع $ICI(N)$ لتشكيل لدينا ما يسمى $IDU(N)$ وهي وسيلة التفاعل بين الطبقتين N و $N+1$ وهي تحوي المعطيات ومعلومات التحكم ، بعد ذلك تُنقل ال $IDU(N)$ إلى الطبقة N عبر نقاط الخدمة SPA وعند مرورها من هذه النقاط تقوم SPA بفصل $SDU(N)$ عن $ICI(N)$ وتحديث عدة عمليات حيث تُأخذ $SDU(N)$ والتي هي المعطيات $Data$ ويضاف لها ال $header$ الخاص بهذه الطبقة $H(N)$ ومجموعة المعطيات $SDU(N)$ مع $H(N)$ نطلق عليها اسم $PDU(N)$ وهي وحدة المعطيات المراد ارسالها الآن إلى الطبقة $N-1$ لذلك تكون $PDU(N)$ هي $SDU(N_1)$ لأنها المعطيات المغلفة بال $header$ والتي نريد ارسالها للطبقة $N-1$ لكن قبل إرسالها تضيف الطبقة N معلومات التحكم الخاصة لعمل الطبقة $N-1$ ومجموع ال $SDU(N_1)$ مع $ICI(N-1)$ يسمى بـ $IDU(N-1)$ وتنقل بعد ذلك $IDU(N-1)$ عبر نقاط الخدمة SPA لتنتقل إلى الطبقة $N-1$ فتتفصل $IDU(N-1)$ بعد مرورها من SPA إلى $SDU(N_1)$ و $ICI(N-1)$ ويحدث تماماً ما حدث في الطبقة N ...

سؤال (1) هل $ICI(N)$ تكافئ $H(N)$ ؟

كما عرفنا سابقاً أن ICI هي معلومات التحكم اللازمة لعمل البروتوكول أو الطبقة وأن H هي $header$ وهي الترويسة التي تضيفها الطبقة ، وبعد مرور IDU على SPA تنفصل ICI عن SDU وتقوم الطبقة بأخذ SDU وتضيف ال $header$ عليها ، ولذلك فإن ICI لا تكافئ H لكن $H(N)$ تحوي بعض المعلومات المستنتجة من $ICI(N)$ بالإضافة إلى معلومات أخرى غير موجودة في $ICI(N)$ ، مثل رقم الطرد موجود في H لكنها غير موجودة في ICI أو مثلاً ACK موجودة في H وقيام باستنتاجها من خلال ICI .

سؤال (2) : ما هي ال $Data(N)$ الموجودة في $PDU(N)$ ؟

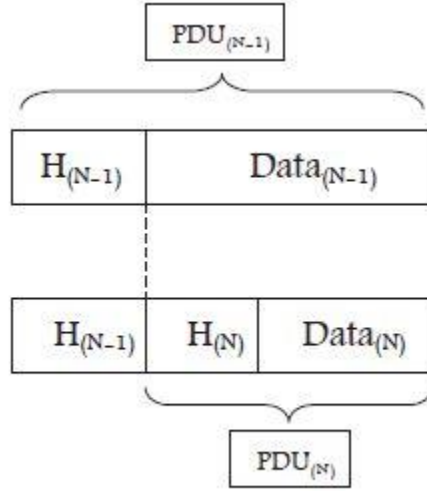
هي عبارة عن $SDU(N)$ لأنه بعد انفصال $SDU(N)$ عن $ICI(N)$ تأخذ الطبقة $SDU(N)$ والتي هي $DATA$ وتضيف عليها $header$ وتكون بمثابة الكتلة التي نريد ارسالها للطبقة $N-1$.

سؤال (3) : ماذا تمثل $SDU(N-1)$ ؟

هي تمثل ال Data التي يجب نقلها إلى الطبقة N-1 وهي عبارة عن PDU(N) أي هي عبارة عن مجموعة Data(N) مع H(N).

سؤال (4): ماذا تمثل PDU(N-1)؟

هي عبارة عن PDU(N) مع H(N-1) ، أي هي عبارة عن Data(N) مع H(N) مع H(N-1) هذا الرسم يوضح أكثر:



أي أن $Data(N-1) = SDU(N-1) = PDU(N)$

ملاحظة: يوجد في ICI معلومات تحكم كما ذكرنا لكن فيها أيضاً قيم Parameters لطلب بروتوكولات معينة .

تعرفنا الآن إلى آلية نقل معطيات من طبقة إلى أخرى ، اما الآن سنتحدث عن آلية طلب خدمة بين طبقتين أو كما ندعوها بـ OSI Service Primitive

OSI Service Primitive

هناك معيار محدد من قبل ال OSI لآلية طلب الخدمة بين طبقتين (N و N+1) من خلال الخطوات التالية:

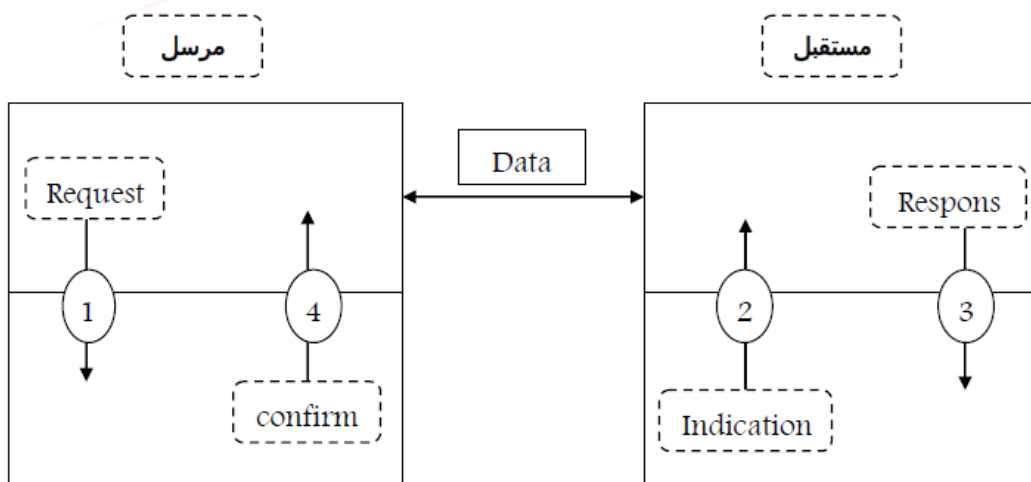
1- تطلب الطبقة N+1 الخدمة (Request) من الطبقة N من خلال (Parameters) المستخدمة في طلب الخدمة من بروتوكول إلى آخر.

2- إشعار بوقوع حدث (Indication).

3- استجابة بوقوع الحدث (Response).

4- تأكيد الاستجابة لطلب الخدمة (Confirm).

سيتم شرح الخطوات السابقة من خلال المخطط التالي:



خطوات كتابة البروتوكول:

أولاً: تحديد وظائف البروتوكول (Functionalities) مثل (فتح رابطة – إعادة الإرسال..).

ثانياً: Interface أي آلية طلب هذا البرتوكول أي عندما طلب طبقة برتوكول من الطبقة التي أدنى منها. وهنا يتم تحديد شكل الHeader وتقسيمات الحقول الخاصة بهذا البرتوكول.

مثال:

بفرض أنه لدينا برتوكول يقيم رابطة أي هو برتوكول موثوق ويقوم بتصحيح الأخطاء.

وبفرض أن هذه البرتوكول موجود لدى المرسل في الطبقة N والطبقة N+1 عند المرسل تريد إنشاء رابطة مع N+1 عند المستقبل فهي بحاجة إلى برتوكول موجود في الطبقة N. لنشرح كيف يتم إنشاء هذه الرابطة:

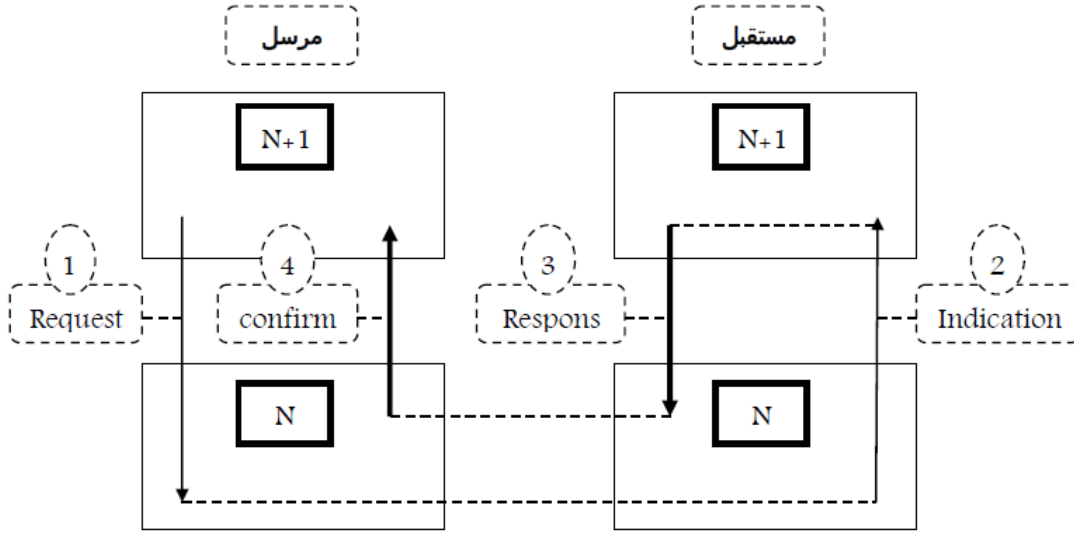
تتم وفق الخطوات التالية بما أن البرتوكول موثوق:

-1 Data request : يقصد به (address ,Data , length).

-2 Data Indication

-3 Data response

-4 Data Confirm



في المخطط السابق إن عملية فتح رابطة بين الطبقتين $N+1$ عند المرسل والطبقة $N+1$ عند المستقبل تتم عن طريق البرتوكول المسؤول عن فتح الرابطة الموجود في N أي أنه تتم عملية الرابطة في الطبقة $N+1$ عن طريق الرابطة N لأن البرتوكول الذي يفتح الرابطة موجود فيها. تتم وفق المراحل التالية:

المرحلة الأولى: تقوم الطبقة $N+1$ عند المرسل بطلب البرتوكول المسؤول عن إقامة الرابطة الموجود في الطبقة N عند المرسل فتُرسل له request بذلك. وتنتقل هذه ال Request من البرتوكول في الطبقة N عند المرسل إلى البرتوكول في الطبقة N عند المستقبل.

المرحلة الثانية: يرسل البرتوكول في الطبقة N عند المستقبل تنبيهاً (Indication) إلى الطبقة $N+1$ عند المستقبل وهي الطبقة إقامة الرابطة معها وهذا التنبيه بإنشاء الرابطة.

المرحلة الثالثة: ترسل الطبقة $N+1$ عند المستقبل إلى الطبقة N عند المستقبل أيضاً Response وهي استجابة لإقامة الرابطة وتنتقل هذه الاستجابة من الطبقة N عند المستقبل إلى الطبقة N عند المرسل.

المرحلة الرابعة: بعد وصول ال Response إلى N عند المرسل تقوم N بإرسال تأكيد للاستجابة Confirm لإقامة الرابطة إلى الطبقة $N+1$ وهي تعني إن الطبقة $N+1$ الموجودة في المستقبل قد استجابت لإقامة الرابطة.

وهكذا تكون قد تم إنشاء الرابطة بين الطبقة $N+1$ عند المرسل والطبقة $N+1$ عند المستقبل.

هنا مخطط توضيحي آخر لما سبق:

